aws

Guia do desenvolvedor

Amazon Polly

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Polly: Guia do desenvolvedor

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o Amazon Polly? 1
Como funciona 1
Benefícios 2
Você é um usuário iniciante do ? 3
Trabalhando com AWS SDKs 3
Introdução 5
Inscrevendo-se para AWS 5
Inscreva-se para um Conta da AWS 6
Configurando o AWS CLI 6
Reconfigurando o AWS CLI 7
Exemplo de síntese de fala 9
Streaming bidirecional 13
SynthesizeSpeech e StartSpeechSynthesisStream comparado 13
Enviando texto e recebendo áudio 15
Abra o stream 15
Enviar texto 15
Receba áudio 17
Feche o stream 17
Exemplos de código 17
Vozes no Amazon Polly 27
Vozes disponíveis 27
Vozes de marca 36
Vozes bilíngues 36
Vozes bilíngues com sotaque 36
Vozes totalmente bilíngues 37
Aplicar a voz de âncora 38
Ouvir as vozes 41
Sincronizar a velocidade da voz 41
Modificar a velocidade da voz 42
Idiomas no Amazon Polly 44
Árabe (arb) 48
Árabe do Golfo (ar-AE) 52
Catalão (ca-ES) 58
Chinês (cantonês) (yue-CN) 62

Chinês (mandarim) (zh-CN) 67
Tcheco (cs-CZ) 71
Dinamarquês (da-DK) 75
Holandês belga (nl-BE) 80
Holandês (nl-NL) 84
Inglês (EUA) (en-US) 88
Inglês australiano (en-AU) 92
Inglês britânico (en-GB) 97
Inglês indiano (en-IN) 101
Inglês (Irlanda) (en-IE) 106
Inglês (neozelandês) (en-NZ) 110
Inglês (singapurense) (en-SG) 116
Inglês (África do Sul) (en-ZA) 120
Inglês galês (en-GB-WLS) 125
Finlandês (Fi-Fi) 129
Francês (fr-FR) 134
Francês (belga) (fr-BE) 138
Francês canadense (fr-CA) 142
Alemão (de-DE) 146
Alemão (Austríaco) (de-AT) 150
Alemão (padrão suíço) (de-CH) 155
Hindi (hi-IN) 160
Islandês (is-IS) 164
Italiano (it-IT) 168
Japonês (ja-JP) 172
Coreano (ko-KR) 176
Norueguês (nb-NO) 179
Polonês (pl-PL) 183
Português (pt-PT) 187
Português brasileiro (pt-BR) 190
Romeno (ro-RO) 194
Russo (ru-RU) 197
Espanhol (es-ES) 201
Espanhol mexicano (es-MX) 204
Espanhol dos EUA (es-US) 207
Sueco (sv-SE) 210

Turco (tr-TR) 215
Galês (cy-GB) 219
Mecanismos de voz 225
Mecanismo generativo 225
Vozes generativas disponíveis 226
Compatibilidade de recursos e regiões 229
Long-form motor 230
Vozes de forma longa disponíveis 231
Compatibilidade de recursos e regiões 232
Mecanismo neural 232
Vozes neurais disponíveis 233
Compatibilidade de recursos e regiões 237
Mecanismo padrão 239
Vozes padrão disponíveis 239
Compatibilidade de recursos e regiões 242
Escolher um mecanismo de voz 244
Marcas de fala 245
Tipos de marca de fala 245
Visemas e Amazon Polly 246
Saída da marca de fala 247
Solicitar marcas de fala 248
Exemplo de marcas de fala sem SSML 250
Exemplo de marcas de fala com SSML 251
Uso de SSML 253
Caracteres reservados 254
Usar SSML no console 256
Como usar SSML com o comando Synthesize-Speech 258
Sintetizar um documento aprimorado por SSML 259
Etiquetas SSML compatíveis 260
Identificar um texto aprimorado por SSML 263
Adicionar uma pausa 263
Enfatizar palavras 264
Especificar outro idioma para palavras específicas 265
Colocar uma tag personalizada no texto 266
Adicionar uma pausa entre parágrafos 267
Usar a pronúncia fonética 267

Controlar o volume, a velocidade e o tom da fala ... 269
Configurar uma duração máxima para fala sintetizada ... 272
Adicionar uma pausa entre as frases ... 275
Controlar como tipos especiais de palavras são ditas ... 276
Pronunciar acrônimos e abreviaturas ... 280
Melhorar a pronúncia especificando partes da fala ... 280
Adicionar o som da respiração ... 282
Estilo de locutor ... 286
Adicionar compactação de intervalo dinâmico ... 287
Falar suavemente ... 289
Controlar o timbre ... 289
Sussurrar ... 291
Gerenciar léxicos ... 293
Usar vários léxicos ... 294
Fazer upload de um léxico ... 295
Aplicar léxicos (sintetização de fala) ... 301
Filtrar a lista de léxicos no console ... 304
Fazer download dos léxicos no console ... 306
Excluir um léxico ... 307
Arquivos de áudio longos ... 309
Configurar a política do IAM para a síntese assíncrona ... 310
Criar arquivos de áudio longos ... 310
Cotas ... 315
Regiões compatíveis ... 316
Cotas e taxas de controle de utilização ... 316
Solicitações simultâneas ... 317
Melhores práticas para mitigar o controle de utilização ... 317
Léxicos de pronúncia ... 318
SynthesizeSpeech Operações de API ... 318
SpeechSynthesisTask Operações de API ... 319
Speech Synthesis Markup Language (SSML) ... 319
Exemplo de código e aplicações ... 321
Exemplos em Java ... 321
DeleteLexicon ... 322
DescribeVoices ... 324
GetLexicon ... 325

ListLexicons 327
PutLexicon 328
StartSpeechSynthesisTask 331
Marcas de fala 334
SynthesizeSpeech 337
Exemplos de Python 339
DeleteLexicon 339
GetLexicon 340
ListLexicon 342
PutLexicon 342
StartSpeechSynthesisTask 344
SynthesizeSpeech 345
Exemplo de Java 345
Exemplo do Python 350
Exemplo em Python: index.html 352
Exemplo em Python: server.py 356
Exemplo para iOS 363
Exemplo para Android 365
Exemplos de código 368
Conceitos básicos 369
Ações 369
Cenários 421
Converter texto em fala e de volta em texto 421
Criar um aplicativo de sincronização labial 422
Criar uma aplicação para analisar o feedback dos clientes 423
Introdução à síntese de conversão de texto em voz 429
Segurança 437
Proteção de dados 438
Criptografia em repouso 438
Criptografia em trânsito 439
Privacidade do tráfego entre redes 439
Gerenciamento de Identidade e Acesso 439
Público 439
Autenticação com identidades 440
Gerenciar o acesso usando políticas 441
Como o Amazon Polly funciona com o IAM 443

Identity-based exemplos de políticas ... 449
Referência de permissões da API do Amazon Polly ... 457
Solução de problemas ... 458
Registro em log e monitoramento ... 460
Validação de conformidade ... 461
Resiliência ... 462
Segurança da infraestrutura ... 462
Práticas recomendadas de segurança ... 462
Usar VPC endpoints de interface ... 463
Disponibilidade ... 463
Criar uma política de endpoint da VPC para o Amazon Polly ... 464
Como testar a conexão entre a VPC e o Amazon Polly ... 464
Controlar o acesso ao endpoint da VPC ao Amazon Polly ... 464
Compatibilidade com chaves de contexto da VPC ... 465
Registrar chamadas de API do Amazon Polly em log com o AWS CloudTrail ... 466
Informações sobre o Amazon Polly no CloudTrail ... 466
Exemplo: entradas de arquivo de log do Amazon Polly ... 467
Integração do CloudWatch ... 470
Obter métricas do CloudWatch (Console) ... 470
Obter métricas do CloudWatch na AWS CLI ... 470
Métricas do Amazon Polly ... 471
Dimensões para métricas do Amazon Polly ... 472
Referência da API ... 474
Ações ... 474
DeleteLexicon ... 475
DescribeVoices ... 477
GetLexicon ... 481
GetSpeechSynthesisTask ... 484
ListLexicons ... 487
ListSpeechSynthesisTasks ... 490
PutLexicon ... 493
StartSpeechSynthesisStream ... 496
StartSpeechSynthesisTask ... 502
SynthesizeSpeech ... 510
Tipos de dados ... 517
AudioEvent ... 518

CloseStreamEvent 519
FlushStreamConfiguration 520
Lexicon 521
LexiconAttributes 522
LexiconDescription 524
StartSpeechSynthesisStreamActionStream 525
StartSpeechSynthesisStreamEventStream 526
StreamClosedEvent 528
SynthesisTask 529
TextEvent 534
ThrottlingReason 535
ValidationExceptionField 536
Voice 537
Tipos de erro comuns 539
Parâmetros gerais 542
Histórico do documento 545
..... dlxiii

O que é o Amazon Polly?

O Amazon Polly é um serviço na nuvem que converte texto em fala realista. Você pode usar o Amazon Polly para desenvolver aplicações que aumentam o envolvimento e a acessibilidade. O Amazon Polly oferece suporte a vários idiomas e inclui uma variedade de vozes realistas. Com o Amazon Polly, você pode criar aplicações compatíveis com fala que funcionam em vários locais e usar a voz ideal para os seus clientes. Além disso, você paga somente pelo texto que sintetizar. Você também pode armazenar em cache e reproduzir as falas geradas pelo Amazon Polly sem custo adicional.

O Amazon Polly oferece muitas opções de voz, incluindo opções generativas, longas, neurais e padrão text-to-speech (TTS). Essas vozes oferecem melhorias inovadoras na qualidade da fala usando a nova tecnologia de aprendizado de máquina para oferecer as vozes mais naturais e humanas possíveis text-to-speech. A tecnologia de TTS neural também é compatível com um estilo de âncora personalizado para uso em narração jornalística.

Usos comuns do Amazon Polly incluem, entre outros, aplicativos móveis, como leitores de notícias, jogos, plataformas de e-learning, aplicações de acessibilidade para deficientes visuais e o segmento em rápido crescimento da Internet das Coisas (IoT).

O Amazon Polly é certificado para uso com cargas de trabalho regulamentadas para a HIPAA (Lei de Portabilidade e Responsabilidade de Seguros de Saúde de 1996) e o Padrão de Segurança de Dados do Setor de Cartões de Pagamento (PCI DSS).

Como o Amazon Polly funciona

O Amazon Polly converte o texto inserido em uma fala realista. Para usar uma voz do Amazon Polly, escolha um mecanismo de voz, chame um método de síntese de voz, forneça o texto que você deseja sintetizar e especifique um formato de saída de áudio. Em seguida, o Amazon Polly sintetizará o texto fornecido em um fluxo de áudio de fala de alta qualidade.

- Texto de entrada: forneça o texto que deseja sintetizar e o Amazon Polly retornará um fluxo de áudio. É possível fornecer a entrada como texto sem formatação ou em formato SSML (Speech Synthesis Markup Language). Com SSML, você pode controlar diversos aspectos da fala, como pronúncia, volume, tom e velocidade da fala. Para obter mais informações, consulte Gerar fala com base em documentos SSML.
- Vozes disponíveis: o Amazon Polly oferece um portfólio de diversos idiomas e uma variedade de vozes, incluindo uma voz bilíngue para inglês e hindi. Para a maioria dos idiomas, é possível

escolher entre várias vozes, masculinas e femininas. Ao iniciar uma ação de síntese de fala, você especifica o nome do ID da voz e o Amazon Polly a usa para converter o texto em fala. O Amazon Polly não é um serviço de tradução. A fala é sintetizada no idioma do texto. Os números representados como dígitos (por exemplo, 53 em vez de cinquenta e três) são sintetizados no idioma da voz e não do texto. Para obter mais informações, consulte Vozes do Amazon Polly.

- Formato de saída: o Amazon Polly pode fornecer fala sintetizada em vários formatos. Selecione o formato de áudio que atenda às suas necessidades. Por exemplo, você pode solicitar a fala no formato Ogg Vorbis MP3 ou no formato Ogg Vorbis para consumo por aplicativos web e móveis. Ou você pode solicitar o formato de saída PCM para consumo por AWS IoT dispositivos e soluções de telefonia. Para aplicativos de telefonia, você pode usar os formatos Mu-law ou A-law.

> **Note**
>
> Para ouvir exemplos de vozes do Amazon Polly no navegador, consulte a visão geral de produtos do Amazon Polly.

Benefícios

Os benefícios de usar o Amazon Polly incluem:

- Alta qualidade — O Amazon Polly oferece vozes generativas, longas, neurais e de alta qualidade (TTS) de alto desempenho. text-to-speech Essas tecnologias sintetizam fala natural com alta precisão de pronúncia (incluindo abreviações, expansões de acrônimos, interpretações de data/hora e desambiguação de homógrafos).
- Baixa latência: o Amazon Polly garante tempos de resposta rápidos, o que faz dele uma opção viável para casos de uso de baixa latência, como sistemas de diálogo.
- Suporte a um amplo catálogo de idiomas e vozes: o Amazon Polly é compatível com dezenas de vozes e idiomas, oferecendo opções de vozes masculinas e femininas para a maioria dos idiomas. Esse número continuará a aumentar à medida que colocarmos mais vozes neurais online. As vozes do inglês dos EUA, Matthew e Joanna, também podem usar o estilo de fala neural de locutor, semelhante ao que você pode ouvir de um locutor de notícias profissional.
- Econômico — o pay-per-use modelo do Amazon Polly significa que não há custos de configuração. Comece aos poucos e aumente a escala verticalmente à medida que a aplicação crescer.

- Solução baseada em nuvem: as soluções de TTS no próprio dispositivo exigem recursos de computação significativos, particularmente capacidade de CPU, RAM e espaço em disco. Isso pode resultar em custo de desenvolvimento mais alto e maior consumo de energia em dispositivos como tablets, smartphones etc. Em contraste, a conversão de TTS feita no reduz Nuvem AWS drasticamente os requisitos de recursos locais. Isso permite o suporte a todos os idiomas e vozes disponíveis com qualidade excelente. Além disso, os aprimoramentos da fala são disponibilizados instantaneamente a todos os usuários finais e não exigem atualizações adicionais nos dispositivos.

> **Note**
>
> Para ouvir exemplos de vozes do Amazon Polly no navegador, consulte a visão geral de produtos do Amazon Polly.

Você é um usuário iniciante do ?

Caso seja iniciante no Amazon Polly, recomendamos que você leia as seguintes seções na ordem listada:

1. Como o Amazon Polly funciona: esta seção apresenta várias entradas e opções do Amazon Polly com as quais você pode criar uma experiência simples.
2. Conceitos básicos do Amazon Polly: aqui você abre sua conta e testa a síntese de fala do Amazon Polly.
3. Exemplo de código e aplicações para o Amazon Polly: esta seção apresenta exemplos adicionais que você pode usar para explorar o Amazon Polly.

Usando o Amazon Polly com um SDK AWS

AWS kits de desenvolvimento de software (SDKs) estão disponíveis para muitas linguagens de programação populares. Cada SDK fornece uma API, exemplos de código e documentação que permitem que os desenvolvedores criem facilmente aplicações em seu idioma de preferência.

Documentação do SDK	Exemplos de código
AWS SDK para C++	AWS SDK para C++ exemplos de código

Documentação do SDK	Exemplos de código
AWS CLI	AWS CLI exemplos de código
AWS SDK para Go	AWS SDK para Go exemplos de código
AWS SDK para Java	AWS SDK para Java exemplos de código
AWS SDK para JavaScript	AWS SDK para JavaScript exemplos de código
AWS SDK para Kotlin	AWS SDK para Kotlin exemplos de código
AWS SDK para .NET	AWS SDK para .NET exemplos de código
AWS SDK para PHP	AWS SDK para PHP exemplos de código
Ferramentas da AWS para PowerShell	Ferramentas da AWS para PowerShell exemplos de código
AWS SDK para Python (Boto3)	AWS SDK para Python (Boto3) exemplos de código
AWS SDK para Ruby	AWS SDK para Ruby exemplos de código
AWS SDK para Rust	AWS SDK para Rust exemplos de código
AWS SDK para SAP ABAP	AWS SDK para SAP ABAP exemplos de código
AWS SDK para Swift	AWS SDK para Swift exemplos de código

Exemplo de disponibilidade

Não consegue encontrar o que precisa? Solicite um exemplo de código usando o link Fornecer feedback na parte inferior desta página.

Conceitos básicos do Amazon Polly

O Amazon Polly fornece várias operações de API que você pode integrar com facilidade a aplicações existentes. Para ver uma lista das operações suportadas, consulte Ações.

Você pode realizar quase todas as mesmas operações no console do Amazon Polly e na AWS CLI. No entanto, você não pode ouvir a fala sintetizada no. AWS CLI Para trabalhar com áudio na AWS CLI, salve o texto em um arquivo. Depois, abra o arquivo em uma aplicação de áudio de sua escolha.

Você pode usar uma das opções a seguir:

- AWS SDKs — Ao usar os SDKs, suas solicitações para o Amazon Polly são automaticamente assinadas e autenticadas usando as credenciais que você fornece. Esta é a escolha recomendada para a criação de suas aplicações.
- AWS CLI — Você pode usar o AWS CLI para usar o Amazon Polly sem escrever nenhum código.

Antes de usar o Amazon Polly pela primeira vez, você deverá se cadastrar na AWS. Quando você se inscreve no Amazon Web Services (AWS), sua AWS conta é automaticamente cadastrada em todos os serviços AWS, incluindo o Amazon Polly. Você será cobrado apenas pelos serviços e recursos que usar. Se você é um novo AWS cliente, pode começar a usar o Amazon Polly gratuitamente. Para mais informações, consulte Nível de uso gratuito da AWS.

As seções a seguir descrevem como começar a usar o Amazon Polly.

Tópicos

- Inscrevendo-se para AWS
- Configurando o AWS CLI
- Reconfigurando o AWS CLI

Inscrevendo-se para AWS

Antes de usar qualquer AWS serviço, incluindo o Amazon Polly, você deve se inscrever no. AWS

Inscreva-se para um Conta da AWS

Para começar AWS, você precisa de um Conta da AWS. Para obter informações sobre como criar um Conta da AWS, consulte Introdução a um Conta da AWS no Guia de AWS Gerenciamento de contas referência.

Para obter mais informações sobre IAM, consulte o seguinte:

- AWS Identity and Access Management (IAM)
- Introdução ao IAM
- Guia do usuário do IAM

> **Note**
>
> Anote o ID AWS da sua conta. Você precisará dele nas próximas etapas.

Configurando o AWS CLI

Siga estas etapas para baixar e configurar o AWS CLI para funcionar com o Amazon Polly.

Para configurar o AWS Command Line Interface

1. Faça download e configure a AWS CLI. Para obter instruções, consulte os seguintes tópicos no Manual do usuário do AWS Command Line Interface :
   - Configurando o AWS Command Line Interface
   - Configurando o AWS Command Line Interface
2. Adicione um perfil nomeado para o usuário administrador no AWS CLI AWS arquivo Config. Você pode usar esse perfil ao executar os AWS CLI comandos. Para obter mais informações sobre perfis nomeados, consulte Perfis nomeados no Guia do usuário da AWS Command Line Interface .

```
[profile adminuser]
    aws_access_key_id = adminuser access key ID
    aws_secret_access_key = adminuser secret access key
    region = aws-region
```

Para obter uma lista das AWS regiões disponíveis e aquelas suportadas pelo Amazon Polly, consulte Regiões e endpoints no. Referência geral da Amazon Web Services

> **Note**
>
> Se você estiver usando uma região suportada pelo Amazon Polly que você especificou ao configurar o AWS CLI, omita a seguinte linha dos exemplos de código. AWS CLI
>
> ```
> --region aws-region
> ```

3. Verifique a configuração digitando o comando de ajuda a seguir no prompt de comando.

   ```
   aws help
   ```

   Uma lista de AWS comandos válidos deve aparecer na AWS CLI janela.

Reconfigurando o AWS CLI

Se você já baixou e configurou o AWS CLI, o Amazon Polly pode estar indisponível, a menos que você reconfigure o. AWS CLI O procedimento a seguir verifica se isso é necessário.

Para reativar o Amazon Polly a partir do AWS CLI

1. Verifique a disponibilidade do Amazon Polly digitando o seguinte comando de ajuda no prompt de comando. AWS CLI

   ```
   aws polly help
   ```

   Se você ver uma descrição do Amazon Polly e uma lista de comandos válidos aparecer na AWS CLI janela, você poderá usar o Amazon Polly imediatamente. AWS CLI Nesse caso, você poderá ignorar o restante deste procedimento. Se isso não for exibido, vá para a etapa 2.
2. Ative o Amazon Polly usando uma das duas opções a seguir:

   a. Desinstale e reinstale o. AWS CLI

      Para instruções, consulte Instalar a AWS Command Line Interface no Guia do usuário da AWS Command Line Interface .

or

b. Faça download do arquivo service-2.json.

No prompt de comando, execute o seguinte comando da .

```
aws configure add-model --service-model file://service-2.json --service-name
 polly
```

3. Verifique novamente a disponibilidade do Amazon Polly.

```
aws polly help
```

A descrição do Amazon Polly deve estar visível.

Exemplo de síntese de fala com o Amazon Polly

Esta página apresenta um breve exemplo de síntese de fala realizada no console AWS CLI, no e com Python. Este exemplo executa a síntese de fala com base em texto simples, não SSML.

Console

Sintetizar a fala no console

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Polly em. https://console.aws.amazon.com/polly/
2. Selecione a guia Text-to-Speech. O campo de texto será carregado com um texto de exemplo para você experimentar rapidamente o Amazon Polly.
3. Desative o SSML.
4. Digite ou cole este texto na caixa de entrada.

```
He was caught up in the game. In the middle of the 10/3/2014 W3C meeting he
 shouted, "Score!" quite loudly.
```

5. Em Mecanismo, escolha Generativo, Forma longa, Neural ou Padrão.
6. Escolha um idioma e uma AWS região e, em seguida, escolha uma voz. Se você selecionar Neural em Mecanismo, somente os idiomas e as vozes compatíveis com NTTS estarão disponíveis. Todas as vozes padrão e de forma longa estarão desabilitadas.
7. Para ouvir a fala imediatamente, selecione Ouvir.
8. Para salvar a fala em um arquivo, siga uma das etapas a seguir:
   a. Escolha Baixar.
   b. Para mudar para um formato de arquivo diferente, expanda Configurações adicionais, ative as configurações de formato de arquivo de fala, escolha o formato de arquivo desejado (como MP3 OGG, PCM, Mu-law ou A-law) e escolha Baixar.

AWS CLI

Neste exercício, chame a operação `SynthesizeSpeech` ao inserir texto de entrada. Você pode salvar o áudio resultante como arquivo e verificar seu conteúdo.

1. Execute o `synthesize-speech` AWS CLI comando para sintetizar texto de amostra em um arquivo de áudio ()`hello.mp3`.

   O AWS CLI exemplo a seguir está formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um acento circunflexo (^) e use aspas completas (") ao redor do texto de entrada com aspas simples (') para tags internas.

   ```
   aws polly synthesize-speech \
       --output-format mp3 \
       --voice-id Joanna \
       --text 'Hello, my name is Joanna. I learned about the W3C on 10/3 of last
    year.' \
       hello.mp3
   ```

   Na chamada para `synthesize-speech`, forneça um texto de amostra para ser sintetizado por uma voz de sua escolha. Você deve fornecer um ID de voz (explicado na etapa a seguir) e um formato de saída. O comando salva o áudio resultante no arquivo `hello.mp3`. Além do MP3 arquivo, a operação envia a seguinte saída para o console.

   ```
   {
           "ContentType": "audio/mpeg",
           "RequestCharacters": "71"
   }
   ```

2. Reproduza o arquivo `hello.mp3` resultante para verificar a fala sintetizada.

Python

Para testar o código de exemplo do Python, você precisará do AWS SDK para Python (Boto). Para obter instruções, consulte AWS SDK para Python (Boto3).

O código Python nesse exemplo realiza as seguintes ações:

- Invoca o AWS SDK para Python (Boto) para enviar uma `SynthesizeSpeech` solicitação ao Amazon Polly (fornecendo algum texto como entrada).
- Acessa o fluxo de áudio resultante na resposta e salva o áudio em um arquivo (`speech.mp3`) no seu disco local.
- Reproduz o arquivo de áudio com o player de áudio padrão do seu sistema local.

Salve o código em um arquivo (exemple.py) e execute-o.

```
"""Getting Started Example for Python 2.7+/3.3+"""
from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError
from contextlib import closing
import os
import sys
import subprocess
from tempfile import gettempdir

# Create a client using the credentials and region defined in the [adminuser]
# section of the AWS credentials file (~/.aws/credentials).
session = Session(profile_name="adminuser")
polly = session.client("polly")

try:
    # Request speech synthesis
    response = polly.synthesize_speech(Text="Hello world!", OutputFormat="mp3",
                                       VoiceId="Joanna")
except (BotoCoreError, ClientError) as error:
    # The service returned an error, exit gracefully
    print(error)
    sys.exit(-1)

# Access the audio stream from the response
if "AudioStream" in response:
    # Note: Closing the stream is important because the service throttles on the
    # number of parallel connections. Here we are using contextlib.closing to
    # ensure the close method of the stream object will be called automatically
    # at the end of the with statement's scope.
        with closing(response["AudioStream"]) as stream:
           output = os.path.join(gettempdir(), "speech.mp3")

           try:
            # Open a file for writing the output as a binary stream
                with open(output, "wb") as file:
                   file.write(stream.read())
           except IOError as error:
              # Could not write to file, exit gracefully
              print(error)
              sys.exit(-1)

else:
```

```
    # The response didn't contain audio data, exit gracefully
    print("Could not stream audio")
    sys.exit(-1)

# Play the audio using the platform's default player
if sys.platform == "win32":
    os.startfile(output)
else:
    # The following works on macOS and Linux. (Darwin = mac, xdg-open = linux).
    opener = "open" if sys.platform == "darwin" else "xdg-open"
    subprocess.call([opener, output])
```

Para exemplos mais detalhados, consulte os tópicos a seguir:

- Usar SSML no console
- Aplicar léxicos (sintetização de fala)
- Exemplo de código e aplicações para o Amazon Polly

Sintetizando fala com streaming bidirecional

O Amazon Polly fornece uma `StartSpeechSynthesisStream` operação que estabelece uma HTTP/2 conexão com comunicação bidirecional entre seu aplicativo e o serviço. O texto flui do seu aplicativo para o Amazon Polly enquanto o áudio sintetizado retorna. Você envia o texto assim que ele se torna disponível, e o Amazon Polly retorna o áudio à medida que ele é sintetizado, sem que nenhum dos lados espere que o outro termine.

Isso é útil quando o texto é produzido progressivamente, em vez de todo de uma vez. Por exemplo, um chatbot de atendimento ao cliente baseado em um modelo básico no Amazon Bedrock gera sua resposta token por token. Com o streaming bidirecional, seu aplicativo pode encaminhar cada fragmento de texto para o Amazon Polly à medida que o modelo o produz e começar a reproduzir o áudio de volta para o chamador enquanto o modelo ainda está gerando o restante da resposta.

Essa operação requer o mecanismo generativo e um AWS SDK que ofereça suporte a fluxos de HTTP/2 eventos. O áudio chega como uma sequência de partes que seu aplicativo acumula em uma saída de áudio completa. As marcas de fala não são suportadas por esta operação.

> **Note**
>
> Não há suporte para AWS CLI (v1 e v2), AWS Ferramentas para PowerShell (v4 e v5), Python e.NET v3. Você pode usar a API de streaming bidirecional com os seguintes SDKs: SDK for AWS Java 2.x JavaScript , v3, .NET v4, C++, Go v2, Kotlin, PHP v3, Ruby v3, Rust e Swift.

Tópicos

- SynthesizeSpeech e StartSpeechSynthesisStream comparado
- Enviando texto e recebendo áudio
- Exemplos de código

SynthesizeSpeech e StartSpeechSynthesisStream comparado

SynthesizeSpeeché uma operação de solicitação-resposta. Você fornece o texto completo em uma única solicitação e recebe o áudio sintetizado completo em uma única resposta. Ele suporta todos os mecanismos (padrão, neural, de formato longo, generativo), todos os formatos de saída, incluindo

marcas de fala, e tem um limite de texto de 6.000 caracteres no total (dos quais não mais do que 3.000 caracteres podem ser cobrados) por solicitação. A resposta retorna o áudio assim que os primeiros bytes estão disponíveis. Use essa operação quando você tiver todo o texto disponível antecipadamente.

StartSpeechSynthesisStreamé uma operação de streaming bidirecional. Ele abre uma HTTP/2 conexão pela qual você envia texto de forma incremental e recebe áudio à medida que é sintetizado. Não há limite de texto por solicitação, pois o texto é transmitido continuamente. Ele requer o mecanismo generativo e não suporta marcas de fala. Use essa operação quando o texto chegar incrementalmente e você quiser que a saída de áudio comece antes que todas as entradas estejam disponíveis. Os cenários comuns incluem:

- IA conversacional e assistentes de voz. Um modelo de linguagem grande gera texto de resposta em pequenos pedaços (tokens). Encaminhe cada trecho de texto para o Amazon Polly assim que ele chegar, para que o usuário ouça a fala enquanto o modelo ainda está sendo gerado.
- Real-time tradução. Um sistema de tradução produz texto traduzido segmento por segmento. Transmita cada segmento para síntese sem esperar que a tradução completa seja concluída.
- Long-form conteúdo que excede SynthesizeSpeech os limites. Textos com mais de 6.000 caracteres podem ser transmitidos continuamente sem se dividir em várias solicitações ou gerenciar limites de fragmentos.

Comparação de SynthesizeSpeech e StartSpeechSynthesisStream

Aspecto	SynthesizeSpeech	StartSpeechSynthesisStream
Protocolo	Request-response	Fluxo de eventos bidirecional () HTTP/2
Entrega de texto	Texto completo no corpo da solicitação	Streaming de texto de entrada por meio TextEvent de mensagens
Entrega de áudio	Resposta de streaming de áudio via corpo de resposta HTTP	Streaming de resposta de áudio por meio AudioEvent de mensagens
Suporte do motor	padrão, neural, de formato longo, generativo	somente generativo

Aspecto	SynthesizeSpeech	StartSpeechSynthesisStream
Suporte SSML	Sim (todos os mecanismos; as tags suportadas variam de acordo com o mecanismo)	Sim (somente etiquetas de mecanismo generativas)
Léxicos	Sim	Sim
Marcas de fala	Sim	Não
Limite de texto	Total de 6.000 caracteres (3.000 cobrados) por solicitação	Total de 6.000 caracteres (3.000 cobrados) por TextEvent
AWS CLI apoio	Sim	Não (o streaming bidirecional requer um SDK)

Enviando texto e recebendo áudio

Uma sessão de streaming bidirecional envolve abrir uma conexão, enviar texto e receber áudio simultaneamente e, em seguida, fechar a transmissão quando a entrada for concluída. As seções a seguir descrevem cada fase em detalhes.

Abra o stream

Seu aplicativo chama a StartSpeechSynthesisStreamoperação por meio do SDK, especificando os parâmetros de síntese (`Engine`,, `VoiceId`, `OutputFormat`, e opcionalmente,`LanguageCode`,`LexiconNames`). `SampleRate` O SDK estabelece uma HTTP/2 conexão e o fluxo bidirecional está pronto para aceitar eventos de entrada.

Enviar texto

O cliente envia uma ou mais TextEventmensagens no fluxo de entrada. Cada evento pode ser enviado assim que o texto estiver disponível, sem esperar que a entrada completa esteja pronta. Os eventos de texto não precisam estar alinhados com os limites da frase ou da pontuação. O Amazon Polly reagrupa o texto internamente e produz uma fala com som natural, independentemente de como a entrada é dividida entre os eventos.

> **Note**
>
> Ao usar SSML, cada documento SSML deve ser independente em um único documento. `TextEvent` Você não pode dividir tags SSML em vários eventos. No entanto, você pode misturar eventos de texto simples e eventos SSML no mesmo fluxo.

A transmissão está sujeita aos seguintes limites de tempo:

- Duração máxima da transmissão: 10 minutos. O Amazon Polly fecha o stream após 10 minutos, independentemente da atividade. Se seu conteúdo exigir mais tempo, abra um novo stream para o texto restante.
- Tempo limite de inatividade entre eventos consecutivos: 5 segundos. Se nenhum evento de entrada for enviado por 5 segundos, o Amazon Polly fechará o stream. Se sua fonte de texto tiver pausas de mais de 5 segundos, envie um keep-alive `TextEvent` com uma string vazia ou com espaço em branco para evitar o tempo limite.

Forçando a síntese de texto em buffer com descarga

Por padrão, o Amazon Polly decide quando sintetizar texto em buffer com base nos limites da linguagem natural. Isso produz a melhor qualidade de áudio, mas significa que o áudio pode não ser retornado imediatamente após o envio de um`TextEvent`.

A lavagem permite que você controle quando a síntese acontece. Quando você limpa, o Amazon Polly sintetiza imediatamente todo o texto armazenado em buffer até o momento, independentemente de o texto terminar em um limite natural. Isso é útil quando sua fonte de texto faz uma pausa entre as seções lógicas e você deseja fornecer áudio para o que foi enviado até o momento.

Para lavar, defina o. FlushStreamConfiguration `Force`parâmetro para `true` um`TextEvent`. Você também pode enviar um vazio `TextEvent` com o sinalizador flush definido para acionar a síntese sem adicionar novo conteúdo.

Lavar é uma troca. `Force`A configuração para o `true` meio da frase pode afetar a pronúncia e a entonação porque o sintetizador não tem contexto sobre o que se segue. Para obter melhores resultados, permita que o Amazon Polly atenda aos limites naturais sempre que possível e só force a síntese quando os requisitos de latência exigirem.

Receba áudio

Conforme o Amazon Polly sintetiza texto, ele retorna AudioEventmensagens no fluxo de saída. Cada evento contém uma parte dos dados de áudio. Seu aplicativo deve acumular esses fragmentos (por exemplo, gravando-os sequencialmente em um arquivo ou buffer de áudio) para produzir a saída de áudio completa. Eventos de áudio podem chegar enquanto você ainda está enviando eventos de texto.

Feche o stream

Quando todo o texto de entrada for enviado, o cliente envia um CloseStreamEvent. O Amazon Polly conclui o processamento de qualquer texto armazenado em buffer restante, envia os eventos finais de áudio e retorna um StreamClosedEventque contém o número total de caracteres sintetizados. Sempre envie um `CloseStreamEvent` em vez de depender do flushing para encerrar a transmissão. O fechamento garante que todo o texto armazenado em buffer seja sintetizado e retornado.

Para obter detalhes completos sobre parâmetros de solicitação, tipos de eventos e erros, consulte a referência da StartSpeechSynthesisStream API.

Exemplos de código

O exemplo a seguir demonstra uma sessão completa de streaming bidirecional. O programa cria um cliente Amazon Polly assíncrono, abre um stream configurado com o mecanismo generativo e a saída MP3, envia texto como uma série `TextEvent` de mensagens, acumula os fragmentos `AudioEvent` retornados em um arquivo de saída e fecha o stream com um. `CloseStreamEvent` Como a entrada e a saída acontecem simultaneamente, os dados de áudio começam a chegar antes que todo o texto seja enviado.

Java

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyAsyncClient;
import software.amazon.awssdk.services.polly.model.*;

import java.io.FileOutputStream;
import java.io.OutputStream;
import java.util.concurrent.CompletableFuture;
import java.util.concurrent.Semaphore;
import org.reactivestreams.Publisher;
```

```
import org.reactivestreams.Subscription;

public class BidirectionalStreamExample {

    public static void main(String[] args) throws Exception {
        try (PollyAsyncClient pollyClient = PollyAsyncClient.builder()
                .region(Region.US_EAST_1)
                .build()) {

        StartSpeechSynthesisStreamRequest request =
 StartSpeechSynthesisStreamRequest.builder()
                .engine(Engine.GENERATIVE)
                .voiceId(VoiceId.TIFFANY)
                .outputFormat(OutputFormat.MP3)
                .sampleRate("24000")
                .build();

        try (OutputStream audioOutput = new FileOutputStream("output.mp3")) {

        StartSpeechSynthesisStreamResponseHandler responseHandler =
                StartSpeechSynthesisStreamResponseHandler.builder()

 .subscriber(StartSpeechSynthesisStreamResponseHandler.Visitor.builder()
                                .onAudioEvent(audioEvent -> {
                                    try {
                                        byte[] audio =
 audioEvent.audioChunk().asByteArray();
                                        System.out.println("Received AudioEvent: " +
 audio.length + " bytes");
                                        audioOutput.write(audio);
                                    } catch (Exception e) {
                                        throw new RuntimeException(e);
                                    }
                                })
                                .onStreamClosedEvent(closedEvent -> {
                                    System.out.println("Stream closed. Characters
 synthesized: "
                                            + closedEvent.requestCharacters());
                                })
                                .onDefault(event -> {})
                                .build())
                        .onError(error -> {
                            System.err.println("Stream error: " +
 error.getMessage());
```

```
                        })
                        .build();

        String[] textChunks = {
                "The weather forecast for today shows clear skies ",
                "with temperatures reaching twenty five degrees. ",
                "Tomorrow we expect some cloud cover in the morning ",
                "but it should clear up by the afternoon. ",
                "The rest of the week looks mostly sunny ",
                "with a slight chance of rain on Friday. ",
                "Overall a great week to spend time outdoors."
        };

        Publisher<StartSpeechSynthesisStreamActionStream> inputPublisher =
subscriber -> {
            subscriber.onSubscribe(new Subscription() {
                private final Semaphore permits = new Semaphore(0);
                private volatile boolean cancelled = false;

                @Override
                public void request(long n) {
                    permits.release((int) Math.min(n, Integer.MAX_VALUE));
                }

                @Override
                public void cancel() {
                    cancelled = true;
                    permits.release();
                }

                {
                    new Thread(() -> {
                        for (String chunk : textChunks) {
                            try { permits.acquire(); } catch (InterruptedException
e) { return; }
                            if (cancelled) return;
                            System.out.println("Sending TextEvent: " +
chunk.trim());

subscriber.onNext(StartSpeechSynthesisStreamActionStream.textEventBuilder()
                                    .text(chunk).textType(TextType.TEXT).build());
                            // Simulate delay between chunks (e.g. waiting for LLM
tokens)
```

```
                            try { Thread.sleep(300); } catch (InterruptedException
 e) { return; }
                        }
                        if (!cancelled) {
                            subscriber.onNext(StartSpeechSynthesisStreamActionStream
                                    .closeStreamEventBuilder().build());
                            subscriber.onComplete();
                        }
                    }).start();
                }
            });
        };

        CompletableFuture<Void> future = pollyClient.startSpeechSynthesisStream(
                request, inputPublisher, responseHandler);

        future.join();
        } // audioOutput closed
        } // pollyClient closed
    }
}
```

JavaScript

```
import { PollyClient, StartSpeechSynthesisStreamCommand } from "@aws-sdk/client-
polly";
import { createWriteStream } from "fs";

const client = new PollyClient({ region: "us-east-1" });

const textChunks = [
  "The weather forecast for today shows clear skies ",
  "with temperatures reaching twenty five degrees. ",
  "Tomorrow we expect some cloud cover in the morning ",
  "but it should clear up by the afternoon. ",
  "The rest of the week looks mostly sunny ",
  "with a slight chance of rain on Friday. ",
  "Overall a great week to spend time outdoors.",
];

const sleep = (ms) => new Promise((resolve) => setTimeout(resolve, ms));

async function* createInputEvents() {
```

```
  for (const chunk of textChunks) {
    console.log(`Sending TextEvent: ${chunk.trim()}`);
    yield {
      TextEvent: {
        Text: chunk,
        TextType: "text",
      },
    };
    // Simulate delay between chunks (e.g. waiting for LLM tokens)
    await sleep(300);
  }

  yield { CloseStreamEvent: {} };
}

async function synthesizeStream() {
  const command = new StartSpeechSynthesisStreamCommand({
    Engine: "generative",
    VoiceId: "Tiffany",
    OutputFormat: "mp3",
    SampleRate: "24000",
    ActionStream: createInputEvents(),
  });

  const response = await client.send(command);
  const outputStream = createWriteStream("output.mp3");

  for await (const event of response.EventStream) {
    if (event.AudioEvent) {
      console.log(`Received AudioEvent: ${event.AudioEvent.AudioChunk.length}
 bytes`);
      outputStream.write(event.AudioEvent.AudioChunk);
    } else if (event.StreamClosedEvent) {
      console.log(
        `Stream closed. Characters synthesized:
 ${event.StreamClosedEvent.RequestCharacters}`
      );
    }
  }

  outputStream.end();
}
```

```
synthesizeStream().catch(console.error);
```

Go

```
package main

import (
 "context"
 "fmt"
 "os"
 "strings"
 "time"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/polly"
 "github.com/aws/aws-sdk-go-v2/service/polly/types"
)

func main() {
 ctx := context.Background()

 cfg, err := config.LoadDefaultConfig(ctx, config.WithRegion("us-east-1"))
 if err != nil {
  fmt.Fprintf(os.Stderr, "Failed to load config: %v\n", err)
  os.Exit(1)
 }

 client := polly.NewFromConfig(cfg)

 output, err := client.StartSpeechSynthesisStream(ctx,
 &polly.StartSpeechSynthesisStreamInput{
  Engine:       types.EngineGenerative,
  VoiceId:      types.VoiceIdTiffany,
  OutputFormat: types.OutputFormatMp3,
  SampleRate:   aws.String("24000"),
 })
 if err != nil {
  fmt.Fprintf(os.Stderr, "Failed to start stream: %v\n", err)
  os.Exit(1)
 }

 stream := output.GetStream()
```

```
defer stream.Close()

audioFile, err := os.Create("output.mp3")
if err != nil {
 fmt.Fprintf(os.Stderr, "Failed to create output file: %v\n", err)
 os.Exit(1)
}
defer audioFile.Close()

textChunks := []string{
 "The weather forecast for today shows clear skies ",
 "with temperatures reaching twenty five degrees. ",
 "Tomorrow we expect some cloud cover in the morning ",
 "but it should clear up by the afternoon. ",
 "The rest of the week looks mostly sunny ",
 "with a slight chance of rain on Friday. ",
 "Overall a great week to spend time outdoors.",
}

// Send text events in a goroutine
go func() {
 for _, chunk := range textChunks {
  fmt.Printf("Sending TextEvent: %s\n", strings.TrimSpace(chunk))
  err := stream.Send(ctx,
&types.StartSpeechSynthesisStreamActionStreamMemberTextEvent{
   Value: types.TextEvent{
    Text:     aws.String(chunk),
    TextType: types.TextTypeText,
   },
  })
  if err != nil {
   fmt.Fprintf(os.Stderr, "Failed to send text event: %v\n", err)
   return
  }
  // Simulate delay between chunks (e.g. waiting for LLM tokens)
  time.Sleep(300 * time.Millisecond)
 }

 // Signal end of input
 stream.Send(ctx,
&types.StartSpeechSynthesisStreamActionStreamMemberCloseStreamEvent{
  Value: types.CloseStreamEvent{},
 })
}()
```

```
 // Receive audio events
 for event := range stream.Events() {
  switch v := event.(type) {
  case *types.StartSpeechSynthesisStreamEventStreamMemberAudioEvent:
   fmt.Printf("Received AudioEvent: %d bytes\n", len(v.Value.AudioChunk))
   audioFile.Write(v.Value.AudioChunk)
  case *types.StartSpeechSynthesisStreamEventStreamMemberStreamClosedEvent:
   fmt.Printf("Stream closed. Characters synthesized: %d\n",
 v.Value.RequestCharacters)
  }
 }

 if err := stream.Err(); err != nil {
  fmt.Fprintf(os.Stderr, "Stream error: %v\n", err)
  os.Exit(1)
 }
}
```

Rust

```
use aws_sdk_polly::types::{
    CloseStreamEvent, Engine, OutputFormat,
    StartSpeechSynthesisStreamActionStream,
    StartSpeechSynthesisStreamEventStream, TextEvent, TextType, VoiceId,
};
use aws_sdk_polly::Client;
use std::fs::File;
use std::io::Write;
use tokio::time::{sleep, Duration};

#[tokio::main]
async fn main() {
    let config = aws_config::defaults(aws_config::BehaviorVersion::latest())
        .region("us-east-1")
        .load()
        .await;

    let client = Client::new(&config);

    let text_chunks = vec![
        "The weather forecast for today shows clear skies ",
        "with temperatures reaching twenty five degrees. ",
```

```
        "Tomorrow we expect some cloud cover in the morning ",
        "but it should clear up by the afternoon. ",
        "The rest of the week looks mostly sunny ",
        "with a slight chance of rain on Friday. ",
        "Overall a great week to spend time outdoors.",
    ];

    let input_stream = async_stream::stream! {
        for chunk in &text_chunks {
            println!("Sending TextEvent: {}", chunk.trim());
            yield Ok(StartSpeechSynthesisStreamActionStream::TextEvent(

TextEvent::builder().text(*chunk).text_type(TextType::Text).build().unwrap(),
            ));
            // Simulate delay between chunks (e.g. waiting for LLM tokens)
            sleep(Duration::from_millis(300)).await;
        }
        yield Ok(StartSpeechSynthesisStreamActionStream::CloseStreamEvent(
            CloseStreamEvent::builder().build(),
        ));
    };

    let mut output = client
        .start_speech_synthesis_stream()
        .engine(Engine::Generative)
        .voice_id(VoiceId::Tiffany)
        .output_format(OutputFormat::Mp3)
        .sample_rate("24000")
        .action_stream(input_stream.into())
        .send()
        .await
        .expect("Failed to start stream");

    let mut audio_file = File::create("output.mp3").expect("Failed to create output
file");

    while let Ok(Some(event)) = output.event_stream.recv().await {
        match event {
            StartSpeechSynthesisStreamEventStream::AudioEvent(audio_event) => {
                if let Some(chunk) = audio_event.audio_chunk() {
                    let bytes = chunk.as_ref();
                    println!("Received AudioEvent: {} bytes", bytes.len());
                    audio_file.write_all(bytes).unwrap();
                }
```

```
            }
            StartSpeechSynthesisStreamEventStream::StreamClosedEvent(closed_event)
 => {
                println!(
                    "Stream closed. Characters synthesized: {}",
                    closed_event.request_characters()
                );
            }
            _ => {}
        }
    }
}
```

Vozes no Amazon Polly

O Amazon Polly oferece dezenas de vozes realistas e compatíveis em vários idiomas. Cada voz é criada usando falantes nativos, portanto, há variações entre as vozes, ainda que no mesmo idioma. Você também pode usar o Console de gerenciamento da AWS para testar cada voz com o texto de sua escolha. Na maioria dos casos, haverá pelo menos uma voz masculina e uma feminina, muitas vezes mais de uma de cada. Alguns poucos idiomas têm apenas uma voz.

O inventário de vozes e o número de idiomas incluído estão sempre sendo atualizados para incluir opções adicionais. Para sugerir um novo idioma ou voz, deixe um comentário nesta página. Não temos informações sobre o lançamento de novos idiomas neste momento.

> **Note**
>
> Para ouvir exemplos de vozes do Amazon Polly no navegador, consulte a visão geral de produtos do Amazon Polly.

Tópicos

- Vozes disponíveis
- Vozes bilíngues
- Aplicar a voz de âncora
- Ouvir as vozes
- Sincronizar a velocidade da voz
- Modificar a velocidade da voz

Vozes disponíveis

O Amazon Polly fornece uma variedade de vozes realistas em vários idiomas para usar ao sintetizar fala com base em texto. A tabela a seguir mostra todas as vozes que o Amazon Polly oferece.

	Idiomas e variantes linguísticas	Código do idioma	Nome/ ID	Gender	Voz generativa	Voz de forma longa	Voz neural	Voz padrão
1	Árabe	arb	Zeina	Feminino	Não	Não	Não	Sim
2	Árabe (Golfo)	ar-AE	Hala*	Feminino	Não	Não	Sim	Não
			Zayd*	Masculino	Não	Não	Sim	Não
3	Holandês (belga)	nl-BE	Lisa	Feminino	Sim	Não	Sim	Não
4	Catalão	ca-ES	Arlet	Feminino	Não	Não	Sim	Não
5	Tcheco	cs-CZ	Jitka	Feminino	Não	Não	Sim	Não
6	Chinês (cantonês)	yue-CN	Hiujin	Feminino	Não	Não	Sim	Não
7	Chinês (mandarin)	cmn-CN	Zhiyu	Feminino	Não	Não	Sim	Sim
8	Dinamarquês	da-DK	Naja	Feminino	Não	Não	Não	Sim
			Mads	Masculino	Não	Não	Não	Sim
			Sofie	Feminino	Não	Não	Sim	Não
9	Holandês	nl-NL	Laura	Feminino	Sim	Não	Sim	Não
			Lotte	Feminino	Não	Não	Não	Sim
			Ruben	Masculino	Não	Não	Não	Sim

	Idiomas e variantes linguísticas	Código do idioma	Nome/ ID	Gender	Voz generativa	Voz de forma longa	Voz neural	Voz padrão
10	Inglês (australiano)	en-AU	Nicole	Feminino	Não	Não	Não	Sim
			Olivia	Feminino	Sim	Não	Sim	Não
			Russell	Masculino	Não	Não	Não	Sim
11	Inglês (britânico)	en-GB	Amy	Feminino	Sim	Não	Sim	Sim
			Emma	Feminino	Não	Não	Sim	Sim
			Brian	Masculino	Sim	Não	Sim	Sim
			Arthur	Masculino	Não	Não	Sim	Não
12	Inglês (indiano)	en-IN	Aditi*	Feminino	Não	Não	Não	Sim
			Raveena	Feminino	Não	Não	Não	Sim
			Kajal*	Feminino	Sim	Não	Sim	Não
13	Inglês (Irlanda)	en-IE	Niamh	Feminino	Sim	Não	Sim	Não
14	Inglês (neozelandês)	en-NZ	Aria	Feminino	Sim	Não	Sim	Não
15	Inglês (singapurense)	en-SG	Jasmine	Feminino	Sim	Não	Sim	Não
16	Inglês (sul-africano)	en-ZA	Ayanda	Feminino	Sim	Não	Sim	Não

	Idiomas e variantes linguísticas	Código do idioma	Nome/ ID	Gender	Voz generativa	Voz de forma longa	Voz neural	Voz padrão
17	Inglês (EUA)	en-US	Danielle	Feminino	Sim	Sim	Sim	Não
			Gregory	Masculino	Não	Sim	Sim	Não
			Ivy	Feminino (criança)	Não	Não	Sim	Sim
			Joanna**		Sim	Não	Sim	Sim
			Kendra	Feminino	Não	Não	Sim	Sim
			Kimberly	Feminino	Não	Não	Sim	Sim
			Salli	Feminino	Sim	Não	Sim	Sim
			Joey	Feminino	Não	Não	Sim	Sim
			Justin	Masculino (criança)	Não	Não	Sim	Não
			Kevin		Não	Não	Sim	Sim
			Matthew**	Masculino	Sim	Não	Sim	Não
			Ruth		Sim	Sim	Sim	Não
			Stephen	Masculino	Sim	Não	Sim	Não
			Tiffany	Feminino	Sim	Não	Não	Não
			Patrick	Masculino	Não	Sim	Não	Não
				Feminino				
				Masculino				

	Idiomas e variantes linguísticas	Código do idioma	Nome/ ID	Gender	Voz generativa	Voz de forma longa	Voz neural	Voz padrão
18	Inglês (País de Gales)	en-GB-WLS	Geraint	Masculino	Não	Não	Não	Sim
19	Finlandês	fi-FI	Suvi	Feminino	Não	Não	Sim	Não
20	Francês	fr-FR	Ambre	Feminino	Sim	Não	Não	Não
			Céline/ Celine	Feminino	Sim	Não	Não	Sim
			Florian	Masculino	Sim	Não	Não	Não
			Léa	Feminino	Sim	Não	Sim	Sim
			Mathieu	Masculino	Não	Não	Não	Sim
			Rémi	Masculino	Sim	Não	Sim	Não
21	Francês (belga)	fr-BE	Isabelle	Feminino	Sim	Não	Sim	Não
22	Francês (canadense)	fr-CA	Chantal	Feminino	Não	Não	Não	Sim
			Gabrielle	Feminino	Sim	Não	Sim	Não
			Liam	Masculino	Sim	Não	Sim	Não

	Idiomas e variantes linguísticas	Código do idioma	Nome/ ID	Gender	Voz generativa	Voz de forma longa	Voz neural	Voz padrão
23	Alemão	de-DE	Marlene	Feminino	Não	Não	Não	Sim
			Vicki	Feminino	Sim	Não	Sim	Sim
			Hans	Masculino	Não	Não	Não	Sim
			Daniel	Masculino	Sim	Não	Sim	Não
			Lennart	Masculino	Sim	Não	Não	Não
24	Alemão (austríaco)	de-AT	Hannah	Feminino	Sim	Não	Sim	Não
25	Alemão (suíço)	de-CH	Sabrina	Feminino	Sim	Não	Sim	Não
26	Hindi	hi-IN	Aditi*	Feminino	Não	Não	Não	Sim
			Kajal*	Feminino	Não	Não	Sim	Não
27	Islandês	is-IS	Dóra/ Dora	Feminino	Não	Não	Não	Sim
			Karl	Masculino	Não	Não	Não	Sim

	Idiomas e variantes linguísticas	Código do idioma	Nome/ ID	Gender	Voz generativa	Voz de forma longa	Voz neural	Voz padrão
28	Italiano	it-IT	Beatriz	Feminino	Sim	Não	Não	Não
			Carla	Feminino	Não	Não	Não	Sim
			Bianca	Feminino	Sim	Não	Sim	Sim
			Lorenzo	Masculino	Sim	Não	Não	Não
			Giorgio	Masculino	Não	Não	Não	Sim
			Adriano	Masculino	Não	Não	Sim	Não
29	Japonês	ja-JP	Mizuki	Feminino	Não	Não	Não	Sim
			Takumi	Masculino	Não	Não	Sim	Sim
			Kazuha	Feminino	Não	Não	Sim	Não
			Tomoko	Feminino	Não	Não	Sim	Não
30	Coreano	ko-KR	Seoyeon	Feminino	Sim	Não	Sim	Sim
			Jihye	Feminino	Não	Não	Sim	Não
31	Norueguê	nb-NO	Liv	Feminino	Não	Não	Não	Sim
			Ida	Feminino	Não	Não	Sim	Não

	Idiomas e variantes linguísticas	Código do idioma	Nome/ ID	Gender	Voz generativa	Voz de forma longa	Voz neural	Voz padrão
32	Polonês	pl-PL	Ewa	Feminino	Sim	Não	Não	Sim
			Maja	Feminino	Não	Não	Não	Sim
			Jacek	Masculino	Não	Não	Não	Sim
			Jan	Masculino	Não	Não	Não	Sim
			Ola	Feminino	Sim	Não	Sim	Não
33	Português (brasileiro)	pt-BR	Camila	Feminino	Sim	Não	Sim	Sim
			Vitória/ Vitoria	Feminino	Não	Não	Sim	Sim
			Ricardo	Masculino	Não	Não	Não	Sim
			Thiago	Masculino	Não	Não	Sim	Não
34	Português (europeu)	pt-PT	Inês/ Ines	Feminino	Não	Não	Sim	Sim
			Cristiano	Masculino	Não	Não	Não	Sim
35	Romeno	ro-RO	Carmen	Feminino	Não	Não	Não	Sim
36	Russo	ru-RU	Tatyana	Feminino	Não	Não	Não	Sim
			Maxim	Masculino	Não	Não	Não	Sim

	Idiomas e variantes linguísticas	Código do idioma	Nome/ ID	Gender	Voz generativa	Voz de forma longa	Voz neural	Voz padrão
37	Espanhol (Espanha)	es-ES	Conchita	Feminino	Não	Não	Não	Sim
			Lúcia	Feminino	Sim	Não	Sim	Sim
			Alba	Feminino	Não	Sim	Não	Não
			Enrique	Masculino	Não	Não	Não	Sim
			Sergio	Masculino	Sim	Não	Sim	Não
			Raúl	Masculino	Não	Sim	Não	Não
38	Espanhol (mexicano)	es-MX	Mia	Feminino	Sim	Não	Sim	Sim
			Andrés	Masculino	Sim	Não	Sim	Não
39	Espanhol (EUA)	es-US	Lupe**	Feminino	Sim	Não	Sim	Sim
			Penélope/ Penelope	Feminino	Não	Não	Não	Sim
			Miguel	Masculino	Não	Não	Não	Sim
			Pedro	Masculino	Sim	Não	Sim	Não
40	Sueco	sv-SE	Astrid	Feminino	Não	Não	Não	Sim
			Elin	Feminino	Não	Não	Sim	Não
41	Turco	tr-TR	Filiz	Feminino	Não	Não	Não	Sim
			Burcu	Feminino	Não	Não	Sim	Não
42	Galês	cy-GB	Gwyneth	Feminino	Não	Não	Não	Sim

*Essa voz é bilíngue. Para obter mais informações, consulte Vozes bilíngues.

**Essas vozes podem ser usadas com o estilo de locutor quando utilizadas no formato neural. Para obter mais informações, consulte Aplicar a voz de âncora.

Cada mecanismo de voz do Amazon Polly tem recursos exclusivos. Saiba mais sobre os recursos e a disponibilidade regional dos mecanismos de voz oferecidos pelo Amazon Polly:

- Vozes generativas
- Vozes de forma longa
- Vozes neurais
- Vozes padrão

Vozes de marca

Além das vozes disponíveis listadas na tabela anterior, você pode usar o Amazon Polly para criar uma voz personalizada de acordo com a identidade da sua marca. Com uma voz de marca, você pode oferecer vozes únicas e exclusivas aos seus clientes. Para saber mais sobre as vozes de marca do Amazon Polly, consulte Brand Voice.

Vozes bilíngues

O Amazon Polly tem duas formas de produzir vozes bilíngues:

- Vozes bilíngues com sotaque
- Vozes totalmente bilíngues

Vozes bilíngues com sotaque

Vozes bilíngues com sotaque podem ser criadas usando qualquer voz do Amazon Polly, mas somente ao usar tags SSML.

Normalmente, todas as palavras no texto de entrada são faladas no idioma padrão da voz especificada que você está usando.

Por exemplo, se você estiver usando a voz de Joanna (que fala inglês dos EUA), o Amazon Polly falará a seguinte frase na voz da Joanna, sem sotaque francês:

```
<speak>
     Why didn't she just say, 'Je ne parle pas français?'
</speak>
```

Neste caso, as palavras Je ne parle pas français são ditas como seriam se fossem em inglês.

No entanto, se você usar a voz da Joanna com a tag **<lang>**, o Amazon Polly falará a frase com a voz da Joanna em francês com sotaque americano:

```
<speak>
     Why didn't she just say, <lang xml:lang="fr-FR">'Je ne parle pas français?'</
lang>.
</speak>
```

Como Joanna não é uma voz nativa do Francês, a pronúncia se baseia no idioma nativo dela, que é o Inglês dos EUA. Por exemplo, apesar da pronúncia perfeita em Francês ter uma vibrante uvular /R/ na palavra français, a voz em inglês dos EUA da Joanna pronuncia esse fonema como o som /r/ correspondente.

Se você usar a voz de Giorgio, que fala italiano, com o seguinte texto, o Amazon Polly falará a frase na voz de Giorgio com uma pronúncia italiana:

```
<speak>
     Mi piace Bruce Springsteen.
</speak>
```

Vozes totalmente bilíngues

Uma voz totalmente bilíngue como Aditi (inglês indiano e hindi) pode falar fluentemente dois idiomas. Isso oferece a capacidade de usar palavras e frases de ambos os idiomas em um único texto usando a mesma voz.

Atualmente, Aditi, Kajal, Hala e Zayd são as únicas vozes totalmente bilíngues disponíveis.

Como usar uma voz bilíngue (por exemplo, Aditi)

Aditi fala ambos os idiomas inglês indiano (en-IN) e hindi (hi-IN) com fluência. Você pode sintetizar fala em inglês e hindi, e a voz pode alternar entre os dois idiomas, mesmo dentro da mesma frase.

O hindi pode ser usado em duas formas diferentes:

- Devanagari: "उसेन कहाँ, खेल तोह अब शुर होगा"
- Romanagari (usando o alfabeto latino): "Usne kahan, khel toh ab shuru hoga"

Além disso, é possível mistura inglês e hindi de uma ou ambas as formas em uma única frase:

- Devanagari + inglês: "This is the song कभी कभी अदिति"
- Romanagari + inglês: "This is the song from the movie Jaane Tu Ya Jaane Na."
- Devanagari + Romanagari + inglês: "This is the song कभी कभी अदिति from the movie Jaane Tu Ya Jaane Na."

ComoAditi é uma voz bilíngue, o texto em todos esses casos será lido corretamente, já que o Amazon Polly consegue diferenciar entre os idiomas e scripts.

O Amazon Polly também é compatível com números, datas, horas e expansão de moedas em inglês (algarismos árabes) e hindi (algarismos Devanagari). Por padrão, os algarismos árabes são lidos em inglês indiano. Para fazer com que o Amazon Polly os leia em hindi, é necessário usar o parâmetro de código de idioma `hi-IN`.

Aplicar a voz de âncora

As pessoas usam estilos de fala diferentes, dependendo do contexto. Uma conversa informal, por exemplo, parece muito diferente de um noticiário de TV ou de rádio. No entanto, devido à forma como essas vozes são feitas, elas não podem produzir diferentes estilos de fala. No entanto, vozes neurais podem. Elas podem ser treinadas para um estilo de locução específico, com variações e ênfases em determinadas partes da fala inerentes a esse estilo.

Além das vozes neurais padrão, o Amazon Polly fornece um estilo de fala de locutor que usa o sistema neural para gerar fala no estilo de um apresentador de TV ou rádio. O estilo de locutor está disponível com as vozes de Matthew e Joanna, em inglês dos EUA (en-US), a voz de Lupe, em espanhol dos EUA (es-US) e a voz de Amy em inglês britânico (en-GB).

Para usar o estilo de locutor, primeiro escolha o mecanismo neural e depois use a sintaxe descrita nas etapas a seguir no texto de entrada.

> **Note**
>
> - Para usar qualquer estilo de fala neural, é necessário usar uma das Regiões AWS que oferecem suporte a vozes neurais. Essa opção não está disponível em todas as regiões. Para obter mais informações, consulte Compatibilidade de recursos e regiões.

Console

Como aplicar o estilo de âncora

1. Abra o console do Amazon Polly em https://console.aws.amazon.com/polly/.
2. Verifique se você está usando uma Região AWS onde vozes neurais são compatíveis.
3. Na página Conversão de texto em fala, em Mecanismo, escolha Neural.
4. Escolha o idioma e a voz que deseja usar. Apenas Matthew e Joanna para inglês dos EUA (en-US), Lupe para espanhol dos EUA (es-US) e Amy para inglês britânico (en-GB) estão disponíveis para a voz de locutor.
5. Ative o SSML.
6. Adicione o texto de entrada à solicitação de conversão de texto em fala usando a sintaxe SSML no estilo de locutor.

```
<amazon:domain name="news">text</amazon:domain>
```

Por exemplo, você pode usar a tag de locutor da seguinte forma:

```
<speak>
<amazon:domain name="news">
From the Tuesday, April 16th, 1912 edition of The Guardian newspaper:

The maiden voyage of the White Star liner Titanic, the largest ship ever
 launched
ended in disaster.

The Titanic started her trip from Southampton for New York on Wednesday. Late
 on
Sunday night she struck an iceberg off the Grand Banks of Newfoundland. By
wireless telegraphy she sent out signals of distress, and several liners were
near enough to catch and respond to the call.
```

```
</amazon:domain>
</speak>
```

7. Escolha Ouvir.

AWS CLI

Como aplicar o estilo de âncora

1. Na solicitação de API, inclua o parâmetro do mecanismo com o valor `neural`:

```
  --engine neural
```

2. Adicione o texto de entrada à solicitação de API usando a sintaxe SSML no estilo de locutor.

```
<amazon:domain name="news">text</amazon:domain>
```

Por exemplo, você pode usar a tag de locutor da seguinte forma:

```
<speak>
<amazon:domain name="news">
From the Tuesday, April 16th, 1912 edition of The Guardian newspaper:

The maiden voyage of the White Star liner Titanic, the largest ship ever
 launched
ended in disaster.

The Titanic started her trip from Southampton for New York on Wednesday. Late
 on
Sunday night she struck an iceberg off the Grand Banks of Newfoundland. By
wireless telegraphy she sent out signals of distress, and several liners were
near enough to catch and respond to the call.
</amazon:domain>
</speak>
```

Para obter mais informações sobre SSML, consulte Etiquetas SSML compatíveis.

Ouvir as vozes

Depois de configurar o Amazon Polly, você pode testar vozes usando texto personalizado no console.

Como ouvir as vozes do Amazon Polly no console

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Polly em https://console.aws.amazon.com/polly/.
2. Selecione a guia Text-to-Speech.
3. Para Mecanismo, escolha Generativo, Forma longa, Neural ou Padrão.
4. Selecione um idioma e uma região. Depois, escolha uma voz.
5. Insira o texto a ser falado pela voz ou use a frase padrão. Em seguida, escolha Ouvir.

Sincronizar a velocidade da voz

Devido à variação natural entre as vozes, cada voz disponível fala em velocidades ligeiramente diferentes. Por exemplo, no caso das vozes em inglês dos EUA, Ivy e Joanna são um pouco mais rápidas do que Matthew e consideravelmente mais rápidas que Joey. Como há muitas variações entre as vozes, não há velocidade padrão (palavras por minuto) disponível para as vozes do Amazon Polly. No entanto, você pode descobrir o tempo necessário para a voz dizer o texto selecionado usando as marcações de fala.

Como cronometrar a duração de uma passagem de texto falada

1. Abra a AWS CLI.
2. Execute o código a seguir, preenchendo conforme o necessário.

```
aws polly synthesize-speech \
    --language-code optional language code if needed
    --output-format json \
    --voice-id [name of desired voice] \
    --text '[desired text]' \
    --speech-mark-types='["viseme"]' \
    LengthOfText.txt
```

3. Abra o `LengthOfText.txt`.

Se o texto fosse "Mary had a little lamb", as últimas linhas retornadas pelo Amazon Polly seriam:

```
     {"time":882,"type":"viseme","value":"t"}
     {"time":964,"type":"viseme","value":"a"}
     {"time":1082,"type":"viseme","value":"p"}
```

O último visema, essencialmente o som para as últimas letras em "lamb", começa 1.082 milissegundos após o início da fala. Embora essa não seja exatamente a duração do áudio, é próxima e pode servir como base de comparação entre vozes.

Modificar a velocidade da voz

Para determinados aplicativos, você pode preferir a voz escolhida com mais ou menos velocidade. Se a velocidade da voz for uma preocupação, o Amazon Polly oferece a capacidade de modificar isso usando tags SSML. Por exemplo, se sua organização está criando uma aplicação que lê livros para imigrantes, talvez você queira variar a velocidade da voz. O público pode até falar inglês, mas a fluência é limitada. O Amazon Polly ajuda você a diminuir a velocidade da fala usando a etiqueta SSML **<**prosody**>**.

Você pode usar uma porcentagem:

```
<speak>
     In some cases, it might help your audience to <prosody rate="85%">slow
     the speaking rate slightly to aid in comprehension.</prosody>
</speak>
```

Ou uma velocidade predefinida:

```
<speak>
     In some cases, it might help your audience to <prosody rate="slow">slow
     the speaking rate slightly to aid in comprehension.</prosody>
</speak>
```

Duas opções de velocidade estão disponíveis ao usar SSML com o Amazon Polly:

- Velocidades predefinidas: `x-slow`, `slow`, `medium`, `fast` e `x-fast`. Nesses casos, a velocidade de cada opção é aproximada, dependendo da sua voz de preferência. A opção `medium` é a velocidade normal da voz.

- n% da velocidade de fala: pode ser usada qualquer porcentagem da velocidade de fala, entre 20% e 200%. Nesses casos, você pode escolher exatamente a velocidade desejada. No entanto, a velocidade real da voz é aproximada, dependendo da voz escolhida. A velocidade normal da voz é 100%.

> **Note**
>
> Teste a voz selecionada em várias velocidades. A velocidade de cada opção é aproximada e depende da voz escolhida.

Para ter mais informações sobre como usar a etiqueta `prosody`, consulte Controlar o volume, a velocidade e o tom da fala.

Idiomas no Amazon Polly

Os seguintes idiomas são compatíveis com o Amazon Polly e podem ser usados para sintetizar a fala. Cada idioma tem um código de idioma exclusivo. Esses códigos de idioma são etiquetas de identificação de idioma W3C (*ISO 639-3* para o nome do idioma e *ISO 3166* para o código do país).

Selecione um idioma na tabela a seguir para conferir detalhes sobre os fonemas e visemas fornecidos pela Amazon Polly.

Idioma	Código do idioma
Árabe	arb
Árabe (Golfo)	Ar-ae
Catalã	ca-ES
Chinês (cantor)	yue-CN
Chinês (manda)	cmn-CN
Tchecc	cs-CZ
Dinamê	da-DK
Holand (belga)	nl-BE

Idioma	Código do idioma
Holand	nl-NL
Inglês (austra ano)	en-AU
Inglês (britâni o)	en-GB
Inglês (indian	en-IN
Inglês (neoze dês)	en-NZ
Inglês (singap ense)	en-SG
Inglês (sul-afri cano)	en-ZA
Inglês (EUA)	en-US
Inglês (País de Gales)	en-GB-WLS
Finlanc	fi-FI

Idioma	Código do idioma
francês	fr-FR
Francê (belga)	fr-Be
Francê (canad e)	fr-CA
Hindi	hi-IN
Alemãc	de-DE
Alemãc (austría o)	de-AT
Alemãc (padrãc suíço)	de-CH
Islandê	is-IS
Italianc	it-IT
Japonê	ja-JP
Corear	ko-KR
Norueç	nb-NO
Polonê	pl-PL

Idioma	Código do idioma
Portugu (brasile ro)	pt-BR
Portugu (europe	pt-PT
Romen	ro-RO
Russo	ru-RU
Espanh (Espan	es-ES
Espanh (mexica)	es-MX
Espanh (EUA)	es-US
Sueco	sv-SE
Turco	tr-TR
galês	cy-GB

Árabe (arb)

A tabela a seguir lista os fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para a voz árabe da Zeina, que é compatível com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
ʔ	?	oclusiva glotal	أنا	
ʕ	?\	fricativa faríngea sonora	عُمَر	k
b	b	oclusiva bilabial sonora	بَلَد	p
d	d	oclusiva alveolar sonora	داري	t
dˤ	d_?\	oclusiva alveolar sonora enfática	ضَوء	t
ʤ	dZ	africada pós-alveo lar sonora	جَميل	S
ð	D	fricativa dental sonora	ذلِكَ	T
ðˤ	D_?\	fricativa dental sonora enfática	ظَلام	T
f	f	fricativa labiodental surda	فَصل	f
g	g	oclusiva velar sonora	إنجلترا	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɣ	G	fricativa velar sonora	برغَ	k
h	h	fricativa glotal surda	اذه	k
j	j	aproximante palatal	يشمَيَ	i
k	k	oclusiva velar surda	بلَكَ	k
l	l	aproximante lateral alveolar	ىقال	t
lˠ	l_G	aproximante lateral alveolar enfática	هلدابع	t
m	m	nasal bilabial	اذام	p
n	n	nasal alveolar	رون	t
p	p	oclusiva bilabial surda	سبَحَ	p
q	q	oclusiva uvular surda	بيرَقَ	k
r	r	vibrante alveolar	رَمل	r
s	s	fricativa alveolar surda	سُؤال	s
sˤ	s_?\	fricativa alveolar surda enfática	بحاص	s
ʃ	S	fricativa pós-alveo lar surda	رشُك	S

IPA	X-SAMPA	Descrição	Exemplo	Visema
t	t	oclusiva alveolar surda	تَمر	t
tˤ	t_?\	oclusiva alveolar surda enfática	طالِب	t
θ	T	fricativa dental surda	ثَلاث	T
v	v	fricativa labiodental sonora	فيتامين	f
w	w	Aproximante labiovelar	وَلَد	u
x	x	fricativa velar surda	خَوْف	k
ħ	X\	fricativa faríngea surda	حَوْلَ	k
z	z	fricativa alveolar sonora	زُهور	s
Vogais				
a	a	vogal anterior aberta não arredondada	بَرد	a
a:	a:	vogal anterior aberta longa não arredondada	دار	a
ɑˤ	A_?\	vogal posterior aberta enfática não arredondada	طَبل	a

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɑˤː	A_?\:	vogal posterior aberta longa enfática não arredondada	ظالِم	a
u	u	vogal posterior fechada arredondada	شُرب	u
uː	u:	vogal posterior fechada longa arredondada	سور	u
uˤ	u_?\	vogal posterior fechada enfática arredondada	بُدّ	u
uˤː	u_?\:	vogal posterior fechada longa enfática arredondada	طول	u
i	i	vogal anterior fechada não arredondada	بِنت	i
iː	i:	vogal anterior fechada longa não arredondada	حَزين	i
iˤ	i_?\	vogal anterior fechada enfática não arredondada	ضِدّ	i

IPA	X-SAMPA	Descrição	Exemplo	Visema
iˤː	i_?\:	vogal anterior fechada longa enfática não arredondada	يضام	i
p	p	vogal anterior semifechada não arredondada	تكرام	p
eː	p:	vogal anterior média fechada longa não arredondada	ليدوم	p
ɔ	U	vogal posterior média aberta arredondada	يجولونكت	U
ɔː	U:	vogal posterior média aberta longa arredondada	نويزفيلت	U

Árabe do Golfo (ar-AE)

A tabela a seguir lista os fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para a voz árabe da Hala que é compatível com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Pronúncia	Visema
Consoantes					
b	b	oclusiva bilabial sonora	دلب	/ " b a . l a d /	b

IPA	X-SAMPA	Descrição	Exemplo	Pronúncia	Visema
d	d	oclusiva alveolar sonora	رد	/ " r a d d /	d
dˤ	d_?\	plosivo alveolar sonora faringealizado	ضوء	/ " d_?\ a w ? /	D
f	f	fricativa labiodental surda	نرف	/ " f I . r I n /	f
g	g	oclusiva velar sonora	لاق	/ " g a: l /	k
j	j	aproximante palatina sonora	يشميي	/ " j I m . S i: /	i
k	k	oclusiva velar surda	لماك	/ " k a: . m i l /	k
l	l	aproximante lateral alveolar sonora	ليل	/ " l e: l /	t
lˤ	l_G	aproximante lateral alveolar sonora faringealizado	هلدابع	/ ?\ a b . " d A_?\ l_G . l_G A_?\ /	t
m	m	parada nasal bilabial	ةئم	/ " m I j . j a /	p
n	n	parada nasal alveolar	رون	/ " n u: r /	t
p	p	oclusiva bilabial surda	ارأوب	/ " ? O . p e . r a: /	p

IPA	X-SAMPA	Descrição	Exemplo	Pronúncia	Visema
q	q	oclusiva uvular surda	رصق	/ " q A_?\ s_?\ r /	k
r	r	vibrante alveolar	لمر	/ " r a . m l l /	r
s	s	fricativa alveolar surda	مسمس	/ " s l m . s l m /	s
sˤ	s_?\	fricativa alveolar faringealizada sem voz	بحاص	/ " s_?\ A_?: . X \ l b /	s
t	t	oclusiva alveolar surda	رمت	/ "t a . m a r /	t
tˤ	t_?\	fricativa alveolar faringealizada sem voz	بلاط	/ " t_?\ A_?: . l l b /	t
v	v	fricativa labiodental sonora	نيماتيڤ	/ v i: . t A . " m i: n /	f
w	w	aproximante labiovelar sonora	دياو	/ " w a: . j l d /	u
x	x	fricativa velar surda	فوروخ	/ x a . " r u: f /	k
z	z	fricativa velar surda	روهز	/ " z h u: r /	s

IPA	X-SAMPA	Descrição	Exemplo	Pronúncia	Visema
ð	D	fricativa interdental sonora	كلذ	/ " D a: . l l k /	D
ðˤ	D_?\	fricativa interdent al sonora faringealizada	ظلام	/ D_?\ A_?\ . " l a: m /	D
ħ	X\	fricativa faríngea surda	نيجلا	/ ? a l . " X\ i: n /	k
ŋ	N	parada nasal velar	غنوكغنوه	/ h O N . " k O N g /	k
ɣ	G	fricativa velar sonora	ةبيرغ	/ G I . " r i: . b a /	k
ʃ	S	fricativa pós-alveolar surda	سمش	/ " S a m s /	S
ʒ	Z	fricativa pós-alveolar sonora	تيكاج	/ Z a . " k e: t /	S
ʔ	?	oclusiva glotal	ةسسؤم	/ m u . " ? a s . s a . s a /	
ʕ	?\	fricativa faríngea sonora	عام	/ " ?\ a: m m /	k
ʤ	dZ	africada pós-alveolar sonora	ةعماج	/ " dZ a: m . ?\ a /	S
θ	T	fricativa interdental sonora	ةثالث	/ T a . " l a: . T a /	T

IPA	X-SAMPA	Descrição	Exemplo	Pronúncia	Visema
ɦ	h	fricativa glotal sonora	هلال	/ " h l a: l /	k
Vogais					
æ	a	vogal anterior média aberta longa não arredondada	سفر	/ " s a . f a r /	a
ɑˤ	A_?\	vogal curta faringealizada, aberta, não arredondada	صلب	/ " s_?\ A_?\ l b /	a
æː	a:	vogal anterior média aberta longa não arredondada	باب	/ " b a: b /	a
ɑˤː	A_?\:	vogal posterior aberta longa enfática não arredondada	ضمان	/ " n A_?: . D_? \ i_?\ dZ /	a
a	A	vogal aberta central não arredondada	wifi	/ " w A j . f A j /	a
i	i	vogal curta não arredondada frontal tensa (MSA)	إسحاق	/ ? i s . " X\ A_? \: q /	i
ɪ	eu	vogal anterior fechada não arredondada	بنت	/ " b I n t /	i

IPA	X-SAMPA	Descrição	Exemplo	Pronúncia	Visema
iˤ	i_?\	vogal curta não arredondada frontal faringeal izada	لطف	/ " t_?\ i_?\ f l /	i
iː	iː	vogal anterior fechada não arredondada	ليبس	/ s a . " b iː l /	i
iˤː	i_?:	vogal anterior fechada longa enfática não arredondada	بيطر	/ r A_?\ . " t_?\ i_?: b /	i
u	u	vogal curta arredondada (MSA) tensa fechada para trás	عرتخم	/ " m u x . t a . r i ?\ /	u
ʊ	U	vogal posterior fechada arredondada	موسر	/ r U . " s uː m /	u
uˤ	u_?\	vogal curta arredondada faringealizada com dorso fechado	عوفصر	/ ?\ u_?\ s_?\ . " f uː r /	u
uː	uː	vogal posterior fechada longa arredondada	توت	/ " t uː t /	u

IPA	X-SAMPA	Descrição	Exemplo	Pronúncia	Visema
uˤː	u_?\:	vogal posterior fechada longa enfática arredondada	صور	/ " s_?\ u_?\: r /	u
p	p	vogal curta média anterior não arredondada	إِنْتَرْنِت	/ " s e n t /	p
p:	p:	vogal anterior média longa não arredondada	شاي	/ " ? e: S /	p
ɔ	U	vogal posterior média aberta arredondada	دولار	/ d O . " l A r /	U
ɔː	U:	vogal posterior média aberta longa arredondada	لون	/ " l O: n /	U

Catalão (ca-ES)

A tabela a seguir lista os fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para a voz catalã da Arlet que é compatível com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				

IPA	X-SAMPA	Descrição	Exemplo	Visema
p	p	oclusiva bilabial surda	pato	p
t	t	oclusiva alveolar surda	montanha	t
k	k	oclusiva velar surda	corpo	k
b	b	oclusiva bilabial sonora	bata	p
d	d	oclusiva alveolar sonora	dor	t
g	g	oclusiva velar sonora	galo	k
m	m	sonora nasal bilabial	mato	p
n	n	nasal alveolar sonora	narina	t
ɲ	J	sonora palatal nasal	alumínio	J
ŋ	N	velar nasal sonora	anão	k
ɬ	5	aproximante lateral alveolar velarizada sonora (escuro l)	Brasil	l
ʎ	L	aproximante lateral palatal sonora	ralho	J
r	r	vibrante alveolar sonora	contra	r

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɾ	4	toque alveolar sonora	para	t
f	f	fricativa labiodental surda	fácil	f
s	s	fricativa alveolar surda	osso	s
z	z	fricativa alveolar sonora	Estados Unidos	s
ʃ	S	fricativa pós-alveolar surda	chave	S
ʒ	Z	fricativa pós-alveolar sonora	loja	S
͡tʃ	tS	africada pós-alveolar surda	chamar	S
͡dʒ	dZ	africada pós-alveolar sonora	grande	S
β	B	aproximante bilabial sonora	obert	B
ð	D	aproximante dentária sonora	marketing	T
j	j	aproximante palatina sonora	noia	i
ɣ	G	aproximante velar sonora	pega	k
v	v	fricativa labiodental sonora	afgà	f

IPA	X-SAMPA	Descrição	Exemplo	Visema
w	w	aproximante labiovelar sonora	aigua	u
x	x	fricativa velar surda	Jiménez	k
ʝ	j\	fricativa palatal sonora	yeso	J
l	l	aproximante lateral alveolar sonora	alondra	t
θ	T	fricativa dental surda	González	T
Vogais				
a	a	vogal traseira aberta	casa	a
p	p	vogal anterior semifechada não arredondada	llenya	p
ɛ	E	vogal anterior média aberta não arredondada	xec	E
i	i	vogal anterior fechada não arredondada	visca	i
o	o	vogal posterior semifechada arredondada	gos	o

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɔ	U	vogal posterior média aberta arredondada	joc	U
u	u	vogal posterior fechada arredondada	un	u
ə	@	vogal central média	casa	@
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Chinês (cantonês) (yue-CN)

A tabela a seguir lista os fonemas do Jyutping e do Alfabeto Fonético Internacional (AFI) para a voz do chinês cantonês que é compatível com o Amazon Polly. Jyutping é um sistema de romanização do cantonês comumente usado no meio acadêmico e entre falantes de cantonês. O AFI e o X-SAMPA não são comumente usados, mas estão disponíveis para suporte em inglês. Os símbolos do AFI e do X-SAMPA na tabela são apenas para referência e não devem ser usadas para transcrição do chinês. Os exemplos do Jyutping e os visemas correspondentes também são mostrados.

Para fazer com que o Amazon Polly use a pronúncia fonética com o Jyutping, use a tag `phoneme alphabet="x-amazon-jyutping"`.

Os exemplos a seguir mostram isso com cada padrão.

Jyutping:

```
<speak>
     ## <phoneme alphabet="x-amazon-jyutping" ph="sing2">#</phoneme>#
```

```
     ## <phoneme alphabet="x-amazon-jyutping" ph="seng2">#</phoneme>#
</speak>
```

AFI:

```
<speak>
     ## <phoneme alphabet="ipa" ph="p##k##n">pecan</phoneme>#
     ## <phoneme alphabet="ipa" ph="#pi.kæn">pecan</phoneme>#
</speak>
```

X-SAMPA:

```
<speak>
     ## <phoneme alphabet='x-sampa' ph='pI"kA:n'>pecan</phoneme>#
     ## <phoneme alphabet='x-sampa' ph='"pi.k{n'>pecan</phoneme>#
</speak>
```

> **Note**
>
> O Amazon Polly aceita entradas do cantonês codificadas somente em UTF-8.

Tabela de fonemas/visemas

Jyutping	IPA	X-SAMPA	Descrição	Exemplo de Jyutping	Visema
Consoantes					
b	p	p	oclusiva bilabial surda	巴, baa1	p
c	ts^h	ts_h	africada alveolar surda aspirada	叉, caa1	s
d	t	t	oclusiva alveolar surda	打, daa2	t
f	f	f	fricativa labiodental surda	花, faa1	f
g	k	k	oclusiva velar surda	家, gaa1	k

Jyutping	IPA	X-SAMPA	Descrição	Exemplo de Jyutping	Visema
gw	k^w	k_w	plosiva velar surda labializada	瓜, gwaa1	u
h	h	h	fricativa glotal surda	哈, haa1	k
k	k^h	k_h	aspirated voiceless velar plosive	卡, kaa1	k
kw	k^{wh}	k_wh	plosiva velar surda aspirada	誇, kwaa1	u
l	l	l	aproximante lateral alveolar	啦, laa1	t
m	m	m	nasal bilabial	媽, maa1	p
m	m	m=	nasal bilabial silábica	唔, m4	p
ng	ŋ	N	nasal velar	牙, ngaa4	k
ng	ŋ	N=	velar nasal silábica	吳, ng4	k
n	n	n	nasal alveolar	拿, naa4	t
p	p^h	p_h	oclusiva bilabial surda aspirada	趴, paa1	p
s	s	s	fricativa alveolar surda	沙, saa1	s
t	t^h	t_h	oclusiva alveolar surda aspirada	他, taa1	t
w	w	w	Aproximante labiovelar	娃, waa1	u
y	j	j	aproximante palatal	也, jaa5	i
z	ts	ts	africada alveolar surda	渣, zaa1	s

Jyutping	IPA	X-SAMPA	Descrição	Exemplo de Jyutping	Visema
Vogais					
a	ɐ	6	vogal central quase aberta	吉, gat1	a
aa	ɑ	A	vogal posterior aberta não arredondada	家, gaa1	a
aai	ɑi	AI	ditongo	街, gaai1	a
aau	ɑu	au	ditongo	交, gaau1	a
ai	ɐi	6i	ditongo	雞, gai1	a
au	ɐu	6u	ditongo	溝, kau1	a
p	ɛ	E	vogal anterior média aberta não arredondada	爹, de1	E
ei	ei	ei	ditongo	基, gei1	p
eo	ɵ	8	vogal média central fechada arredondada	春, ceon1	o
eoi	ɵy	8y	ditongo	居, geoi1	o
eu	ɛu	Eu	ditongo	掉 in 掉垃圾, deu6	E
i	i	i	vogal anterior fechada não arredondada	斯, si1	i
i	eu	I	vogal quase anterior quase fechada não arredondada	激, gik1	i
iu	iu	iu	ditongo	驕, giu1	i

Jyutping	IPA	X-SAMPA	Descrição	Exemplo de Jyutping	Visema
o	ɔ	U	vogal posterior média aberta arredondada	哥, go1	U
oe	œ	9	vogal anterior média aberta arredondada	鋸, goe3	U
oi	ɔi	Oi	ditongo	該, goi1	U
ou	ou	ou	ditongo	高, gou1	o
u	u	u	vogal posterior fechada arredondada	姑, gu1	u
u	ʊ	U	vogal quase posterior quase fechada arredondada	谷, guk5	u
ui	ui	ui	ditongo	勾, gui6	u
yu	y	y	vogal anterior fechada arredondada	於, jyu1	u
Marcas de tom e símbolos adicionais					
1			alto nível	詩, si1	
2			aumento médio	史, si2	
3			nível médio	試, si3	
4			nível muito baixo	時, si4	
5			baixa ascensão	市, si5	
6			nível baixo	是, si6	
-	.	.	separação silábica	語音 jyu5-jam1	

Chinês (mandarim) (zh-CN)

A tabela a seguir lista os fonemas do Pinyin e do Alfabeto Fonético Internacional (AFI) para a voz do chinês mandarim que é compatível com o Amazon Polly. O Pinyin é o padrão internacional padrão para romanização do chinês padrão. O AFI e o X-SAMPA não são comumente usados, mas estão disponíveis para suporte em inglês. Os símbolos do AFI e do X-SAMPA na tabela são apenas para referência e não devem ser usadas para transcrição do chinês. Os exemplos do Pinyin e os visemas correspondentes também são mostrados.

Para que o Amazon Polly use a pronúncia fonética com Pinyin, use a tag `phoneme alphabet="x-amazon-`*`phonetic standard used`*`"`.

Os exemplos a seguir mostram isso com cada padrão.

Pinyin:

```
<speak>
     ## <phoneme alphabet="x-amazon-pinyin" ph="bo2">#</phoneme>#
     ## <phoneme alphabet="x-amazon-pinyin" ph="bao2">#</phoneme>#
</speak>
```

AFI:

```
<speak>
     ## <phoneme alphabet="ipa" ph="p##k##n">pecan</phoneme>#
     ## <phoneme alphabet="ipa" ph="#pi.kæn">pecan</phoneme>#
</speak>
```

X-SAMPA:

```
<speak>
     ## <phoneme alphabet='x-sampa' ph='pI"kA:n'>pecan</phoneme>#
     ## <phoneme alphabet='x-sampa' ph='"pi.k{n'>pecan</phoneme>#
</speak>
```

> **Note**
>
> O Amazon Polly aceita apenas entradas em chinês mandarim codificadas em UTF-8. No momento, o padrão de codificação GB 18030 não tem suporte no Amazon Polly.

Tabela de fonemas/visemas

Pinyin	IPA	X-SAMPA	Descrição	Exemplo do Pinyin	Visema
Consoantes					
f	f	f	fricativa labiodental surda	发, fa1	f
h	h	h	fricativa glotal surda	和, he2	k
g	k	k	oclusiva velar surda	古, gu3	k
k	kʰ	k_h	aspirated voiceless velar plosive	苦, ku3	k
l	l	l	aproximante lateral alveolar	拉, la1	t
m	m	m	nasal bilabial	骂, ma4	p
n	n	n	nasal alveolar	那, na4	t
ng	ŋ	N	nasal velar	正, zheng4	k
b	p	p	oclusiva bilabial surda	爸, ba4	p
p	pʰ	p_h	oclusiva bilabial surda aspirada	怕, pa4	p
s	s	s	fricativa alveolar surda	四, si4	s
x	ɕ	s\	fricativa palato-alveolar surda	西, xi1	J
sh	ʂ	s`	fricativa retroflexa surda	是, shi4	S
d	t	t	oclusiva alveolar surda	打, da3	t

Pinyin	IPA	X-SAMPA	Descrição	Exemplo do Pinyin	Visema
t	tʰ	t_h	oclusiva alveolar surda aspirada	他, ta1	t
zh	͡tʂ	t`s`	africada retroflexa surda	之, zhi1	S
ch	͡tʂʰ	t`s`_h	africada retroflexa aspirada surda	吃, chi1	S
s	͡ts	ts	africada alveolar surda	字, zi4	s
j	͡tɕ	ts\	africada palato-alveolar surda	鸡, ji1	J
q	͡tɕʰ	ts_h	africada palato-alveolar surda aspirada	七, qi1	J
c	͡tsʰ	ts_h	africada alveolar surda aspirada	次, ci4	s
w	w	w	Aproximante labiovelar	我, wo3	u
r	ʐ	z`	fricativa retroflexa sonora	日, ri4	S
"sílabas coloridas "er" e "r					
er	ɚ	@`	vogal média central colorida pelo r	二, er4	@
-r			sílaba colorida r	馅儿, xianr4	@
Vogais					
p	ɤ	7	vogal posterior semifechada não arredondada	恶, e4	p

Pinyin	IPA	X-SAMPA	Descrição	Exemplo do Pinyin	Visema
p	ə	@	vogal central média	恩, en1	@
a	a	a	vogal anterior aberta não arredondada	安, an1	a
ai	aɪ	aI	ditongo	爱, ai4	a
ao	aʊ	aU	ditongo	奥, ao4	a
ei	eɪ	p	ditongo	诶, ei4	p
p	ɛ	E	vogal anterior média aberta não arredondada	姐, jie3	E
i	i	i	vogal anterior fechada não arredondada	鸡, ji1	i
ou	oʊ	oU	ditongo	欧, ou1	o
o	ɔ	U	vogal posterior média aberta arredondada	哦, o4	o
u	u	u	vogal posterior fechada arredondada	主, zhu3	u
yu	y	y	vogal anterior fechada arredondada	于, yu2	u
Marcas de tom e símbolos adicionais					
1			tom de alto nível	淤, yu1	
2			tom crescente	鱼, yu2	
3			tom baixo (descrescente-crescente)	语, yu3	
4			tom decrescente	育, yu4	

Pinyin	IPA	X-SAMPA	Descrição	Exemplo do Pinyin	Visema
0			tom neutro	的, de0	
-	.	.	separação silábica	语音 yu3-yin1	

Tcheco (cs-CZ)

A tabela a seguir lista os fonemas do alfabeto fonético internacional (AFI), os símbolos do alfabeto fonético dos métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do tcheco que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
p	p	oclusiva bilabial surda	pes	p
t	t	oclusiva alveolar surda	tok	t
c	c	oclusiva palatal surda	t’uk	J
k	k	oclusiva velar surda	kos	k
b	b	oclusiva bilabial sonora	bez	p
d	d	oclusiva alveolar sonora	dok	t
ɟ	J\	oclusiva palatal sonora	d’as	J

IPA	X-SAMPA	Descrição	Exemplo	Visema
g	g	oclusiva velar sonora	gum	k
f	f	fricativa labiodental surda	film	f
v	v	fricativa labiodental sonora	ves	f
s	s	fricativa alveolar surda	sen	s
z	z	fricativa alveolar sonora	zel	s
ʃ	S	fricativa pós-alveolar surda	šel	S
ʒ	Z	fricativa pós-alveolar sonora	žen	S
x	x	fricativa velar surda	chat	k
ɦ	h	fricativa glotal sonora	hus	k
t͡s	ts	africada alveolar surda	co	s
t͡ʃ	tS	africada pós-alveolar surda	čin	S
d͡z	dz	africada alveolar sonora	špicberský	s
d͡ʒ	dZ	africada alveolar surda	džin	S

IPA	X-SAMPA	Descrição	Exemplo	Visema
m	m	nasal bilabial	mor	p
n	n	nasal alveolar	nos	t
ɲ	J	nasal palatal	ňader	J
ŋ	N	nasal velar	banka	k
r	r	vibrante alveolar surda	rys	r
r̠	r_r	trinado fricativo alveolar elevado com voz	řez	r
r̠̊	r_0_r	trinado fricativo alveolar elevado sem voz	keř	r
l	l	aproximante lateral alveolar	les	t
j	j	aproximante palatal	jen	i
w	w	aproximante labiovelar	Watson	u
r̩	r_=	trinado alveolar silábico com voz	krk	r
l̩	l_=	aproximante lateral alveolar silábica	vlna	t
Vogais				

IPA	X-SAMPA	Descrição	Exemplo	Visema
a	a	vogal anterior aberta não arredondada	lan	a
aː	a:	vogal anterior aberta longa não arredondada	lán	a
ɛ	E	vogal anterior média aberta não arredondada	let	E
ɛː	E:	vogal anterior média aberta longa não arredondada	lét	E
ɪ	I	vogal quase anterior quase fechada não arredondada	bit	i
iː	i:	vogal anterior fechada longa não arredondada	bít	i
o	o	vogal posterior semifechada arredondada	hol	o
oː	o:	vogal posterior média fechada longa arredondada	gól	o
u	u	vogal posterior fechada arredonda da	pul	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
uː	u:	vogal posterior fechada longa arredondada	půl	u
a͡u	au	ditongo	auto	a
ɛ͡u	Eu	ditongo	euro	E
o͡u	ou	ditongo	mouk	o
Símbolos adicionais				
ˈ	"	tonicidade primária		
.	.	separação silábica		

Dinamarquês (da-DK)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do dinamarquês que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	boca	p
d	d	oclusiva alveolar sonora	dor	t
ð	D	fricativa dental sonora	azar, xícara	T

IPA	X-SAMPA	Descrição	Exemplo	Visema
f	f	fricativa labiodental surda	fácil	f
g	g	oclusiva velar sonora	guerra	k
h	h	fricativa glotal surda	rio	k
j	j	aproximante palatal	sandália	i
k	k	oclusiva velar surda	queijo	k
l	l	aproximante lateral alveolar	limpo	t
m	m	nasal bilabial	montanha	p
n	n	nasal alveolar	negócio	t
ŋ	N	nasal velar	tanque	k
p	p	oclusiva bilabial surda	pato	p
r	r	vibrante alveolar	barco, estória	r
ʁ	R	fricativa uvular sonora	carro	k
s	s	fricativa alveolar surda	sat	s
t	t	oclusiva alveolar surda	teto	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
v	v	fricativa labiodental sonora	vinho	f
w	w	aproximante labiovelar	hav, weekend	u
Vogais				
ø	2	vogal anterior semifechada arredondada	øst	o
ø:	2:	vogal anterior semifechada longa arredondada	øse	o
ɐ	6	vogal central quase aberta	mor	a
œ	9	vogal anterior média aberta arredondada	skøn, grønt	U
œ:	9:	vogal anterior média aberta longa arredondada	høne, gøre	U
ə	@	vogal central média	pasta	@
æ:	{:	vogal anterior quase aberta longa não arredondada	paz	a
a	a	vogal anterior aberta não arredondada	man	a

IPA	X-SAMPA	Descrição	Exemplo	Visema
æ	{	vogal anterior quase aberta não arredondada	adresse	a
ɑ	A	vogal posterior aberta não arredondada	lak, tak	a
ɑ:	A:	vogal posterior aberta longa não arredondada	rase	a
p	p	vogal anterior semifechada não arredondada	midt	p
p:	p:	vogal anterior média fechada longa não arredondada	mele	p
ɛ	E	vogal anterior média aberta não arredondada	mæt	E
ɛ:	E:	vogal anterior média aberta longa não arredondada	mæle	E
i	i	vogal anterior fechada não arredondada	mit	i
i:	i:	vogal anterior fechada longa não arredondada	mile	i

IPA	X-SAMPA	Descrição	Exemplo	Visema
o	o	vogal posterior semifechada arredondada	foto	o
oː	o:	vogal posterior média fechada longa arredondada	mole	o
ɔ	U	vogal posterior média aberta arredondada	mund	U
ɔː	U:	vogal posterior média aberta longa arredondada	måle	U
ɒː	Q:	vogal posterior aberta longa arredondada	morse	U
u	u	vogal posterior fechada arredondada	lusk	u
uː	u:	vogal posterior fechada longa arredondada	mule	u
ʌ	V	posterior média aberta não arredondada	kører	E
y	y	vogal anterior fechada arredondada	yt	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
y:	y:	vogal anterior fechada longa arredondada	hyle	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Holandês belga (nl-BE)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do português brasileiro que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bak	p
d	d	oclusiva alveolar sonora	dak	t
͡dʒ	dZ	africada pós-alveolar sonora	manager	S
f	f	fricativa labiodental surda	fel	f

IPA	X-SAMPA	Descrição	Exemplo	Visema
g	g	oclusiva velar sonora	goal	k
ɣ	G	fricativa velar sonora	hoed	k
ɦ	h\	fricativa glotal sonora	hand	k
j	j	aproximante palatal	ja	i
k	k	oclusiva velar surda	kap	k
l	l	aproximante lateral alveolar	land	t
m	m	nasal bilabial	met	p
n	n	nasal alveolar	net	t
ŋ	N	nasal velar	bang	k
p	p	oclusiva bilabial surda	pak	p
r	r	vibrante alveolar	rand	r
s	s	fricativa alveolar surda	sein	s
ʃ	S	fricativa pós-alveolar surda	show	S
t	t	oclusiva alveolar surda	tak	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
v	v	fricativa labiodental sonora	vel	f
ʋ	v\	aproximante labiodental	wit	f
x	x	fricativa velar surda	toch	k
z	z	fricativa alveolar sonora	ziin	s
ʒ	Z	fricativa pós-alveolar sonora	bagage	S
Vogais				
øː	2:	vogal anterior semifechada longa arredondada	neus	o
œy	9y	ditongo	buit	U
ə	@	vogal central média	de	@
aː	a:	vogal anterior aberta longa não arredondada	baad	a
ɑː	A	vogal posterior aberta não arredondada	bad	a
pː	p:	vogal anterior média fechada longa não arredondada	beet	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɜː	3:	vogal média central aberta longa não arredondada	barrière	E
ɛ	E	vogal anterior média aberta não arredondada	bed	E
ɛi	Ei	ditongo	beet	E
i	i	vogal anterior fechada não arredondada	vier	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	pit	i
oː	o:	vogal posterior média fechada longa arredondada	boot	o
ɔ	U	vogal posterior média aberta arredondada	pot	U
u	u	vogal posterior fechada arredonda da	hoed	u
ʌu	Vu	ditongo	fout	E
yː	y:	vogal anterior fechada longa arredondada	fuut	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʏ	S	vogal quase anterior quase fechada arredonda da	hut	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Holandês (nl-NL)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do holandês que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bak	p
d	d	oclusiva alveolar sonora	dak	t
ʤ	dZ	africada pós-alveolar sonora	manager	S

IPA	X-SAMPA	Descrição	Exemplo	Visema
f	f	fricativa labiodental surda	fel	f
g	g	oclusiva velar sonora	goal	k
ɣ	G	fricativa velar sonora	hoed	k
ɦ	h\	fricativa glotal sonora	hand	k
j	j	aproximante palatal	ja	i
k	k	oclusiva velar surda	kap	k
l	l	aproximante lateral alveolar	land	t
m	m	nasal bilabial	met	p
n	n	nasal alveolar	net	t
ŋ	N	nasal velar	bang	k
p	p	oclusiva bilabial surda	pak	p
r	r	vibrante alveolar	rand	r
s	s	fricativa alveolar surda	sein	s
ʃ	S	fricativa pós-alveo lar surda	show	S

IPA	X-SAMPA	Descrição	Exemplo	Visema
t	t	oclusiva alveolar surda	tak	t
v	v	fricativa labiodental sonora	vel	f
ʋ	v\	aproximante labiodental	wit	f
x	x	fricativa velar surda	toch	k
z	z	fricativa alveolar sonora	ziin	s
ʒ	Z	fricativa pós-alveo lar sonora	bagage	S
Vogais				
øː	2:	vogal anterior semifechada longa arredondada	neus	o
œy	9y	ditongo	buit	U
ə	@	vogal central média	de	@
aː	a:	vogal anterior aberta longa não arredondada	baad	a
ɑː	A	vogal posterior aberta não arredondada	bad	a

IPA	X-SAMPA	Descrição	Exemplo	Visema
p:	p:	vogal anterior média fechada longa não arredondada	beet	p
ɜː	3:	vogal média central aberta longa não arredondada	barrière	E
ɛ	E	vogal anterior média aberta não arredondada	bed	E
ɛi	Ei	ditongo	beet	E
i	i	vogal anterior fechada não arredondada	vier	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	pit	i
o:	o:	vogal posterior média fechada longa arredondada	boot	o
ɔ	U	vogal posterior média aberta arredondada	pot	U
u	u	vogal posterior fechada arredonda da	hoed	u
ʌu	Vu	ditongo	fout	E

IPA	X-SAMPA	Descrição	Exemplo	Visema
y:	y:	vogal anterior fechada longa arredondada	fuut	u
ʏ	S	vogal quase anterior quase fechada arredondada	hut	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Inglês (EUA) (en-US)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do inglês americano que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʤ	dZ	africada pós-alveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental surda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal surda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar surda	cat	k
l	l	aproximante lateral alveolar	lay	l
m	m	nasal bilabial	mouse	p
n	n	nasal alveolar	nap	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial surda	speak	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar surda	seem	s

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʃ	S	fricativa pós-alveolar surda	ship	S
t	t	oclusiva alveolar surda	trap	t
͡tʃ	tS	africada pós-alveolar surda	chart	S
θ	T	fricativa dental surda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante labiovelar	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa pós-alveolar sonora	vision	S
Vogais				
ə	@	vogal central média	arena	@
ɚ	@`	vogal média central colorida pelo r	reader	@
æ	{	vogal anterior quase aberta não arredondada	trap	a
aɪ	aI	ditongo	price	a
aʊ	aU	ditongo	mouth	a

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɑ	A	vogal posterior aberta longa não arredondada	father	a
eɪ	eI	ditongo	face	p
ɝ	3`	vogal aberta média central não arredondada colorida pelo r	nurse	E
ɛ	E	vogal anterior média aberta não arredondada	dress	E
i	i	vogal anterior fechada longa não arredondada	fleece	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	kit	i
oʊ	oU	ditongo	goat	o
ɔ	U	vogal posterior média aberta longa arredondada	thought	U
ɔɪ	OI	ditongo	choice	U
u	u	vogal posterior fechada longa arredondada	goose	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʊ	U	vogal quase posterior quase fechada arredondada	foot	u
ʌ	V	vogal posterior média aberta não arredondada	strut	E
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Inglês australiano (en-AU)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do inglês australiano que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʤ	dZ	africada pós-alveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental surda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal surda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar surda	cat	k
l	l	aproximante lateral alveolar	lay	t
l̩	l=	aproximante lateral alveolar silábica	battle	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	N	nasal alveolar silábica	botão	t
ŋ	N	nasal velar	thing	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
p	p	oclusiva bilabial surda	pin	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar surda	seem	s
ʃ	S	fricativa pós-alveolar surda	ship	S
t	t	oclusiva alveolar surda	task	t
͡tʃ	tS	africada pós-alveolar surda	chart	S
Θ	T	fricativa dental surda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante labiovelar	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa pós-alveolar sonora	vision	S
Vogais				
ə	@	vogal central média	arena	@
əʊ	@U	ditongo	goat	@

IPA	X-SAMPA	Descrição	Exemplo	Visema
æ	{	vogal anterior quase aberta não arredondada	trap	a
aɪ	aI	ditongo	price	a
aʊ	aU	ditongo	mouth	a
ɑː	A:	vogal posterior aberta longa não arredondada	father	a
eɪ	eI	ditongo	face	p
ɜː	3:	vogal média central aberta longa não arredondada	nurse	E
ɛ	E	vogal anterior média aberta não arredondada	dress	E
ɛə	E@	ditongo	square	E
iː	i	vogal anterior fechada longa não arredondada	fleece	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	kit	i
ɪə	I@	ditongo	near	i

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɔː	OI	vogal posterior média aberta longa arredondada	thought	U
ɔɪ	OI	Ditongo	choice	U
ɒ	Q	vogal posterior aberta arredonda da	lot	U
uː	u:	vogal posterior fechada longa arredondada	goose	u
ʊ	U	vogal quase posterior quase fechada arredonda da	foot	u
ʊə	U@	ditongo	cure	u
ʌ	V	Vogal posterior média aberta não arredondada	strut	E
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Inglês britânico (en-GB)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do inglês britânico que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada pós-alveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental surda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal surda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar surda	cat	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
l	l	aproximante lateral alveolar	lay	t
l̩	l=	aproximante lateral alveolar silábica	battle	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	N	nasal alveolar silábica	botão	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial surda	pin	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar surda	seem	s
ʃ	S	fricativa pós-alveolar surda	ship	S
t	t	oclusiva alveolar surda	task	t
͡tʃ	tS	africada pós-alveolar surda	chart	S
ɵ	T	fricativa dental surda	thin	T

IPA	X-SAMPA	Descrição	Exemplo	Visema
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante labiovelar	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa pós-alveolar sonora	vision	S
Vogais				
ə	@	vogal central média	arena	@
əʊ	@U	ditongo	goat	@
æ	{	vogal anterior quase aberta não arredondada	trap	a
aɪ	aI	ditongo	price	a
aʊ	aU	ditongo	mouth	a
ɑː	A:	vogal posterior aberta longa não arredondada	father	a
eɪ	eI	ditongo	face	p
ɜː	3:	vogal média central aberta longa não arredondada	nurse	E

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɛ	E	vogal anterior média aberta não arredondada	dress	E
ɛə	E@	ditongo	square	E
iː	i	vogal anterior fechada longa não arredondada	fleece	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	kit	i
ɪə	I@	ditongo	near	i
ɔː	U:	vogal posterior média aberta longa arredondada	thought	U
ɔɪ	OI	Ditongo	choice	U
ɒ	Q	vogal posterior aberta arredonda da	lot	U
uː	u:	vogal posterior fechada longa arredondada	goose	u
ʊ	U	vogal quase posterior quase fechada arredonda da	foot	u
ʊə	U@	ditongo	cure	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʌ	V	Vogal posterior média aberta não arredondada	strut	E
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Inglês indiano (en-IN)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do inglês indiano que são compatíveis com o Amazon Polly.

Para obter fonemas adicionais usados em conjunto com o inglês indiano, consulte Hindi (hi-IN).

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada pós-alveolar sonora	jump	S

IPA	X-SAMPA	Descrição	Exemplo	Visema
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental surda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal surda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar surda	cat	k
l	l	aproximante lateral alveolar	lay	t
l̩	l=	aproximante lateral alveolar silábica	battle	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	N	nasal alveolar silábica	nap	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial surda	pin	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar surda	seem	s
ʃ	S	fricativa pós-alveolar surda	ship	S
t	t	oclusiva alveolar surda	task	t
͡tʃ	tS	africada pós-alveolar surda	chart	S
Θ	T	fricativa dental surda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante labiovelar	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa pós-alveolar sonora	vision	S
Vogais				
ə	@	vogal central média	arena	@
əʊ	@U	ditongo	goat	@

IPA	X-SAMPA	Descrição	Exemplo	Visema
æ	{	vogal anterior quase aberta não arredondada	trap	a
aɪ	aI	ditongo	price	a
aʊ	aU	ditongo	mouth	a
ɑː	A:	vogal posterior aberta longa não arredondada	father	a
eɪ	eI	ditongo	face	p
ɜː	3:	vogal média central aberta longa não arredondada	nurse	E
ɛ	E	vogal anterior média aberta não arredondada	dress	E
ɛə	E@	ditongo	square	E
iː	i	vogal anterior fechada longa não arredondada	fleece	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	kit	i
ɪə	I@	ditongo	near	i

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɔː	OI	vogal posterior média aberta longa arredondada	thought	U
ɔɪ	OI	Ditongo	choice	U
ɒ	Q	vogal posterior aberta arredonda da	lot	U
uː	u:	vogal posterior fechada longa arredondada	goose	u
ʊ	U	vogal quase posterior quase fechada arredonda da	foot	u
ʊə	U@	ditongo	cure	u
ʌ	V	Vogal posterior média aberta não arredondada	strut	E
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Inglês (Irlanda) (en-IE)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do inglês irlandês que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada pós-alveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental surda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal surda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar surda	cat	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
l	l	aproximante lateral alveolar	lay	t
m	m	nasal bilabial	mouse	p
n	n	nasal alveolar	nap	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial surda	speak	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar surda	seem	s
ʃ	S	fricativa pós-alveolar surda	ship	S
t	t	oclusiva alveolar surda	trap	t
͡tʃ	tS	africada pós-alveolar surda	chart	S
θ	T	fricativa dental surda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante labiovelar	west	u
z	z	fricativa alveolar sonora	zero	s

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʒ	Z	fricativa pós-alveolar sonora	vision	S
Vogais				
ə	@	vogal central média	arena	@
ɚ	@`	vogal média central colorida pelo r	reader	@
æ	{	vogal anterior quase aberta não arredondada	trap	a
aɪ	aI	ditongo	price	a
aʊ	aU	ditongo	mouth	a
ɑ	A	vogal posterior aberta longa não arredondada	father	a
eɪ	eI	ditongo	face	p
ɝ	3`	vogal aberta média central não arredondada colorida pelo r	nurse	E
ɛ	E	vogal anterior média aberta não arredondada	dress	E
i	i	vogal anterior fechada longa não arredondada	fleece	i

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɪ	eu	vogal quase anterior quase fechada não arredondada	kit	i
oʊ	oU	ditongo	goat	o
ɔ	U	vogal posterior média aberta longa arredondada	thought	U
ɔɪ	OI	ditongo	choice	U
u	u	vogal posterior fechada longa arredondada	goose	u
ʊ	U	vogal quase posterior quase fechada arredonda da	foot	u
ʌ	V	vogal posterior média aberta não arredondada	strut	E
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Inglês (neozelandês) (en-NZ)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do inglês neozelandês que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada pós-alveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental surda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal surda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar surda	cat	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
l	l	aproximante lateral alveolar	lay	t
l̩	l=	aproximante lateral alveolar silábica	battle	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	N	nasal alveolar silábica	botão	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial surda	pin	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar surda	seem	s
ʃ	S	fricativa pós-alveolar surda	ship	S
t	t	oclusiva alveolar surda	task	t
͡tʃ	tS	africada pós-alveolar surda	chart	S
ɵ	T	fricativa dental surda	thin	T

IPA	X-SAMPA	Descrição	Exemplo	Visema
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante labiovelar	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa pós-alveolar sonora	vision	S
Vogais				
ə	@	vogal central média	arena	@
əʊ	@U	ditongo	goat	@
æ	{	vogal anterior quase aberta não arredondada	trap	a
aɪ	aI	ditongo	price	a
aʊ	aU	ditongo	mouth	a
ɑː	A:	vogal posterior aberta longa não arredondada	father	a
eɪ	eI	ditongo	face	p
ɜː	3:	vogal média central aberta longa não arredondada	nurse	E

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɛ	E	vogal anterior média aberta não arredondada	dress	E
ɛə	E@	ditongo	square	E
iː	i	vogal anterior fechada longa não arredondada	fleece	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	kit	i
ɪə	I@	ditongo	near	i
ɔː	U:	vogal posterior média aberta longa arredondada	thought	U
ɔɪ	OI	Ditongo	choice	U
ɒ	Q	vogal posterior aberta arredonda da	lot	U
uː	u:	vogal posterior fechada longa arredondada	goose	u
ʊ	U	vogal quase posterior quase fechada arredonda da	foot	u
ʊə	U@	ditongo	cure	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʌ	V	Vogal posterior média aberta não arredondada	strut	E
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

A voz de Aria fala inglês neozelandês e oferece suporte limitado para maori. Ela pode pronunciar as palavras e frases em maori a seguir. As frases em maori diferenciam maiúsculas de minúsculas.

Inglês	Maori
Hello/Cheers	Kia ora
Bem-vindo (ao)	Nau mai (ki)
Olá (uma pessoa)/obrigado	Tēnā koe
Olá (três ou mais pessoas)/obrigado	Tēnā koutou
Bom dia	Ata marie
Bom dia	Morena
Obrigado	Ngā mihi
Cuide-se	Nga manaakitanga
Te vejo	Ka kite
Te vejo mais tarde	Mā te wā

Inglês	Maori
Tenha um bom dia	Kia pai tō ra
Feliz Natal	Meri Kirihimete
maori	Māori
Língua maori	o Areo Māori
Semana da língua maori	A wiki do Areo Māori
Nova Zelândia	Nova Zelândia
Ano Novo Maori	Matariki
Cidade na Nova Zelândia/O Dia de Waitangi é o dia nacional da Nova Zelândia	Waitangi
Um	tailandês
Dois	rua
Três	passeio
quatro	o que
Cinco	rima
Seis	ono
Sete	branco
Oito	waru
Nove	iwa
Dez	tekau
Vinte	rua tekau
Trinta	Tour Tekau

Inglês (singapurense) (en-SG)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do inglês singapurense que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada pós-alveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental surda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal surda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar surda	cat	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
l	l	aproximante lateral alveolar	lay	t
l̩	l=	aproximante lateral alveolar silábica	battle	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	N	nasal alveolar silábica	botão	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial surda	pin	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar surda	seem	s
ʃ	S	fricativa pós-alveolar surda	ship	S
t	t	oclusiva alveolar surda	task	t
͡tʃ	tS	africada pós-alveolar surda	chart	S
Θ	T	fricativa dental surda	thin	T

IPA	X-SAMPA	Descrição	Exemplo	Visema
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante labiovelar	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa pós-alveolar sonora	vision	S
Vogais				
ə	@	vogal central média	arena	@
əʊ	@U	ditongo	goat	@
æ	{	vogal anterior quase aberta não arredondada	trap	a
aɪ	aI	ditongo	price	a
aʊ	aU	ditongo	mouth	a
ɑː	A:	vogal posterior aberta longa não arredondada	father	a
eɪ	eI	ditongo	face	p
ɜː	3:	vogal média central aberta longa não arredondada	nurse	E

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɛ	E	vogal anterior média aberta não arredondada	dress	E
ɛə	E@	ditongo	square	E
iː	i	vogal anterior fechada longa não arredondada	fleece	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	kit	i
ɪə	I@	ditongo	near	i
ɔː	U:	vogal posterior média aberta longa arredondada	thought	U
ɔɪ	OI	Ditongo	choice	U
ɒ	Q	vogal posterior aberta arredonda da	lot	U
uː	u:	vogal posterior fechada longa arredondada	goose	u
ʊ	U	vogal quase posterior quase fechada arredonda da	foot	u
ʊə	U@	ditongo	cure	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʌ	V	Vogal posterior média aberta não arredondada	strut	E
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Inglês (África do Sul) (en-ZA)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do inglês sul-africano que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada pós-alveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T

IPA	X-SAMPA	Descrição	Exemplo	Visema
f	f	fricativa labiodental surda	five	f
g	g	oclusiva velar sonora	game	k
h	h	fricativa glotal surda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar surda	cat	k
l	l	aproximante lateral alveolar	lay	t
l̩	l=	aproximante lateral alveolar silábica	battle	t
ɬ̩	K	fricativa palatal surda	Hum hl Manga	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	N	nasal alveolar silábica	botão	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial surda	pin	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɹ	r\	aproximante alveolar	red	r
r	r	vibrante alveolar	spa ou eris	r
s	s	fricativa alveolar surda	seem	s
ʃ	S	fricativa pós-alveo lar surda	ship	S
t	t	oclusiva alveolar surda	task	t
͡tʃ	tS	africada pós-alveo lar surda	chart	S
Θ	T	fricativa dental surda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante labiovelar	west	u
x	x	fricativa velar surda	g) auteng	k
z	z	fricativa alveolar sonora	zero	s
!	!\	clique pós-alveolar	gq eberha	k
|	|\	clique dentário	em um tubo de cm	t
||	||\	clique lateral	xh rosa	t
Vogais				

IPA	X-SAMPA	Descrição	Exemplo	Visema
ə	@	vogal central média	arena	@
əi	i	ditongo	nelspruit	i
əʊ	@U	ditongo	goat	@
æ	{	vogal anterior quase aberta não arredondada	trap	a
aɪ	aI	ditongo	price	a
aʊ	aU	ditongo	mouth	a
ɑː	A:	vogal posterior aberta longa não arredondada	father	a
eɪ	eI	ditongo	face	p
ɜː	3:	vogal média central aberta longa não arredondada	nurse	E
ɛ	E	vogal anterior média aberta não arredondada	dress	E
ɛə	E@	ditongo	square	E
iː	i	vogal anterior fechada longa não arredondada	fleece	i
iəi	I@	ditongo	du preez	i

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɪ	eu	vogal quase anterior quase fechada não arredondada	kit	i
ɪə	I@	ditongo	near	i
ɔː	U:	vogal posterior média aberta longa arredondada	thought	U
ɔɪ	OI	Ditongo	choice	U
ɒ	Q	vogal posterior aberta arredonda da	lot	U
uː	u:	vogal posterior fechada longa arredondada	goose	u
ʊ	U	vogal quase posterior quase fechada arredonda da	foot	u
ʊə	U@	ditongo	cure	u
ʌ	V	Vogal posterior média aberta não arredondada	strut	E
y	y	vogal anterior fechada arredonda da	van v uu ren	u
Símbolos adicionais				

IPA	X-SAMPA	Descrição	Exemplo	Visema
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Inglês galês (en-GB-WLS)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do inglês galês que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bed	p
d	d	oclusiva alveolar sonora	dig	t
ʤ	dZ	africada pós-alveolar sonora	jump	S
ð	D	fricativa dental sonora	then	T
f	f	fricativa labiodental surda	five	f
g	g	oclusiva velar sonora	game	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
h	h	fricativa glotal surda	house	k
j	j	aproximante palatal	yes	i
k	k	oclusiva velar surda	cat	k
l	l	aproximante lateral alveolar	lay	t
l̩	l=	aproximante lateral alveolar silábica	battle	t
m	m	nasal bilabial	mouse	p
m̩	m=	nasal bilabial silábica	anthem	p
n	n	nasal alveolar	nap	t
n̩	N	nasal alveolar silábica	nap	t
ŋ	N	nasal velar	thing	k
p	p	oclusiva bilabial surda	pin	p
ɹ	r\	aproximante alveolar	red	r
s	s	fricativa alveolar surda	seem	s
ʃ	S	fricativa pós-alveolar surda	ship	S

IPA	X-SAMPA	Descrição	Exemplo	Visema
t	t	oclusiva alveolar surda	task	t
͡tʃ	tS	africada pós-alveolar surda	chart	S
Θ	T	fricativa dental surda	thin	T
v	v	fricativa labiodental sonora	vest	f
w	w	aproximante labiovelar	west	u
z	z	fricativa alveolar sonora	zero	s
ʒ	Z	fricativa pós-alveolar sonora	vision	S
Vogais				
ə	@	vogal central média	arena	@
əʊ	@U	ditongo	goat	@
æ	{	vogal anterior quase aberta não arredondada	trap	a
aɪ	aI	ditongo	price	a
aʊ	aU	ditongo	mouth	a
ɑː	A:	vogal posterior aberta longa não arredondada	father	a

IPA	X-SAMPA	Descrição	Exemplo	Visema
eɪ	eI	ditongo	face	p
ɜː	3:	vogal média central aberta longa não arredondada	nurse	E
ɛ	E	vogal anterior média aberta não arredondada	dress	E
ɛə	E@	ditongo	square	E
iː	i	vogal anterior fechada longa não arredondada	fleece	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	kit	i
ɪə	I@	ditongo	near	i
ɔː	OI	vogal posterior média aberta longa arredondada	thought	U
ɔɪ	OI	Ditongo	choice	U
ɒ	Q	vogal posterior aberta arredonda da	lot	U
uː	u:	vogal posterior fechada longa arredondada	goose	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʊ	U	vogal quase posterior quase fechada arredonda da	foot	u
ʊə	U@	ditongo	cure	u
ʌ	V	Vogal posterior média aberta não arredondada	strut	E
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Finlandês (Fi-Fi)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do finlandês que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes finlandesas				
p	p	oclusiva bilabial surda	[p] anquim	p
t	t	oclusiva alveolar surda	[t] também	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
k	k	oclusiva velar surda	[k]aali	k
d	d	oclusiva alveolar sonora	[d]ata	t
s	s	fricativa alveolar surda	[s] ali	s
h	h	fricativa glotal surda	[h] tchau	k
ʋ	v\	aproximante labiodental	[v] aiva	v
j	j	aproximante palatal	[j] oki	i
l	l	aproximante lateral alveolar	[l] Roma	t
r	r	vibrante alveolar surda	[r] iita	r
m	m	nasal bilabial	[m] ato	p
n	n	nasal alveolar	[n] enää	t
ŋ	N	nasal velar	o [n] ki	k
Consoantes encontradas em palavras emprestadas				
b	b	oclusiva bilabial sonora	[b] russo	p
f	f	fricativa labiodental surda	[f] firma	v

IPA	X-SAMPA	Descrição	Exemplo	Visema
w	w	aproximante labiovelar	[w] wiki	u
z	z	fricativa alveolar sonora	[z]	s
g	g	oclusiva velar sonora	[g] aala	k
ʃ	S	fricativa pós-alveolar surda	[sh] akki	S
ʒ	Z	fricativa pós-alveolar sonora	[g] gênero	S
θ	T	fricativa dental surda	ano [th]	T
ð	D	fricativa dental sonora	ei [o]	T
Vogais curtas				
i	i	vogal anterior fechada não arredondada	k[i]lo	i
ɛ	E	vogal anterior média aberta não arredondada	k[e]sä	E
æ	{	vogal anterior quase aberta não arredondada	k[ä]ly	A

IPA	X-SAMPA	Descrição	Exemplo	Visema
y	y	vogal anterior fechada arredondada	[k] ou	u
ø	2	vogal anterior semifechada arredondada	[ö] ly	U
u	u	vogal posterior fechada arredondada	k[u]lo	u
ɔ	U	vogal posterior média aberta arredondada	k[o]lo	U
ɑ	A	vogal posterior aberta não arredondada	k[a]la	A
Vogais longas				
iː	iː	vogal anterior fechada longa não arredondada	[sii] li	i
ɛː	E:	vogal anterior média aberta longa não arredondada	[ee] tu	E
æː	{:	vogal anterior quase aberta longa não arredondada	[tää] mais	A
y:	y:	vogal anterior fechada longa não arredondada	[yy] li	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
øː	2:	vogal anterior semifechada longa arredondada	[töö] lö	U
uː	u:	vogal posterior fechada longa arredondada	[tuu] li	u
ɔː	U:	vogal posterior média aberta longa arredondada	[roo] li	U
ɑː	A:	vogal posterior aberta longa não arredondada	k[aa]su	A
Ditongos				
ɛi	Ei	ditongo	[li] pä	E
æi	{i	ditongo	[äi]ti	A
ui	ui	ditongo	k[ui]n	u
ɑi	AI	ditongo	k[ai]kki	A
ɔi	Oi	ditongo	[poi] ka	U
øi	2i	ditongo	[söin]	U
yi	yi	ditongo	[yi] jy	u
ɑu	au	ditongo	[sauna]	A
ɔu	Ou	ditongo	k[ou]lu	U
ɛu	Eu	ditongo	[ruína]	E
iu	iu	ditongo	[viu] lu	i

IPA	X-SAMPA	Descrição	Exemplo	Visema
æy	{y	ditongo	t[äy]nnä	A
øy	2y	ditongo	k[öy]hä	U
əy	Olho	ditongo	pes [ey] tyä	E
IY	IY	ditongo	kär [iy] tyä	i
i.	IE	ditongo	[morra]	i
yø	y2	ditongo	[yö]	u
e	Uo	ditongo	[tuo]	u
Vogais encontradas em palavras emprestadas em inglês				
ɪ	eu	vogal quase anterior quase fechada não arredondada	[b]	i
ʊ	U	vogal quase posterior quase fechada arredonda da	[livro]	u
ə	@	vogal central média	[a] sobre	@
ʌ	V	vogal posterior média aberta não arredondada	[cut]	E

Francês (fr-FR)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do francês que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	boire	p
d	d	oclusiva alveolar sonora	madame	t
f	f	fricativa labiodental surda	femme	f
g	g	oclusiva velar sonora	grand	k
ɥ	H	aproximante labiopalatal	bruit	u
j	j	aproximante palatal	meilleur	i
k	k	oclusiva velar surda	quatre	k
l	l	aproximante lateral alveolar	malade	t
m	m	nasal bilabial	maison	p
n	n	nasal alveolar	astronome	t
ɲ	J	nasal palatal	baigner	J
ŋ	N	nasal velar	parking	k
p	p	oclusiva bilabial surda	pomme	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʁ	R	fricativa uvular sonora	amoureux	k
s	s	fricativa alveolar surda	santé	s
ʃ	S	fricativa pós-alveolar surda	chat	S
t	t	oclusiva alveolar surda	téléphone	t
v	v	fricativa labiodental sonora	vrai	f
w	w	aproximante labiovelar	soir	u
z	z	fricativa alveolar sonora	raison	s
ʒ	Z	fricativa pós-alveolar sonora	aubergine	S
Vogais				
ø	2	vogal anterior semifechada arredondada	deux	o
œ	9	vogal anterior média aberta arredondada	neuf	U
œ̃	9~	vogal anterior média aberta nasal arredondada	brun	U

IPA	X-SAMPA	Descrição	Exemplo	Visema
ə	@	vogal central média	je	@
a	a	vogal anterior aberta não arredondada	table	a
ɑ̃	A~	vogal posterior aberta nasal não arredondada	camembert	a
p	p	vogal anterior semifechada não arredondada	marché	p
ɛ	E	vogal anterior média aberta não arredondada	neige	E
ɛ̃	E~	vogal anterior média aberta nasal não arredondada	sapin	E
i	i	vogal anterior fechada não arredondada	mille	i
o	o	vogal posterior semifechada arredondada	hôpital	o
ɔ	U	vogal posterior média aberta arredondada	homme	U
ɔ̃	U~	vogal posterior média aberta nasal arredondada	bon	U

IPA	X-SAMPA	Descrição	Exemplo	Visema
u	u	vogal posterior fechada arredonda da	sous	u
y	y	vogal anterior fechada arredonda da	dur	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Francês (belga) (fr-BE)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do francês belga que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	boire	p
d	d	oclusiva alveolar sonora	madame	t
f	f	fricativa labiodental surda	femme	f

IPA	X-SAMPA	Descrição	Exemplo	Visema
g	g	oclusiva velar sonora	grand	k
ɥ	H	aproximante labiopalatal	bruit	u
j	j	aproximante palatal	meilleur	i
k	k	oclusiva velar surda	quatre	k
l	l	aproximante lateral alveolar	malade	t
m	m	nasal bilabial	maison	p
n	n	nasal alveolar	astronome	t
ɲ	J	nasal palatal	baigner	J
ŋ	N	nasal velar	parking	k
p	p	oclusiva bilabial surda	pomme	p
ʁ	R	fricativa uvular sonora	amoureux	k
s	s	fricativa alveolar surda	santé	s
ʃ	S	fricativa pós-alveo lar surda	chat	S
t	t	oclusiva alveolar surda	téléphone	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
v	v	fricativa labiodental sonora	vrai	f
w	w	aproximante labiovelar	soir	u
z	z	fricativa alveolar sonora	raison	s
ʒ	Z	fricativa pós-alveolar sonora	aubergine	S
Vogais				
ø	2	vogal anterior semifechada arredondada	deux	o
œ	9	vogal anterior média aberta arredondada	neuf	U
œ̃	9~	vogal anterior média aberta nasal arredondada	brun	U
ə	@	vogal central média	je	@
a	a	vogal anterior aberta não arredondada	table	a
ɑ̃	A~	vogal posterior aberta nasal não arredondada	camembert	a

IPA	X-SAMPA	Descrição	Exemplo	Visema
p	p	vogal anterior semifechada não arredondada	marché	p
ɛ	E	vogal anterior média aberta não arredondada	neige	E
ɛ̃	E~	vogal anterior média aberta nasal não arredondada	sapin	E
i	i	vogal anterior fechada não arredondada	mille	i
o	o	vogal posterior semifechada arredondada	hôpital	o
ɔ	U	vogal posterior média aberta arredondada	homme	U
ɔ̃	U~	vogal posterior média aberta nasal arredondada	bon	U
u	u	vogal posterior fechada arredondada	sous	u
y	y	vogal anterior fechada arredondada	dur	u
Símbolos adicionais				

IPA	X-SAMPA	Descrição	Exemplo	Visema
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Francês canadense (fr-CA)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do francês canadense que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	boire	p
d	d	oclusiva alveolar sonora	madame	t
f	f	fricativa labiodental surda	femme	f
g	g	oclusiva velar sonora	grand	k
ɥ	H	aproximante labiopalatal	bruit	u
j	j	aproximante palatal	meilleur	i

IPA	X-SAMPA	Descrição	Exemplo	Visema
k	k	oclusiva velar surda	quatre	k
l	l	aproximante lateral alveolar	malade	t
m	m	nasal bilabial	maison	p
n	n	nasal alveolar	astronome	t
ɲ	J	nasal palatal	baigner	J
ŋ	N	nasal velar	parking	k
p	p	oclusiva bilabial surda	pomme	p
ʁ	R	fricativa uvular sonora	amoureux	k
s	s	fricativa alveolar surda	santé	s
ʃ	S	fricativa pós-alveolar surda	chat	S
t	t	oclusiva alveolar surda	téléphone	t
v	v	fricativa labiodental sonora	vrai	f
w	w	aproximante labiovelar	soir	u
z	z	fricativa alveolar sonora	raison	s

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʒ	Z	fricativa pós-alveolar sonora	aubergine	S
Vogais				
ø	2	vogal anterior semifechada arredondada	deux	o
œ	9	vogal anterior média aberta arredondada	neuf	U
œ̃	9~	vogal anterior média aberta nasal arredondada	brun	U
ə	@	vogal central média	je	@
a	a	vogal anterior aberta não arredondada	table	a
ɑ̃	A~	vogal posterior aberta nasal não arredondada	camembert	a
p	p	vogal anterior semifechada não arredondada	marché	p
ɛ	E	vogal anterior média aberta não arredondada	neige	E

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɛ̃	E~	vogal anterior média aberta nasal não arredondada	sapin	E
i	i	vogal anterior fechada não arredondada	mille	i
o	o	vogal posterior semifechada arredondada	hôpital	o
ɔ	U	vogal posterior média aberta arredondada	homme	U
ɔ̃	U~	vogal posterior média aberta nasal arredondada	bon	U
u	u	vogal posterior fechada arredondada	sous	u
y	y	vogal anterior fechada arredondada	dur	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Alemão (de-DE)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do alemão que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
ʔ	?	oclusiva glotal		
b	b	oclusiva bilabial sonora	Bier	p
d	d	oclusiva alveolar sonora	Dach	t
ç	C	fricativa palatal surda	ich	k
͡dʒ	dZ	africada pós-alveolar sonora	Dschungel	S
f	f	Fricativa labiodental surda	Vogel	f
g	g	Oclusiva velar sonora	Gabel	k
h	h	Fricativa glotal surda	Haus	k
j	j	Fricativa glotal surda	jemand	i
k	k	Oclusiva velar surda	Kleid	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
l	l	Aproximante lateral alveolar	Loch	t
m	m	Nasal bilabial	Milch	p
n	n	Nasal alveolar	Natur	t
ŋ	N	Nasal velar	klingen	k
p	p	Oclusiva bilabial surda	Park	p
͡pf	pf	Africada labiodental surda	Apfel	
ʀ	R	Vibrante uvular	Regen	
s	s	fricativa alveolar surda	Messer	s
ʃ	S	Fricativa pós-alveolar surda	Fischer	S
t	t	Oclusiva alveolar surda	Topf	T
͡ts	Ts	Africada alveolar surda	Zahl	
͡tʃ	tS	Africada pós-alveolar surda	deutsch	S
v	v	Fricativa labiodental sonora	Wasser	f
x	x	Fricativa velar surda	kochen	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
z	z	Fricativa alveolar sonora	Consulte	s
ʒ	Z	Fricativa pós-alveolar sonora	Orange	S
Vogais				
øː	2:	vogal anterior semifechada longa arredondada	böse	o
ɐ	6	vogal central quase aberta	besser	a
ɐ̯	6_^	vogal central quase aberta não silábica	Klar	a
œ	9	vogal anterior média aberta arredondada	können	U
ə	@	vogal central média	Rede	@
a	a	vogal anterior aberta não arredondada	Salz	a
aː	a:	vogal anterior aberta longa não arredondada	Sahne	a
aɪ	aI	ditongo	nein	a
aʊ	aU	ditongo	Augen	a

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɑ̃	A~	vogal posterior aberta nasal não arredondada	Restaurant	a
p:	p:	vogal anterior média fechada longa não arredondada	Rede	p
ɛ	E	vogal anterior média aberta não arredondada	Keller	E
ɛ̃	E~	vogal anterior média aberta nasal não arredondada	Terrain	E
i:	i:	vogal anterior fechada longa não arredondada	Lied	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	bitte	i
o:	o:	vogal posterior média fechada longa arredondada	Kohl	o
ɔ	U	vogal posterior média aberta arredondada	Koffer	U
ɔ̃	U~	vogal posterior média aberta nasal arredondada	Annonce	U

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɔʏ	OY	ditongo	neu	U
u:	u:	vogal posterior fechada longa arredondada	Bruder	u
ʊ	U	vogal quase posterior quase fechada arredonda da	Wunder	u
y:	y:	vogal anterior fechada longa arredondada	kühl	u
ʏ	S	vogal quase anterior quase fechada arredonda da	Küche	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Alemão (Austríaco) (de-AT)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do alemão austríaco que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
ʔ	?	oclusiva glotal		
b	b	oclusiva bilabial sonora	Bier	p
d	d	oclusiva alveolar sonora	Dach	t
ç	C	fricativa palatal surda	ich	k
͡dʒ	dZ	africada pós-alveolar sonora	Dschungel	S
f	f	Fricativa labiodental surda	Vogel	f
g	g	Oclusiva velar sonora	Gabel	k
h	h	Fricativa glotal surda	Haus	k
j	j	Fricativa glotal surda	jemand	i
k	k	Oclusiva velar surda	Kleid	k
l	l	Aproximante lateral alveolar	Loch	t
m	m	Nasal bilabial	Milch	p
n	n	Nasal alveolar	Natur	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
ŋ	N	Nasal velar	klingen	k
p	p	Oclusiva bilabial surda	Park	p
p͡f	pf	Africada labiodental surda	Apfel	
ʀ	R	Vibrante uvular	Regen	
s	s	fricativa alveolar surda	Messer	s
ʃ	S	Fricativa pós-alveolar surda	Fischer	S
t	t	Oclusiva alveolar surda	Topf	T
t͡s	Ts	Africada alveolar surda	Zahl	
t͡ʃ	tS	Africada pós-alveolar surda	deutsch	S
v	v	Fricativa labiodental sonora	Wasser	f
x	x	Fricativa velar surda	kochen	k
z	z	Fricativa alveolar sonora	Consulte	s
ʒ	Z	Fricativa pós-alveolar sonora	Orange	S
Vogais				

IPA	X-SAMPA	Descrição	Exemplo	Visema
øː	2:	vogal anterior semifechada longa arredondada	böse	o
ɐ	6	vogal central quase aberta	besser	a
ɐ̯	6_^	vogal central quase aberta não silábica	Klar	a
œ	9	vogal anterior média aberta arredondada	können	U
ə	@	vogal central média	Rede	@
a	a	vogal anterior aberta não arredondada	Salz	a
aː	a:	vogal anterior aberta longa não arredondada	Sahne	a
aɪ	aI	ditongo	nein	a
aʊ	aU	ditongo	Augen	a
ɑ̃	A~	vogal posterior aberta nasal não arredondada	Restaurant	a
pː	p:	vogal anterior média fechada longa não arredondada	Rede	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɛ	E	vogal anterior média aberta não arredondada	Keller	E
ɛ̃	E~	vogal anterior média aberta nasal não arredondada	Terrain	E
i:	i:	vogal anterior fechada longa não arredondada	Lied	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	bitte	i
o:	o:	vogal posterior média fechada longa arredondada	Kohl	o
ɔ	U	vogal posterior média aberta arredondada	Koffer	U
ɔ̃	U~	vogal posterior média aberta nasal arredondada	Annonce	U
ɔʏ	OY	ditongo	neu	U
u:	u:	vogal posterior fechada longa arredondada	Bruder	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʊ	U	vogal quase posterior quase fechada arredondada	Wunder	u
y:	y:	vogal anterior fechada longa arredondada	kühl	u
ʏ	S	vogal quase anterior quase fechada arredondada	Küche	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Alemão (padrão suíço) (de-CH)

A tabela a seguir lista os fonemas do alfabeto fonético internacional (AFI), os símbolos do alfabeto fonético dos métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do alemão (padrão suíço) que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
ʔ	?	oclusiva glotal		

IPA	X-SAMPA	Descrição	Exemplo	Visema
b	b	oclusiva bilabial sonora	Bier	p
d	d	oclusiva alveolar sonora	Dach	t
ç	C	fricativa palatal surda	ich	k
͡dʒ	dZ	africada pós-alveolar sonora	Dschungel	S
f	f	Fricativa labiodental surda	Vogel	f
g	g	Oclusiva velar sonora	Gabel	k
h	h	Fricativa glotal surda	Haus	k
j	j	Fricativa glotal surda	jemand	i
k	k	Oclusiva velar surda	Kleid	k
l	l	Aproximante lateral alveolar	Loch	t
m	m	Nasal bilabial	Milch	p
n	n	Nasal alveolar	Natur	t
ŋ	N	Nasal velar	klingen	k
p	p	Oclusiva bilabial surda	Park	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
p͡f	pf	Africada labiodental surda	Apfel	
ʀ	R	Vibrante uvular	Regen	
s	s	fricativa alveolar surda	Messer	s
ʃ	S	Fricativa pós-alveolar surda	Fischer	S
t	t	Oclusiva alveolar surda	Topf	T
t͡s	Ts	Africada alveolar surda	Zahl	
t͡ʃ	tS	Africada pós-alveolar surda	deutsch	S
v	v	Fricativa labiodental sonora	Wasser	f
x	x	Fricativa velar surda	kochen	k
z	z	Fricativa alveolar sonora	Consulte	s
ʒ	Z	Fricativa pós-alveolar sonora	Orange	S
Vogais				
øː	2:	vogal anterior semifechada longa arredondada	böse	o

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɐ	6	vogal central quase aberta	besser	a
ɐ̯	6_^	vogal central quase aberta não silábica	Klar	a
œ	9	vogal anterior média aberta arredondada	können	U
ə	@	vogal central média	Rede	@
a	a	vogal anterior aberta não arredondada	Salz	a
a:	a:	vogal anterior aberta longa não arredondada	Sahne	a
aɪ	aI	ditongo	nein	a
aʊ	aU	ditongo	Augen	a
ɑ̃	A~	vogal posterior aberta nasal não arredondada	Restaurant	a
p:	p:	vogal anterior média fechada longa não arredondada	Rede	p
ɛ	E	vogal anterior média aberta não arredondada	Keller	E

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɛ̃	E~	vogal anterior média aberta nasal não arredondada	Terrain	E
i:	i:	vogal anterior fechada longa não arredondada	Lied	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	bitte	i
o:	o:	vogal posterior média fechada longa arredondada	Kohl	o
ɔ	U	vogal posterior média aberta arredondada	Koffer	U
ɔ̃	U~	vogal posterior média aberta nasal arredondada	Annonce	U
ɔʏ	OY	ditongo	neu	U
u:	u:	vogal posterior fechada longa arredondada	Bruder	u
ʊ	U	vogal quase posterior quase fechada arredonda da	Wunder	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
y:	y:	vogal anterior fechada longa arredondada	kühl	u
ʏ	S	vogal quase anterior quase fechada arredonda da	Küche	u
Símbolos adicionais				
ˈ	“	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Hindi (hi-IN)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e o tipo de som do fonema para vozes do hindi que são compatíveis com o Amazon Polly.

Para obter fonemas adicionais usados em conjunto com o hindi, consulte Inglês indiano (en-IN).

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo
Consoantes			
pʰ	p_h	oclusiva bilabial aspirada surda	फूल (phool)
bʱ	b_h	oclusiva bilabial aspirada sonora	भारी (bhaari)

IPA	X-SAMPA	Descrição	Exemplo
t̪	t_d	oclusiva dental surda	तापमान (taapmaan)
t̪ʰ	t_d_h	oclusiva dental aspirada surda	थोड़ा (thoda)
d̪	d_d	oclusiva dental sonora	दिल्ली (dilli)
d̪ʰ	d_d_h	oclusiva dental aspirada sonora	धोबी (dhobi)
ʈ	t`	oclusiva retroflexa surda	कटोरा (katora)
ʈʰ	t`_h	oclusiva retroflexa aspirada surda	ठंड (thand)
ɖ	d`	oclusiva retroflexa sonora	डर (darr)
ɖʰ	d`_h	oclusiva retroflexa aspirada sonora	ढाल (dhal)
tʃʰ	tS_h	africada palatal aspirada surda	छाल (chaal)
dʒʰ	dZ_h	africada palatal aspirada sonora	झाल (jhaal)
kʰ	k_h	oclusiva velar aspirada surda	खान (khan)
gʰ	g_h	oclusiva velar aspirada sonora	घान (ghaan)
ɳ	n`	nasal retroflexa	क्षण (kshan)
ɾ	4	flape alveolar	राम (ram)
ɽ	r`	flape retroflexa simples	बड़ा (bada)

IPA	X-SAMPA	Descrição	Exemplo
ɽʱ	r`_h	flape retroflexa aspirada sonora	बढ़ी (barhi)
ʋ	v\	aproximadamente bilabial	वसूल (wasool)
Vogais			
ə	@_o	vogal central média	अच्छा (achhaa)
ə̃	@~	vogal central média nasal	हँसना (hansnaa)
a	A_o	vogal anterior aberta não arredondada	आग (aag)
ã	A~	vogal anterior aberta não arredondada nasal	घ़डयिँा (ghariyaan)
ɪ	I_o	vogal quase anterior quase fechada não arredondada	इक्कीस (ikkees)
ɪ̃	I~	vogal quase anterior quase fechada não arredondada nasal	संचिाई (sinchai)
i	i_o	vogal anterior fechada não arredondada	बल्लिी (billee)
ĩ	i~	vogal anterior fechada não arredondada nasal	नही (nahin)
ʊ	U_o	vogal quase posterior quase fechada arredonda da	उलूल (ullu)
ʊ̃	U~	vogal quase posterior quase fechada arredonda da nasal	मुँह (munh)

IPA	X-SAMPA	Descrição	Exemplo
u	u_o	vogal posterior fechada arredondada	फूल (phool)
ũ	u~	vogal posterior fechada arredondada nasal	ऊँट (oont)
ɔ	O_o	vogal posterior média aberta arredondada	कौन (kaun)
ɔ̃	U~	vogal posterior média aberta arredondada nasal	भौ (bhaun)
o	o	vogal posterior semifechada arredondada	सोना (sona)
õ	o~	vogal posterior semifechada arredondada nasal	क्यो (kyon)
ɛ	E_o	vogal anterior média aberta não arredondada	पैसा (paisa)
ɛ̃	E~	vogal anterior média aberta não arredondada nasal	मैं (main)
p	p	vogal anterior semifechada não arredondada	एक (ek)
p̃	p~	vogal anterior semifechada não arredondada nasal	कितबें (kitabein)

Islandês (is-IS)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do islandês que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	grasbakkanum	0
c	c	oclusiva palatal surda	pakkin	k
c^h	c_h	oclusiva palatal surda aspirada	anarkistai	k
ç	C	fricativa palatal surda	héðan	k
d	d	oclusiva alveolar sonora	bóndi	t
ð	D	fricativa dental sonora	borð	T
f	f	fricativa labiodental surda	duft	f
g	g	oclusiva velar sonora	holgóma	k
ɣ	G	fricativa velar sonora	hugur	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
h	h	fricativa glotal surda	heili	k
j	j	aproximante palatal	jökull	i
kʰ	k_h	aspirated voiceless velar plosive	ósköpunum	k
l	l	aproximante lateral alveolar	gólf	t
l̥	l_0	aproximante lateral alveolar surda	fólk	t
m	m	nasal bilabial	setembro	p
m̥	m_0	nasal bilabial surda	kompa	p
n	n	nasal alveolar	númer	t
n̥	n_0	nasal alveolar surda	pöntun	t
ɲ	J	nasal palatal	pælingar	J
ŋ	N	nasal velar	söngvarann	k
ŋ̊	N_0	nasal velar surda	frænka	k
pʰ	p_h	oclusiva bilabial surda aspirada	afplánun	p
r	r	vibrante alveolar	afskrifta	r
r̥	r_0	muda alveolar sonora	andvörpum	r

IPA	X-SAMPA	Descrição	Exemplo	Visema
s	s	fricativa alveolar surda	baðhús	s
t^h	t_h	oclusiva alveolar surda aspirada	tanki	t
θ	T	fricativa dental surda	þeldökki	T
v	v	fricativa labiodental sonora	silfur	f
w	w	aproximante labiovelar		u
x	x	fricativa velar surda	samfélags	k
Vogais				
œ	9	vogal anterior média aberta arredondada	þröskuldinum	U
œ:	9:	vogal anterior média aberta longa arredondada	tvö	U
a	a	vogal anterior aberta não arredondada	nefna	a
a:	a:	vogal anterior aberta longa não arredondada	fara	a
au	au	ditongo	átta	a
au:	au:	ditongo	átján	a

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɛ	E	vogal anterior média aberta não arredondada	kennari	E
ɛː	E:	vogal anterior média aberta longa não arredondada	dreka	E
i	i	vogal anterior fechada não arredondada	Gúlíver	i
iː	i:	vogal anterior fechada longa não arredondada	þrír	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	samspil	i
ɪː	I:	vogal quase anterior quase fechada longa não arredondada	stig	i
ɔ	U	vogal posterior média aberta arredondada	regndropar	U
ɔː	U:	vogal posterior média aberta longa arredondada	ullarbolur	U
ɔu	Ou	ditongo	tólf	U
ɔuː	Ou:	ditongo	fjórir	U

IPA	X-SAMPA	Descrição	Exemplo	Visema
u	u	vogal posterior fechada arredonda da	stúlkan	u
u:	u:	vogal posterior fechada longa arredondada	frú	u
ʏ	S	vogal quase anterior quase fechada arredonda da	tíu	u
ʏː	S	vogal quase anterior quase fechada longa arredondada	gruninn	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Italiano (it-IT)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do italiano que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bacca	p
d	d	oclusiva alveolar sonora	dama	t
͡dz	dz	africada alveolar sonora	zero	s
͡dʒ	dZ	africada pós-alveolar sonora	giro	S
f	f	fricativa labiodental surda	famiglia	f
g	g	oclusiva velar sonora	gatto	k
h	h	fricativa glotal surda	horror	k
j	j	aproximante palatal	dieci	i
k	k	oclusiva velar surda	campo	k
l	l	aproximante lateral alveolar	lido	t
ʎ	L	aproximante lateral palatal	aglio	J
m	m	nasal bilabial	mille	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
n	n	nasal alveolar	nove	t
ɲ	J	nasal palatal	lasagne	J
p	p	oclusiva bilabial surda	pizza	p
r	r	vibrante alveolar	risata	r
s	s	fricativa alveolar surda	sei	s
ʃ	S	fricativa pós-alveolar surda	scienza	S
t	t	oclusiva alveolar surda	tavola	t
͡ts	ts	africada alveolar surda	forza	s
͡tʃ	tS	africada pós-alveolar surda	cielo	S
v	v	fricativa labiodental sonora	venti	f
w	w	aproximante labiovelar	quattro	u
z	z	fricativa alveolar sonora	bisogno	s
ʒ	Z	fricativa pós-alveolar sonora	bijou	S
Vogais				

IPA	X-SAMPA	Descrição	Exemplo	Visema
a	a	vogal anterior aberta não arredondada	arco	a
p	p	vogal anterior semifechada não arredondada	tre	p
ɛ	E	vogal anterior média aberta não arredondada	ettaro	E
i	i	vogal anterior fechada não arredondada	impero	i
o	o	vogal posterior semifechada arredondada	cento	o
ɔ	U	vogal posterior média aberta arredondada	otto	U
u	u	vogal posterior fechada arredonda da	uno	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Japonês (ja-JP)

O Amazon Polly é compatível com os alfabetos de pronúncia Kana e Yomigana para japonês. Para fazer com que o Amazon Polly use a pronúncia fonética com esses alfabetos, use o atributo de fonema `alphabet="x-amazon-`*`phonetic standard used`*`"`.

- `x-amazon-pron-kana`: indica que a pronúncia Kana é usada. Pronúncia Kana são caracteres Katakana especiais usados para transcrição fonética e podem codificar o sotaque de tom.
- `x-amazon-yomigana`: indica que o Yomigana é usado. Yomigana pode ser um alfabeto convencional em Katakana, Hiragana e latim, interpretado como romanização Hepburn.

Os exemplos a seguir mostram o uso.

Pronúncia Kana

```
<speak>
     ###<phoneme alphabet="x-amazon-pron-kana" ph="###'#">##</phoneme>###
</speak>
```

Yomigana

```
<speak>
     ###<phoneme alphabet="x-amazon-yomigana" ph="####">##</phoneme>###
     ###<phoneme alphabet="x-amazon-yomigana" ph="####">##</phoneme>###
     ###<phoneme alphabet="x-amazon-yomigana" ph="Hirokazu">##</phoneme>###
</speak>
```

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do japonês que são compatíveis com o Amazon Polly.

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
ɾ	4	flape alveolar	練習, renshuu	t
ʔ	?	oclusiva glotal	あっ, atsu'	

IPA	X-SAMPA	Descrição	Exemplo	Visema
b	b	oclusiva bilabial sonora	舞踊, buyou	p
β	B	fricativa bilabial sonora	ヴィンテージ, vinteeji	B
c	c	oclusiva palatal surda	ききょう, kikyou	k
ç	C	fricativa palatal surda	人, hito	k
d	d	oclusiva alveolar sonora	濁点, dakuten	t
͡dʑ	dz\	africada palato-alveolar sonora	純, jun	J
g	g	oclusiva velar sonora	ご飯, gohan	k
h	h	fricativa glotal surda	本, hon	k
j	j	aproximante palatal	屋根, yane	i
ɟ	J\	oclusiva palatal sonora	行儀, gyougi	J
k	k	oclusiva velar surda	漢字, kanji	k
ɺ	l\	flape lateral alveolar	釣り, tsuri	r

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɺj	l\j	flape lateral alveolar, aproximante palatal	流行, ryuukou	r
m	m	nasal bilabial	飯, meshi	p
n	n	nasal alveolar	猫, neko	t
ɲ	J	nasal palatal	日本, nippon	J
ɴ	N\	nasal uvular	缶, kan	k
p	p	oclusiva bilabial surda	パン, pan	p
ɸ	p\	fricativa bilabial surda	福, huku	f
s	s	fricativa alveolar surda	層, sou	s
ɕ	s\	fricativa palato-alveolar surda	書簡, shokan	J
t	t	oclusiva alveolar surda	手紙, tegami	t
͡ts	ts	africada alveolar surda	釣り, tsuri	s
͡tɕ	ts\	africada palato-alveolar surda	吉, kichi	J
w	w	aproximante labiovelar	電話, denwa	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
z	z	fricativa alveolar sonora	座敷, zashiki	s
Vogais				
äː	a:_"	vogal central aberta longa não arredondada	羽蟻, haari	a
ä	a_"	vogal aberta central não arredondada	仮名, kana	a
eː	e:_o	vogal anterior média longa não arredondada	学生, gakusei	@
p	e_o	vogal anterior média não arredondada	歴, reki	@
i	i	vogal anterior fechada não arredondada	気, ki	i
iː	i:	vogal anterior fechada longa não arredondada	詩歌, shiika	i
ɯ	M	vogal posterior fechada não arredondada	運, un	i
ɯː	M:	vogal posterior fechada longa não arredondada	宗教, shuukyou	i

IPA	X-SAMPA	Descrição	Exemplo	Visema
oː	o:_o	vogal anterior média longa arredondada	購読, koodoku	o
o	o_o	vogal anterior média arredondada	読者, dokusha	o

Coreano (ko-KR)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do coreano que são compatíveis com o Amazon Polly.

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
k	k	oclusiva velar surda	강, [g]ang	k
k#	k_t	oclusiva velar surda forte	깨, [kk]e	k
n	n	nasal alveolar	남, [n]am	t
t	t	oclusiva alveolar surda	도, [d]o	t
t#	t_t	oclusiva alveolar surda forte	때, [tt]e	t
ɾ	4	flape alveolar	사랑, sa[r]ang	t
l	l	aproximante lateral alveolar	돌, do[l]	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
m	m	nasal bilabial	무, [m]u	p
p	p	oclusiva bilabial surda	봄, [b]om	p
p#	p_t	oclusiva bilabial surda forte	뻘, [pp]eol	p
s	s	fricativa alveolar surda	새, [s]e	s
s#	s_t	fricativa alveolar surda forte	씨, [ss]i	s
ŋ	N	nasal velar	방, ba[ng]	k
t͡ɕ	ts\	africada palato-alveolar surda	조, [j]o	J
t͡ɕ#	ts_t	africada palato-alveolar surda forte	찌, [jj]i	J
t͡ɕʰ	ts_h	africada palato-alveolar surda aspirada	차, [ch]a	J
kʰ	k_h	aspirated voiceless velar plosive	코, [k]o	k
tʰ	t_h	oclusiva alveolar surda aspirada	통, [t]ong	t
pʰ	p_h	oclusiva bilabial surda aspirada	패, [p]e	p
h	h	fricativa glotal surda	힘, [h]im	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
j	j	aproximante palatal	양, [y]ang	i
w	w	aproximante labiovelar	왕, [w]ang	u
ɯj	M\	aproximadamente velar>	의, [wj]i	i
Vogais				
a	a	vogal anterior aberta não arredondada	밥, b[a]b	a
ʌ	V	vogal posterior média aberta não arredondada	정, j[eo]ng	E
ɛ	E	vogal anterior média aberta não arredondada	배, b[e]	E
o	o	vogal posterior semifechada arredondada	노, n[o]	o
u	u	vogal posterior fechada arredondada	둘, d[u]l	u
ɯ	M	vogal posterior fechada não arredondada	은, [eu]n	i

IPA	X-SAMPA	Descrição	Exemplo	Visema
i	i	vogal anterior fechada não arredondada	김, k[i]m	i

Norueguês (nb-NO)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes, conforme compatíveis com o Amazon Polly para vozes do norueguês.

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
ɾ	4	flape alveolar	prøv	t
b	b	oclusiva bilabial sonora	labb	p
ç	C	fricativa palatal surda	kino	k
d	d	oclusiva alveolar sonora	ladd	t
ɖ	d`	oclusiva retroflexa sonora	verdi	t
f	f	fricativa labiodental surda	fot	f
g	g	oclusiva velar sonora	tagg	k
h	h	fricativa glotal surda	ha	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
j	j	aproximante palatal	gi	i
k	k	oclusiva velar surda	takk	k
l	l	aproximante lateral alveolar	fall, ball	t
ɭ	l`	aproximante lateral retroflexa	ærlig	t
m	m	nasal bilabial	lam	p
n	n	nasal alveolar	vann	t
ɳ	n`	nasal retroflexa	garn	t
ŋ	N	nasal velar	sang	k
p	p	oclusiva bilabial surda	hopp	p
s	s	fricativa alveolar surda	lass	s
ʂ	s`	fricativa retroflexa surda	års	S
ʃ	S	fricativa pós-alveolar surda	skyt	S
t	t	oclusiva alveolar surda	lat	t
ʈ	t`	oclusiva retroflexa surda	hardt	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʋ	v\	aproximante labiodental	vin	f
w	w	aproximante labiovelar	will	x
Vogais				
øː	2:	vogal anterior semifechada longa arredondada	søt	o
œ	9	vogal anterior média aberta arredondada	søtt	U
ə	@	vogal central média	ape	@
æː	{:	vogal anterior quase aberta longa não arredondada	vær	a
ʉ	}	vogal fechada central arredonda da	lund	u
ʉː	}:	vogal fechada central longa arredondada	lun	u
æ	{	vogal anterior quase aberta não arredondada	vært	a
ɑ	A	vogal posterior aberta não arredondada	hatt	a

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɑː	A:	vogal posterior aberta longa não arredondada	hat	a
pː	p:	vogal anterior média fechada longa não arredondada	sen	p
ɛ	E	vogal anterior média aberta não arredondada	send	E
iː	i:	vogal anterior fechada longa não arredondada	vin	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	vind	i
oː	o:	vogal posterior média fechada longa arredondada	våt	o
ɔ	U	vogal posterior média aberta arredondada	vått	U
uː	u:	vogal posterior fechada longa arredondada	bok	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʊ	U	vogal quase posterior quase fechada arredondada	bukk	u
y:	y:	vogal anterior fechada longa arredondada	lyn	u
ʏ	S	vogal quase anterior quase fechada arredondada	lynne	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Polonês (pl-PL)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do polonês que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bobas, belka	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
d	d	oclusiva alveolar sonora	dar, do	t
͡dz	dz	africada alveolar sonora	dzwon, widzowie	s
͡dʑ	dz\	africada palato-alveolar sonora	dźwięk	J
͡dʐ	dz`	africada retroflexa sonora	dżem, dżungla	S
f	f	fricativa labiodental surda	furtka, film	f
g	g	oclusiva velar sonora	gazeta, waga	k
h	h	fricativa glotal surda	chleb, handel	k
j	j	aproximante palatal	jak, maja	i
k	k	oclusiva velar surda	kura, marek	k
l	l	aproximante lateral alveolar	lipa, alicja	t
m	m	nasal bilabial	matka, molo	p
n	n	nasal alveolar	norka	t
ɲ	J	nasal palatal	koń, toruń	J
p	p	oclusiva bilabial surda	pora, stop	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
r	r	vibrante alveolar	rok, park	r
s	s	fricativa alveolar surda	sum, pas	s
ɕ	s\	fricativa palato-alveolar surda	śruba, śnieg	J
ʂ	s`	fricativa retroflexa surda	szum, masz	S
t	t	oclusiva alveolar surda	tok, stół	t
t͡s	ts	africada alveolar surda	car, co	s
t͡ɕ	ts\	africada palato-alveolar surda	ćma, mieć	J
t͡ʂ	ts`	africada retroflexa surda	czas, raczej	S
v	v	fricativa labiodental sonora	worek, mewa	f
w	w	aproximante labiovelar	łaska, mało	u
z	z	fricativa alveolar sonora	zero	s
ʑ	z\	fricativa palato-alveolar sonora	źrebię, bieliźnie	J
ʐ	z`	fricativa retroflexa sonora	żar, żona	S

IPA	X-SAMPA	Descrição	Exemplo	Visema
Vogais				
a	a	vogal anterior aberta não arredondada	ja	a
ɛ	E	vogal anterior média aberta não arredondada	echo	E
ɛ̃	E~	vogal anterior média aberta nasal não arredondada	węże	E
i	i	vogal anterior fechada não arredondada	ile	i
ɔ	U	vogal posterior média aberta arredondada	oczy	U
ɔ̃	U~	vogal posterior média aberta nasal arredondada	wąż	U
u	u	vogal posterior fechada arredonda da	uczta	u
ɨ	1	vogal central fechada não arredondada	byk	i
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	

IPA	X-SAMPA	Descrição	Exemplo	Visema
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Português (pt-PT)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do português europeu que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
ɾ	4	flape alveolar	pira	t
b	b	oclusiva bilabial sonora	dato	p
d	d	oclusiva alveolar sonora	dato	t
f	f	fricativa labiodental surda	facto	f
g	g	oclusiva velar sonora	gato	k
j	j	aproximante palatal	paraguay	i
k	k	oclusiva velar surda	cacto	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
l	l	aproximante lateral alveolar	galo	t
ʎ	L	aproximante lateral palatal	galho	J
m	m	nasal bilabial	mato	p
n	n	nasal alveolar	nato	t
ɲ	J	nasal palatal	pinha	J
p	p	oclusiva bilabial surda	pato	p
ʁ	R\	vibrante uvular	barroso	k
s	s	fricativa alveolar surda	saca	s
ʃ	S	fricativa pós-alveolar surda	chato	S
t	t	oclusiva alveolar surda	tacto	t
v	v	fricativa labiodental sonora	vaca	f
w	w	aproximante labiovelar	mau	u
z	z	fricativa alveolar sonora	zaca	s
ʒ	Z	fricativa pós-alveolar sonora	jacto	S

IPA	X-SAMPA	Descrição	Exemplo	Visema
Vogais				
a	a	vogal anterior aberta não arredondada	parto	a
ã	a~	vogal anterior aberta nasal não arredondada	pega	a
p	p	vogal anterior semifechada não arredondada	pega	p
p̃	p~	vogal anterior nasal semifechada não arredondada	movem	p
ɛ	E	vogal anterior média aberta não arredondada	café	E
i	i	vogal anterior fechada não arredondada	lingueta	i
ĩ	i~	vogal anterior nasal fechada não arredondada	cinto	i
o	o	vogal posterior semifechada arredondada	poder	o
õ	o~	vogal posterior nasal semifechada arredondada	compra	o

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɔ	U	vogal posterior média aberta arredondada	cotó	U
u	u	vogal posterior fechada arredondada	fui	u
ũ	u~	vogal posterior fechada nasal arredondada	sunto	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Português brasileiro (pt-BR)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do português brasileiro que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
ɾ	4	flape alveolar	pira	t
b	b	oclusiva bilabial sonora	bato	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
d	d	oclusiva alveolar sonora	dato	t
͡dʒ	dZ	africada pós-alveolar sonora	idade	S
f	f	fricativa labiodental surda	facto	f
g	g	oclusiva velar sonora	gato	k
j	j	aproximante palatal	paraguay	i
k	k	oclusiva velar surda	cacto	k
l	l	aproximante lateral alveolar	galo	t
ʎ	L	aproximante lateral palatal	galho	J
m	m	nasal bilabial	mato	p
n	n	nasal alveolar	nato	t
ɲ	J	nasal palatal	pinha	J
p	p	oclusiva bilabial surda	pato	p
s	s	fricativa alveolar surda	saca	s
ʃ	S	fricativa pós-alveolar surda	chato	S

IPA	X-SAMPA	Descrição	Exemplo	Visema
t	t	oclusiva alveolar surda	tacto	t
͡tʃ	tS	africada pós-alveolar surda	noite	S
v	v	fricativa labiodental sonora	vaca	f
w	w	aproximante labiovelar	mau	u
χ	X	fricativa uvular surda	carro	k
z	z	fricativa alveolar sonora	zaca	s
ʒ	Z	fricativa pós-alveolar sonora	jacto	S
Vogais				
a	a	vogal anterior aberta não arredondada	parto	a
ã	a~	vogal anterior aberta nasal não arredondada	pensamos	a
p	p	vogal anterior semifechada não arredondada	pega	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
p̃	p~	vogal anterior nasal semifechada não arredondada	movem	p
ɛ	E	vogal anterior média aberta não arredondada	café	E
i	i	vogal anterior fechada não arredondada	lingueta	i
ĩ	i~	vogal anterior nasal fechada não arredondada	cinto	i
o	o	vogal posterior semifechada arredondada	poder	o
õ	o~	vogal posterior nasal semifechada arredondada	compra	o
ɔ	U	vogal posterior média aberta arredondada	cotó	U
u	u	vogal posterior fechada arredondada	fui	u
ũ	u~	vogal posterior fechada nasal arredondada	sunto	u
Símbolos adicionais				

IPA	X-SAMPA	Descrição	Exemplo	Visema
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Romeno (ro-RO)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do romeno que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bubă	p
d	d	oclusiva alveolar sonora	după	t
͡dʒ	dZ	africada pós-alveolar sonora	george	S
f	f	fricativa labiodental surda	afacere	f
g	g	oclusiva velar sonora	agri#	k
h	h	fricativa glotal surda	harpă	k

IPA	X-SAMPA	Descrição	Exemplo	Visema
j	j	aproximante palatal	baie	i
k	k	oclusiva velar surda	co#	k
l	l	aproximante lateral alveolar	lampa	t
m	m	nasal bilabial	mama	p
n	n	nasal alveolar	nor	t
p	p	oclusiva bilabial surda	pilă	p
r	r	vibrante alveolar	rampă	r
s	s	fricativa alveolar surda	soare	s
ʃ	S	fricativa pós-alveolar surda	ma#ină	S
t	t	oclusiva alveolar surda	tata	t
͡ts	ts	africada alveolar surda	#ară	s
͡tʃ	tS	africada pós-alveolar surda	ceai	S
v	v	fricativa labiodental sonora	via#ă	f
w	w	aproximante labiovelar	beau	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
z	z	fricativa alveolar sonora	mozol	s
ʒ	Z	fricativa pós-alveolar sonora	joacă	S
Vogais				
ə	@	vogal central média	babă	@
a	a	vogal anterior aberta não arredondada	casa	a
p	p	vogal anterior semifechada não arredondada	elan	p
p̯	p_^	vogal anterior média fechada não silábica não arredondada	beau	p
i	i	vogal anterior fechada não arredondada	mie	i
o	o	vogal posterior média fechada arredondada	oră	o
oa	o_^a	ditongo	oare	o
u	u	vogal posterior fechada arredondada	unde	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɨ	1	vogal central fechada não arredondada	România	i
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Russo (ru-RU)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do russo que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	борт	p
bʲ	b'	oclusiva bilabial sonora palatalizada	бюро	p
d	d	oclusiva alveolar sonora	дом	t
dʲ	d'	oclusiva alveolar sonora palatalizada	дядя	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
f	f	fricativa labiodental surda	флаг	f
f^j	f'	fricativa labiodental surda palatalizada	февраль	f
g	g	oclusiva velar sonora	нога	k
g^j	g'	oclusiva velar sonora palatalizada	герой	k
j	j	aproximante palatal	дизайн, ящик	i
k	k	oclusiva velar surda	кот	k
k^j	k'	oclusiva velar surda palatalizada	кино	k
l	l	aproximante lateral alveolar	лампа	t
l^j	l'	aproximante lateral alveolar palataliz ada	лес	t
m	m	nasal bilabial	мама	p
m^j	m'	nasal bilabial palatalizada	мяч	p
n	n	nasal alveolar	нос	t
n^j	n'	nasal alveolar palatalizada	няня	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
p	p	oclusiva bilabial surda	папа	p
pʲ	p'	oclusiva bilabial surda palatalizada	перо	p
r	r	vibrante alveolar	роза	r
rʲ	r'	vibrante alveolar palatalizada	рюмка	r
s	s	fricativa alveolar surda	сыр	s
sʲ	s'	fricativa alveolar surda palatalizada	сердце, русь	s
ɕː	s\:	fricativa palato-al veolar surda longa	щека	J
ʂ	s`	fricativa retroflexa surda	шум	S
t	t	oclusiva alveolar surda	точка	t
tʲ	t'	oclusiva alveolar surda palatalizada	тётя	t
t͡s	ts	africada alveolar surda	царь	s
t͡ɕ	ts\	africada palato-al veolar surda	час	J
v	v	fricativa labiodental sonora	вор	f

IPA	X-SAMPA	Descrição	Exemplo	Visema
vʲ	v'	fricativa labiodental sonora palatalizada	верфь	f
x	x	fricativa velar surda	хор	k
xʲ	x'	fricativa velar surda palatalizada	химия	k
z	z	fricativa alveolar sonora	зуб	s
zʲ	z'	fricativa alveolar sonora palatalizada	зима	s
ʑː	z\:	fricativa palato-alveolar sonora longa	уезжать	J
ʐ	z`	fricativa retroflexa sonora	жена	S
Vogais				
ə	@	vogal central média	канарейка	@
a	a	vogal anterior aberta não arredondada	два, яблоко	a
p	p	vogal anterior semifechada não arredondada	печь	p
ɛ	E	vogal anterior média aberta não arredondada	это	E

IPA	X-SAMPA	Descrição	Exemplo	Visema
i	i	vogal anterior fechada não arredondada	один, четыре	i
o	o	vogal posterior semifechada arredondada	кот	o
u	u	vogal posterior fechada arredondada	муж, вьюга	u
ɨ	1	vogal central fechada não arredondada	мышь	i

Espanhol (es-ES)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do espanhol que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
ɾ	4	flape alveolar	pero, bravo, amor, eterno	t
b	b	oclusiva bilabial sonora	bestia	p
β	B	fricativa bilabial sonora	bebé	B

IPA	X-SAMPA	Descrição	Exemplo	Visema
d	d	oclusiva alveolar sonora	cuando	t
ð	D	fricativa dental sonora	arder	T
f	f	fricativa labiodental surda	fase, café	f
g	g	oclusiva velar sonora	gato, lengua, guerra	k
ɣ	G	fricativa velar sonora	trigo, Argos	k
j	j	aproximante palatal	hacia, tierra, radio, viuda	i
ʝ	j\	fricativa palatal sonora	enhielar, sayo, inyectado, desyerba	J
k	k	oclusiva velar surda	caña, laca, quisimos	k
l	l	aproximante lateral alveolar	lino, calor, principal	t
ʎ	L	aproximante lateral palatal	llave, pollo	J
m	m	nasal bilabial	madre, comer, anfibio	p
n	n	nasal alveolar	nido, anillo, sin	t
ɲ	J	nasal palatal	cabaña, ñoquis	J

IPA	X-SAMPA	Descrição	Exemplo	Visema
ŋ	N	nasal velar	cinco, venga	k
p	p	oclusiva bilabial surda	pozo, topo	p
r	r	vibrante alveolar	perro, enrachado	r
s	s	fricativa alveolar surda	saco, casa, puertas	s
t	t	oclusiva alveolar surda	tamiz, átomo	t
͡tʃ	tS	africada pós-alveolar surda	chubasco	S
θ	T	fricativa dental surda	cereza, zorro, lacero, paz	T
w	w	aproximante labiovelar	fuego, fuimos, cuota, cuadro	u
x	x	fricativa velar surda	jamón, general, suje, reloj	k
z	z	fricativa alveolar sonora	rasgo, mismo	s
Vogais				
a	a	vogal anterior aberta não arredondada	tanque	a
p	p	vogal anterior semifechada não arredondada	peso	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
i	i	vogal anterior fechada não arredondada	cinco	i
o	o	vogal posterior semifechada arredondada	bosque	o
u	u	vogal anterior semifechada não arredondada	publicar	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Espanhol mexicano (es-MX)

A tabela a seguir lista os fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para a voz do espanhol mexicano que é compatível com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
ɾ	4	flape alveolar	pero, bravo, amor, eterno	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
b	b	oclusiva bilabial sonora	bestia	p
β	B	fricativa bilabial sonora	bebé	B
d	d	oclusiva alveolar sonora	cuando	t
ð	D	fricativa dental sonora	arder	T
f	f	fricativa labiodental surda	fase, café	f
g	g	oclusiva velar sonora	gato, lengua, guerra	k
ɣ	G	fricativa velar sonora	trigo, Argos	k
j	j	aproximante palatal	hacia, tierra, radio, viuda	i
ʝ	j\	fricativa palatal sonora	enhielar, sayo, inyectado, desyerba	J
k	k	oclusiva velar surda	caña, laca, quisimos	k
l	l	Aproximante alveolar lateral	lino, calor, principal	t
m	m	nasal bilabial	madre, comer, anfibio	p
n	n	nasal alveolar	nido, anillo, sin	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɲ	J	nasal palatal	cabaña, ñoquis	J
ŋ	N	nasal velar	angosto, increíble	k
p	p	oclusiva bilabial surda	pozo, topo	p
r	r	vibrante alveolar	perro, enrachado	r
s	s	fricativa alveolar surda	saco, casa, puertas	s
ʃ	S	fricativa pós-alveolar surda	show, flash	S
t	t	oclusiva alveolar surda	tamiz, átomo	t
⁀tʃ	tS	africada pós-alveolar surda	chubasco	S
w	w	aproximante labiovelar	fuego, fuimos, cuota, cuadro	u
x	x	fricativa velar surda	jamón, general, peaje, reloj	k
z	z	fricativa alveolar sonora	rasgo, mismo	s
Vogais				
a	a	vogal central aberta não arredondada	tanque	a

IPA	X-SAMPA	Descrição	Exemplo	Visema
p	p	vogal anterior semifechada não arredondada	peso	p
i	i	vogal anterior fechada não arredondada	cinco	i
o	o	vogal posterior semifechada arredondada	bosque	o
u	u	vogal posterior fechada arredondada	publicar	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Espanhol dos EUA (es-US)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do espanhol dos EUA que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɾ	4	flape alveolar	pero, bravo, amor, eterno	t
b	b	oclusiva bilabial sonora	bestia	p
β	B	fricativa bilabial sonora	bebé	B
d	d	oclusiva alveolar sonora	cuando	t
ð	D	fricativa dental sonora	arder	T
f	f	fricativa labiodental surda	fase, café	f
g	g	oclusiva velar sonora	gato, lengua, guerra	k
ɣ	G	fricativa velar sonora	trigo, Argos	k
j	j	aproximante palatal	hacia, tierra, radio, viuda	i
ʝ	j\	fricativa palatal sonora	enhielar, sayo, inyectado, desyerba	J
k	k	oclusiva velar surda	caña, laca, quisimos	k
l	l	Aproximante alveolar lateral	lino, calor, principal	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
m	m	nasal bilabial	madre, comer, anfibio	p
n	n	nasal alveolar	nido, anillo, sin	t
ɲ	J	nasal palatal	cabaña, ñoquis	J
ŋ	N	nasal velar	angosto, increíble	k
p	p	oclusiva bilabial surda	pozo, topo	p
r	r	vibrante alveolar	perro, enrachado	r
s	s	fricativa alveolar surda	saco, casa, puertas	s
ʃ	S	fricativa pós-alveolar surda	show, flash	S
t	t	oclusiva alveolar surda	tamiz, átomo	t
͡tʃ	tS	africada pós-alveolar surda	chubasco	S
w	w	aproximante labiovelar	fuego, fuimos, cuota, cuadro	u
x	x	fricativa velar surda	jamón, general, peaje, reloj	k
z	z	fricativa alveolar sonora	rasgo, mismo	s
Vogais				

IPA	X-SAMPA	Descrição	Exemplo	Visema
a	a	vogal central aberta não arredondada	tanque	a
p	p	vogal anterior semifechada não arredondada	peso	p
i	i	vogal anterior fechada não arredondada	cinco	i
o	o	vogal posterior semifechada arredondada	bosque	o
u	u	vogal posterior fechada arredonda da	publicar	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Sueco (sv-SE)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do sueco que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	bil	p
d	d	oclusiva alveolar sonora	dal	t
ɖ	d`	oclusiva retroflexa sonora	bord	t
f	f	fricativa labiodental surda	fil	f
g	g	oclusiva velar sonora	gås	k
h	h	fricativa glotal surda	hal	k
j	j	aproximante palatal	jag	i
k	k	oclusiva velar surda	kal	k
l	l	aproximante lateral alveolar	lös	t
ɭ	l`	aproximante lateral retroflexa	härlig	t
m	m	nasal bilabial	mil	p
n	n	nasal alveolar	nålar	t
ɳ	n`	nasal retroflexa	barn	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
ŋ	N	nasal velar	ring	k
p	p	oclusiva bilabial surda	pil	p
r	r	vibrante alveolar	ris	r
s	s	fricativa alveolar surda	sil	s
ɕ	s\	fricativa palato-al veolar surda	tjock	J
ʂ	s`	fricativa retroflexa surda	fors, schlager	S
t	t	oclusiva alveolar surda	teto	t
ʈ	t`	oclusiva retroflexa surda	hjort	t
v	v	fricativa labiodental sonora	vår	f
w	w	aproximante labiovelar	aula, airways	u
ɧ	x\	fricativa palatal velar surda	sjuk	k
Vogais				
ø	2	vogal anterior semifechada arredondada	föll, förr	o

IPA	X-SAMPA	Descrição	Exemplo	Visema
ø	2:	vogal anterior semifechada longa arredondada	föl, nöt, för	o
ɵ	8	vogal média central fechada arredonda da	buss, full	o
ə	@	vogal central média	pojken	@
ʉː	}:	vogal fechada central longa arredondada	hus, ful	u
a	a	vogal anterior aberta não arredondada	hall, matt	a
æ	{	vogal anterior quase aberta não arredondada	herr	a
ɑː	A:	vogal posterior aberta longa não arredondada	hal, mat	a
p:	p:	vogal anterior média fechada longa não arredondada	vet, hel	p
ɛ	E	vogal anterior média aberta não arredondada	vett, rätt, hetta, häll	E

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɛː	E:	vogal anterior média aberta longa não arredondada	säl, häl, här	E:
iː	i:	vogal anterior fechada longa não arredondada	vit, sil	i:
ɪ	eu	vogal quase anterior quase fechada não arredondada	vitt, sill	i
oː	o:	vogal posterior média fechada longa arredondada	hål, mål	o
ɔ	U	vogal posterior média aberta arredondada	håll, moll	U
uː	u:	vogal posterior fechada longa arredondada	sol, bot	u
ʊ	U	vogal quase posterior quase fechada arredondada	bott	u
y	y	vogal anterior fechada arredondada	bytt	u
yː	y:	vogal anterior fechada longa arredondada	syl, syl	u

IPA	X-SAMPA	Descrição	Exemplo	Visema
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Turco (tr-TR)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do turco que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
ɾ	4	flape alveolar	durum	t
r̥	4_0_r	flape alveolar fricado surdo	bir	t
ɾ̞	4_r	flape alveolar fricado	raf	t
b	b	oclusiva bilabial sonora	raf	p
c	c	oclusiva palatal surda	kedi	k
d	d	oclusiva alveolar sonora	dede	t

IPA	X-SAMPA	Descrição	Exemplo	Visema
͡dʒ	dZ	africada pós-alveolar sonora	cam	S
f	f	fricativa labiodental surda	fare	f
g	g	oclusiva velar sonora	galibi	k
h	h	fricativa glotal surda	hasta	k
j	j	aproximante palatal	yat	i
ɟ	J\	oclusiva palatal sonora	genç	J
k	k	oclusiva velar surda	akıl	k
l	l	aproximante lateral alveolar	lale	t
ɫ	5	aproximante lateral alveolar velarizada	labirent	t
m	m	nasal bilabial	maaş	p
n	n	nasal alveolar	anı	t
p	p	oclusiva bilabial surda	ip	p
s	s	fricativa alveolar surda	ses	s

IPA	X-SAMPA	Descrição	Exemplo	Visema
ʃ	S	fricativa pós-alveolar surda	aşı	S
t	t	oclusiva alveolar surda	ütü	t
ʧ	tS	africada pós-alveolar surda	çaba	S
v	v	fricativa labiodental sonora	ekvator, kahveci, akvaryum, isveçli, teşviki, cetvel	f
z	z	fricativa alveolar sonora	ver	s
ʒ	Z	fricativa pós-alveolar sonora	azık	S
Vogais				
ø	2	vogal anterior semifechada arredondada	göl	0
œ	9	vogal anterior média aberta arredondada	banliyö	U
a	a	vogal anterior aberta não arredondada	kal	a
a:	a:	vogal anterior aberta longa não arredondada	davacı	a

IPA	X-SAMPA	Descrição	Exemplo	Visema
æ	{	vogal anterior quase aberta não arredondada	özlem, güvenlik, gürel, somersault	a
p	p	vogal anterior semifechada não arredondada	keçi	p
ɛ	E	vogal anterior média aberta não arredondada	dede	E
i	i	vogal anterior fechada não arredondada	bir	i
i:	i:	vogal anterior fechada longa não arredondada	izah	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	keçi	i
ɯ	M	vogal posterior fechada não arredondada	kıl	i
o	o	vogal posterior semifechada arredondada	kol	o
o:	o:	vogal posterior média fechada longa arredondada	dolar	o

IPA	X-SAMPA	Descrição	Exemplo	Visema
u	u	vogal posterior fechada arredondada	durum	u
u:	u:	vogal posterior fechada longa arredondada	ruhum	u
ʊ	U	vogal quase posterior quase fechada arredondada	dolu	u
y	y	vogal anterior fechada arredondada	güvenlik	u
ʏ	S	vogal quase anterior quase fechada arredondada	aşı	u
Símbolos adicionais				
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Galês (cy-GB)

A tabela a seguir lista o conjunto completo dos fonemas do Alfabeto Fonético Internacional (AFI), os símbolos do Alfabeto fonético de métodos de avaliação da fala (X-SAMPA) e os visemas correspondentes para vozes do galês que são compatíveis com o Amazon Polly.

Tabela de fonemas/visemas

IPA	X-SAMPA	Descrição	Exemplo	Visema
Consoantes				
b	b	oclusiva bilabial sonora	baban	p
d	d	oclusiva alveolar sonora	deg	t
ʤ	dZ	africada pós-alveolar sonora	garej	S
ð	D	fricativa dental sonora	deuddeg	T
f	f	fricativa labiodental surda	ffacs	f
g	g	oclusiva velar sonora	gadael	k
h	h	fricativa glotal surda	haearn	k
j	j	aproximante palatal	astudio	i
k	k	oclusiva velar surda	cant	k
l	l	aproximante lateral alveolar	lan	t
ɬ	K	fricativa lateral alveolar surda	llan	t
m	m	nasal bilabial	mae	p

IPA	X-SAMPA	Descrição	Exemplo	Visema
m̥	m_0	nasal bilabial surda	ymhen	p
n	n	nasal alveolar	naw	t
n̥	n_0	nasal alveolar surda	anhawster	t
ŋ	N	nasal velar	argyfwng	k
ŋ̊	N_0	nasal velar surda	anghenion	k
p	p	oclusiva bilabial surda	pump	p
r	r	vibrante alveolar	rhoi	r
r̥	r_0	vibrante alveolar surda	garw	r
s	s	fricativa alveolar surda	saith	s
ʃ	S	fricativa pós-alveolar surda	siawns	S
t	t	oclusiva alveolar surda	tegan	t
t͡ʃ	tS	africada pós-alveolar surda	cytsain	S
θ	T	fricativa dental surda	aberth	T
v	v	fricativa labiodental sonora	prawf	f

IPA	X-SAMPA	Descrição	Exemplo	Visema
w	w	aproximante labiovelar	rhagweld	u
χ	X	fricativa uvular surda	chwech	k
z	z	fricativa alveolar sonora	aids	s
ʒ	Z	fricativa pós-alveolar sonora	rouge	S
Vogais				
ə	@	vogal central média	ychwanega	@
a	a	vogal anterior aberta não arredondada	acen	a
ai	ai	ditongo	dau	a
au	au	ditongo	awdur	a
ɑː	A:	vogal posterior aberta longa não arredondada	mab	a
ɑːɨ	A:1	ditongo	aelod	a
pː	p:	vogal anterior média fechada longa não arredondada	peth	p
ɛ	E	vogal anterior média aberta não arredondada	pedwar	E

IPA	X-SAMPA	Descrição	Exemplo	Visema
ɛi	Ei	ditongo	beic	E
i:	i:	vogal anterior fechada longa não arredondada	tri	i
ɪ	eu	vogal quase anterior quase fechada não arredondada	miliwn	i
ɨu	1u	ditongo	unigryw	i
o:	o:	vogal posterior média fechada longa arredondada	oddi	o
ɔ	U	vogal posterior média aberta arredondada	oddieithr	U
ɔi	Oi	ditongo	troi	U
ɔu	Ou	ditongo	rownd	U
u:	u:	vogal posterior fechada longa arredondada	cwch	u
ʊ	U	vogal quase posterior quase fechada arredonda da	acwstig	u
ʊi	Ui	ditongo	wyth	u
Símbolos adicionais				

IPA	X-SAMPA	Descrição	Exemplo	Visema
ˈ	"	tonicidade primária	Alabama	
ˌ	%	tonicidade secundária	Alabama	
.	.	separação silábica	A.la.ba.ma	

Mecanismos de voz do Amazon Polly

O Amazon Polly tem quatro mecanismos de voz que convertem o texto de entrada em fala realista. Isso inclui: Generativo Long-form, Neural e Padrão. Para usar uma voz do Amazon Polly, selecione um mecanismo e uma operação de API de síntese de fala. Depois, forneça texto de entrada para o mecanismo sintetizar e selecione um formato de saída de áudio. Considerando essas informações, o Amazon Polly sintetizará o texto fornecido em um fluxo de áudio de fala de alta qualidade.

As seções a seguir incluem detalhes sobre os mecanismos de voz oferecidos pelo Amazon Polly.

Tópicos

- Vozes generativas
- Long-form vozes
- Vozes neurais
- Vozes padrão
- Escolher um mecanismo de voz

Vozes generativas

O mecanismo generativo de conversão de texto em fala (TTS) do Amazon Polly oferece as vozes de conversação mais humanas, emocionais e adaptáveis disponíveis para uso por meio do console do Amazon Polly.

O mecanismo generativo é o maior modelo de TTS do Amazon Polly até o momento. Ele implanta um transformador de bilhões de parâmetros que converte texto bruto em códigos de fala, seguido por um decodificador baseado em convolução que converte esses códigos de fala em formas de onda de maneira incremental e transmissível. Esse método mostra as capacidades emergentes amplamente relatadas de grandes modelos de linguagens (LLMs) quando treinados em volumes crescentes de dados proprietários e disponíveis publicamente, abrangendo uma variedade de vozes, idiomas e estilos.

O mecanismo generativo cria uma fala sintética que é emocional, assertiva e altamente coloquial de uma maneira consideravelmente semelhante a uma voz humana. Você pode usar essas vozes como um experiente assistente para os clientes, treinador virtual ou anunciante com um discurso sintético quase humano.

> **Note**
>
> A tecnologia de ponta por trás dessas vozes se enquadra no paradigma da IA generativa para modelagem de linguagem e voz. Um efeito colateral da tecnologia é que qualquer atualização nos dados de treinamento e no modelo pode resultar em pequenas variações na forma como as vozes soam, mesmo no caso de sua qualidade geral melhorar com as atualizações do modelo. Isso pode ter um impacto nos casos de uso em que diferentes partes do conteúdo são sintetizadas por um longo período, por exemplo, uma temporada de podcasts.

Vozes generativas disponíveis

Atualmente, o Amazon Polly oferece 43 vozes em uma variante generativa.

	Idioma	Código do idioma	Name/ID	Gender
1	Inglês (australiano)	en-AU	Olivia	Feminino
2	Inglês (britânico)	en-GB	Amy Brian	Feminino Masculino
3	Inglês (indiano)	en-IN	Kajal	Feminino
4	Inglês (Irlanda)	en-IE	Niamh	Feminino
5	Inglês (neozelandês)	en-NZ	Aria	Feminino
6	Inglês (singapurense)	en-SG	Jasmine	Feminino
7	Inglês (sul-africano)	en-ZA	Ayanda	Feminino
8	Inglês (EUA)	en-US	Danielle	Feminino

	Idioma	Código do idioma	Name/ID	Gender
			Joanna	Feminino
			Matthew	Masculino
			Ruth	Feminino
			Salli	Feminino
			Stephen	Masculino
			Tiffany	Feminino
9	Holandês (Bélgica)	nl-BE	Lisa	Feminino
10	Holandês (Países Baixos)	nl-NL	Laura	Feminino
11	Francês (belga)	fr-BE	Isabelle	Feminino
12	Francês (canadense)	fr-CA	Gabrielle	Feminino
			Liam	Masculino
13	Francês (França)	fr-FR	Ambre	Feminino
			Céline	Feminino
			Florian	Masculino
			Léa	Feminino
			Rémi	Masculino
14	Alemão (Áustria)	de-AT	Hannah	Feminino

	Idioma	Código do idioma	Name/ID	Gender
15	Alemão (Alemanha)	de-DE	Daniel	Masculino
			Lennart	Masculino
			Vicki	Feminino
16	Alemão (suíço)	de-CH	Sabrina	Feminino
17	Italiano (Itália)	it-IT	Beatriz	Feminino
			Bianca	Feminino
			Lorenzo	Masculino
18	Coreano (Coréia)	ko-KR	Seoyeon	Feminino
19	Polonês (Polônia)	pl-PL	Ewa	Feminino
			Ola	Feminino
20	Português (brasileiro)	pt-BR	Camila	Feminino
21	Espanhol (mexicano)	es-MX	Andrés	Masculino
			Mía	Feminino
22	Espanhol (Espanha)	es-ES	Lúcia	Feminino
			Sergio	Masculino
23	Espanhol (EUA)	es-US	Lupe	Feminino
			Pedro	Masculino

> **Note**
>
> O custo das vozes generativas é especificado na página de informações sobre preços do Amazon Polly.

Compatibilidade de recursos e regiões

As vozes generativas do Amazon Polly estão disponíveis nas seguintes regiões:

- Leste dos EUA (Norte da Virgínia): us-east-1
- Europa (Frankfurt): eu-central-1
- Oeste dos EUA (Oregon): us-west-2
- Ásia-Pacífico (Tóquio): ap-northeast-1
- Ásia-Pacífico (Seul): ap-northeast-2
- Ásia-Pacífico (Singapura): ap-southeast-1
- Europa (Londres): eu-west-2
- Canadá (Central): ca-central-1
- Europa (Zurique): eu-central-2
- Outras regiões não estão disponíveis

Os seguintes recursos são compatíveis com as vozes generativas:

- A API de streaming bidirecional agora é oferecida no Generative Engine e permite streaming de entrada e saída ao mesmo tempo. Essa API está disponível nas seguintes AWS regiões: Leste dos EUA (Norte da Virgínia), Europa (Frankfurt), Oeste dos EUA (Oregon), Ásia-Pacífico (Cingapura), Europa (Londres), Canadá (Central) e Europa (Zurique). Visite a documentação para saber mais sobre como usá-lo.
- Real-time e operações assíncronas de síntese de fala.
- O estilo de fala de âncora não é compatível com o mecanismo Generativo.
- Muitas (mas não todas) etiquetas SSML são compatíveis com o Amazon Polly. Para obter mais informações sobre tags NTTS-supported SSML, consulte Tags SSML suportadas
- Assim como ocorre com vozes padrão, é possível escolher entre várias taxas de amostragem para otimizar a largura de banda e a qualidade do áudio para seu aplicativo. As taxas de amostragem

válidas para vozes padrão e neurais são 8 kHz, 16 kHz, 22 kHz ou 24 kHz. O padrão para vozes padrão é 22 kHz. O padrão para vozes generativas é 24 kHz. O Amazon Polly é compatível com os formatos de stream de áudio MP3, ogg_vorbis e PCM bruto.

Ainda não há suporte à geração de marcas de fala.

> **Note**
>
> Atualmente, as regiões da Europa (Londres), Canadá (Central) e Europa (Zurique) suportam apenas as seguintes vozes generativas: Joanna (en-US), Ruth (en-US), Salli (en-US), Stephen (en-US), Tiffany (en-US), Amy (en-GB), Brian (en-GB), Olivia (en-AU), Florian (fr-FR), Ambre (fr-FR) -fr), Lorenzo (it-IT), Beatrice (it-IT), Jasmine (en-SG), Aria (pt-NZ), Lennart (de-DE), Vicki (de-DE), Sabrina (de-CH), Hannah (de-AT), Niamh (pt-BR), Camila (pt-BR), Lisa (nl-BE) e Seoyeon (Ko-KR)

> **Note**
>
> Para o caso improvável de alucinação do modelo (e considerando o comportamento do modelo do mecanismo generativo de renderizar a fala token por token), há um mecanismo de parada de emergência imposta em vigor. O mecanismo integrado impede que o modelo continue a renderizar a fala. Esse recurso de segurança é baseado na análise de dados em que o modelo tem o potencial de alucinar, geralmente no final da frase.
> Pode haver casos em que o modelo pensa que vai alucinar e acaba cortando uma palavra durante uma etapa de geração, renderizando assim metade da palavra. Isso pode gerar resultados inadequados.

Long-form vozes

O Amazon Polly tem um Long-form motor que produz vozes humanas, altamente expressivas e emocionalmente adequadas. Long-form as vozes são projetadas para cativar a atenção dos ouvintes para conteúdos mais longos, como artigos de notícias, materiais de treinamento ou vídeos de marketing.

As Long-form vozes do Amazon Polly são desenvolvidas com uma tecnologia TTS de aprendizado profundo de ponta. O modelo aprende a replicar fonemas, prosódia, entonação e outros aspectos fonéticos e acústicos da linguagem humana, resultando em uma saída de fala altamente natural.

O Long-form mecanismo usa incorporações de texto para interpretar o significado de um texto. Usando incorporações de texto, o Long-form mecanismo pode gerar a ênfase, as pausas e o tom corretos de uma voz natural. O resultado é uma voz que combina toda a gama de elementos emocionais presentes na comunicação humana. Isso inclui a imitação da surpresa ou a diferenciação entre diálogo e narração. Juntos, isso cria um produto de fala premium que soa como um ser humano vivo.

> **Note**
>
> A tecnologia de ponta por trás dessas vozes se enquadra no paradigma da IA generativa para modelagem de linguagem e voz. Um efeito colateral da tecnologia é que qualquer atualização nos dados de treinamento e no modelo pode resultar em pequenas variações na forma como as vozes soam, mesmo no caso de sua qualidade geral melhorar com as atualizações do modelo. Isso pode ter um impacto nos casos de uso em que diferentes partes do conteúdo são sintetizadas por um longo período, por exemplo, uma temporada de podcasts.

Vozes de forma longa disponíveis

Atualmente, o Amazon Polly oferece quatro vozes em formato longo en-US e duas es-ES. Ambos os idiomas têm vozes femininas e masculinas disponíveis. As vozes Daniel, Gregory e Ruth, que são em formato longo e em inglês, também estão disponíveis em uma variante NTTS conversacional.

	Idioma	Código do idioma	Name/ID	Gender
1	Inglês (EUA)	en-US	Danielle	Feminino
			Gregory	Masculino
			Ruth	Feminino
			Patrick	Masculino

	Idioma	Código do idioma	Name/ID	Gender
2	Espanhol (Espanha)	es-ES	Alba Raúl	Feminino Masculino

Compatibilidade de recursos e regiões

As vozes de forma longa do Amazon Polly estão disponíveis nas seguintes regiões:

- Leste dos EUA (Norte da Virgínia): us-east-1
- Outras regiões não disponíveis

O Long-form mecanismo Amazon Polly oferece suporte aos seguintes recursos:

- Real-time e operações assíncronas de síntese de fala.
- Todas as marcas de fala.
- Muitas (mas não todas) etiquetas SSML são compatíveis com o Amazon Polly. Para obter mais informações sobre tags NTTS-supported SSML, consulte Tags SSML suportadas
- Assim como ocorre com vozes padrão, é possível escolher entre várias taxas de amostragem para otimizar a largura de banda e a qualidade do áudio para seu aplicativo. As taxas de amostragem válidas para vozes padrão, em formato longo e neurais são 8 kHz, 16 kHz, 22 kHz ou 24 kHz. O padrão para vozes padrão é 22 kHz. O padrão para vozes em formato longo e neurais é 24 kHz. O Amazon Polly é compatível com os formatos de stream de áudio MP3, ogg_vorbis e PCM bruto.

> **Note**
>
> Long-form o custo das vozes é especificado na página de informações de preços do Amazon Polly.

Vozes neurais

O Amazon Polly tem um mecanismo de conversão de texto em fala neural (NTTS) capaz de produzir vozes ainda melhores do que as vozes padrão. As vozes do TTS padrão usam síntese

concatenativa. O mecanismo padrão concatena os fonemas da fala gravada, produzindo uma fala sintetizada muito natural. No entanto, as variações inevitáveis na fala e as técnicas usadas para segmentar as formas de onda limitam a qualidade da fala. O mecanismo de NTTS do Amazon Polly não usa a síntese concatenativa padrão para produzir fala. Ele tem duas partes:

- Uma rede neural, que converte uma sequência de fonemas (as unidades mais básicas da linguagem) em uma sequência de espectrogramas. Os espectogramas são snapshots dos níveis de energia em diferentes bandas de frequência.
- Um vocoder, que converte espectrogramas em um sinal de áudio contínuo.

O primeiro componente do sistema TTS neural é um modelo de sequência a sequência. Esse modelo não cria seus resultados exclusivamente a partir da entrada correspondente, mas também considera como a sequência dos elementos da entrada funcionam em conjunto. O modelo escolhe os espectrogramas que ele gera para que suas faixas de frequência enfatizem os recursos acústicos que o cérebro humano usa ao processar a fala.

A saída desse modelo passa para um vocoder neural. Isso converte os espectrogramas em formas de onda de fala. Quando o treinamento é feito nos grandes conjuntos de dados usados para criar sistemas de síntese concatenativa de uso geral, essa abordagem de sequência a sequência produz vozes de melhor qualidade e mais naturais.

Vozes neurais disponíveis

As vozes neurais estão disponíveis em 36 idiomas e variantes linguísticas. A tabela a seguir lista a vozes.

	Idiomas e variantes linguísticas	Código do idioma	Name/ID	Gender
1	Árabe (Golfo)	ar-AE	Hala	Feminino
			Zayd	Masculino
2	Holandês belga (flamengo)	nl-BE	Lisa	Feminino
3	Catalão	ca-ES	Arlet	Feminino

	Idiomas e variantes linguísticas	Código do idioma	Name/ID	Gender
4	Tcheco	cs-CZ	Jitka	Feminino
5	Chinês (cantonês)	yue-CN	Hiujin	Feminino
6	Chinês (mandarim)	cmn-CN	Zhiyu	Feminino
7	Dinamarquês	da-DK	Sofie	Feminino
8	Holandês	nl-NL	Laura	Feminino
9	Inglês (australiano)	en-AU	Olivia	Feminino
10	Inglês (britânico)	en-GB	Amy* Emma Brian Arthur	Feminino Feminino Masculino Masculino
11	Inglês (indiano)	en-IN	Kajal	Feminino
12	Inglês (irlandês)	en-IE	Niamh	Feminino
13	Inglês (neozelandês)	en-NZ	Aria	Feminino
14	Inglês (singapurense)	en-SG	Jasmine	Feminino
15	Inglês (sul-africano)	en-ZA	Ayanda	Feminino

	Idiomas e variantes linguísticas	Código do idioma	Name/ID	Gender
16	Inglês (EUA)	en-US	Danielle Gregory Ivy Joanna* Kendra Kimberly Salli Joey Justin Kevin Matthew* Ruth Stephen	Feminino Masculino Feminino (infantil) Feminino Feminino Feminino Feminino Masculino Masculino (criança) Masculino (criança) Masculino Feminino Masculino
17	Finlandês	fi-FI	Suvi	Feminino
18	Francês (belga)	fr-BE	Isabelle	Feminino
19	Francês (canadense)	fr-CA	Gabrielle Liam	Feminino Masculino

	Idiomas e variantes linguísticas	Código do idioma	Name/ID	Gender
20	Francês	fr-FR	Léa Rémi	Feminino Masculino
21	Alemão	de-DE	Vicki Daniel	Feminino Masculino
22	Alemão (austríaco)	de-AT	Hannah	Feminino
23	Alemão (suíço)	de-CH	Sabrina	Feminino
24	Hindi	hi-IN	Kajal	Feminino
25	Italiano	it-IT	Bianca Adriano	Feminino Masculino
26	Japonês	ja-JP	Takumi Kazuha Tomoko	Masculino Feminino Feminino
27	Coreano	ko-KR	Seoyeon Jihye	Feminino Feminino
28	Norueguês	nb-NO	Ida	Feminino
29	Polonês	pl-PL	Ola	Feminino
30	Português (brasileiro)	pt-BR	Camila Vitó ria/Vitoria Thiago	Feminino Feminino Masculino

	Idiomas e variantes linguísticas	Código do idioma	Name/ID	Gender
31	Português (europeu)	pt-PT	Inê s/Ines	Feminino
32	Espanhol (Espanha)	es-ES	Lúcia Sergio	Feminino Masculino
33	Espanhol (mexicano)	es-MX	Mia Andrés	Feminino Masculino
34	Espanhol (EUA)	es-US	Lupe* Pedro	Feminino Masculino
35	Sueco	sv-SE	Elin	Feminino
36	Turco	tr-TR	Burcu	Feminino

*As vozes Amy, Joanna, Lupe e Matthew podem ser usadas com o estilo de fala de locutor. Para obter mais informações, consulte Aplicar a voz de âncora.

Compatibilidade de recursos e regiões

As vozes neurais não estão disponíveis em todas as AWS regiões, nem oferecem suporte a todos os recursos do Amazon Polly.

As vozes neurais têm suporte nas seguintes regiões:

- Leste dos EUA (Norte da Virgínia): us-east-1
- Oeste dos EUA (Oregon): us-west-2
- África (Cidade do Cabo): af-south-1
- Ásia-Pacífico (Tóquio): ap-northeast-1
- Ásia-Pacífico (Seul): ap-northeast-2
- Ásia-Pacífico (Osaka): ap-northeast-3

- Ásia-Pacífico (Mumbai): ap-south-1
- Ásia-Pacífico (Singapura): ap-southeast-1
- Ásia-Pacífico (Sydney): ap-southeast-2
- Ásia-Pacífico (Malásia): ap-southeast-5
- Ásia-Pacífico (Tailândia): ap-southeast-7
- Canadá (Central): ca-central-1
- Europa (Frankfurt): eu-central-1
- Europa (Irlanda): eu-west-1
- Europa (Londres): eu-west-2
- Europa (Paris): eu-west-3
- Europa (Espanha) eu-south-2
- Europa (Zurique): eu-central-2
- AWS GovCloud (US-West): us-gov-west-1

Os endpoints e protocolos para essas regiões são idênticos aos usados para vozes padrão. Para obter mais informações, consulte Endpoints e cotas do Amazon Polly.

Os seguintes recursos são compatíveis com vozes neurais:

- Real-time e operações assíncronas de síntese de fala.
- Estilo de locutor. Para mais informações sobre estilos de fala, consulte Aplicar a voz de âncora.
- Todas as marcas de fala.
- Muitas (não todas) tags SSML são compatíveis com o Amazon Polly. Para obter mais informações sobre tags NTTS-supported SSML, consulte Tags suportadas.

Assim como ocorre com vozes padrão, é possível escolher entre várias taxas de amostragem para otimizar a largura de banda e a qualidade do áudio para seu aplicativo. As taxas de amostragem válidas para vozes padrão e neurais são 8 kHz, 16 kHz, 22 kHz ou 24 kHz. O padrão para vozes padrão é 22 kHz. O padrão para vozes neurais é 24 kHz. O Amazon Polly é compatível com os formatos de stream de áudio MP3, ogg_vorbis e PCM bruto.

Vozes padrão

O Amazon Polly tem um mecanismo padrão que usa síntese concatenativa. O mecanismo padrão concatena os fonemas da fala gravada, produzindo uma fala sintetizada muito natural.

Vozes padrão disponíveis

No momento, o Amazon Polly oferece 40 vozes padrão femininas e 20 masculinas em 29 idiomas e variantes linguísticas.

	Idioma	Código do idioma	Name/ID	Gender
1	Árabe	arb	Zeina	Feminino
2	Chinês (mandarim)	cmn-CN	Zhiyu	Feminino
3	Dinamarquês	da-DK	Naja Mads	Feminino Masculino
4	Holandês	nl-NL	Lotte Ruben	Feminino Masculino
5	Inglês (australiano)	en-AU	Nicole Russell	Feminino Masculino
6	Inglês (britânico)	en-GB	Amy Emma Brian	Feminino Feminino Masculino
7	Inglês (indiano)	en-IN	Aditi Raveena	Feminino Feminino
8	Inglês (EUA)	en-US	Ivy	Feminino

	Idioma	Código do idioma	Name/ID	Gender
			Joanna	Feminino
			Kendra	Feminino
			Kimberly	Feminino
			Salli	Feminino
			Joey	Masculino
			Kevin	Masculino
9	Inglês (País de Gales)	en- GB-WLS	Geraint	Masculino
10	Francês	fr-FR	Cé line/Celine	Feminino
			Léa	Feminino
			Mathieu	Masculino
11	Francês (canadense)	fr-CA	Chantal	Feminino
12	Alemão	de-DE	Marlene	Feminino
			Vicki	Feminino
			Hans	Masculino
13	Hindi	hi-IN	Aditi	Feminino
14	Islandês	is-IS	Dó ra/Dora	Feminino
			Karl	Masculino

	Idioma	Código do idioma	Name/ID	Gender
15	Italiano	it-IT	Carla	Feminino
			Bianca	Feminino
			Giorgio	Masculino
16	Japonês	ja-JP	Mizuki	Feminino
			Takumi	Masculino
17	Coreano	ko-KR	Seoyeon	Feminino
18	Norueguês	nb-NO	Liv	Feminino
19	Polonês	pl-PL	Ewa	Feminino
			Maja	Feminino
			Jacek	Masculino
			Jan	Masculino
20	Português (brasileiro)	pt-BR	Camila	Feminino
			Vitó ria/Vitoria	Feminino
			Ricardo	Masculino
21	Português (europeu)	pt-PT	Inê s/Ines	Feminino
			Cristiano	Masculino
22	Romeno	ro-RO	Carmen	Feminino
23	Russo	ru-RU	Tatyana	Feminino
			Maxim	Masculino

	Idioma	Código do idioma	Name/ID	Gender
24	Espanhol (Espanha)	es-ES	Conchita Lúcia Enrique	Feminino Feminino Masculino
25	Espanhol (mexicano)	es-MX	Mia	Feminino
26	Espanhol (EUA)	es-US	Lupe Pené lope/Pene lope Miguel	Feminino Feminino Masculino
27	Sueco	sv-SE	Astrid	Feminino
28	Turco	tr-TR	Filiz	Masculino
29	Galês	cy-GB	Gwyneth	Feminino

Compatibilidade de recursos e regiões

As vozes padrão do Amazon Polly estão disponíveis nas seguintes regiões do Amazon Polly:

- Leste dos EUA (Norte da Virgínia): us-east-1
- Leste dos EUA (Ohio): us-east-2
- Oeste dos EUA (N. da Califórnia): us-west-1
- Oeste dos EUA (Oregon): us-west-2
- África (Cidade do Cabo): af-south-1
- Ásia-Pacífico (Hong Kong): ap-east-1
- Ásia-Pacífico (Tóquio): ap-northeast-1
- Ásia-Pacífico (Seul): ap-northeast-2

- Ásia-Pacífico (Osaka): ap-northeast-3
- Ásia-Pacífico (Mumbai): ap-south-1
- Ásia-Pacífico (Singapura): ap-southeast-1
- Ásia-Pacífico (Sydney): ap-southeast-2
- Ásia-Pacífico (Malásia): ap-southeast-5
- China (Ningxia): cn-northwest-1
- Canadá (Central): ca-central-1
- Europa (Frankfurt): eu-central-1
- Europa (Irlanda): eu-west-1
- Europa (Londres): eu-west-2
- Europa (Paris): eu-west-3
- Europa (Espanha) eu-south-2
- Europa (Estocolmo): eu-north-1
- Oriente Médio (Bahrein): me-south-1
- América do Sul (São Paulo): sa-east-1
- AWS GovCloud (US-West): us-gov-west-1

Os endpoints e protocolos para essas regiões são idênticos aos usados para vozes neurais. Para obter mais informações, consulte Endpoints e cotas do Amazon Polly.

O mecanismo padrão do Amazon Polly oferece suporte aos seguintes recursos (TBD):

- Real-time e operações assíncronas de síntese de fala.
- Todas as marcas de fala.
- Muitas (mas não todas) etiquetas SSML são compatíveis com o Amazon Polly. Para obter mais informações sobre tags NTTS-supported SSML, consulte Tags SSML suportadas.
- É possível escolher entre várias taxas de amostragem para otimizar a largura de banda e a qualidade do áudio para sua aplicação. As taxas de amostragem padrão para vozes padrão são 22 kHz. O Amazon Polly é compatível com os formatos de stream de áudio MP3, ogg_vorbis e PCM bruto.

> **Note**
>
> O custo de vozes padrão é especificado na página de informações sobre preços do Amazon Polly.

Escolher um mecanismo de voz

Você pode acessar as vozes do Amazon Polly por meio do console do Amazon Polly ou. AWS CLI

Como escolher um mecanismo de voz no console

1. Abra o console do Amazon Polly em. https://console.aws.amazon.com/polly/
2. No console do Amazon Polly, escolha o mecanismo de voz desejado.
3. Escolha a voz desejada no menu suspenso de voz.
4. Gere o áudio de TTS com o texto de sua escolha.

Para escolher um mecanismo de voz no AWS CLI, especifique o `Engine` e `VoiceId` nas `SynthesizeSpeech` operações `StartSpeechSynthesisTask` da API. Para ver alguns exemplos, consulte os exemplos de código de início rápido e os exemplos em Python.

Marcas de fala

Marcas de fala são metadados que descrevem a fala que você sintetiza, como onde uma frase ou palavra começa e termina no fluxo de áudio. Ao solicitar marcas de fala para seu texto, o Amazon Polly retorna esses metadados em vez de fala sintetizada. Ao usar marcas de fala em conjunto com um fluxo de áudio de fala sintetizada, você pode fornecer às suas aplicações uma experiência visual melhor.

Por exemplo, combinar os metadados com o fluxo de áudio do seu texto permite sincronizar a fala com uma animação facial (sincronização labial) ou destacar palavras escritas à medida que são faladas.

As marcas de fala estão disponíveis ao usar mecanismos neurais, de formato longo ou padrão. text-to-speech

Tópicos

- Tipos de marca de fala
- Visemas e Amazon Polly
- Saída da marca de fala
- Solicitar marcas de fala
- Exemplo de marcas de fala sem SSML
- Exemplo de marcas de fala com SSML

Tipos de marca de fala

Você solicita marcas de fala usando a SpeechMarkTypesopção para os StartSpeechSynthesisTaskcomandos SynthesizeSpeechou. Você especifica os elementos de metadados que deseja retornar do texto de entrada. Você pode solicitar até quatro tipos de metadados, mas deve especificar pelo menos um por solicitação. Nenhuma saída de áudio é gerada com a solicitação.

No AWS CLI, por exemplo:

```
--speech-mark-types='["sentence", "word", "viseme", "ssml"]'
```

O Amazon Polly gera marcas de fala usando os seguintes elementos.

- sentence – Indica um elemento de frase no texto de entrada.
- word – Indica um elemento de palavra no texto.
- visema: descreve os movimentos faciais e orais correspondentes a cada fonema falado. Para obter mais informações, consulte Visemas e Amazon Polly.
- ssml: descreve um elemento <mark> do texto de entrada SSML. Para obter mais informações, consulte Gerar fala com base em documentos SSML.

Visemas e Amazon Polly

Um visema representa a posição da face e da boca ao dizer uma palavra. Ele é o equivalente visual a um fonema, que é a unidade acústica básica da qual uma palavra é formada. Visemas são os blocos básicos da construção visual da fala.

Cada idioma tem um conjunto de visemas correspondentes a fonemas específicos. Em um idioma, cada fonema tem um visema correspondente que representa a forma que a boca deve tomar para formar o som. No entanto, nem todos os visemas podem ser mapeados a um fonema em particular, pois diversos fonemas parecem ser iguais quando falados, ainda que soem diferentes. Por exemplo, em inglês, as palavras "pet" e "bet" são acusticamente diferentes. No entanto, quando observado visualmente (sem som), eles são exatamente iguais.

A tabela a seguir lista mostra uma lista parcial dos fonemas do Alfabeto Fonético Internacional (AFI) e os símbolos do Speech Assessment Methods Phonetic Alphabet (X-SAMPA, Alfabeto fonético de métodos de avaliação da fala), bem como os visemas correspondentes para vozes do idioma inglês (EUA).

Para obter a tabela completa e tabelas de todos os idiomas disponíveis, consulte Idiomas no Amazon Polly.

IPA	X-SAMPA	Description	Exemplo	Visema
Consoantes				
b	b	Oclusiva bilabial sonora	bed	p
d	d	Oclusiva alveolar sonora	dig	t

IPA	X-SAMPA	Description	Exemplo	Visema
ʤ	dZ	Africada pós-alveolar sonora	jump	S
ð	D	Fricativa dental sonora	then	T
f	f	Fricativa labiodental surda	five	f
g	g	Oclusiva velar sonora	game	k
h	h	Fricativa glotal surda	house	k
...	...	...	...	...

Saída da marca de fala

O Amazon Polly retorna objetos de marca de fala em um fluxo JSON delimitado por linha. Um objeto de marca de fala contém os seguintes campos:

- time – a data e a hora em milissegundos desde o início do fluxo de áudio correspondente
- Tipo: tipo de marca de fala (sentença, palavra, visema ou ssml).
- Início: deslocamento em bytes do início do objeto no texto de entrada (não inclui marcas visema)
- Término: deslocamento em bytes (não caracteres) do fim do objeto no texto de entrada (não inclui marcas viseme)
- value – isso varia de acordo com o tipo de marca de fala
  - SSML: tag <mark> de SSML
  - viseme: o nome do visema
  - word ou sentence: uma substring do texto de entrada, conforme delimitado pelos campos de início e fim

Por exemplo, o Amazon Polly gera o seguinte objeto de marca da fala `word` do texto "Maria tinha um carneirinho":

```
{"time":373,"type":"word","start":5,"end":8,"value":"had"}
```

A palavra descrita ("had") começa 373 milissegundos após o fluxo de áudio começar, e começa no byte 5 e termina no byte 8 do texto de entrada.

> **Note**
>
> Esses metadados são para a voice-id `Joanna`. Se você usar outra voz com o mesmo texto de entrada, os metadados do texto podem ser diferente.

Solicitar marcas de fala

É possível usar o console ou o comando `synthesize-speech` para solicitar marcas de fala do Amazon Polly. Em seguida, visualize os metadados ou salve-os em um arquivo.

Console

Como gerar marcas de fala no console

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Polly em. https://console.aws.amazon.com/polly/
2. Selecione a guia Text-to-Speech.
3. Ative o SSML para usar o SSML.
4. Digite ou cole o texto na caixa de entrada.
5. Para Idioma, escolha o idioma do texto.
6. Para Voz, escolha a voz que você deseja usar.
7. Para alterar a pronúncia do texto, expanda Configurações adicionais, ative Personalizar pronúncia e, em Aplicar léxico, escolha o léxico desejado.
8. Para verificar a fala, escolha Ouvir.
9. Ative as Configurações de formato de arquivo de voz.

> **Note**
>
> Os formatos de download MP3 OGG, PCM, Mu-law ou A-law não gerarão marcas de fala.

10. Para Formato de arquivo, escolha Marcas de fala.
11. Para Tipos de marca de fala, escolha os tipos de marcas de fala a serem gerados. A opção para escolher metadados SSML só está disponível quando o SSML está ativo. Para obter mais informações sobre como usar SSML com o Amazon Polly, consulte Gerar fala com base em documentos SSML.
12. Escolha Baixar.

AWS CLI

Além do texto de entrada, os seguintes elementos são necessários para retornar esses metadados:

- `output-format`

  O Amazon Polly é compatível apenas com o formato JSON ao retornar marcas de fala.

  ```
  --output-format json
  ```

  Se você usar um formato de saída não compatível, o Amazon Polly lançará uma exceção.

- `voice-id`

  Para garantir que os metadados correspondam ao fluxo de áudio associado, especifique a mesma voz usada para gerar o fluxo de áudio de fala sintetizado. As vozes disponíveis não têm velocidades de fala idênticas. Se você usar uma voz diferente daquela usado para gerar a fala, os metadados não corresponderão ao fluxo de áudio.

  ```
  --voice-id Joanna
  ```

- `speech-mark-types`

  Especifique os tipos de marcas de fala desejados. Você pode solicitar qualquer um ou todos os tipos de marca de fala, mas deve especificar pelo menos um tipo.

```
--speech-mark-types='["sentence", "word", "viseme", "ssml"]'
```

- `text-type`

  Texto sem formatação é o padrão para texto de entrada no Amazon Polly. Você deve usar `text-type ssml` se quiser retornar marcas de fala em SSML.

- `outfile`

  Especifique o arquivo de saída ao qual os metadados são gravados.

  ```
  MaryLamb.txt
  ```

O AWS CLI exemplo a seguir está formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um acento circunflexo (^) e use aspas completas (") ao redor do texto de entrada com aspas simples (') para tags internas.

```
aws polly synthesize-speech \
  --output-format json \
  --voice-id Voice ID \
  --text 'Input text' \
  --speech-mark-types='["sentence", "word", "viseme"]' \
  outfile
```

Exemplo de marcas de fala sem SSML

O exemplo a seguir mostra como são os metadados solicitados na sua tela para a frase simples: "Mary had a little lamb". Para simplificar, não incluímos marcas de fala SSML neste exemplo.

O AWS CLI exemplo a seguir está formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um acento circunflexo (^) e use aspas completas (") ao redor do texto de entrada com aspas simples (') para tags internas.

```
aws polly synthesize-speech \
  --output-format json \
  --voice-id Joanna \
  --text 'Mary had a little lamb.' \
```

```
  --speech-mark-types='["viseme", "word", "sentence"]' \
  MaryLamb.txt
```

Ao fazer essa solicitação, o Amazon Polly retorna o seguinte no arquivo .txt:

```
{"time":0,"type":"sentence","start":0,"end":23,"value":"Mary had a little lamb."}
{"time":6,"type":"word","start":0,"end":4,"value":"Mary"}
{"time":6,"type":"viseme","value":"p"}
{"time":73,"type":"viseme","value":"E"}
{"time":180,"type":"viseme","value":"r"}
{"time":292,"type":"viseme","value":"i"}
{"time":373,"type":"word","start":5,"end":8,"value":"had"}
{"time":373,"type":"viseme","value":"k"}
{"time":460,"type":"viseme","value":"a"}
{"time":521,"type":"viseme","value":"t"}
{"time":604,"type":"word","start":9,"end":10,"value":"a"}
{"time":604,"type":"viseme","value":"@"}
{"time":643,"type":"word","start":11,"end":17,"value":"little"}
{"time":643,"type":"viseme","value":"t"}
{"time":739,"type":"viseme","value":"i"}
{"time":769,"type":"viseme","value":"t"}
{"time":799,"type":"viseme","value":"t"}
{"time":882,"type":"word","start":18,"end":22,"value":"lamb"}
{"time":882,"type":"viseme","value":"t"}
{"time":964,"type":"viseme","value":"a"}
{"time":1082,"type":"viseme","value":"p"}
```

Nessa saída, cada parte do texto é dividida em termos de marcas de fala:

- A frase "Mary had a little lamb".
- Cada palavra no texto: "Mary", "had", "a", "little" e "lamb".
- O visema de cada som no fluxo de áudio correspondente: "p", "E", "r", "i", e assim por diante. Para obter mais informações sobre visemas, consulte Visemas e Amazon Polly.

Exemplo de marcas de fala com SSML

O processo de gerar marcas de fala com base em texto aprimorado para SSML é semelhante ao processo quando o SSML não estiver presente. Use o comando `synthesize-speech` e especifique o texto aprimorado por SSML e o tipo de marcas de fala desejado, conforme mostrado

no exemplo a seguir. Para facilitar a leitura do exemplo, nós não incluímos marcas de fala de visemas, mas elas também podem ser incluídas.

O AWS CLI exemplo a seguir está formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um acento circunflexo (^) e use aspas completas (") ao redor do texto de entrada com aspas simples (') para tags internas.

```
aws polly synthesize-speech \
  --output-format json \
  --voice-id Joanna \
  --text-type ssml \
  --text '<speak><prosody volume="+20dB">Mary had <break time="300ms"/>a little <mark
 name="animal"/>lamb</prosody></speak>' \
  --speech-mark-types='["sentence", "word", "ssml"]' \
  output.txt
```

Ao fazer essa solicitação, o Amazon Polly retorna o seguinte no arquivo .txt:

```
{"time":0,"type":"sentence","start":31,"end":95,"value":"Mary had <break time=\"300ms
\"\/>a little <mark name=\"animal\"\/>lamb"}
{"time":6,"type":"word","start":31,"end":35,"value":"Mary"}
{"time":325,"type":"word","start":36,"end":39,"value":"had"}
{"time":897,"type":"word","start":40,"end":61,"value":"<break time=\"300ms\"\/>"}
{"time":1291,"type":"word","start":61,"end":62,"value":"a"}
{"time":1373,"type":"word","start":63,"end":69,"value":"little"}
{"time":1635,"type":"ssml","start":70,"end":91,"value":"animal"}
{"time":1635,"type":"word","start":91,"end":95,"value":"lamb"}
```

Gerar fala com base em documentos SSML

É possível usar o Amazon Polly para gerar fala a partir de texto sem formatação ou de documentos marcados com o SSML. Usar um texto de entrada aprimorado por SSML permite exercer controle adicional sobre como o Amazon Polly gera a fala com base no texto fornecido.

Com as tags do SSML, você pode personalizar e controlar diversos aspectos da fala, como pronúncia, volume, tom e velocidade. No Console de gerenciamento da AWS, o texto aprimorado por SSML que você deseja converter em áudio é inserido na guia SSML da página. Text-to-Speech Embora o texto inserido sem formatação conte com configurações padrão para o idioma e a voz escolhidos, o texto aprimorado com SSML informa ao Amazon Polly não apenas o que você deseja dizer, mas como deseja dizê-lo. Exceto para as tags SSML adicionadas, o Amazon Polly sintetiza textos aprimorados por SSML da mesma maneira que ele sintetiza texto sem formatação. Consulte Exemplo de síntese de fala com o Amazon Polly para obter mais informações.

Ao usar SSML, coloque todo o texto em uma tag `<speak>` para que o Amazon Polly saiba que você está usando SSML. Por exemplo:

```
<speak>Hi! My name is Joanna. I will read any text you type here.</speak>
```

Em seguida, use tags SSML específicas no texto dentro das tags `<speak>` para personalizar a maneira como você deseja que o texto soe. Você pode adicionar uma pausa, alterar o ritmo da fala, diminuir ou aumentar o volume da voz ou adicionar muitas outras personalizações para que o texto soe da forma adequada para você. Para obter uma lista completa das tags SSML que você pode usar, consulte Etiquetas SSML compatíveis.

Por exemplo, você pode incluir uma pausa longa dentro do texto ou alterar a velocidade ou o tom da fala. Outras opções incluem:

- enfatizar palavras ou frases específicas
- usar a pronúncia fonética
- incluir sons de respiração
- sussurrar
- usar o estilo de locutor.

Para obter detalhes completos sobre as tags SSML compatíveis com o Amazon Polly e como usá-las, consulte Etiquetas SSML compatíveis.

Ao usar SSML, há vários caracteres reservados que exigem tratamento especial. Isso ocorre porque o SSML usa esses caracteres como parte de seu código. Para usá-los, você usa uma entidade específica para escapá-los. Para obter mais informações, consulte Caracteres reservados no SSML.

O Amazon Polly fornece esses tipos de controle com um subconjunto de tags SSML definidas pela Recomendação W3C para a Linguagem de marcação de síntese de fala, versão 1.1.

É possível usar SSML no console do Amazon Polly ou usando a AWS CLI. Os tópicos a seguir mostram como você pode usar o SSML para gerar a fala e controlar a saída de modo que ele atenda perfeitamente às suas necessidades.

Tópicos

- Caracteres reservados no SSML
- Usar SSML no console
- Como usar SSML com o comando Synthesize-Speech
- Sintetizar um documento aprimorado por SSML
- Etiquetas SSML compatíveis

Caracteres reservados no SSML

Existem cinco caracteres predefinidos que normalmente não podem ser usados em uma instrução SSML. Essas entidades são reservadas pela especificação da linguagem. Esses caracteres são os seguintes:

Caractere	Código de escape
"	" (aspas duplas)
&	& comercial
'	' apóstrofo ou

Caractere	Código de escape
"	"
'	'
&	&
<	<
>	>

Como o SSML usa esses caracteres como parte de seu código, para usar esses símbolos no SSML, você deve escapar o caractere ao usá-lo. Use o código de escape em vez do caractere real para que ele seja exibido corretamente enquanto cria um documento SSML válido. Por exemplo, a frase a seguir

```
We're using the lawyer at Peabody & Chambers, attorneys-at-law.
```

seria renderizada em SSML como

```
<speak>
We&apos;re using the lawyer at Peabody &amp; Chambers, attorneys-at-law.
</speak>
```

Nesse caso, os caracteres especiais apóstrofo e E comercial precisam de escape para que o documento SSML permaneça válido.

Para os símbolos &, <, e > os códigos de escape são sempre necessários quando você usa SSML. Além disso, ao usar o apóstrofo/aspas simples (') como apóstrofo, você também deve usar o código de escape.

No entanto, quando você usa as aspas duplas (") ou as apostrophe/single aspas (') como aspas, o uso ou não do código de escape depende do contexto.

Aspas duplas

- Deve ser escapado quando em um valor de atributo delimitado por aspas duplas. Por exemplo, no AWS CLI código a seguir

```
--text "Pete &quot;Maverick&quot; Mitchell"
```

- Não precisa de escape quando estiver no contexto textual. Por exemplo, no seguinte

```
He said, "Turn right at the corner."
```

- Não precisa de escape quando em um valor de atributo delimitado por aspas simples. Por exemplo, no seguinte código da AWS CLI .

```
--text 'Pete "Maverick" Mitchell'
```

Aspas simples

- Precisa de escape quando usado como apóstrofo. Por exemplo, no seguinte

```
We&apos;ve got to leave quickly.
```

- Não precisa de escape quando estiver no contexto textual. Por exemplo, no seguinte

```
"And then I said, 'Don't quote me.'"
```

- Não precisa de escape quando em um atributo de código delimitado por aspas duplas. Por exemplo, no seguinte código da AWS CLI .

```
--text "Pete 'Maverick' Mitchell"
```

Usar SSML no console

No exemplo a seguir, você usa uma tag SSML para informar o Amazon Polly sobre a substituição do "World Wide Web Consortium" por "W3C" quando se refere a um parágrafo curto. Você também pode usar tags para inserir uma pausa e sussurrar uma palavra. Compare os resultados deste exercício com os do Aplicar léxicos (sintetização de fala) .

Para obter mais informações sobre o SSML por meio de exemplos, consulte Etiquetas SSML compatíveis.

Como sintetizar a fala com base no texto aprimorado por SSML (console)

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Polly em. https://console.aws.amazon.com/polly/
2. Selecione a guia Text-to-Speech (Texto para fala) se ela ainda não estiver sendo exibida.
3. Ative o SSML.
4. Digite ou cole o texto a seguir na caixa de texto:

```
<speak>
     He was caught up in the game.<break time="1s"/> In the middle of the
     10/3/2014 <sub alias="World Wide Web Consortium">W3C</sub> meeting,
     he shouted, "Nice job!" quite loudly. When his boss stared at him, he
 repeated
     <amazon:effect name="whispered">"Nice job,"</amazon:effect> in a
     whisper.
</speak>
```

   As tags SSML informam o Amazon Polly como renderizar o texto:

   - `<break time="1s"/>` diz ao Amazon Polly para pausar um segundo entre as duas primeiras frases.
   - `<sub alias="World Wide Web Consortium">W3C</sub>` diz ao Amazon Polly para substituir "World Wide Web Consortium" pelo acrônimo "W3C".
   - `<amazon:effect name="whispered">Nice job</amazon:effect>` diz ao Amazon Polly para sussurrar "Bom trabalho" pela segunda vez.

   > **Note**
   >
   > Ao usar o AWS CLI, você coloca o texto de entrada entre aspas para diferenciá-lo do código ao redor. O console do Amazon Polly não mostra o código, para que você não coloque um texto de entrada entre aspas quando usá-lo.

5. Em Idioma, escolha Inglês, EUA e uma voz.
6. Para ouvir a fala, selecione Ouvir.
7. Para salvar o arquivo de fala, selecione Baixar. Se quiser salvá-lo em um formato diferente, expanda Configurações adicionais, ative as Configurações de formato de arquivo de fala, escolha o formato desejado e escolha Baixar.

Como usar SSML com o comando Synthesize-Speech

Este exemplo mostra como usar o comando `synthesize-speech` com uma string SSML. Quando você usa o comando `synthesize-speech`, normalmente fornece estas informações:

- O texto de entrada (obrigatório)
- Tags de abertura e fechamento (obrigatórias)
- O formato da saída
- Uma voz

Neste exemplo, você especifica uma string de texto simples entre aspas juntamente com as tags `<speak></speak>` de abertura e fechamento necessárias.

Important

Embora você não use aspas ao redor do texto de entrada no console do Amazon Polly, é necessário usá-las na AWS CLI . Também é importante que você diferencie as aspas ao redor do texto de entrada e as aspas necessárias para tags individuais.
Por exemplo, você pode usar aspas padrão (") ao redor do texto de entrada e aspas simples (') em tags internas, ou ao contrário. Essas opções funcionam para Unix, Linux e macOS. No entanto, no Windows, é necessário colocar o texto de entrada em aspas padrão e usar aspas simples para as tags.
Em todos os sistemas operacionais, você pode usar aspas padrão (") ao redor do texto de entrada e aspas simples (') em tags internas. Por exemplo:

```
--text "<speak>Hello <break time='300ms'/> World</speak>"
```

Nos sistemas operacionais Unix, Linux e macOS, você também pode usar essas opções ao contrário, ou seja, aspas simples (') ao redor do texto de entrada e aspas padrão (") em tags internas:

```
--text '<speak>Hello <break time="300ms"/> World</speak>'
```

O AWS CLI exemplo a seguir está formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um acento circunflexo (^) e use aspas completas (") ao redor do texto de entrada com aspas simples (') para tags internas.

```
aws polly synthesize-speech \
--text-type ssml \
--text '<speak>Hello world</speak>' \
--output-format mp3 \
--voice-id Joanna \
speech.mp3
```

Para ouvir a fala sintetizada, reproduza o arquivo `speech.mp3` resultante usando qualquer reprodutor de áudio.

Sintetizar um documento aprimorado por SSML

Em textos de entrada mais longos, convém salvar o conteúdo SSML em um arquivo e simplesmente especificar o nome do arquivo no comando `synthesize-speech`. Por exemplo, você pode salvar os comandos a seguir em um arquivo chamado `example.xml`:

```
<?xml version="1.0"?>
<speak version="1.1"
       xmlns="http://www.w3.org/2001/10/synthesis"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.w3.org/2001/10/synthesis http://www.w3.org/TR/
speech-synthesis11/synthesis.xsd"
       xml:lang="en-US">Hello World</speak>
```

O atributo `xml:lang` especifica `en-US` (inglês EUA) como o idioma do texto de entrada. Para obter informações sobre como o idioma do texto de entrada e o idioma da voz selecionada afetam a operação `SynthesizeSpeech`, consulte Especificar outro idioma para palavras específicas.

Como executar um arquivo aprimorado por SSML

1. Salve o SSML em um arquivo (por exemplo, `example.xml`).
2. Execute o seguinte comando `synthesize-speech` no caminho onde o arquivo XML está armazenado e especifique o arquivo SSML como entrada, substituindo `file:\\example.xml` pelo texto de entrada. Como esse comando direciona para um arquivo em vez de conter o texto de entrada em si, não são necessárias aspas.

> **Note**
>
> O AWS CLI exemplo a seguir está formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um circunflexo (^).

```
aws polly synthesize-speech \
--text-type ssml \
--text file://example.xml \
--output-format mp3 \
--voice-id Joanna \
speech.mp3
```

3. Para ouvir a fala sintetizada, reproduza o arquivo `speech.mp3` resultante usando qualquer reprodutor de áudio.

Etiquetas SSML compatíveis

Todas as tags, exceto `<amazon:domain name="news">`, são compatíveis com vozes padrão. A disponibilidade de tags para outras vozes é fornecida na tabela a seguir.

O Amazon Polly é compatível com as seguintes tags SSML:

Ação	Etiqueta SSML	Disponibilidade de voz neural	Disponibilidade de voz em formato longo	Disponibilidade de voz generativa
Adicionar uma pausa	<break>	Disponibilidade total	Disponibilidade total	Disponibilidade total
Enfatizar palavras	<emphasis>	Não disponível	Não disponível	Não disponível

Ação	Etiqueta SSML	Disponibilidade de voz neural	Disponibilidade de voz em formato longo	Disponibilidade de voz generativa
Especificar outro idioma para palavras específicas	<lang>	Disponibilidade total	Disponibilidade total	Disponibilidade total
Colocar uma tag personalizada no texto	<mark>	Disponibilidade total	Disponibilidade total	Disponibilidade parcial
Adicionar uma pausa entre parágrafos	<p>	Disponibilidade total	Disponibilidade total	Disponibilidade total
Usar a pronúncia fonética	<phoneme>	Disponibilidade total	Disponibilidade total	Disponibilidade parcial
Controlar o volume, a velocidade e o tom da fala	<prosody>	Disponibilidade parcial	Disponibilidade parcial	Disponibilidade parcial
Configurar uma duração máxima para fala sintetizada	<prosody amazon:max-duration>	Não disponível	Não disponível	Não disponível
Adicionar uma pausa entre as frases	<s>	Disponibilidade total	Disponibilidade total	Disponibilidade total
Controlar como tipos especiais de palavras são ditas	<say-as>	Disponibilidade parcial	Disponibilidade total	Disponibilidade total
Identificar um texto aprimorado por SSML	<speak>	Disponibilidade total	Disponibilidade total	Disponibilidade total

Ação	Etiqueta SSML	Disponibilidade de voz neural	Disponibilidade de voz em formato longo	Disponibilidade de voz generativa
Pronunciar acrônimos e abreviaturas	<sub>	Disponibilidade total	Disponibilidade total	Disponibilidade total
Melhorar a pronúncia especificando partes da fala	<w>	Disponibilidade total	Disponibilidade total	Disponibilidade total
Adicionar o som da respiração	<amazon:auto-breaths>	Não disponível	Não disponível	Não disponível
Estilo de locutor	<amazon:domain name="news">	Selecionar somente vozes neurais	Não disponível	Não disponível
Adicionar compactação de intervalo dinâmico	<amazon:effect name="drc">	Disponibilidade total	Disponibilidade total	Indisponível
Falar suavemente	<amazon:effect phonation="soft">	Não disponível	Não disponível	Não disponível
Controlar o timbre	<amazon : efeito > vocal-tract-length	Não disponível	Não disponível	Não disponível
Sussurrar	<amazon:effect name="whispered">	Não disponível	Não disponível	Não disponível

> **Note**
>
> Se você usar tags SSML incompatíveis no formato padrão, neural ou de forma longa, receberá um erro.

Identificar um texto aprimorado por SSML

<speak>

Essa etiqueta é compatível com TTS generativa, de forma longa, neural e padrão.

A tag `<speak>` é o elemento raiz de todos os textos SSML do Amazon Polly. Todos os textos aprimorados por SSML devem estar entre tags **<speak>**.

```
<speak>Mary had a little lamb.</speak>
```

Adicionar uma pausa

<break>

Essa etiqueta é compatível com TTS generativa, de forma longa, neural e padrão.

Para adicionar uma pausa ao texto, use a tag **<break>**. Você pode definir uma pausa com base na intensidade (equivalente à pausa após uma vírgula, uma frase ou um parágrafo), ou configurá-la para um determinado período em segundos ou milissegundos. Se você não especificar um atributo para determinar o tamanho da pausa, o Amazon Polly usará o padrão, que é `<break strength="medium"/>`, que adiciona uma pausa do tamanho de uma pausa após uma vírgula.

Valores de atributo `strength`:

- `none`: nenhuma pausa. Use `none` para remover uma pausa comum, como a que ocorre depois de um ponto final.
- `x-weak`: tem a mesma função de `none`; nenhuma pausa.
- `weak`: define uma pausa com a mesma duração da pausa após uma vírgula.
- `medium`: tem a mesma função de `weak`.
- `strong`: define uma pausa com a mesma duração da pausa após uma frase.
- `x-strong`: define uma pausa com a mesma duração da pausa após um parágrafo.

Valores de atributo `time`:

- *`[number]`*`s`: a duração da pausa, em segundos. A duração máxima é de `10s`.

- *[number]*ms: a duração da pausa em milissegundos. A duração máxima é de 10000ms.

Por exemplo:

```
<speak>
     Mary had a little lamb <break time="3s"/>Whose fleece was white as snow.
</speak>
```

Se você não usar um atributo com a tag break, o resultado variará dependendo do texto:

- Se não houver mais nenhuma pontuação ao lado da tag break, ela criará uma <break strength="medium"/> (pausa de duração de vírgula).
- Se a tag estiver ao lado de uma vírgula, ela atualizará a tag para uma <break strength="strong"/> (pausa de duração de frase).
- Se a tag estiver ao lado de um ponto final, ela atualizará a tag para uma <break strength="x-strong"/> (pausa de duração de parágrafo).

Enfatizar palavras

<emphasis>

Essa tag é compatível somente com o formato TTS padrão.

Para enfatizar palavras, use a tag <emphasis>. Quando há ênfase de palavras, a taxa e o volume da fala mudam. Uma ênfase maior significa que o texto será falado pelo Amazon Polly em um volume mais alto e velocidade mais lenta. Uma ênfase menor significa que a fala será mais tranquila e mais lenta. Para especificar o grau de ênfase, use o atributo level.

Valores de atributo level:

- Strong: aumenta o volume e desacelera a taxa da fala para que ela soe mais alta e mais lentamente.
- Moderate: aumenta o volume e diminui a taxa da fala, mas menos do que strong. Moderate é o padrão.
- Reduced: diminui o volume e aumenta a taxa da fala. A fala é mais suave e mais rápida.

> **Note**
>
> A taxa e o volume normais de uma voz ficam entre os níveis `moderate` e `reduced`.

Por exemplo:

```
<speak>I already told you I <emphasis level="strong">really like</emphasis> that
 person.</speak>
```

Especificar outro idioma para palavras específicas

<lang>

Essa etiqueta é compatível com TTS generativa, de forma longa, neural e padrão. Para vozes generativas, a `<lang>` tag só pode ser usada em torno de frases completas.

Especifique outro idioma para uma determinada palavra, frase ou frase com a tag **<lang>**. Frases e palavras em idioma estrangeiro geralmente são ditas com melhor sonoridade quando são inseridas dentro de um par de tags `<lang>`. Para especificar o idioma, use o atributo `xml:lang`. Para obter uma lista completa dos idiomas disponíveis, consulte Idiomas no Amazon Polly.

A menos que você aplique a tag `<lang>`, todas as palavras no texto de entrada serão faladas no idioma da voz especificada no `voice-id`. Se você aplicar a tag `<lang>`, as palavras serão faladas no idioma selecionado.

Por exemplo, se `voice-id` for Joanna (que fala inglês dos EUA), o Amazon Polly falará a seguinte frase na voz dela sem sotaque francês:

```
<speak>
     Je ne parle pas français.
</speak>
```

Se você usar a voz da Joanna com a tag `<lang>`, o Amazon Polly falará a frase com a voz dela em francês com sotaque americano:

```
<speak>
```

```
     <lang xml:lang="fr-FR">Je ne parle pas français.</lang>.
</speak>
```

Como Joanna não é uma voz nativa do Francês, a pronúncia se baseia no idioma nativo dela, que é o Inglês dos EUA. Por exemplo, apesar da pronúncia perfeita em Francês ter uma vibrante uvular /R/ na palavra français, a voz em Inglês dos EUA da Joanna pronuncia esse fonema como o som /r/ correspondente.

Se você usar o `voice-id` de Giorgio, que fala italiano, com o seguinte texto, o Amazon Polly falará a frase na voz dele com pronúncia italiana:

```
<speak>
     Mi piace Bruce Springsteen.
</speak>
```

Se você usar a mesma voz com a tag `<lang>`, o Amazon Polly pronunciará Bruce Springsteen em Inglês com sotaque italiano:

```
<speak>
     Mi piace <lang xml:lang="en-US">Bruce Springsteen.</lang>
</speak>
```

Essa tag também pode ser usada como substituto da DefaultLangCodeopção opcional ao sintetizar a fala. No entanto, isso requer que você formate o texto usando SSML.

Colocar uma tag personalizada no texto

<mark>

Essa tag é compatível com os formatos TTS de forma longa, padrão e neural. Essa tag não faz nada com relação a vozes generativas porque as marcas de fala não estão disponíveis para esse tipo de voz.

Para colocar uma tag personalizada no texto, use a tag **<mark>**. O Amazon Polly não realiza nenhuma ação na tag, mas retorna a localização dela nos metadados do SSML. Essa tag pode ter qualquer nome que você desejar, desde que ela mantenha o seguinte formato:

```
<mark name="tag_name"/>
```

Por exemplo, se o nome da tag for "animal" e o texto de entrada for:

```
<speak>
     Mary had a little <mark name="animal"/>lamb.
</speak>
```

O Amazon Polly poderá retornar os seguintes metadados de SSML:

```
{"time":767,"type":"ssml","start":25,"end":46,"value":"animal"}
```

Adicionar uma pausa entre parágrafos

<p>

Essa etiqueta é compatível com TTS generativa, de forma longa, neural e padrão.

Para adicionar uma pausa entre parágrafos no texto, use a tag <p>. O uso dessa tag fornece uma pausa maior do que aquelas geralmente colocadas por falantes nativos em vírgulas ou final de frases. Use a tag <p> para incluir o parágrafo:

```
<speak>
     <p>This is the first paragraph. There should be a pause after this text is
 spoken.</p>
     <p>This is the second paragraph.</p>
</speak>
```

Isso é equivalente a especificar uma pausa usando <break strength="x-strong"/>.

Usar a pronúncia fonética

<phoneme>

Essa tag é compatível com os formatos TTS de forma longa, padrão e neural.

Para que o Amazon Polly use a pronúncia fonética em um texto específico, use a tag <phoneme>.

Dois atributos são necessários com a tag <phoneme>. Eles indicam o alfabeto fonético que o Amazon Polly usa e os símbolos fonéticos da pronúncia corrigida:

- `alphabet`
  - `ipa` – indica que o sistema do Alfabeto Fonético Internacional (AFI) será usado.
  - `x-sampa` – indica que o sistema Extended Speech Assessment Methods Phonetic Alphabet (X-SAMPA, Alfabeto fonético estendido de métodos de avaliação da fala) será usado.
- ph
  - Especifica os símbolos fonéticos da pronúncia. Para obter mais informações, consulte Idiomas no Amazon Polly.

Com a tag <phoneme>, o Amazon Polly usa a pronúncia especificada pelo atributo ph em vez da pronúncia padrão associada ao idioma da voz selecionada.

Por exemplo, a palavra "pecan" pode ser pronunciada de duas formas. No exemplo a seguir, uma pronúncia personalizada diferente em cada linha é atribuída à palavra "pecan". O Amazon Polly pronuncia "pecan" conforme especificado nos atributos ph, em vez de usar a pronúncia padrão.

Alfabeto Fonético Internacional (AFI)

```
<speak>
     You say, <phoneme alphabet="ipa" ph="p##k##n">pecan</phoneme>.
     I say, <phoneme alphabet="ipa" ph="#pi.kæn">pecan</phoneme>.
</speak>
```

Extented Speech Assessment Methods Phonetic Alphabet (X-SAMPA, Alfabeto fonético de métodos de avaliação da fala).

```
<speak>
     You say, <phoneme alphabet='x-sampa' ph='pI"kA:n'>pecan</phoneme>.
     I say, <phoneme alphabet='x-sampa' ph='"pi.k{n'>pecan</phoneme>.
</speak>
```

O chinês mandarim usa Pinyin para pronúncia fonética.

Pinyin

```
<speak>
```

```
     ## <phoneme alphabet="x-amazon-pinyin" ph="bo2">#</phoneme>#
     ## <phoneme alphabet="x-amazon-pinyin" ph="bao2">#</phoneme>#
</speak>
```

O japonês usa Yomigana e a pronúncia Kana.

Yomigana

```
<speak>
     ###<phoneme alphabet="x-amazon-yomigana" ph="####">##</phoneme>###
     ###<phoneme alphabet="x-amazon-yomigana" ph="####">##</phoneme>###
     ###<phoneme alphabet="x-amazon-yomigana" ph="Hirokazu">##</phoneme>###
</speak>
```

Pronúncia Kana

```
<speak>
     ###<phoneme alphabet="x-amazon-pron-kana" ph="##'##">##</phoneme>###
</speak>
```

Controlar o volume, a velocidade e o tom da fala

<prosody>

Os atributos das tags de prosódia são totalmente compatíveis com as vozes TTS padrão. As vozes generativas, neurais e em formato longo são compatíveis com os atributos `volume` e `rate`, mas não com o atributo `pitch`. Para vozes generativas, a tag de prosódia só pode ser usada em frases completas.

Para controlar o volume, a velocidade ou o tom da voz selecionada, use a tag `prosody`.

O volume, a taxa e o tom da fala dependem da voz selecionada. Além das diferenças entre vozes em idiomas diferentes, há diferenças entre vozes individuais falando o mesmo idioma. Por isso, embora os atributos sejam semelhantes em todos os idiomas, há variações nítidas entre idiomas e nenhum valor absoluto estará disponível.

A tag `prosody` tem três atributos, e cada um deles têm vários valores disponíveis para definir o atributo. Todos os atributos usam a mesma sintaxe:

```
<prosody attribute="value"></prosody>
```

- `volume`
  - `default`: redefine o volume como o nível padrão da voz atual.
  - `silent`, `x-soft`, `soft`, `medium`, `loud`, `x-loud`: define o volume como um valor predefinido para a voz atual.
  - `+ndB`, `-ndB`: altera o volume com relação ao nível atual. Um valor de `+0dB` significa nenhuma alteração, `+6dB` significa aproximadamente o dobro do volume atual e `-6dB` significa aproximadamente metade do volume atual.

  Por exemplo, você pode definir o volume para um trecho das seguintes maneiras:

```
<speak>
     Sometimes it can be useful to <prosody volume="loud">increase the volume
     for a specific speech.</prosody>
</speak>
```

  Ou você pode definir assim:

```
<speak>
     And sometimes a lower volume <prosody volume="-6dB">is a more effective way of
     interacting with your audience.</prosody>
</speak>
```

- `rate`
  - `x-slow`, `slow`, `medium`, `fast` e `x-fast` especificam o tom para um valor predefinido da voz selecionada.
  - n%: uma alteração de porcentagem não negativa na taxa da fala. Por exemplo, um valor de 100% significa nenhuma alteração na taxa da fala, 200% significa que a taxa é o dobro da taxa padrão e 50% que taxa é metade da taxa padrão. Esse valor varia de 20 a 200%.

  Por exemplo, você pode definir a taxa da fala para um trecho das seguintes maneiras:

```
<speak>
     For dramatic purposes, you might wish to <prosody rate="slow">slow up the
 speaking
     rate of your text.</prosody>
</speak>
```

  Ou você pode definir assim:

```
<speak>
     Although in some cases, it might help your audience to <prosody rate="85%">slow
     the speaking rate slightly to aid in comprehension.</prosody>
</speak>
```

- `pitch`
  - `default`: redefine o tom como o nível padrão da voz atual.
  - `x-low`, `low`, `medium`, `high`, `x-high`: Define o tom do valor predefinido para a voz atual.
  - `+n%` ou `-n%`: Ajusta o tom por uma porcentagem relativa. Por exemplo, um valor de `+0%` significa nenhuma alteração no tom da linha de base, `+5%` fornece um tom de linha de base um pouco maior e `-5%` resulta em um tom de linha de base um pouco menor.

  Por exemplo, você pode definir o tom para um trecho das seguintes maneiras:

```
<speak>
     Do you like sythesized speech <prosody pitch="high">with a pitch that is higher
     than normal?</prosody>
</speak>
```

  Ou você pode definir assim:

```
<speak>
     Or do you prefer your speech <prosody pitch="-10%">with a somewhat lower pitch?
</prosody>
</speak>
```

A tag **<prosody>** deve conter pelo menos um atributo, mas pode incluir mais atributos na mesma tag.

```
<speak>
     Each morning when I wake up, <prosody volume="loud" rate="x-slow">I speak
     quite slowly and deliberately until I have my coffee.</prosody>
</speak>
```

Ela também pode ser combinada com tags aninhadas da seguinte forma:

```
<speak>
     <prosody rate="85%">Sometimes combining attributes <prosody pitch="-10%">can
```

```
    change the impression your audience has of a voice</prosody> as well.</prosody>

</speak>
```

> **Note**
>
> Atualmente <prosody> está parcialmente disponível para as vozes generativas.

Configurar uma duração máxima para fala sintetizada

<prosody amazon:max-duration>

No momento, essa tag é compatível somente com o formato TTS padrão.

Para controlar quanto tempo você deseja que dure uma fala quando ela for sintetizada, use a tag <prosody> com o atributo amazon:max-duration.

A duração da fala sintetizada varia levemente, dependendo da voz selecionada. Isso pode dificultar a correspondência da fala sintetizada com recursos visuais ou outras atividades que exigem uma sincronização precisa. Esse problema é ampliado em aplicativos de tradução, pois o tempo usado para dizer certas frases pode variar amplamente entre diferentes idiomas.

A tag <prosody amazon:max-duration> corresponde a fala sintetizada à quantidade de tempo que você deseja que ela leve (a duração).

Essa tag usa a seguinte sintaxe:

```
<prosody amazon:max-duration="time duration">
```

Com a tag <prosody amazon:max-duration>, é possível especificar a duração em segundos ou milissegundos:

- *n*s: a duração máxima em segundos
- *n*ms: a duração máxima em milissegundos

Por exemplo, o seguinte texto falado tem uma duração máxima de 2 segundos:

```
<speak>
```

```
     <prosody amazon:max-duration="2s">
          Human speech is a powerful way to communicate.
     </prosody>
</speak>
```

Texto colocado dentro da tag não excede a duração especificada. Se a voz ou o idioma escolhido normalmente levam mais tempo do que aquela duração, o Amazon Polly acelera a fala para que ela se encaixe na duração especificada.

Se a duração especificada leva mais tempo do que o necessário para ler o texto em uma velocidade normal, o Amazon Polly lê a fala normalmente. Ele não diminui a velocidade da fala ou adiciona silêncio, portanto o áudio resultante fica mais curto do que o solicitado.

> **Note**
>
> O Amazon Polly aumenta a velocidade não mais do que cinco vezes a velocidade normal. Se o texto for falado mais rápido do que isso, normalmente não fará sentido. Se uma fala não couber na duração especificada, mesmo quando acelerada ao máximo, o áudio será acelerado mas durará mais do que a duração especificada.

É possível incluir uma única frase ou várias frases em uma tag `<prosody amazon:max-duration>`, e também é possível usar várias tags `<prosody amazon:max-duration>` em seu texto.

Por exemplo:

```
<speak>
     <prosody amazon:max-duration="2400ms">
        Human speech is a powerful way to communicate.
     </prosody>
     <break strength="strong"/>
     <prosody amazon:max-duration="5100ms">
        Even a simple ‘Hello’ can convey a lot of information depending on the pitch,
 intonation, and tempo.
     </prosody>
     <break strength="strong"/>
     <prosody amazon:max-duration="8900ms">
        We naturally understand this information, which is why speech is ideal for
 creating applications where
```

```
        a screen isn’t practical or possible, or simply isn’t convenient.
    </prosody>
</speak>
```

Usar a tag <prosody amazon:max-duration> pode aumentar a latência quando o Amazon Polly retornar a fala sintetizada. O grau de latência depende da passagem e da duração. Recomendamos usar um texto composto de passagens relativamente curtas.

Limitações

Há limitações em como você usa a tag <prosody amazon:max-duration> e em como ela funciona com outras tags SSML:

- O texto dentro de uma tag <prosody amazon:max-duration> não pode ter mais de 1500 caracteres.
- Não é possível aninhar tags <prosody amazon:max-duration>. Se você colocar uma tag <prosody amazon:max-duration> dentro de outra, o Amazon Polly ignorará a tag interna.

  No seguinte exemplo, a tag <prosody amazon:max-duration="5s"> é ignorada:

  ```
  <speak>
      <prosody amazon:max-duration="16s">
          Human speech is a powerful way to communicate.

          <prosody amazon:max-duration="5s">
              Even a simple ‘Hello’ can convey a lot of information depending on the
   pitch, intonation, and tempo.
          </prosody>

          We naturally understand this information, which is why speech is ideal for
   creating applications where a screen isn’t practical or possible, or simply isn’t
   convenient.
      </prosody>
  </speak>
  ```

- Não é possível usar as tags <prosody> com o atributo rate dentro de uma tag <prosody amazon:max-duration>. Isso ocorre porque as duas afetam a velocidade de fala do texto.

  No seguinte exemplo, o Amazon Polly ignora a tag <prosody rate="2">:

```
<speak>
     <prosody amazon:max-duration="7500ms">
          Human speech is a powerful way to communicate.

          <prosody rate="2">
               Even a simple ‘Hello’ can convey a lot of information depending on the
 pitch, intonation, and tempo.
          </prosody>
     </prosody>
</speak>
```

Pausas e **`max-duration`**

Ao usar a tag `max-duration`, você ainda poderá inserir pausas dentro do texto. No entanto, o Amazon Polly inclui a duração da pausa ao calcular a duração máxima da fala. Além disso, o Amazon Polly preserva as pausas breves que ocorrem onde vírgulas e pontos são colocados em uma passagem e inclui na duração máxima.

Por exemplo, no seguinte bloco, a pausa de 600 milissegundos e as pausas causadas pelas vírgulas e pontos ocorrem dentro da fala de 8 segundos:

```
<speak>
     <prosody amazon:max-duration="8s">
          Human speech is a powerful way to communicate.
          <break time="600ms"/>
          Even a simple ‘Hello’ can convey a lot of information depending on the pitch,
 intonation, and tempo.
     </prosody>
</speak>
```

Adicionar uma pausa entre as frases

<s>

Essa etiqueta é compatível com TTS generativa, de forma longa, neural e padrão.

Para adicionar uma pausa entre linhas ou frases no texto, use a tag <s>. Usar essa tag gera o mesmo resultado destas ações:

- Finalizar uma frase com um ponto (.)
- Especificar uma pausa com `<break strength="strong"/>`

Ao contrário da tag `<break>`, a tag **<s>** inclui a frase. Isso é útil ao sintetizar falas organizadas em linhas, em vez de frases, como poesias.

No exemplo a seguir, a tag `<s>` cria uma curta pausa após a primeira e a segunda frase. A frase final não tem a tag `<s>`, mas também tem uma pausa curta depois, pois contém um ponto final.

```
<speak>
     <s>Mary had a little lamb</s>
     <s>Whose fleece was white as snow</s>
     And everywhere that Mary went, the lamb was sure to go.
</speak>
```

Controlar como tipos especiais de palavras são ditas

<say-as>

A tag `<say-as>` é compatível com mecanismos de TTS generativos, de forma longa, neural e padrão. Entretanto, se o Amazon Polly estiver usando uma voz neural e encontrar a tag `<say-as>` com a opção `characters` no runtime, a frase afetada será sintetizada usando a voz padrão relacionada. No entanto, a frase afetada ainda será cobrada como se usasse uma voz neural.

Use a tag `<say-as>` com o atributo `interpret-as` para informar ao Amazon Polly como dizer determinados caracteres, palavras e números. Isso permite que você forneça contexto adicional para eliminar qualquer ambiguidade sobre como o Amazon Polly deve renderizar o texto.

A tag `<say-as>` usa um atributo, `interpret-as`, que usa um número de possíveis valores disponíveis. Todos usam a mesma sintaxe:

```
<say-as interpret-as="value">[text to be interpreted]</say-as>
```

Os valores a seguir estão disponíveis com o `interpret-as`:

- `characters` `spell-out`: Soletra cada letra do texto, como em. a-b-c

> **Note**
>
> No momento, essa opção não é compatível com vozes neurais. Se você estiver usando uma voz neural e esse código SSML for encontrado pelo Amazon Polly no runtime, a frase afetada será sintetizada usando a voz padrão relacionada. No entanto, essa frase ainda será cobrada como se usasse uma voz neural.

- `cardinal` ou `number`: interpreta o texto numérico como um número cardinal, por exemplo, 1.234.
- `ordinal`: interpreta o texto numérico como um número ordinal, por exemplo, 1.234º.
- `digits`: soletra cada dígito individualmente, como em 1-2-3-4.
- `fraction`: interpreta o texto numérico como uma fração. Isso funciona tanto em frações comuns, como 3/20, quanto em mistas, como 2 ½. Veja a seguir mais informações.
- `unit`: interpreta um texto numérico como medida. O valor deve ser um número ou uma fração que, por sua vez, é seguida por uma unidade sem espaço no meio, como em `1/2inch`, ou por apenas uma unidade, como em `1meter`.
- `date`: interpreta o texto como uma data. O formato de data deve ser especificado com o atributo do formato. Veja a seguir mais informações.
- `time`: interpreta o texto numérico como uma duração, em minutos e segundos, como em `1'21"`.
- `address`: interpreta o texto como parte de um endereço.
- `expletive`: aumenta o som do conteúdo incluído na tag.
- `telephone`: interpreta o texto numérico como um número de telefone de 7 ou 10 dígitos, como em `2025551212`. Você também pode usar esse valor para lidar com ramais telefônicos, como em `2025551212x345`. Veja a seguir mais informações.

> **Note**
>
> Atualmente, a opção `telephone` não está disponível para todos os idiomas. No entanto, está disponível para vozes que falam variantes do idioma inglês (en-AU, en-GB, en-IN, en-US e en-GB-WLS), variantes do idioma espanhol (es-ES, es-MX e es-US), variantes do idioma francês (fr-FR e fr-CA) e variantes do português (pt-BR e pt-PT), bem como alemão (de-DE), italiano (it-IT), japonês (ja-JP) e russo (ru-ru). Também deve ser observado que, em alguns casos, idiomas como o árabe (arb) manipulam automaticamente o número definido como um número de telefone e, portanto, não implementam a etiqueta SSML `telephone` de fato.

Frações

O Amazon Polly interpretará os valores da tag `say-as` com o atributo `interpret-as="fraction"` como frações comuns. Veja a seguir a sintaxe de frações.

- Fração

  Sintaxe:*cardinal number*/*cardinal number*, como 2/9.

  Por exemplo: `<say-as interpret-as="fraction">2/9</say-as>` é pronunciado "dois nonos."
- Número misto não negativo

  Sintaxe: *cardinal number* +*cardinal number*/*cardinal number*, como 3+1/2.

  Por exemplo: `<say-as interpret-as="fraction">3+1/2</say-as>` é pronunciado "três e meio".

  > **Note**
  >
  > Deve haver um + entre "3" e "1/2". O Amazon Polly não é compatível com um número misto sem o +, como "3 1/2".

Datas

Quando `interpret-as` for definido como `date`, você também precisará indicar o formato da data.

Neste caso, usa-se a seguinte sintaxe:

```
<say-as interpret-as="date" format="format">[date]</say-as>
```

Por exemplo:

```
<speak>
     I was born on <say-as interpret-as="date" format="mdy">12-31-1900</say-as>.
</speak>
```

Os formatos a seguir podem ser usados com o atributo `date`.

- `mdy`: Month-day-year.

- dmy: Day-month-year.
- ymd: Year-month-day.
- md: mês-dia.
- dm: dia-mês.
- ym: ano-mês.
- my: mês-ano.
- d: dia.
- m: Mês.
- y: Ano.
- yyyymmdd: Year-month-day. Se você usar esse formato, poderá fazer com que o Amazon Polly pule partes da data usando pontos de interrogação.

  Por exemplo, o Amazon Polly renderiza o seguinte exemplo como "22 de setembro":

  ```
  <say-as interpret-as="date">????0922</say-as>
  ```

  Format não é necessários.

Telefone

O Amazon Polly tenta interpretar o texto fornecido corretamente com base na formatação do texto, mesmo sem a tag <say-as>. Por exemplo, se o texto incluir 202-555-1212, o Amazon Polly o interpretará como um número de telefone de dez dígitos e dirá cada dígito individualmente, com uma breve pausa para cada traço. Nesse caso, você não precisa usar o <say-as interpret-as="telephone">. No entanto, se você fornecer o texto 2025551212 e quiser que o Amazon Polly o diga como número de telefone, será necessário especificar <say-as interpret-as="telephone">.

A lógica para a interpretação de cada elemento é específica de cada idioma. Por exemplo, o Inglês dos EUA e o Inglês do Reino Unido se diferenciam na pronúncia dos números de telefone (no Inglês do Reino Unido, as sequências do mesmo dígito são agrupadas, como "double five" [dois números cinco] ou "triple four" [três números quatro]). Para ver a diferença, teste o exemplo a seguir com uma voz dos EUA e com uma voz do Reino Unido:

```
<speak>
     Richard's number is <say-as interpret-as="telephone">2122241555</say-as>
```

```
</speak>
```

Pronunciar acrônimos e abreviaturas

<sub>

Essa etiqueta é compatível com TTS generativa, de forma longa, neural e padrão.

Use a tag `<sub>` com o atributo `alias` para substituir uma palavra (ou pronúncia) diferente do texto selecionado, como um acrônimo ou abreviação.

Neste caso, usa-se sintaxe:

```
<sub alias="new word">abbreviation</sub>
```

No exemplo a seguir, o nome "Mercúrio" é substituído pelo símbolo químico do elemento para tornar o conteúdo do áudio mais claro.

```
<speak>
     My favorite chemical element is <sub alias="Mercury">Hg</sub>, because it looks so
 shiny.
</speak>
```

Melhorar a pronúncia especificando partes da fala

<w>

Essa etiqueta é compatível com TTS generativa, de forma longa, neural e padrão.

Você pode usar a tag **<w>** para personalizar a pronúncia de palavras especificando a parte da fala da palavra ou significado alternativo. Isso é feito usando o atributo `role`.

Essa tag usa a seguinte sintaxe:

```
<w role="attribute">text</w>
```

Os valores a seguir podem ser usados para o atributo `role`:

Para especificar a parte da fala:

- `amazon:VB`: interpreta a palavra como verbo (presente simples).
- `amazon:VBD`: interpreta a palavra como verbo no pretérito.
- `amazon:DT`: interpreta a palavra como um determinante.
- `amazon:IN`: interpreta a palavra como uma preposição.
- `amazon:JJ`: interpreta a palavra como um adjetivo.
- `amazon:NN`: interpreta a palavra como um substantivo.

Por exemplo, dependendo da parte da fala, a pronúncia do inglês dos EUA da palavra "read" varia de acordo com a tag:

```
<speak>
     The word <say-as interpret-as="characters">read</say-as> may be interpreted
     as either the present simple form <w role="amazon:VB">read</w>, or the past
     participle form <w role="amazon:VBD">read</w>.
</speak>
```

Como definir um significado específico:

- `amazon:DEFAULT`: usa o sentido padrão da palavra.
- `amazon:SENSE_1`: usa o sentido não padrão da palavra, quando presente. Por exemplo, o termo "bass" é pronunciado de forma diferente dependendo de seu significado. O significado padrão é a parte mais baixa da escala musical. O significado alternativo é uma espécie de peixe de água doce também chamado de "bass", mas com pronúncia diferente. O uso de `<w role="amazon:SENSE_1">bass</w>` renderiza a pronúncia não padrão (peixe de água fresca) para o texto de áudio.

Essa diferença poderá ser ouvida se você sintetizar o seguinte:

```
<speak>
    Depending on your meaning, the word <say-as interpret-as="characters">bass</say-
as>
    may be interpreted as either a musical element: bass, or as its alternative
 meaning,
    a freshwater fish <w role="amazon:SENSE_1">bass</w>.
</speak>
```

> **Note**
>
> Alguns idiomas podem ter uma seleção diferente de partes da fala com suporte.

Adicionar o som da respiração

<amazon:breath> e <amazon:auto-breaths>

Essa tag é compatível somente com o formato TTS padrão.

Um fala natural inclui corretamente palavras faladas e sons de respiração. Ao adicionar sons de respiração à fala sintetizada, você pode torná-la mais natural. As tags `<amazon:breath>` e `<amazon:auto-breaths>` fornecem respirações. Você tem as seguintes opções:

- Modo manual: defina o local, o comprimento e o volume de um som de respiração no texto
- Modo automatizado: o Amazon Polly insere automaticamente sons de respiração na saída de fala
- Modo misto: você e o Amazon Polly adicionam sons de respiração

Modo manual

No modo manual, coloque a tag `<amazon:breath/>` no texto de entrada em que você deseja colocar uma respiração. Você pode personalizar o comprimento e volume de respirações com os atributos `duration` e `volume` respectivamente:

- `duration`: controla o comprimento da respiração. Os valores válidos são: `default`, `x-short`, `short`, `medium`, `long`, `x-long`. O valor padrão é `medium`.
- `volume`: controla o volume de sons de respiração. Os valores válidos são: `default`, `x-soft`, `soft`, `medium`, `loud`, `x-loud`. O valor padrão é `medium`.

> **Note**
>
> O comprimento e o volume exatos de cada valor de atributo dependem da voz do Amazon Polly usada.

Para definir um som de respiração usando os valores padrão, use <amazon:breath/> sem atributos.

Por exemplo, para usar atributos para definir a duração e o volume de uma respiração como média, você deve definir os atributos da seguinte maneira:

```
<speak>
     Sometimes you want to insert only <amazon:breath duration="medium" volume="x-
loud"/>a single breath.
</speak>
```

Para usar os valores padrão, você pode simplesmente usar a tag:

```
<speak>
     Sometimes you need <amazon:breath/>to insert one or more average breaths
 <amazon:breath/> so that the
     text sounds correct.
</speak>
```

Você pode adicionar sons de respiração individuais em um trecho da seguinte forma:

```
<speak>
     <amazon:breath duration="long" volume="x-loud"/> <prosody rate="120%"> <prosody
 volume="loud">
     Wow! <amazon:breath duration="long" volume="loud"/> </prosody> That was quite
 fast. <amazon:breath
     duration="medium" volume="x-loud"/> I almost beat my personal best time on this
 track. </prosody>
</speak>
```

Modo automatizado

No modo automatizado, use a tag <amazon:auto-breaths> para instruir o Amazon Polly a criar automaticamente ruídos de respiração em intervalos apropriados. Você pode definir a frequência dos intervalos, o volume e a duração. Coloque a tag </amazon:auto-breaths> no início do texto em que você deseja aplicar a respiração automatizada e feche a tag no final.

> **Note**
>
> Ao contrário da tag do modo manual, `<amazon:breath/>`, a tag `<amazon:auto-breaths>` requer uma tag de fechamento (`</amazon:auto-breaths>`).

Você pode usar os seguintes atributos opcionais com a tag `<amazon:auto-breaths>`:

- `volume`: controla o volume de sons de respiração. Os valores válidos são: `default`, `x-soft`, `soft`, `medium`, `loud`, `x-loud`. O valor padrão é `medium`.
- `frequency`: controla a frequência com que sons respiratórios ocorrem no texto. Os valores válidos são: `default`, `x-low`, `low`, `medium`, `high`, `x-high`. O valor padrão é `medium`.
- `duration`: controla o comprimento da respiração. Os valores válidos são: `default`, `x-short`, `short`, `medium`, `long`, `x-long`. O valor padrão é `medium`.

Por padrão, a frequência de sons de respiração depende do texto de entrada. No entanto, muitas vezes, sons de respiração ocorrem após vírgulas e pontos.

Os exemplos a seguir mostram como usar a tag `<amazon:auto-breaths>`. Para decidir quais opções usar para o seu conteúdo, copie os exemplos aplicáveis para o console do Amazon Polly e escute as diferenças.

- Usando o modo automatizado sem parâmetros opcionais.

```
<speak>
     <amazon:auto-breaths>Amazon Polly is a service that turns text into lifelike
 speech,
     allowing you to create applications that talk and build entirely new categories
 of speech-
     enabled products. Amazon Polly is a text-to-speech service that uses advanced
 deep learning
     technologies to synthesize speech that sounds like a human voice. With dozens of
 lifelike
     voices across a variety of languages, you can select the ideal voice and build
 speech-
     enabled applications that work in many different countries.</amazon:auto-
breaths>
</speak>
```

- Usando o modo automatizado com controle de volume. O parâmetros não especificados (`duration` e `frequency`) são definidos como os valores padrão (`medium`).

```
<speak>
     <amazon:auto-breaths volume="x-soft">Amazon Polly is a service that turns text
 into lifelike
     speech, allowing you to create applications that talk and build entirely new
 categories of
     speech-enabled products. Amazon Polly is a text-to-speech service, that uses
 advanced deep
     learning technologies to synthesize speech that sounds like a human voice. With
 dozens of
     lifelike voices across a variety of languages, you can select the ideal voice
 and build speech-
     enabled applications that work in many different countries.</amazon:auto-
breaths>
</speak>
```

- Usando o modo automatizado com controle de frequência. O parâmetros não especificados (`duration` e `volume`) são definidos como os valores padrão (`medium`).

```
<speak>
     <amazon:auto-breaths frequency="x-low">Amazon Polly is a service that turns text
 into lifelike
     speech, allowing you to create applications that talk and build entirely new
 categories of
     speech-enabled products. Amazon Polly is a text-to-speech service, that uses
 advanced deep
     learning technologies to synthesize speech that sounds like a human voice. With
 dozens of
     lifelike voices across a variety of languages, you can select the ideal voice
 and build speech-
     enabled applications that work in many different countries.</amazon:auto-
breaths>
</speak>
```

- Usando o modo automatizado com parâmetros opcionais. Para o parâmetro `Duration` não especificado, o Amazon Polly usa o valor padrão (`medium`).

```
<speak>
     <amazon:auto-breaths volume="x-loud" frequency="x-low">Amazon Polly is a service
 that turns
```

```
    text into lifelike speech, allowing you to create applications that talk and
 build entirely new
    categories of speech-enabled products. Amazon Polly is a text-to-speech service,
 that uses
    advanced deep learning technologies to synthesize speech that sounds like a
 human voice. With
    dozens of lifelike voices across a variety of languages, you can select the
 ideal voice and build
    speech-enabled applications that work in many different countries.</amazon:auto-
breaths>
</speak>
```

Estilo de locutor

<amazon:domain name="news">

O estilo de locutor pode ser usado somente com as vozes de Matthew e Joanna, disponíveis apenas em inglês dos EUA (en-US), e Lupe em espanhol dos EUA (es-US). Ele só é compatível ao usar o formato `Neural`.

Para usar o estilo de locutor, use tags SSML e a seguinte sintaxe:

```
<amazon:domain name="news">text</amazon:domain>
```

Por exemplo, você poderá usar o estilo de locutor com a voz de Matthew da seguinte forma:

```
<speak>
<amazon:domain name="news">
From the Tuesday, April 16th, 1912 edition of The Guardian newspaper:

The maiden voyage of the White Star liner Titanic, the largest ship ever launched, has
 ended in disaster.

The Titanic started her trip from Southampton for New York on Wednesday. Late on Sunday
 night she struck
an iceberg off the Grand Banks of Newfoundland. By wireless telegraphy she sent out
 signals of distress,
and several liners were near enough to catch and respond to the call.
</amazon:domain>
</speak>
```

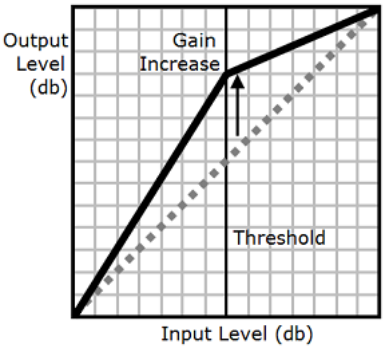
Adicionar compactação de intervalo dinâmico

<amazon:effect name="drc">

Essa tag é compatível com os formatos TTS de forma longa, padrão e neural.

Dependendo do texto, do idioma e da voz usados em um arquivo de áudio, os sons variam de baixo a alto. Os sons ambientais, como o som de um veículo em movimento, muitas vezes podem ocultar os sons mais baixos, o que dificulta ouvir a faixa de áudio com clareza. Para aumentar o volume de determinados sons no seu arquivo de áudio, use a tag de compactação de alcance dinâmico (`drc`).

A tag `drc` define um limite de "intensidade" de médio alcance para seu áudio e aumenta o volume (o ganho) dos sons em torno desse limite. Ele aplica o maior aumento de ganho mais próximo do limiar, e o aumento de ganho é diminuído muito além do limiar.



Isso facilita ouvir os sons de meio alcance em um ambiente com barulho, o que aumenta a clareza do arquivo de áudio todo.

A tag `drc` é um parâmetro booleano (presente ou ausente). Ela usa a sintaxe: `<amazon:effect name="drc">` e é fechada com `</amazon:effect>`.

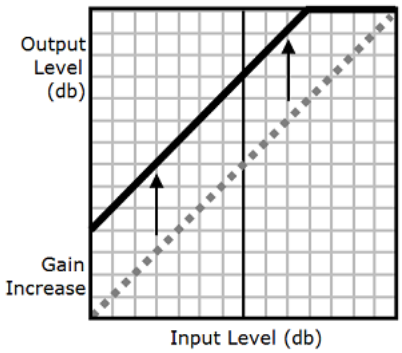
Você pode usar a tag `drc` com qualquer voz ou idioma compatível com o Amazon Polly. Você pode aplicá-la a uma seção inteira da gravação ou em apenas algumas palavras. Por exemplo:

```
<speak>
     Some audio is difficult to hear in a moving vehicle, but <amazon:effect
 name="drc"> this audio
     is less difficult to hear in a moving vehicle.</amazon:effect>
</speak>
```

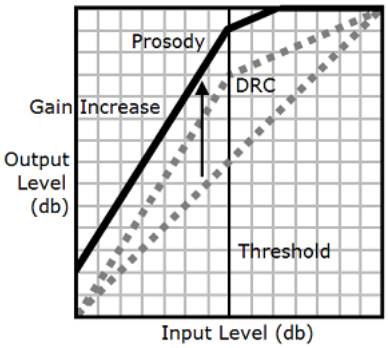
> **Note**
>
> Quando você usa "`drc`" na sintaxe , há distinção entre letras maiúsculas e minúsculas.`amazon:effect`

Usar **`drc`** com a tag **`prosody volume`**

Assim como mostra o gráfico a seguir, a tag `prosody volume` aumenta uniformemente o volume de um arquivo de áudio inteiro a partir do nível original (linha pontilhada) até um nível ajustado (linha contínua). Para aumentar ainda mais o volume de certas partes do arquivo, use a tag `drc` com a tag `prosody volume`. Combinar tags não afeta as configurações da tag `prosody volume`.



Ao usar as tags `drc` e `prosody volume` em conjunto, o Amazon Polly aplica a tag `drc` primeiro, aumentando os sons de alcance médio (aqueles próximos ao limite). Em seguida, ele aplica a tag `prosody volume` e aumenta ainda mais o volume da faixa de áudio toda de forma uniforme.



Para usar as tags em conjunto, aninhe uma dentro da outra. Por exemplo:

```
<speak>
     <prosody volume="loud">This text needs to be understandable and loud.
 <amazon:effect name="drc">
```

```
    This text also needs to be more understandable in a moving car.</amazon:effect></
prosody>
</speak>
```

Neste texto, a tag `prosody volume` aumenta o volume do trecho inteiro para "alto". A tag `drc` aprimora o volume de valores de alcance médio na segunda frase.

> **Note**
>
> Ao usar as tags `drc` e `prosody volume` em conjunto, use as práticas de XML padrão para agrupar tags.

Falar suavemente

<amazon:effect phonation="soft">

No momento, essa tag é compatível somente com o formato TTS padrão.

Para especificar que o texto de entrada deve ser falado em softer-than-normal voz, use a <amazon:effect phonation="soft">tag.

Neste caso, usa-se sintaxe:

```
<amazon:effect phonation="soft">text</amazon:effect>
```

Por exemplo, você pode usar essa tag com a voz de Matthew da seguinte forma:

```
<speak>
    This is Matthew speaking in my normal voice. <amazon:effect phonation="soft">This
    is Matthew speaking in my softer voice.</amazon:effect>
</speak>
```

Controlar o timbre

<amazon : efeito > vocal-tract-length

No momento, essa tag é compatível somente com o formato TTS padrão.

Timbre é a qualidade tonal de uma voz que ajuda a distinguir a diferença entre as vozes, mesmo quando elas têm o mesmo tom e intensidade. Uma das características fisiológicas mais importantes que contribui com o timbre da fala timbre é o comprimento do trato vocal. O trato vocal é uma cavidade de ar que se estende desde o topo das pregas vocais até a borda dos lábios.

Para controlar o timbre da saída de fala no Amazon Polly, use a tag `vocal-tract-length`. Esta tag tem o efeito de alterar o comprimento do trato vocal do orador, que soa como uma alteração no tamanho do orador. Ao aumentar o `vocal-tract-length`, o orador soa fisicamente maior. Ao diminuí-lo, o orador soa menor. Você pode usar essa tag com qualquer uma das vozes no portfólio do Amazon Polly. Text-to-Speech

Para alterar o timbre, use os seguintes valores:

- `+n%` ou `-n%`: ajusta comprimento do trato vocal por uma alteração percentual na voz atual. Por exemplo, +4% ou -2%. Os valores válidos variam de +100% a -50%. Valores fora deste intervalo são cortados. Por exemplo, sons +111%, como+100% e -60% soam como -50%.
- `n%`: altera o comprimento do trato vocal a uma porcentagem absoluta do comprimento do trato da voz atual. Por exemplo, 110% ou 75%. Um valor absoluto de 110% é equivalente a um valor relativo de +10%. Um valor absoluto de 100% é o mesmo que o valor padrão da voz atual.

O exemplo a seguir mostra como alterar o comprimento do trato vocal para alterar o timbre:

```
<speak>
     This is my original voice, without any modifications. <amazon:effect vocal-tract-
length="+15%">
     Now, imagine that I am much bigger. </amazon:effect> <amazon:effect vocal-tract-
length="-15%">
     Or, perhaps you prefer my voice when I'm very small. </amazon:effect> You can also
 control the
     timbre of my voice by making minor adjustments. <amazon:effect vocal-tract-
length="+10%">
     For example, by making me sound just a little bigger. </
amazon:effect><amazon:effect
     vocal-tract-length="-10%"> Or, making me sound only somewhat smaller. </
amazon:effect>
</speak>
```

Combinar várias tags

Você pode combinar a tag `vocal-tract-length` com qualquer outra tag de SSML compatível com o Amazon Polly. Visto que o timbre (comprimento do trato vocal) e a altura estão intimamente conectados, você pode ter os melhores resultados usando ambas as tags `vocal-tract-lengthe <prosody pitch>`. Para produzir a voz mais realista, recomendamos que você use diferentes porcentagens de alterações para as duas tags. Experimente com várias combinações para obter os resultados desejados.

O exemplo a seguir mostra como combinar tags.

```
<speak>
     The pitch and timbre of a person's voice are connected in human speech.
     <amazon:effect vocal-tract-length="-15%"> If you are going to reduce the vocal
 tract length,
     </amazon:effect><amazon:effect vocal-tract-length="-15%"> <prosody pitch="+20%">
 you
     might consider increasing the pitch, too. </prosody></amazon:effect>
     <amazon:effect vocal-tract-length="+15%"> If you choose to lengthen the vocal
 tract,
     </amazon:effect> <amazon:effect vocal-tract-length="+15%"> <prosody pitch="-10%">
     you might also want to lower the pitch. </prosody></amazon:effect>
</speak>
```

Sussurrar

<amazon:effect name="whispered">

No momento, essa tag é compatível somente com o formato TTS padrão.

Essa tag indica que o texto de entrada deve ser falado em uma voz sussurrada, em vez da fala normal. Isso pode ser usado com qualquer uma das vozes no portfólio do Amazon Polly. Text-to-Speech

Neste caso, usa-se a seguinte sintaxe:

```
<amazon:effect name="whispered">text</amazon:effect>
```

Por exemplo:

```
<speak>
     <amazon:effect name="whispered">If you make any noise, </amazon:effect>
```

```
     she said, <amazon:effect name="whispered">they will hear us.</amazon:effect>
</speak>
```

Neste caso, a fala sintetizada dita pelo personagem será sussurrada, mas a frase "Ela disse" será dita na fala normal sintetizada da voz do Amazon Polly selecionada.

É possível aprimorar o efeito "sussurro" diminuindo a velocidade da prosódia em até 10%, dependendo do efeito desejado.

Por exemplo:

```
<speak>
     When any voice is made to whisper, <amazon:effect name="whispered">
     <prosody rate="-10%">the sound is slower and quieter than normal speech
     </prosody></amazon:effect>
</speak>
```

Ao gerar marcas de fala para uma voz sussurrada, o fluxo de áudio também deverá incluir a voz sussurrada para garantir que as marcas de fala correspondam ao fluxo de áudio.

Gerenciar léxicos

Léxicos de pronúncia permitem que você personalize a pronúncia de palavras. O Amazon Polly fornece operações de API que você pode usar para armazenar léxicos em uma região da AWS. Esses léxicos são específicos para essa região em particular. Você pode usar um ou mais dos léxicos dessa região quando sintetizar o texto usando a operação `SynthesizeSpeech`. Isso aplica o léxico específico ao texto de entrada antes de a síntese começar. Para obter mais informações, consulte SynthesizeSpeech.

> **Note**
>
> Esses léxicos devem estar em conformidade com a recomendação PLS (Pronunciation Lexicon Specification, Especificação do léxico de pronúncia), do W3C. Para obter mais informações, consulte Pronunciation Lexicon Specification (PLS) Version 1.0 no site do W3C.

Veja a seguir exemplos de maneiras de usar léxicos com mecanismos de síntese de fala:

- As palavras comuns são, às vezes, estilizadas com números tomando o lugar de letras, como em "g3t sm4rt" ("get smart", ou "fique esperto"). Seres humanos conseguem ler essas palavras corretamente. No entanto, um mecanismo de texto para fala (TTS) lê o texto literalmente, pronunciando a palavra exatamente como está escrita. É nesse ponto que você pode utilizar léxicos para personalizar a fala sintetizada usando o Amazon Polly. Neste exemplo, você pode especificar um alias (get smart) para a palavra "g3t sm4rt" no léxico.
- Seu texto pode incluir um acrônimo, como W3C. Você pode usar um léxico para definir um alias para a palavra W3C, de forma que ela seja lida no formato completo e expandido (World Wide Web Consortium).

Os léxicos oferecem a você mais controle sobre como o Amazon Polly pronuncia palavras incomuns no idioma selecionado. Por exemplo, você pode especificar a pronúncia usando um alfabeto fonético. Para obter mais informações, consulte Pronunciation Lexicon Specification (PLS) Version 1.0 no site do W3C.

Tópicos

- Usar vários léxicos
- Fazer upload de um léxico

- Aplicar léxicos (sintetização de fala)
- Filtrar a lista de léxicos no console
- Fazer download dos léxicos no console
- Excluir um léxico

Usar vários léxicos

Você pode aplicar até cinco léxicos ao texto. Se o mesmo grafema aparecer em mais de um léxico aplicável ao seu texto, a ordem na qual eles são aplicados pode fazer a diferença na fala resultante. Por exemplo, no texto: "Olá, meu nome é Bob". e dois lexemas em diferentes léxicos nos quais ambos usem o grafema Bob.

LexA

```
<lexeme>
   <grapheme>Bob</grapheme>
   <alias>Robert</alias>
</lexeme>
```

LexB

```
<lexeme>
   <grapheme>Bob</grapheme>
   <alias>Bobby</alias>
</lexeme>
```

Se os léxicos estiverem listados na ordem LexA e depois LexB, a fala sintetizada será "Hello, my name is Robert". Se estiverem listados na ordem LexB e depois LexA, a fala sintetizada será "Hello, my name is Bobby".

Example– Aplicação do LexA antes do LexB

```
aws polly synthesize-speech \
--lexicon-names LexA LexB \
--output-format mp3 \
--text 'Hello, my name is Bob' \
--voice-id Justin \
bobAB.mp3
```

Resultado da fala: "Hello, my name is Robert."

Example– Aplicação do LexB antes do LexA

```
aws polly synthesize-speech \
--lexicon-names LexB LexA \
--output-format mp3 \
--text 'Hello, my name is Bob' \
--voice-id Justin \
bobBA.mp3
```

Resultado da fala: "Hello, my name is Bobby."

Para obter informações sobre a aplicação de léxicos usando o console do Amazon Polly, consulte Aplicar léxicos (sintetização de fala).

Fazer upload de um léxico

Os léxicos que você usa devem estar em conformidade com a recomendação PLS (Pronunciation Lexicon Specification) do W3C. Para obter mais informações, consulte Pronunciation Lexicon Specification (PLS) Version 1.0 no site do W3C.

Console - Lexicons tab

Para usar um léxico de pronúncia, é preciso primeiro fazer o upload. Há dois locais no console do qual você pode fazer upload de um léxico: a guia Text-to-Speech e a guia Lexicons.

Os processos a seguir descrevem como adicionar léxicos que você pode usar para personalizar como palavras e frases incomuns ao idioma escolhido são pronunciadas.

Como adicionar um léxico com a guia Léxicos

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Polly em https://console.aws.amazon.com/polly/.
2. Escolha a guia Lexicons.
3. Escolha Carregar léxico.
4. Forneça um nome para o léxico e, em seguida, use Escolher um arquivo de léxico para encontrar o léxico a ser carregado. Você só pode fazer upload de arquivos PLS com extensões .pls ou .xml.

5. Escolha Carregar léxico. Se um léxico com o mesmo nome (.pls ou .xml) já existir, o carregamento do léxico substituirá o existente.

Console - TTS tab

Como adicionar um léxico por meio da guia Texto para fala

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Polly em https://console.aws.amazon.com/polly/.
2. Selecione a guia Text-to-Speech.
3. Expanda Configurações adicionais, ative Personalizar pronúncia e escolha Carregar léxico.
4. Forneça um nome para o léxico e, em seguida, use Escolher um arquivo de léxico para encontrar o léxico a ser carregado. Você só pode usar arquivos PLS que com as extensões .pls e .xml.
5. Escolha Carregar léxico. Se um léxico com o mesmo nome (seja arquivo .pls ou .xml) já existir, carregar o léxico substituirá o existente.

AWS CLI - one lexeme

Com o Amazon Polly, você pode usar PutLexicon para armazenar léxicos de pronúncia em uma região da AWS específica para sua conta. Em seguida, você pode especificar um ou mais desses léxico armazenados na sua solicitação SynthesizeSpeech que você deseja aplicar antes de o serviço começar a sintetizar o texto. Para obter mais informações, consulte Gerenciar léxicos.

Considere o léxico compatível com a PLS do W3C a seguir.

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa"
      xml:lang="en-US">
  <lexeme>
    <grapheme>W3C</grapheme>
    <alias>World Wide Web Consortium</alias>
  </lexeme>
```

```
</lexicon>
```

Observe o seguinte:

- Os dois atributos especificados no elemento `<lexicon>`:
  - O atributo `xml:lang` especifica o código do idioma, `en-US`, ao qual o léxico se aplica. O Amazon Polly pode usar esse léxico de exemplo se a voz que você especificar na chamada `SynthesizeSpeech` tiver o mesmo código de idioma (en-US).

    > **Note**
    >
    > Você pode usar a operação `DescribeVoices` para localizar o código de idioma associado a uma voz.

  - O atributo `alphabet` especifica `IPA`, o que significa que o Alfabeto Fonético Internacional (AFI; IPA na sigla em inglês, que significa "International Phonetic Alphabet") é usado para pronúncias. O AFI é um dos alfabetos para escrever pronúncias. O Amazon Polly também é compatível com Alfabeto fonético de métodos de avaliação da fala (X-SAMPA).

- O elemento `<lexeme>` descreve o mapeamento entre `<grapheme>` (ou seja, uma representação textual da palavra) e `<alias>`.

Para testar esse léxico, faça o seguinte:

1. Salve o léxico como `example.pls`.
2. Execute o comando da `put-lexicon` AWS CLI para armazenar o léxico (com o nome `w3c`), na região us-east-2.

   ```
   aws polly put-lexicon \
   --name w3c \
   --content file://example.pls
   ```

3. Execute o comando `synthesize-speech` para sintetizar o texto de amostra em um fluxo de áudio (`speech.mp3`) e especifique parâmetro `lexicon-name` opcional.

```
aws polly synthesize-speech \
--text 'W3C is a Consortium' \
--voice-id Joanna \
--output-format mp3 \
--lexicon-names="w3c" \
speech.mp3
```

4. Reproduza o `speech.mp3` resultante e observe que a palavra W3C no texto é substituída por World Wide Web Consortium.

O léxico do exemplo anterior usa um alias. O AFI mencionado no léxico não é usado. O léxico a seguir especifica uma pronúncia fonética usando o elemento <phoneme> com o AFI.

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa"
      xml:lang="en-US">
  <lexeme>
    <grapheme>pecan</grapheme>
    <phoneme>p##k##n</phoneme>
  </lexeme>
</lexicon>
```

Siga as mesmas etapas para testar esse léxico. Especifique um texto de entrada que tenha a palavra “pecan” (por exemplo, “Pecan pie is delicious”).

Para conferir mais exemplos de código da operação de API PutLexicon, consulte os seguintes recursos adicionais:

- Amostra de Java: PutLexicon
- Amostra de Python (Boto3): PutLexicon

AWS CLI - multiple lexemes

Com o Amazon Polly, você pode usar PutLexicon para armazenar léxicos de pronúncia em uma região da AWS específica para sua conta. Em seguida, você pode especificar um ou mais desses léxico armazenados na sua solicitação SynthesizeSpeech que você deseja aplicar antes de o serviço começar a sintetizar o texto. Para obter mais informações, consulte Gerenciar léxicos.

Neste exemplo, o lexema que você especificar no léxico se aplica exclusivamente ao texto de entrada para a síntese. Considere o seguinte léxico:

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa" xml:lang="en-US">

  <lexeme>
    <grapheme>W3C</grapheme>
    <alias>World Wide Web Consortium</alias>
  </lexeme>
  <lexeme>
    <grapheme>W3C</grapheme>
    <alias>WWW Consortium</alias>
  </lexeme>
  <lexeme>
    <grapheme>Consortium</grapheme>
    <alias>Community</alias>
  </lexeme>
</lexicon>
```

O léxico especifica três lexemas, dois dos quais definem um alias para o grafema W3C, da seguinte forma:

- O primeiro elemento <lexeme> define um alias (World Wide Web Consortium).
- O segundo <lexeme> define um alias alternativo (WWW Consortium).

O Amazon Polly usa a primeira substituição para determinado grafema em um léxico.

O terceiro <lexeme> define uma substituição (Community) para a palavra Consortium.

Primeiro, vamos testar esse léxico. Suponha que você deseja sintetizar o texto de exemplo a seguir para um arquivo de áudio (`speech.mp3`) e especifica o léxico em uma chamada para `SynthesizeSpeech`.

```
The W3C is a Consortium
```

`SynthesizeSpeech`O primeiro aplica a léxico da seguinte forma:

- De acordo com o primeiro lexema, a palavra W3C é revisada como World Wide Web Consortium. O texto revisado aparece da seguinte forma:

  ```
  The World Wide Web Consortium is a Consortium
  ```

- O alias definido no terceiro lexema se aplica somente à palavra Consortium, que foi parte do texto original, resultando no texto a seguir:

  ```
  The World Wide Web Consortium is a Community.
  ```

Você pode testar isso usando a AWS CLI, da seguinte forma:

1. Salve o léxico como `example.pls`.
2. Execute o comando `put-lexicon` para armazenar o léxico com o nome w3c na região us-east-2.

   ```
   aws polly put-lexicon \
   --name w3c \
   --content file://example.pls
   ```

3. Execute o comando `list-lexicons` para verificar se o léxico w3c está na lista de léxicos retornados.

   ```
   aws polly list-lexicons
   ```

4. Execute o comando `synthesize-speech` para sintetizar o texto de amostra para um arquivo de áudio (`speech.mp3`) e especifique parâmetro `lexicon-name` opcional.

   ```
   aws polly synthesize-speech \
   --text 'W3C is a Consortium' \
   --voice-id Joanna \
   ```

```
--output-format mp3 \
--lexicon-names="w3c" \
speech.mp3
```

5. Reproduza o arquivo `speech.mp3` resultante para verificar se a fala sintetizada reflete as alterações no texto.

Para conferir mais exemplos de código da operação de API PutLexicon, consulte os seguintes recursos adicionais:

- Amostra de Java: PutLexicon
- Amostra de Python (Boto3): PutLexicon

Aplicar léxicos (sintetização de fala)

Os léxicos que você usa devem estar em conformidade com a recomendação PLS (Pronunciation Lexicon Specification) do W3C. Para obter mais informações, consulte Pronunciation Lexicon Specification (PLS) Version 1.0 no site do W3C.

Console

O procedimento a seguir demonstra como aplicar um léxico ao seu texto inserido aplicando o léxico `W3c.pls` para substituir "W3C" por "World Wide Web Consortium". Se você aplicar vários léxicos ao texto, eles serão aplicados de cima para baixo, com a primeira correspondência levando precedência sobre as correspondências posteriores. O léxico só será aplicado ao texto se o idioma especificado nele for o mesmo que o idioma escolhido.

Você pode aplicar um léxico a uma entrada em texto sem formatação ou SSML.

Example– Aplicação do léxico W3C.pls

Para criar o léxico do qual você precisará para este exercício, consulte Fazer upload de um léxico. Use um editor de texto sem formatação para criar o léxico W3C.pls exibido na parte superior do tópico. Lembre-se do local onde salvou o arquivo.

Para aplicar o léxico W3C.pls à sua entrada

Neste exemplo, apresentamos um léxico para substituir "World Wide Web Consortium" por "W3C". Compare os resultados deste exercício com os de Usar SSML no console para inglês norte-americano e outro idioma.

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Polly em https://console.aws.amazon.com/polly/.
2. Execute um destes procedimentos:

   - Selecione a guia SSML e digite ou cole este texto na caixa de entrada de texto.

     ```
     He was caught up in the game.
     In the middle of the 10/3/2014 W3C meeting
     he shouted, "Score!" quite loudly.
     ```

   - Selecione a guia SSML e digite ou cole este texto na caixa de entrada de texto.

     ```
     <speak>He wasn't paying attention.<break time="1s"/>
     In the middle of the 10/3/2014 W3C meeting
     he shouted, "Score!" quite loudly.</speak>
     ```

3. Na lista Idiomas, escolha Inglês, EUA e, em seguida, escolha a voz que deseja usar para esse texto.
4. Expanda Configurações adicionais e ative Personalizar pronúncia.
5. Na lista de léxicos, selecione `W3C (English, US)`.

   Se o léxico `W3C (English, US)` não estiver listado, escolha Upload lexicon e carregue-o; em seguida, escolha-o na lista. Para criar esse léxico, consulte Fazer upload de um léxico.
6. Para ouvir a fala imediatamente, selecione Ouvir.
7. Para salvar a fala em um arquivo,

   a. Escolha Baixar.

   b. Para alterar para um formato de arquivo diferente, selecione Alterar formato do arquivo, selecione o formato de arquivo desejado e, em seguida, selecione Alterar.

Repita as etapas anteriores, mas escolha um idioma diferente e observe a diferença na saída.

AWS CLI

Em uma chamada para `SynthesizeSpeech`, você pode especificar vários léxicos. Neste caso, o primeiro léxico especificado (da esquerda para a direita) sobrepõe quaisquer léxicos anteriores.

Considere os dois léxicos a seguir. Observe que cada léxico descreve aliases diferentes para o mesmo grafema de W3C.

- Léxico 1: `w3c.pls`

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa" xml:lang="en-US">
  <lexeme>
    <grapheme>W3C</grapheme>
    <alias>World Wide Web Consortium</alias>
  </lexeme>
</lexicon>
```

- Léxico 2: `w3cAlternate.pls`

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa" xml:lang="en-US">

  <lexeme>
    <grapheme>W3C</grapheme>
    <alias>WWW Consortium</alias>
  </lexeme>
</lexicon>
```

Vamos supor que você armazene esses léxicos como `w3c` e `w3cAlternate`, respectivamente. Se você especificar léxicos na ordem (`w3c` seguido por `w3cAlternate`) em uma chamada `SynthesizeSpeech`, o alias de W3C definido no primeiro léxico terá precedência sobre o segundo. Para testar os léxicos, faça o seguinte:

1. Salve os léxicos localmente em arquivos chamados `w3c.pls` e `w3cAlternate.pls`.
2. Faça upload desses léxicos usando o comando da `put-lexicon` AWS CLI.

   - Faça upload do léxico `w3c.pls` e armazene-o como `w3c`.

```
aws polly put-lexicon \
--name w3c \
--content file://w3c.pls
```

- Faça upload do léxico `w3cAlternate.pls` no serviço como `w3cAlternate`.

```
aws polly put-lexicon \
--name w3cAlternate \
--content file://w3cAlternate.pls
```

3. Execute o comando `synthesize-speech` para sintetizar texto de exemplo em um fluxo de áudio (`speech.mp3`) e especifique os dois léxicos usando o parâmetro `lexicon-name`.

```
aws polly synthesize-speech \
--text 'PLS is a W3C recommendation' \
--voice-id Joanna \
--output-format mp3 \
--lexicon-names '["w3c","w3cAlternative"]' \
speech.mp3
```

4. Teste o resultante `speech.mp3`. Ele deve ler da seguinte forma:

```
PLS is a World Wide Web Consortium recommendation
```

Filtrar a lista de léxicos no console

O procedimento a seguir descreve como filtrar a lista de léxicos de forma que somente os léxicos de determinado idioma sejam exibidos.

Console

Para filtrar os léxicos listados por idioma

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Polly em https://console.aws.amazon.com/polly/.
2. Escolha a guia Lexicons.
3. Escolha Qualquer idioma.
4. Na lista de idiomas, selecione o idioma que você deseja filtrar.

A lista exibe somente os léxicos para o idioma escolhido.

AWS CLI

O Amazon Polly fornece a operação da API ListLexicons, que pode ser usada para obter a lista de léxicos de pronúncia na sua conta em uma região da AWS específica. A chamada da AWS CLI a seguir lista os léxicos na sua conta na região us-east-2.

```
aws polly list-lexicons
```

A seguir está um exemplo de resposta, mostrando dois léxicos de nome `w3c` e `tomato`. Para cada léxico, a resposta retorna metadados, como o código do idioma para o qual o léxico se aplica, o número de lexemas definido no léxico, o tamanho em bytes, e assim por diante. O código de idioma descreve um idioma e uma localidade aos quais se aplicam os lexemas definidos no léxico.

```
{
    "Lexicons": [
        {
            "Attributes": {
                "LanguageCode": "en-US",
                "LastModified": 1474222543.989,
                "Alphabet": "ipa",
                "LexemesCount": 1,
                "LexiconArn": "arn:aws:polly:aws-region:account-id:lexicon/w3c",
                "Size": 495
            },
            "Name": "w3c"
        },
        {
            "Attributes": {
                "LanguageCode": "en-US",
                "LastModified": 1473099290.858,
                "Alphabet": "ipa",
                "LexemesCount": 1,
                "LexiconArn": "arn:aws:polly:aws-region:account-id:lexicon/tomato",
                "Size": 645
            },
            "Name": "tomato"
```

```
        }
    ]
}
```

Os seguintes recursos contêm informações adicionais sobre a operação ListLexicons:

- Amostra de Java: ListLexicons
- Amostra de Python (Boto3): ListLexicon

Fazer download dos léxicos no console

O processo a seguir descreve como fazer o download de um ou mais léxicos. Você pode adicionar, remover ou modificar entradas do léxico no arquivo e, em seguida, carregá-las novamente para manter seu léxico atualizado.

Console

Para baixar um ou mais léxicos

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Polly em https://console.aws.amazon.com/polly/.
2. Escolha a guia Lexicons.
3. Escolha os léxicos que você deseja baixar.
    a. Para baixar um único léxico, escolha o nome na lista.
    b. Para baixar vários léxicos como um único arquivo compactado, marque a caixa de seleção ao lado de cada entrada na lista que você deseja baixar.
4. Escolha Baixar.
5. Abra a pasta para onde você deseja fazer o download do léxico.
6. Escolha Salvar.

AWS CLI

O Amazon Polly fornece a operação da API GetLexicon para recuperar o conteúdo de um léxico de pronúncia armazenado em sua conta em uma região específica.

O seguinte comando da `get-lexicon` AWS CLI recupera o conteúdo do léxico `example`.

```
aws polly get-lexicon \
--name example
```

Se você ainda não tiver um léxico armazenado em sua conta, pode usar a operação `PutLexicon` para armazenar um. Para obter mais informações, consulte Fazer upload de um léxico.

Veja a seguir uma resposta de exemplo. Além do léxico de conteúdo, a resposta retorna os metadados, como o código de idioma ao qual o léxico se aplica, número de lexemas definido no léxico, o nome de recurso da Amazon (ARN) do recurso e o tamanho do léxico em bytes. O valor `LastModified` é um carimbo de data/hora do Unix.

```
{
    "Lexicon": {
        "Content": "lexicon content in plain text PLS format",
        "Name": "example"
    },
    "LexiconAttributes": {
        "LanguageCode": "en-US",
        "LastModified": 1474222543.989,
        "Alphabet": "ipa",
        "LexemesCount": 1,
        "LexiconArn": "arn:aws:polly:us-east-2:account-id:lexicon/example",
        "Size": 495
    }
}
```

Os recursos a seguir contêm exemplos de código adicionais para a operação GetLexicon:

- Amostra de Java: GetLexicon
- Amostra de Python (Boto3): GetLexicon

Excluir um léxico

O processo a seguir descreve como excluir um léxico. Depois de excluir o léxico, você deve adicioná-lo de volta antes de usá-lo novamente. Você pode excluir um ou mais léxicos ao mesmo tempo marcando as caixas de seleção ao lado de cada léxico.

Console

Para excluir um léxico

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Polly em https://console.aws.amazon.com/polly/.
2. Escolha a guia Lexicons.
3. Escolha um ou mais léxicos que você deseja excluir da lista.
4. Escolha Excluir.
5. Insira o texto de confirmação e escolha Excluir para remover o léxico da região ou Cancelar para mantê-lo.

AWS CLI

O Amazon Polly fornece a operação da API DeleteLexicon para excluir um léxico de pronúncia de uma determinada região da AWS na sua conta. A AWS CLI a seguir exclui o léxico especificado.

O exemplo da AWS CLI a seguir está formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um acento circunflexo (^) e use aspas completas (") ao redor do texto de entrada com aspas simples (') para tags internas.

```
aws polly delete-lexicon \
--name example
```

Os seguintes recursos contêm informações adicionais sobre a operação DeleteLexicon:

- Amostra de Java: DeleteLexicon
- Amostra de Python (Boto3): DeleteLexicon

Arquivos de áudio longos

Para criar arquivos TTS para grandes passagens de texto, use a funcionalidade síntese assíncrona do Amazon Polly. Isso usa os três `SpeechSynthesisTask` APIs:

- `StartSpeechSynthesisTask`: começa uma nova tarefa de síntese.
- `GetSpeechSynthesisTask`: retorna detalhes sobre uma tarefa de síntese enviada anteriormente.
- `ListSpeechSynthesisTasks`: lista todas as tarefas de síntese enviadas.

A operação `SynthesizeSpeech` produz áudio em tempo quase real, com latência relativamente baixa na maioria dos casos. Para fazer isso, a operação pode sintetizar somente 3000 caracteres.

O recurso de Síntese assíncrona do Amazon Polly supera o desafio de processar um grande documento de texto alterando a maneira como documento é sintetizado e retornado. Quando uma solicitação de síntese é feita enviando um texto de entrada com `StartSpeechSynthesisTask`, o Amazon Polly coloca as solicitações em fila e as processa de maneira assíncrona em segundo plano assim que os recursos do sistema ficam disponíveis. Em seguida, o Amazon Polly carrega o fluxo de fala ou marcas de fala diretamente para o bucket do Amazon Simple Storage Service (Amazon S3, obrigatório) e notifica você sobre a disponibilidade do arquivo concluído através do seu tópico do SNS (opcional).

Dessa forma, todas as funcionalidades, exceto o processamento em tempo quase real, estão disponíveis para textos de até 100.000 caracteres faturáveis (ou 200.000 caracteres no total).

Para sintetizar um documento usando esse método, é necessário ter um bucket do Amazon S3 gravável no qual o arquivo de áudio possa ser salvo. Você poderá ser notificado quando o áudio sintetizado estiver pronto, fornecendo um identificador opcional do tópico do SNS. Quando a tarefa de síntese for concluída, o Amazon Polly publicará uma mensagem nesse tópico. Essa mensagem também pode conter informações de erro úteis em casos em que a tarefa de síntese não tenha obtido êxito. Para fazer isso, verifique se o usuário que está criando a tarefa de síntese também pode publicar no tópico do SNS. Consulte a documentação do Amazon SNS para obter mais informações sobre como criar e inscrever-se em um tópico do SNS.

Criptografia

É possível armazenar o arquivo de saída em um formato criptografado no bucket do S3, se desejar. Para fazer isso, habilite a criptografia do bucket do Amazon S3, que usa uma das mais fortes cifras de bloco disponíveis, o Advanced Encryption Standard de 256 bits (AES-256).

Tópicos

- Configurar a política do IAM para a síntese assíncrona
- Criar arquivos de áudio longos

Configurar a política do IAM para a síntese assíncrona

Para usar a funcionalidade de síntese assíncrona, será necessária uma política do IAM que permita o seguinte:

- Usar novas operações do Amazon Polly
- gravar no bucket do S3 de saída
- publicar no tópico do SNS de status [opcional]

A política a seguir concede apenas as permissões necessárias exigidas para a síntese assíncrona e podem ser anexada ao usuário do IAM.

Criar arquivos de áudio longos

É possível usar o console do Amazon Polly para criar falas longas usando a síntese assíncrona com a mesma funcionalidade que você pode usar com a AWS CLI. Isso é feito usando a guia Text-to-Speech (Texto para fala) da mesma maneira que qualquer outra síntese.

Console

A outra funcionalidade de síntese assíncrona também está disponível pelo console. A guia S3 synthesis tasks (Tarefas de síntese do S3) reflete a funcionalidade `ListSpeechSynthesisTasks`, exibindo todas as tarefas salvas no bucket do S3 e permitindo filtrá-las, se desejar. Clicar em uma única tarefa específica mostra seus detalhes, refletindo a funcionalidade `GetSpeechSynthesisTask`.

Como sintetizar um texto grande usando o console do Amazon Polly

1. Faça login no Console de gerenciamento da AWS e abra o console do Amazon Polly em. https://console.aws.amazon.com/polly/
2. Selecione a guia Text-to-Speech. Selecione Formato longo como o mecanismo, se apropriado.
3. Com SSML ativo ou inativo, digite ou cole seu texto na caixa de entrada.
4. Selecione o idioma, a região e a voz para seu texto.
5. Escolha Salvar no S3.

> **Note**
>
> As opções Fazer download e Ouvir estarão esmaecidas se o comprimento do texto estiver acima do limite de 3.000 caracteres da operação `SynthesizeSpeech` em tempo real.

6. O console abre um formulário para que você possa escolher onde armazenar o arquivo de saída.
   a. Preencha o nome do bucket do Amazon S3 de destino.
   b. Se preferir, preencha o prefixo de chaves da saída.

   > **Note**
   >
   > A bucket do S3 de saída deve ser gravável.

   c. Se você deseja ser notificado quando a tarefa de síntese tarefa for concluída, forneça o identificador opcional do tópico do SNS.

   > **Note**
   >
   > O SNS deve estar aberto para publicação pelo usuário atual do console para usar essa opção. Para obter mais informações, consulte Amazon Simple Notification Service (SNS)

   d. Escolha Salvar no S3.

Para recuperar informações sobre suas tarefas de síntese de fala

1. No console, selecione a guia Tarefas de síntese do S3.
2. As tarefas são exibidas por ordem de data. Para filtrar as tarefas por status, escolha Todos os status e defina o status a ser usado.
3. Para visualizar os detalhes de uma tarefa específica, selecione o Task ID (ID da tarefa) vinculado.

AWS CLI

A funcionalidade de síntese assíncrona do Amazon Polly usa três `SpeechSynthesisTask` APIs para trabalhar com grandes quantidades de texto:

- `StartSpeechSynthesisTask`: começa uma nova tarefa de síntese.
- `GetSpeechSynthesisTask`: retorna detalhes sobre uma tarefa de síntese enviada anteriormente.
- `ListSpeechSynthesisTasks`: lista todas as tarefas de síntese enviadas.

Sintetizar grandes quantidades de texto (**`StartSpeechSynthesisTask`**)

Quando quiser criar um arquivo de áudio maior do que pode criar com a operação `SynthesizeSpeech` em tempo real, use a operação `StartSpeechSynthesisTask`. Além dos argumentos necessários para a operação `SynthesizeSpeech`, a `StartSpeechSynthesisTask` também exige o nome de um bucket do Amazon S3. Outros dois argumentos opcionais também estão disponíveis: um prefixo de chaves para o arquivo de saída e o ARN de um tópico do SNS, se quiser receber notificações de status sobre a tarefa.

- `OutputS3BucketName`: o nome do bucket do Amazon S3 para o qual deve ser feito o upload da síntese. Esse bucket deve estar na mesma região que o serviço do Amazon Polly. Além disso, o usuário do IAM que está sendo usado para fazer a chamada deve ter acesso ao bucket. [Obrigatório]
- `OutputS3KeyPrefix`: prefixo de chaves para o arquivo de saída. Use este parâmetro se deseja salvar o arquivo de saída da fala em uma chave personalizada como um diretório em seu bucket. [Opcional]
- `SnsTopicArn`: o ARN do tópico do SNS a ser usado se quiser receber notificações sobre o status da tarefa. Esse tópico do SNS deve estar na mesma região que o serviço do Amazon

Polly. Além disso, o usuário do IAM que está sendo usado para fazer a chamada deve ter acesso ao tópico. (Opcional)

Por exemplo, o exemplo a seguir pode ser usado para executar o `start-speech-synthesis-task` AWS CLI comando na região Leste dos EUA (Ohio):

O AWS CLI exemplo a seguir está formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um acento circunflexo (^) e use aspas completas (") ao redor do texto de entrada com aspas simples (') para tags internas.

```
aws polly start-speech-synthesis-task \
  --region us-east-2 \
  --endpoint-url "https://polly.us-east-2.amazonaws.com/" \
  --output-format mp3 \
  --output-s3-bucket-name your-bucket-name \
  --output-s3-key-prefix optional/prefix/path/file \
  --voice-id Joanna \
  --text file://text_file.txt
```

Isso resultará em uma resposta semelhante a esta:

```
"SynthesisTask":
{
     "OutputFormat": "mp3",
     "OutputUri": "https://s3.us-east-2.amazonaws.com/your-bucket-name/optional/
prefix/path/file.<task_id>.mp3",
     "TextType": "text",
     "CreationTime": [..],
     "RequestCharacters": [..],
     "TaskStatus": "scheduled",
     "TaskId": [task_id],
     "VoiceId": "Joanna"
 }
```

A operação `start-speech-synthesis-task` retorna diversos campos novos:

- `OutputUri`: o local do seu arquivo de saída da fala.
- `TaskId`: um identificador exclusivo para a tarefa de síntese de fala gerado pelo Amazon Polly.
- `CreationTime`: um time stamp de quando a tarefa foi inicialmente enviada.

- `RequestCharacters`: o número de caracteres faturáveis na tarefa.
- `TaskStatus`: fornece informações sobre o status da tarefa enviada.

  Quando a tarefa for enviada, o status inicial será exibido como `scheduled`. Quando o Amazon Polly começar a processar a tarefa, o status será alterado para `inProgress` e depois para `completed` ou `failed`. Se a tarefa falhar, uma mensagem de erro será retornada ao chamar a ListSpeechSynthesisTasks operação GetSpeechSynthesisTask ou.

Quando a tarefa for concluída, o arquivo da fala estará disponível no local especificado em `OutputUri`.

Recuperar informações sobre sua tarefa de síntese de fala

É possível obter informações sobre uma tarefa, como erros, status e assim por diante, usando a operação `GetSpeechSynthesisTask`. Para fazer isso, você precisará do `task-id` retornado pelo `StartSpeechSynthesisTask`.

Por exemplo, o exemplo a seguir pode ser usado para executar o `get-speech-synthesis-task` AWS CLI comando:

```
aws polly get-speech-synthesis-task \
--region us-east-2 \
--endpoint-url "https:// polly.us-east-2.amazonaws.com/" \
--task-id task identifier
```

Também é possível listar todas as tarefas de síntese de fala que você executou na região atual usando a operação `ListSpeechSynthesisTasks`.

Por exemplo, o exemplo a seguir pode ser usado para executar o `list-speech-synthesis-tasks` AWS CLI comando:

```
aws polly list-speech-synthesis-tasks \
--region us-east-2 \
--endpoint-url "https:// polly.us-east-2.amazonaws.com/"
```

Cotas no Amazon Polly

O Amazon Polly aplica cotas ao tráfego de clientes rejeitando solicitações excessivas. A cota padrão para a solicitação `SynthesizeSpeech` com vozes padrão é de 80 transações por segundo (tps), em uma única região, para uma única Conta da AWS. Se os limites não aumentarem e você gerar 100 solicitações de `SynthesizeSpeech` por segundo usando uma voz padrão, 80 solicitações por segundo serão bem-sucedidas e 20 solicitações por segundo serão limitadas pelo controle de utilização do Amazon Polly. Essas solicitações retornarão uma resposta com o status HTTP 400 e um cabeçalho de resposta indicando `ThrottlingException`. O Amazon Polly também controla a utilização do tráfego para todas as operações com base na taxa de solicitação.

Exemplos de limites de síntese de fala

- Sintetizar as primeiras 24 letras do alfabeto inglês, uma letra por vez. Se a síntese de cada letra levar menos de 50 milissegundos, com um limite de operação de 8 tps, sintetizar 24 letras tomará pelo menos três segundos. Nesse tempo você pode sintetizar até oito letras por segundo. Qualquer solicitação adicional será limitada. Como as solicitações duram pouco, serão sintetizadas em série sem sobreposição.
- Sintetizar 16 parágrafos de texto. Se cada parágrafo for sintetizado e totalmente recebido pelo cliente em dois segundos ou menos, com um limite de operação de oito solicitações simultâneas, serão necessários pelo menos quatro segundos para sintetizar todos os 16 artigos. No primeiro segundo, você pode iniciar até oito solicitações. Durante solicitações simultâneas, tentativas de iniciar uma nova síntese serão limitadas pelo controle de utilização devido ao limite de simultaneidade. Você pode sintetizar os oito parágrafos restantes após os primeiros dois segundos, depois da conclusão do primeiro lote de solicitações.

Mantenha os seguintes limites em mente ao usar o Amazon Polly.

Tópicos

- Regiões compatíveis
- Cotas e taxas de controle de utilização
- Léxicos de pronúncia
- SynthesizeSpeech Operações de API
- SpeechSynthesisTask Operações de API
- Speech Synthesis Markup Language (SSML)

Regiões compatíveis

Para obter uma lista das AWS regiões em que o Amazon Polly está disponível, consulte Amazon Polly Endpoints and Quotas no. Referência geral da Amazon Web Services

- Para regiões que oferecem suporte a vozes generativas, consulte Generative voices.
- Para regiões que oferecem suporte a vozes de formato longo, consulte Long-form vozes.
- Para saber quais regiões são compatíveis com vozes neurais, consulte the section called "Compatibilidade de recursos e regiões" para TTS neural.

Cotas e taxas de controle de utilização

A tabela a seguir define as taxas de controle de utilização por operação do Amazon Polly. Você pode usar o Console de gerenciamento da AWS para solicitar aumentos de cota para as cotas ajustáveis quando necessário.

Operation	Limite
Vocabulário	

`DeleteLexicon`
`PutLexicon`
`GetLexicon`
`ListLexicons`
	Quaisquer 2 transações por segundo (tps) entre essas operações combinadas.
	Intermitência máxima permitida de 4 tps.

Fala	
`DescribeVoices`	80 tps com um limite de intermitência de 100 tps
`SynthesizeSpeech`	Voz generativa: 8 tps Long-form voz: 8 tps com um limite de rajada de 10 tps Voz neural: 8 tps com um limite de intermitência de 10 tps Voz padrão: 80 tps com um limite de intermitência de 100 tps

Operation	Limite
`StartSpeechSynthesisTask`	Voz generativa: 1 tps Long-form voz: 1 tps Voz neural: 10 tps Voz padrão: 10 tps com um limite de intermitência de 12 tps
`StartSpeechSynthesisStream`	Voz generativa: 8 tps
`GetSynthesizeSpeechTask` e `ListSynthesizeSpeechTask`	Máximo permitido de 10 tps combinados

Solicitações simultâneas

Para vozes generativas, o Amazon Polly oferece suporte a até 26 solicitações simultâneas. Para vozes em formato longo, o Amazon Polly oferece suporte a até 26 solicitações simultâneas. Para vozes neurais, o Amazon Polly é compatível com 8 tps com um limite de intermitência de 10 tps, para até 18 solicitações simultâneas. O Amazon Polly também é compatível com limites para solicitações simultâneas. Para vozes padrão, o Amazon Polly é compatível com 80 tps para até 80 solicitações simultâneas.

Pois StartSpeechSynthesisStream, o Amazon Polly oferece suporte a até 8 solicitações simultâneas.

Melhores práticas para mitigar o controle de utilização

- Faça uma nova tentativa dos controles de utilização com recuo e tremulação para distribuir a carga por um curto período e lidar com picos inesperados de uso sem comprometer a disponibilidade. O Catálogo de exemplos de código da AWS já está configurado para isso por padrão em muitas linguagens de programação. Visite o comportamento de nova tentativa do recurso para ver os detalhes.
- Use as Métricas do Amazon Polly. O Amazon Polly publica automaticamente CloudWatch para analisar seu uso atual e prever o crescimento do uso.

> **Note**
>
> Antes de solicitar um aumento de cota (quando aplicável), calcule as necessidades de tps seguindo as diretrizes desta página. O Amazon Polly protege somente os recursos computacionais necessários de acordo com a demanda do cliente para manter os custos baixos.

Léxicos de pronúncia

- Você pode armazenar até 100 léxicos por conta.
- Os nomes do léxico podem ser uma string alfanumérica de até 20 caracteres.
- Cada léxico pode ter até 40.000 caracteres de tamanho. (Observe que o tamanho do léxico afeta a latência da SynthesizeSpeech operação.)
- Você pode especificar até 100 caracteres para a substituição de cada <phoneme> ou <alias> em um léxico.

Para obter mais informações sobre o uso de léxicos, consulte Gerenciar léxicos.

SynthesizeSpeech Operações de API

Ao estimar o uso de `SynthesizeSpeech`, lembre-se de que o áudio produzido pelo Amazon Polly, especialmente para aplicativos interativos, geralmente leva alguns segundos para ser reproduzido. Isso reduz a taxa de solicitações ao `SynthesizeSpeech`, mesmo para um grande número de consumidores simultâneos. Além disso, o Amazon Polly faz o controle de utilização de solicitações de `SynthesizeSpeech` pelo número de solicitações simultâneas sintetizadas. Não há uma configuração separada para solicitações simultâneas. O limite de solicitações simultâneas sempre tem o mesmo valor que o número de tps permitido e escala com ele.

Exemplo de aplicativo para contos literários. Você pode usar o Amazon Polly para criar um aplicativo para reproduzir uma série de histórias curtas. Com esse tipo de aplicativo, a primeira história começaria a ser reproduzida, depois a próxima, e assim por diante até que o usuário feche o aplicativo. Cada história levaria cerca de 0,5 segundo para ser sintetizada e dez segundos para ser reproduzida. Nesse cenário, você poderia esperar uma chamada para `SynthesizeSpeech` a cada dez segundos que o cliente passar usando o aplicativo. Isso se traduziria em uma chamada por segundo para cada dez clientes usando o aplicativo simultaneamente. Se você tiver mil

clientes usando o aplicativo simultaneamente, pode esperar uma taxa média de chamadas para `SynthesizeSpeech` de apenas cem transações por segundo.

Observe os seguintes limites relacionados ao uso da operação de API `SynthesizeSpeech`:

- O tamanho do texto de entrada pode ser de até 3000 caracteres cobrados (6000 caracteres no total). Tags SSML não são contadas como caracteres cobrados.
- Você pode especificar até cinco léxicos para aplicar ao texto de entrada.
- O fluxo de áudio de saída (síntese) é limitado a 10 minutos. Depois de ser atingido, qualquer fala restante será cortada.

Para obter mais informações, consulte SynthesizeSpeech.

> **Note**
>
> Algumas limitações de operação de API `SynthesizeSpeech` podem ser ignoradas usando a operação de API `StartSythensizeSpeechTask`. Para obter mais informações, consulte Arquivos de áudio longos.

SpeechSynthesisTask Operações de API

Observe os seguintes limites relacionados ao uso das operações de API `StartSpeechSynthesisTask`, `GetSpeechSynthesisTask` e `ListSpeechSynthesisTasks`:

- O tamanho do texto de entrada pode ser de até 100,000 caracteres cobrados (200,000 caracteres no total). Tags SSML não são contadas como caracteres cobrados.
- Você pode especificar até cinco léxicos para aplicar ao texto de entrada.

Speech Synthesis Markup Language (SSML)

Observe os seguintes limites relacionados ao uso de SSML:

- Não há compatibilidade para as tags `<audio>`, `<lexicon>`, `<lookup>` e `<voice>`.
- Os elementos `<break>` podem especificar uma duração máxima de 10 segundos cada.
- A tag `<prosody>` não é compatível com valores para o atributo de velocidade inferior a -80%.

Para obter mais informações, consulte Gerar fala com base em documentos SSML.

Exemplo de código e aplicações para o Amazon Polly

Esta seção apresenta amostras de código e aplicativos de exemplo que podem ser usados para explorar o Amazon Polly.

O tópico Código de exemplo contém fragmentos de código organizados por linguagem de programação e separados em exemplos de diferentes funcionalidades do Amazon Polly. O tópico Aplicativo de exemplo contém aplicativos organizados por linguagem de programação que podem ser usados de modo independente para explorar o Amazon Polly.

Antes de começar a usar esses exemplos, recomendamos que você primeiro leia Como o Amazon Polly funciona e siga as etapas descritas em Conceitos básicos do Amazon Polly.

Tópicos

- Exemplos em Java
- Exemplos de Python
- Exemplo de Java
- Exemplo de Python (HTML5 cliente e servidor Python)
- Exemplo para iOS
- Exemplo para Android

Exemplos em Java

Os exemplos de código a seguir mostram como usar os aplicativos baseados em Java para realizar várias tarefas com o Amazon Polly. Esses exemplos não são completos, mas podem ser incluídos em aplicativos Java maiores que usam o AWS SDK para Java.

Trechos de código

- DeleteLexicon
- DescribeVoices
- GetLexicon
- ListLexicons
- PutLexicon
- StartSpeechSynthesisTask

- Marcas de fala
- SynthesizeSpeech

DeleteLexicon

A amostra de código Java a seguir mostra como usar aplicativos baseados em Java para excluir um léxico específico armazenado em uma região AWS. Um léxico que foi excluído não está disponível para síntese de voz, nem pode ser recuperado usando as APIs `GetLexicon` ou `ListLexicon`.

Para obter mais informações sobre esta operação, consulte a referência da API `DeleteLexicon`.

SDK v2

```
/*
   Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
   SPDX-License-Identifier: Apache-2.0
*/

package com.example.polly;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.DeleteLexiconRequest;
import software.amazon.awssdk.services.polly.model.DeleteLexiconResponse;
import software.amazon.awssdk.services.polly.model.PollyException ;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DeleteLexiconSample {

    public static void main(String args[]) {

        PollyClient polly = PollyClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
```

```
                .build();

        deleteLexicon(polly) ;
        polly.close();
    }

    private static String LEXICON_NAME = "SampleLexicon";
    public static void deleteLexicon(PollyClient client) {

        try {
            DeleteLexiconRequest deleteLexiconRequest = DeleteLexiconRequest.builder()
                    .name(LEXICON_NAME).build();

            DeleteLexiconResponse deleteLexiconResult =
 client.deleteLexicon(deleteLexiconRequest);

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
            System.exit(1);
        }
    }

}
```

SDK v1

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.DeleteLexiconRequest;

public class DeleteLexiconSample {
    private String LEXICON_NAME = "SampleLexicon";

    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void deleteLexicon() {
        DeleteLexiconRequest deleteLexiconRequest = new
 DeleteLexiconRequest().withName(LEXICON_NAME);

        try {
            client.deleteLexicon(deleteLexiconRequest);
```

```
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

DescribeVoices

A amostra de código Java a seguir mostra como usar aplicativos baseados em Java para produzir uma lista de vozes que estão disponíveis para uso ao solicitar uma síntese de fala. Opcionalmente, você pode especificar um código de idioma para filtrar as vozes disponíveis. Por exemplo, se você especificar en-US, a operação retornará uma lista de todas as vozes disponíveis em inglês dos EUA.

Para obter mais informações sobre esta operação, consulte a referência da API `DescribeVoices`.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.DescribeVoicesRequest;
import com.amazonaws.services.polly.model.DescribeVoicesResult;

public class DescribeVoicesSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void describeVoices() {
        DescribeVoicesRequest allVoicesRequest = new DescribeVoicesRequest();
        DescribeVoicesRequest enUsVoicesRequest = new
 DescribeVoicesRequest().withLanguageCode("en-US");

        try {
            String nextToken;
            do {
                DescribeVoicesResult allVoicesResult =
 client.describeVoices(allVoicesRequest);
                nextToken = allVoicesResult.getNextToken();
                allVoicesRequest.setNextToken(nextToken);

                System.out.println("All voices: " + allVoicesResult.getVoices());
            } while (nextToken != null);

            do {
```

```
                DescribeVoicesResult enUsVoicesResult =
 client.describeVoices(enUsVoicesRequest);
                nextToken = enUsVoicesResult.getNextToken();
                enUsVoicesRequest.setNextToken(nextToken);

                System.out.println("en-US voices: " + enUsVoicesResult.getVoices());
            } while (nextToken != null);
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

GetLexicon

A amostra de código Java a seguir mostra como usar aplicativos baseados em Java para produzir o conteúdo de um léxico específico de pronúncia armazenado em uma região da AWS.

Para obter mais informações sobre esta operação, consulte a referência da API GetLexicon.

SDK v2

```
/*
   Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
   SPDX-License-Identifier: Apache-2.0
*/

package com.example.polly;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.GetLexiconRequest;
import software.amazon.awssdk.services.polly.model.GetLexiconResponse;
import software.amazon.awssdk.services.polly.model.PollyException ;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
```

```
 */
public class GetLexiconSample {

    public static void main(String args[]) {

        PollyClient polly = PollyClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        getLexicon(polly) ;
        polly.close();
    }

    private static String LEXICON_NAME = "SampleLexicon";
    public static void getLexicon(PollyClient client) {

        try {
            GetLexiconRequest getLexiconRequest = GetLexiconRequest.builder()
                    .name(LEXICON_NAME).build();

            GetLexiconResponse getLexiconResult = client.getLexicon(getLexiconRequest);
            System.out.println("The name of the Lexicon is " +
 getLexiconResult.lexicon().name());

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
            System.exit(1);
        }
    }

}
```

SDK v1

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.GetLexiconRequest;
import com.amazonaws.services.polly.model.GetLexiconResult;

public class GetLexiconSample {
```

```
    private String LEXICON_NAME = "SampleLexicon";

    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void getLexicon() {
        GetLexiconRequest getLexiconRequest = new
 GetLexiconRequest().withName(LEXICON_NAME);

        try {
            GetLexiconResult getLexiconResult = client.getLexicon(getLexiconRequest);
            System.out.println("Lexicon: " + getLexiconResult.getLexicon());
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

ListLexicons

O código Java de exemplo a seguir mostra como usar aplicativos baseados em Java para produzir uma lista de léxicos de pronúncia armazenados em uma região da AWS.

Para obter mais informações sobre esta operação, consulte a referência da API `ListLexicons`.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.LexiconAttributes;
import com.amazonaws.services.polly.model.LexiconDescription;
import com.amazonaws.services.polly.model.ListLexiconsRequest;
import com.amazonaws.services.polly.model.ListLexiconsResult;

public class ListLexiconsSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void listLexicons() {
        ListLexiconsRequest listLexiconsRequest = new ListLexiconsRequest();

        try {
            String nextToken;
            do {
```

```
                ListLexiconsResult listLexiconsResult =
 client.listLexicons(listLexiconsRequest);
                nextToken = listLexiconsResult.getNextToken();
                listLexiconsRequest.setNextToken(nextToken);

                for (LexiconDescription lexiconDescription :
 listLexiconsResult.getLexicons()) {
                    LexiconAttributes attributes = lexiconDescription.getAttributes();
                    System.out.println("Name: " + lexiconDescription.getName()
                            + ", Alphabet: " + attributes.getAlphabet()
                            + ", LanguageCode: " + attributes.getLanguageCode()
                            + ", LastModified: " + attributes.getLastModified()
                            + ", LexemesCount: " + attributes.getLexemesCount()
                            + ", LexiconArn: " + attributes.getLexiconArn()
                            + ", Size: " + attributes.getSize());
                }
            } while (nextToken != null);
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

PutLexicon

A amostra de código Java a seguir mostra como usar aplicativos baseados em Java para armazenar um léxico de pronúncia em uma região da AWS.

Para obter mais informações sobre esta operação, consulte a referência da API PutLexicon.

SDK v2

```
/*
   Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
   SPDX-License-Identifier: Apache-2.0
*/

package com.example.polly;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.PutLexiconRequest;
```

```
import software.amazon.awssdk.services.polly.model.PutLexiconResponse;
import software.amazon.awssdk.services.polly.model.PollyException ;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class PutLexiconSample {

    public static void main(String args[]) {

        PollyClient polly = PollyClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        putLexicon(polly) ;
        polly.close();
    }

    private static String LEXICON_CONTENT = "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>" +
            "<lexicon version=\"1.0\" xmlns=\"http://www.w3.org/2005/01/pronunciation-
lexicon\" xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" " +
            "xsi:schemaLocation=\"http://www.w3.org/2005/01/pronunciation-lexicon
 http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd\" " +
            "alphabet=\"ipa\" xml:lang=\"en-US\">" +
            "<lexeme><grapheme>test1</grapheme><alias>test2</alias></lexeme>" +
            "</lexicon>";
    private static String LEXICON_NAME = "SampleLexicon";
    public static void putLexicon(PollyClient client) {

        try {
            PutLexiconRequest putLexiconRequest = PutLexiconRequest.builder()
                    .name(LEXICON_NAME).content(LEXICON_CONTENT).build();

            PutLexiconResponse putLexiconResult = client.putLexicon(putLexiconRequest);

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
```

```
            System.exit(1);
        }
    }

}
```

SDK v1

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.PutLexiconRequest;

public class PutLexiconSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    private String LEXICON_CONTENT = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>" +
            "<lexicon version=\"1.0\" xmlns=\"http://www.w3.org/2005/01/pronunciation-
lexicon\" xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" " +
            "xsi:schemaLocation=\"http://www.w3.org/2005/01/pronunciation-lexicon
 http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd\" " +
            "alphabet=\"ipa\" xml:lang=\"en-US\">" +
            "<lexeme><grapheme>test1</grapheme><alias>test2</alias></lexeme>" +
            "</lexicon>";
    private String LEXICON_NAME = "SampleLexicon";

    public void putLexicon() {
        PutLexiconRequest putLexiconRequest = new PutLexiconRequest()
                .withContent(LEXICON_CONTENT)
                .withName(LEXICON_NAME);

        try {
            client.putLexicon(putLexiconRequest);
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

StartSpeechSynthesisTask

O seguinte exemplo de código Java mostra como usar aplicativos baseados em Java para sintetizar uma fala longa (de até 100 mil caracteres faturados) e armazená-la diretamente em um bucket do Amazon S3.

Para obter mais informações, consulte a referência da API StartSpeechSynthesisTask.

SDK v2

```
/*
   Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
   SPDX-License-Identifier: Apache-2.0
*/

package com.example.polly;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.*;

import java.time.Duration;
import org.awaitility.Durations;
import org.awaitility.Awaitility;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class StartSpeechSynthesisTaskSample {

    public static void main(String args[]) {

        PollyClient polly = PollyClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
```

```
        startSpeechSynthesisTask(polly) ;
        polly.close();
    }

    private static final String PLAIN_TEXT = "This is a sample text to be
 synthesized.";
    private static final String OUTPUT_FORMAT_MP3 = OutputFormat.MP3.toString();
    private static final String OUTPUT_BUCKET = "synth-books-buckets";
    private static final String SNS_TOPIC_ARN = "arn:aws:sns:eu-
west-2:123456789012:synthesize-finish-topic";
    private static final Duration SYNTHESIS_TASK_POLL_INTERVAL =
 Durations.FIVE_SECONDS;
    private static final Duration SYNTHESIS_TASK_POLL_DELAY = Durations.TEN_SECONDS;
    private static final Duration SYNTHESIS_TASK_TIMEOUT = Durations.FIVE_MINUTES;
    public static void startSpeechSynthesisTask(PollyClient client) {

        try {
            StartSpeechSynthesisTaskRequest startSpeechSynthesisTaskRequest =
 StartSpeechSynthesisTaskRequest.builder()

 .outputFormat(OUTPUT_FORMAT_MP3).text(PLAIN_TEXT).textType(TextType.TEXT)

 .voiceId(VoiceId.AMY).outputS3BucketName(OUTPUT_BUCKET).snsTopicArn(SNS_TOPIC_ARN)
                    .engine("neural").build();

            StartSpeechSynthesisTaskResponse startSpeechSynthesisTaskResponse =
                    client.startSpeechSynthesisTask(startSpeechSynthesisTaskRequest);
            String taskId = startSpeechSynthesisTaskResponse.synthesisTask().taskId();

            Awaitility.await().with()
                    .pollInterval(SYNTHESIS_TASK_POLL_INTERVAL)
                    .pollDelay(SYNTHESIS_TASK_POLL_DELAY)
                    .atMost(SYNTHESIS_TASK_TIMEOUT)
                    .until(
                            () -> getSynthesisTaskStatus(client,
 taskId).equals(TaskStatus.COMPLETED.toString())
                    );

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
            System.exit(1);
        }
    }
```

```
    private static String getSynthesisTaskStatus(PollyClient client, String taskId) {
        GetSpeechSynthesisTaskRequest getSpeechSynthesisTaskRequest =
 GetSpeechSynthesisTaskRequest.builder()
                .taskId(taskId).build();
        GetSpeechSynthesisTaskResponse result =
 client.getSpeechSynthesisTask(getSpeechSynthesisTaskRequest);
        return result.synthesisTask().taskStatusAsString();
    }

}
```

SDK v1

```
package com.amazonaws.parrot.service.tests.speech.task;

import com.amazonaws.parrot.service.tests.AbstractParrotServiceTest;
import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.model.*;
import org.awaitility.Duration;

import java.util.concurrent.TimeUnit;

import static org.awaitility.Awaitility.await;

public class StartSpeechSynthesisTaskSample {

    private static final int SYNTHESIS_TASK_TIMEOUT_SECONDS = 300;
    private static final AmazonPolly AMAZON_POLLY_CLIENT =
 AmazonPollyClientBuilder.defaultClient();
    private static final String PLAIN_TEXT = "This is a sample text to be
 synthesized.";
    private static final String OUTPUT_FORMAT_MP3 = OutputFormat.Mp3.toString();
    private static final String OUTPUT_BUCKET = "synth-books-buckets";
    private static final String SNS_TOPIC_ARN = "arn:aws:sns:eu-
west-2:123456789012:synthesize-finish-topic";
    private static final Duration SYNTHESIS_TASK_POLL_INTERVAL = Duration.FIVE_SECONDS;
    private static final Duration SYNTHESIS_TASK_POLL_DELAY = Duration.TEN_SECONDS;

    public static void main(String... args) {
        StartSpeechSynthesisTaskRequest request = new StartSpeechSynthesisTaskRequest()
                .withOutputFormat(OUTPUT_FORMAT_MP3)
                .withText(PLAIN_TEXT)
                .withTextType(TextType.Text)
```

```
                .withVoiceId(VoiceId.Amy)
                .withOutputS3BucketName(OUTPUT_BUCKET)
                .withSnsTopicArn(SNS_TOPIC_ARN)
                .withEngine("neural");

        StartSpeechSynthesisTaskResult result =
 AMAZON_POLLY_CLIENT.startSpeechSynthesisTask(request);
        String taskId = result.getSynthesisTask().getTaskId();

        await().with()
                .pollInterval(SYNTHESIS_TASK_POLL_INTERVAL)
                .pollDelay(SYNTHESIS_TASK_POLL_DELAY)
                .atMost(SYNTHESIS_TASK_TIMEOUT_SECONDS, TimeUnit.SECONDS)
                .until(
                        () ->
 getSynthesisTaskStatus(taskId).equals(TaskStatus.Completed.toString())
                );
    }

    private static SynthesisTask getSynthesisTask(String taskId) {
        GetSpeechSynthesisTaskRequest getSpeechSynthesisTaskRequest = new
 GetSpeechSynthesisTaskRequest()
                .withTaskId(taskId);
        GetSpeechSynthesisTaskResult result
 =AMAZON_POLLY_CLIENT.getSpeechSynthesisTask(getSpeechSynthesisTaskRequest);
        return result.getSynthesisTask();
    }

    private static String getSynthesisTaskStatus(String taskId) {
        GetSpeechSynthesisTaskRequest getSpeechSynthesisTaskRequest = new
 GetSpeechSynthesisTaskRequest()
                .withTaskId(taskId);
        GetSpeechSynthesisTaskResult result
 =AMAZON_POLLY_CLIENT.getSpeechSynthesisTask(getSpeechSynthesisTaskRequest);
        return result.getSynthesisTask().getTaskStatus();
    }

}
```

Marcas de fala

A amostra de código a seguir mostra como usar aplicativos baseados em Java para sintetizar marcas de fala para texto digitado. Essa funcionalidade usa a API SynthesizeSpeech.

Para obter mais informações sobre essa funcionalidade, consulte Marcas de fala.

Para obter mais informações sobre a API, consulte a referência da API `SynthesizeSpeech`.

SDK v2

```
/*
   Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
   SPDX-License-Identifier: Apache-2.0
*/

package com.example.polly;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.*;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SpeechMarksSample {

    public static void main(String args[]) {

        PollyClient polly = PollyClient.builder()
                .region(Region.US_WEST_2)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        speechMarksSample(polly) ;
        polly.close();
    }
```

```
    private static final String OUTPUT_FILE = "./speechMarks.json";
    public static void speechMarksSample(PollyClient client) {

        try {
            SynthesizeSpeechRequest speechMarksSampleRequest =
 SynthesizeSpeechRequest.builder()
                    .outputFormat(OutputFormat.JSON)
                    .speechMarkTypes(SpeechMarkType.VISEME, SpeechMarkType.WORD)
                    .voiceId(VoiceId.JOANNA)
                    .text("This is a sample text to be synthesized")
                    .build();
            try (FileOutputStream outputStream = new FileOutputStream(new
 File(OUTPUT_FILE))) {
                ResponseInputStream<SynthesizeSpeechResponse> synthesizeSpeechResponse
 = client
                        .synthesizeSpeech(speechMarksSampleRequest);
                byte[] buffer = new byte[2 * 1024];
                int readBytes;

                try (InputStream in = synthesizeSpeechResponse){
                    while ((readBytes = in.read(buffer)) > 0) {
                        outputStream.write(buffer, 0, readBytes);
                    }
                }
            } catch (Exception e) {
                System.err.println("Exception caught: " + e);
            }

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
            System.exit(1);
        }
    }

}
```

SDK v1

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.OutputFormat;
```

```
import com.amazonaws.services.polly.model.SpeechMarkType;
import com.amazonaws.services.polly.model.SynthesizeSpeechRequest;
import com.amazonaws.services.polly.model.SynthesizeSpeechResult;
import com.amazonaws.services.polly.model.VoiceId;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;

public class SynthesizeSpeechMarksSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void synthesizeSpeechMarks() {
        String outputFileName = "/tmp/speechMarks.json";

        SynthesizeSpeechRequest synthesizeSpeechRequest = new SynthesizeSpeechRequest()
                .withOutputFormat(OutputFormat.Json)
                .withSpeechMarkTypes(SpeechMarkType.Viseme, SpeechMarkType.Word)
                .withVoiceId(VoiceId.Joanna)
                .withText("This is a sample text to be synthesized.");

        try (FileOutputStream outputStream = new FileOutputStream(new
 File(outputFileName))) {
            SynthesizeSpeechResult synthesizeSpeechResult =
 client.synthesizeSpeech(synthesizeSpeechRequest);
            byte[] buffer = new byte[2 * 1024];
            int readBytes;

            try (InputStream in = synthesizeSpeechResult.getAudioStream()){
                while ((readBytes = in.read(buffer)) > 0) {
                    outputStream.write(buffer, 0, readBytes);
                }
            }
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
}
```

SynthesizeSpeech

A amostra de código Java a seguir mostra como usar aplicativos baseados em Java para sintetizar fala com textos mais curtos para processamento em tempo quase real.

Para obter mais informações, consulte a referência da API SynthesizeSpeech.

```
package com.amazonaws.polly.samples;

import com.amazonaws.services.polly.AmazonPolly;
import com.amazonaws.services.polly.AmazonPollyClientBuilder;
import com.amazonaws.services.polly.model.OutputFormat;
import com.amazonaws.services.polly.model.SynthesizeSpeechRequest;
import com.amazonaws.services.polly.model.SynthesizeSpeechResult;
import com.amazonaws.services.polly.model.VoiceId;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;

public class SynthesizeSpeechSample {
    AmazonPolly client = AmazonPollyClientBuilder.defaultClient();

    public void synthesizeSpeech() {
        String outputFileName = "/tmp/speech.mp3";

        SynthesizeSpeechRequest synthesizeSpeechRequest = new SynthesizeSpeechRequest()
                .withOutputFormat(OutputFormat.Mp3)
                .withVoiceId(VoiceId.Joanna)
                .withText("This is a sample text to be synthesized.")
                .withEngine("neural");

        try (FileOutputStream outputStream = new FileOutputStream(new
 File(outputFileName))) {
            SynthesizeSpeechResult synthesizeSpeechResult =
 client.synthesizeSpeech(synthesizeSpeechRequest);
            byte[] buffer = new byte[2 * 1024];
            int readBytes;

            try (InputStream in = synthesizeSpeechResult.getAudioStream()){
                while ((readBytes = in.read(buffer)) > 0) {
                    outputStream.write(buffer, 0, readBytes);
                }
            }
        } catch (Exception e) {
            System.err.println("Exception caught: " + e);
        }
    }
```

```
}
```

Exemplos de Python

Os exemplos de código a seguir mostram como usar os aplicativos baseados em Python (boto3) para realizar várias tarefas com o Amazon Polly. Esses exemplos não se propõem a ser exemplos completos, mas podem ser incluídos em aplicativos Python maiores que usam o AWS SDK para Python (Boto).

Fragmentos de código

- DeleteLexicon
- GetLexicon
- ListLexicon
- PutLexicon
- StartSpeechSynthesisTask
- SynthesizeSpeech

DeleteLexicon

O exemplo de código Python a seguir usa o AWS SDK para Python (Boto) para excluir um léxico na região especificada na sua configuração local da AWS. O exemplo exclui somente o léxico especificado. Ele pede que você confirme se deseja continuar antes de realmente excluir o léxico.

O exemplo de código a seguir usa as credenciais padrão armazenadas no arquivo de configuração do AWS SDK. Para obter informações sobre como criar o arquivo de configuração, consulte Configurando o AWS CLI.

Para obter mais informações sobre esta operação, consulte a referência da API `DeleteLexicon`.

```
from argparse import ArgumentParser
from sys import version_info

from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError


# Define and parse the command line arguments
cli = ArgumentParser(description="DeleteLexicon example")
```

```
cli.add_argument("name", type=str, metavar="LEXICON_NAME")
arguments = cli.parse_args()

# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")

# Request confirmation
prompt = input if version_info >= (3, 0) else raw_input
proceed = prompt((u"This will delete the \"{0}\" lexicon,"
                  " do you want to proceed? [y,n]: ").format(arguments.name))

if proceed in ("y", "Y"):
    print(u"Deleting {0}...".format(arguments.name))

    try:
        # Request deletion of a lexicon by name
        response = polly.delete_lexicon(Name=arguments.name)
    except (BotoCoreError, ClientError) as error:
        # The service returned an error, exit gracefully
        cli.error(error)

    print("Done.")
else:
    print("Cancelled.")
```

GetLexicon

O código Python a seguir usa o AWS SDK para Python (Boto) para recuperar todos os léxicos armazenados em uma região da AWS. O exemplo aceita um nome de léxico como parâmetro de linha de comando e obtém somente esse léxico, imprimindo o caminho tmp onde foi salvo localmente.

O exemplo de código a seguir usa as credenciais padrão armazenadas no arquivo de configuração do AWS SDK. Para obter informações sobre como criar o arquivo de configuração, consulte Configurando o AWS CLI.

Para obter mais informações sobre esta operação, consulte a referência da API `GetLexicon`.

```
from argparse import ArgumentParser
from os import path
```

```
from tempfile import gettempdir

from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError

# Define and parse the command line arguments
cli = ArgumentParser(description="GetLexicon example")
cli.add_argument("name", type=str, metavar="LEXICON_NAME")
arguments = cli.parse_args()

# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")

print(u"Fetching {0}...".format(arguments.name))

try:
    # Fetch lexicon by name
    response = polly.get_lexicon(Name=arguments.name)
except (BotoCoreError, ClientError) as error:
    # The service returned an error, exit gracefully
    cli.error(error)

# Get the lexicon data from the response
lexicon = response.get("Lexicon", {})

# Access the lexicon's content
if "Content" in lexicon:
    output = path.join(gettempdir(), u"%s.pls" % arguments.name)
    print(u"Saving to %s..." % output)

    try:
        # Save the lexicon contents to a local file
        with open(output, "w") as pls_file:
            pls_file.write(lexicon["Content"])
    except IOError as error:
        # Could not write to file, exit gracefully
        cli.error(error)
else:
    # The response didn't contain lexicon data, exit gracefully
    cli.error("Could not fetch lexicons contents")
```

```
print("Done.")
```

ListLexicon

O exemplo de código Python a seguir usa o AWS SDK para Python (Boto) para listar os léxicos na sua conta na região especificada na configuração local da AWS. Para obter informações sobre como criar o arquivo de configuração, consulte Configurando o AWS CLI.

Para obter mais informações sobre esta operação, consulte a referência da API `ListLexicons`.

```
import sys

from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError

# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")

try:
    # Request the list of available lexicons
    response = polly.list_lexicons()
except (BotoCoreError, ClientError) as error:
    # The service returned an error, exit gracefully
    print(error)
    sys.exit(-1)

# Get the list of lexicons in the response
lexicons = response.get("Lexicons", [])
print("{0} lexicon(s) found".format(len(lexicons)))

# Output a formatted list of lexicons with some of the attributes
for lexicon in lexicons:
    print((u" - {Name} ({Attributes[LanguageCode]}), "
           "{Attributes[LexemesCount]} lexeme(s)").format(**lexicon))
```

PutLexicon

A amostra de código a seguir indica como usar aplicativos baseados em Python (boto3) para armazenar um léxico de pronúncia em uma região da AWS.

Para obter mais informações sobre esta operação, consulte a referência da API `PutLexicon`.

Observe o seguinte:

- Você precisa atualizar o código fornecendo um nome de arquivo de léxico local e um nome de léxico armazenado.
- O exemplo pressupõe que você tenha arquivos de léxico em um subdiretório chamado `pls`. Você precisa atualizar o caminho conforme apropriado.

O exemplo de código a seguir usa as credenciais padrão armazenadas no arquivo de configuração do AWS SDK. Para obter informações sobre como criar o arquivo de configuração, consulte Configurando o AWS CLI.

Para obter mais informações sobre esta operação, consulte a referência da API `PutLexicon`.

```
from argparse import ArgumentParser

from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError

# Define and parse the command line arguments
cli = ArgumentParser(description="PutLexicon example")
cli.add_argument("path", type=str, metavar="FILE_PATH")
cli.add_argument("-n", "--name", type=str, required=True,
                 metavar="LEXICON_NAME", dest="name")
arguments = cli.parse_args()

# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")

# Open the PLS lexicon file for reading
try:
    with open(arguments.path, "r") as lexicon_file:
        # Read the pls file contents
        lexicon_data = lexicon_file.read()

        # Store the PLS lexicon on the service.
        # If a lexicon with that name already exists,
        # its contents will be updated
```

```
        response = polly.put_lexicon(Name=arguments.name,
                                     Content=lexicon_data)
except (IOError, BotoCoreError, ClientError) as error:
    # Could not open/read the file or the service returned an error,
    # exit gracefully
    cli.error(error)

print(u"The \"{0}\" lexicon is now available for use.".format(arguments.name))
```

StartSpeechSynthesisTask

O exemplo de código Python a seguir usa o AWS SDK para Python (Boto) para listar os léxicos na sua conta na região especificada na configuração local da AWS. Para obter informações sobre como criar o arquivo de configuração, consulte Configurando o AWS CLI.

Para obter mais informações, consulte a referência da API `StartSpeechSynthesisTask`.

```
import boto3
import time

polly_client = boto3.Session(
                aws_access_key_id='',
    aws_secret_access_key='',
    region_name='eu-west-2').client('polly')

response = polly_client.start_speech_synthesis_task(VoiceId='Joanna',
                OutputS3BucketName='synth-books-buckets',
                OutputS3KeyPrefix='key',
                OutputFormat='mp3',
                Text='This is a sample text to be synthesized.',
                Engine='neural')

taskId = response['SynthesisTask']['TaskId']

print( "Task id is {} ".format(taskId))

task_status = polly_client.get_speech_synthesis_task(TaskId = taskId)

print(task_status)
```

SynthesizeSpeech

O exemplo de código Python a seguir usa a sintetização de fala do AWS SDK para Python (Boto) com textos mais curtos com processamento quase em tempo real. Para obter mais informações, consulte a referência da operação SynthesizeSpeech.

Este exemplo usa uma sequência curta de texto sem formatação. Você pode usar texto SSML para ter mais controle sobre a saída. Para obter mais informações, consulte Gerar fala com base em documentos SSML.

```
import boto3

polly_client = boto3.Session(
                aws_access_key_id=,
    aws_secret_access_key=,
    region_name='us-west-2').client('polly')

response = polly_client.synthesize_speech(VoiceId='Joanna',
                OutputFormat='mp3',
                Text = 'This is a sample text to be synthesized.',
                Engine = 'neural')

file = open('speech.mp3', 'wb')
file.write(response['AudioStream'].read())
file.close()
```

Exemplo de Java

Este exemplo mostra como usar o Amazon Polly para o fluxo de fala a partir de um aplicativo baseado em Java. O exemplo usa o AWS SDK para Java para ler o texto especificado com uma voz selecionada a partir de uma lista.

O código mostrado abrange as principais tarefas, mas faz apenas a verificação mínima de erros. Se o Amazon Polly encontrar um erro, o aplicativo será encerrado.

Para executar essa aplicação de exemplo, é necessário o seguinte:

- Java Development Kit (JDK) do Java 8
- AWS SDK para Java
- Apache Maven

Para testar a aplicação

1. Certifique-se de que a variável de ambiente JAVA_HOME está definida para o JDK.

   Por exemplo, se você instalou o JDK 1.8.0_121 no Windows em `C:\Program Files\Java\jdk1.8.0_121`, deve digitar o seguinte no prompt de comando:

   ```
   set JAVA_HOME=""C:\Program Files\Java\jdk1.8.0_121""
   ```

   Se você instalou o JDK 1.8.0_121 em Linux no `/usr/lib/jvm/java8-openjdk-amd64`, você deve digitar o seguinte no prompt de comando:

   ```
   export JAVA_HOME=/usr/lib/jvm/java8-openjdk-amd64
   ```

2. Defina as variáveis do ambiente Maven para executar o Maven a partir da linha de comando.

   Por exemplo, se você instalou Maven 3.3.9 no Windows em `C:\Program Files\apache-maven-3.3.9`, deve digitar o seguinte:

   ```
   set M2_HOME=""C:\Program Files\apache-maven-3.3.9""
   set M2=%M2_HOME%\bin
   set PATH=%M2%;%PATH%
   ```

   Se você instalou o Maven 3.3.9 no Linux no `/home/ec2-user/opt/apache-maven-3.3.9`, deve digitar o seguinte:

   ```
   export M2_HOME=/home/ec2-user/opt/apache-maven-3.3.9
   export M2=$M2_HOME/bin
   export PATH=$M2:$PATH
   ```

3. Crie um novo diretório chamado `polly-java-demo`.

4. No diretório `polly-java-demo`, crie um novo arquivo chamado `pom.xml` e cole o seguinte código dentro:

   ```
   <project xmlns="http://maven.apache.org/POM/4.0.0"
                        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/
   maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.amazonaws.polly</groupId>
   ```

```
 <artifactId>java-demo</artifactId>
 <version>0.0.1-SNAPSHOT</version>

 <dependencies>
  <!-- https://mvnrepository.com/artifact/com.amazonaws/aws-java-sdk-polly -->
  <dependency>
   <groupId>com.amazonaws</groupId>
   <artifactId>aws-java-sdk-polly</artifactId>
   <version>1.11.77</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/com.googlecode.soundlibs/jlayer -->
  <dependency>
   <groupId>com.googlecode.soundlibs</groupId>
   <artifactId>jlayer</artifactId>
   <version>1.0.1-1</version>
  </dependency>

 </dependencies>
 <build>
  <plugins>
   <plugin>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>exec-maven-plugin</artifactId>
    <version>1.2.1</version>
    <executions>
     <execution>
      <goals>
       <goal>java</goal>
      </goals>
     </execution>
    </executions>
    <configuration>
     <mainClass>com.amazonaws.demos.polly.PollyDemo</mainClass>
    </configuration>
   </plugin>
  </plugins>
 </build>
</project>
```

5. Crie um novo diretório chamado `polly` em `src/main/java/com/amazonaws/demos`.
6. No diretório `polly`, crie um novo arquivo de código-fonte Java chamado `PollyDemo.java` e cole o seguinte código:

```
package com.amazonaws.demos.polly;

import java.io.IOException;
import java.io.InputStream;

import com.amazonaws.ClientConfiguration;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.polly.AmazonPollyClient;
import com.amazonaws.services.polly.model.DescribeVoicesRequest;
import com.amazonaws.services.polly.model.DescribeVoicesResult;
import com.amazonaws.services.polly.model.OutputFormat;
import com.amazonaws.services.polly.model.SynthesizeSpeechRequest;
import com.amazonaws.services.polly.model.SynthesizeSpeechResult;
import com.amazonaws.services.polly.model.Voice;

import javazoom.jl.player.advanced.AdvancedPlayer;
import javazoom.jl.player.advanced.PlaybackEvent;
import javazoom.jl.player.advanced.PlaybackListener;

public class PollyDemo {

 private final AmazonPollyClient polly;
 private final Voice voice;
 private static final String SAMPLE = "Congratulations. You have successfully built
 this working demo
 of Amazon Polly in Java. Have fun building voice enabled apps with Amazon Polly
 (that's me!), and always
 look at the AWS website for tips and tricks on using Amazon Polly and other great
 services from AWS";

 public PollyDemo(Region region) {
  // create an Amazon Polly client in a specific region
  polly = new AmazonPollyClient(new DefaultAWSCredentialsProviderChain(),
  new ClientConfiguration());
  polly.setRegion(region);
  // Create describe voices request.
  DescribeVoicesRequest describeVoicesRequest = new DescribeVoicesRequest();

  // Synchronously ask Amazon Polly to describe available TTS voices.
  DescribeVoicesResult describeVoicesResult =
 polly.describeVoices(describeVoicesRequest);
```

```
  voice = describeVoicesResult.getVoices().get(0);
 }

 public InputStream synthesize(String text, OutputFormat format) throws IOException
 {
  SynthesizeSpeechRequest synthReq =
  new SynthesizeSpeechRequest().withText(text).withVoiceId(voice.getId())
    .withOutputFormat(format).withEngine("neural");
  SynthesizeSpeechResult synthRes = polly.synthesizeSpeech(synthReq);

  return synthRes.getAudioStream();
 }

 public static void main(String args[]) throws Exception {
  //create the test class
  PollyDemo helloWorld = new PollyDemo(Region.getRegion(Regions.US_EAST_1));
  //get the audio stream
  InputStream speechStream = helloWorld.synthesize(SAMPLE, OutputFormat.Mp3);

  //create an MP3 player
  AdvancedPlayer player = new AdvancedPlayer(speechStream,
    javazoom.jl.player.FactoryRegistry.systemRegistry().createAudioDevice());

  player.setPlayBackListener(new PlaybackListener() {
   @Override
   public void playbackStarted(PlaybackEvent evt) {
    System.out.println("Playback started");
    System.out.println(SAMPLE);
   }

   @Override
   public void playbackFinished(PlaybackEvent evt) {
    System.out.println("Playback finished");
   }
  });


  // play it!
  player.play();

 }
}
```

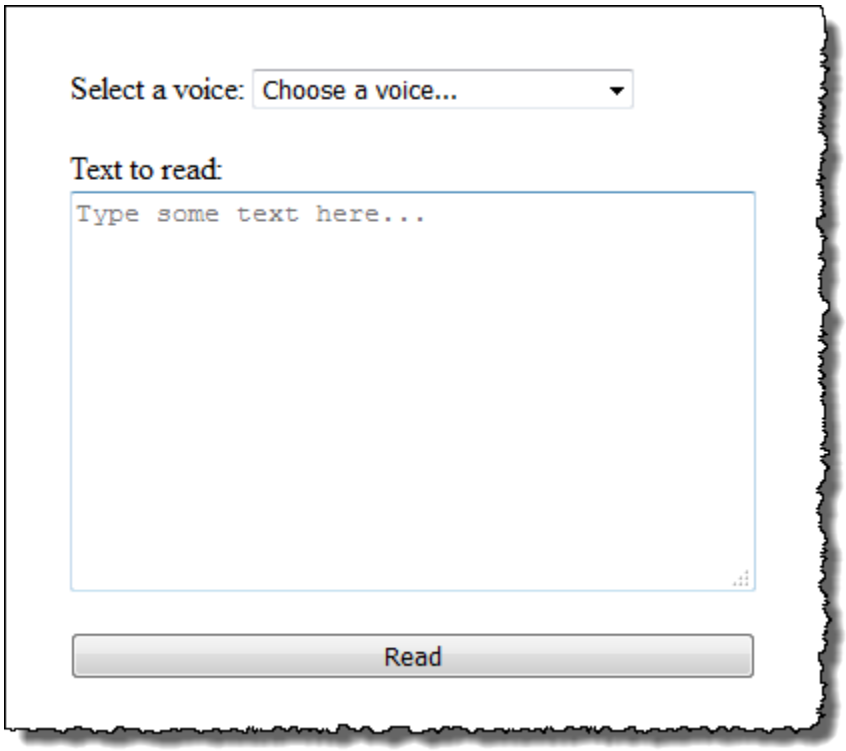
7. Volte para o diretório `polly-java-demo` para limpar, compilar e executar a demonstração:

```
mvn clean compile exec:java
```

Exemplo de Python (HTML5 cliente e servidor Python)

Esta aplicação de exemplo é formada pelo seguinte:

- Um servidor HTTP 1.1 usando a codificação de transferência HTTP (consulte a codificação de transferência em parte ("chunked") (consulte Codificação de transferência em parte)
- Uma interface de HTML5 usuário simples que interage com o servidor HTTP 1.1 (mostrada abaixo):



O objetivo deste exemplo é mostrar como usar o Amazon Polly para transmitir fala de um aplicativo baseado em navegador HTML5 . Consumir o fluxo de áudio produzido pelo Amazon Polly à medida que o texto é sintetizado é a abordagem recomendada para casos de uso em que a agilidade é um fator importante (por exemplo, sistemas de diálogo, leitores de tela, etc.).

Para executar essa aplicação de exemplo, é necessário o seguinte:

- Navegador da Web compatível com os padrões HTML5 e EcmaScript 5 (por exemplo, Chrome 23.0 ou superior, Firefox 21.0 ou superior, Internet Explorer 9.0 ou superior)
- Python versão superior a 3.0

Para testar a aplicação

1. Salve o código do servidor como `server.py`. Para o código, consulte Exemplo em Python: código de servidor Python (server.py).
2. Salve o código do HTML5 cliente como`index.html`. Para o código, consulte Exemplo em Python: interface de HTML5 usuário (index.html).
3. Execute o comando a seguir a partir do caminho onde você salvou o server.py para iniciar a aplicação (em alguns sistemas, você pode precisar usar `python3` em vez de `python` ao executar o comando).

   ```
   $ python  server.py
   ```

   Depois que a aplicação é iniciada, é exibido um URL no terminal.
4. Abra o URL mostrado no terminal em um navegador da web.

   Você pode passar o endereço e a porta para o servidor de aplicações usar como parâmetro para `server.py`. Para obter mais informações, execute `python server.py -h`.
5. Para ouvir a fala, selecione uma voz na lista, digite um texto e, em seguida, escolha Read. A fala começa a ser reproduzida assim que o Amazon Polly transfere o primeiro bloco utilizável de dados de áudio.
6. Para interromper o servidor Python quando terminar de testar o aplicativo, pressione Ctrl+C no terminal onde o servidor está em execução.

> **Note**
>
> O servidor cria um cliente Boto3 usando o AWS SDK para Python (Boto). O cliente usa as credenciais armazenadas no arquivo de AWS configuração do seu computador para assinar e autenticar as solicitações no Amazon Polly. Para obter mais informações sobre como criar o arquivo de AWS configuração e armazenar credenciais, consulte Configurando o AWS Command Line Interface no Guia doAWS Command Line Interface Usuário.

Exemplo em Python: interface de HTML5 usuário (index.html)

Esta seção fornece o código para o HTML5 cliente descrito emExemplo de Python (HTML5 cliente e servidor Python).

```
<html>

<head>
    <title>Text-to-Speech Example Application</title>
    <script>
        /*
         * This sample code requires a web browser with support for both the
         * HTML5 and ECMAScript 5 standards; the following is a non-comprehensive
         * list of compliant browsers and their minimum version:
         *
         * - Chrome 23.0+
         * - Firefox 21.0+
         * - Internet Explorer 9.0+
         * - Edge 12.0+
         * - Opera 15.0+
         * - Safari 6.1+
         * - Android (stock web browser) 4.4+
         * - Chrome for Android 51.0+
         * - Firefox for Android 48.0+
         * - Opera Mobile 37.0+
         * - iOS (Safari Mobile and Chrome) 3.2+
         * - Internet Explorer Mobile 10.0+
         * - Blackberry Browser 10.0+
         */

        // Mapping of the OutputFormat parameter of the SynthesizeSpeech API
        // and the audio format strings understood by the browser
        var AUDIO_FORMATS = {
            'ogg_vorbis': 'audio/ogg',
            'mp3': 'audio/mpeg',
            'pcm': 'audio/wave; codecs=1',
            'mulaw': 'audio/mulaw',
            'alaw': 'audio/alaw'
        };

        /**
         * Handles fetching JSON over HTTP
         */
```

```
        function fetchJSON(method, url, onSuccess, onError) {
            var request = new XMLHttpRequest();
            request.open(method, url, true);
            request.onload = function () {
                // If loading is complete
                if (request.readyState === 4) {
                    // if the request was successful
                    if (request.status === 200) {
                        var data;

                        // Parse the JSON in the response
                        try {
                            data = JSON.parse(request.responseText);
                        } catch (error) {
                            onError(request.status, error.toString());
                        }

                        onSuccess(data);
                    } else {
                        onError(request.status, request.responseText)
                    }
                }
            };

            request.send();
        }

        /**
         * Returns a list of audio formats supported by the browser
         */
        function getSupportedAudioFormats(player) {
            return Object.keys(AUDIO_FORMATS)
                .filter(function (format) {
                    var supported = player.canPlayType(AUDIO_FORMATS[format]);
                    return supported === 'probably' || supported === 'maybe';
                });
        }

        // Initialize the application when the DOM is loaded and ready to be
        // manipulated
        document.addEventListener("DOMContentLoaded", function () {
            var input = document.getElementById('input'),
                voiceMenu = document.getElementById('voice'),
                text = document.getElementById('text'),
```

```
        player = document.getElementById('player'),
        submit = document.getElementById('submit'),
        supportedFormats = getSupportedAudioFormats(player);

    // Display a message and don't allow submitting the form if the
    // browser doesn't support any of the available audio formats
    if (supportedFormats.length === 0) {
        submit.disabled = true;
        alert('The web browser in use does not support any of the' +
              ' available audio formats. Please try with a different' +
              ' one.');
    }

    // Play the audio stream when the form is submitted successfully
    input.addEventListener('submit', function (event) {
        // Validate the fields in the form, display a message if
        // unexpected values are encountered
        if (voiceMenu.selectedIndex <= 0 || text.value.length === 0) {
            alert('Please fill in all the fields.');
        } else {
            var selectedVoice = voiceMenu
                                    .options[voiceMenu.selectedIndex]
                                    .value;

            // Point the player to the streaming server
            player.src = '/read?voiceId=' +
                encodeURIComponent(selectedVoice) +
                '&text=' + encodeURIComponent(text.value) +
                '&outputFormat=' + supportedFormats[0];
            player.play();
        }

        // Stop the form from submitting,
        // Submitting the form is allowed only if the browser doesn't
        // support Javascript to ensure functionality in such a case
        event.preventDefault();
    });

    // Load the list of available voices and display them in a menu
    fetchJSON('GET', '/voices',
        // If the request succeeds
        function (voices) {
            var container = document.createDocumentFragment();
```

```
                // Build the list of options for the menu
                voices.forEach(function (voice) {
                    var option = document.createElement('option');
                    option.value = voice['Id'];
                    option.innerHTML = voice['Name'] + ' (' +
                        voice['Gender'] + ', ' +
                        voice['LanguageName'] + ')';
                    container.appendChild(option);
                });

                // Add the options to the menu and enable the form field
                voiceMenu.appendChild(container);
                voiceMenu.disabled = false;
            },
            // If the request fails
            function (status, response) {
                // Display a message in case loading data from the server
                // fails
                alert(status + ' - ' + response);
            });
    });

</script>
<style>
    #input {
        min-width: 100px;
        max-width: 600px;
        margin: 0 auto;
        padding: 50px;
    }

    #input div {
        margin-bottom: 20px;
    }

    #text {
        width: 100%;
        height: 200px;
        display: block;
    }

    #submit {
        width: 100%;
    }
```

```
    </style>
</head>

<body>
    <form id="input" method="GET" action="/read">
        <div>
            <label for="voice">Select a voice:</label>
            <select id="voice" name="voiceId" disabled>
                <option value="">Choose a voice...</option>
            </select>
        </div>
        <div>
            <label for="text">Text to read:</label>
            <textarea id="text" maxlength="1000" minlength="1" name="text"
                    placeholder="Type some text here..."></textarea>
        </div>
        <input type="submit" value="Read" id="submit" />
    </form>
    <audio id="player"></audio>
</body>

</html>
```

Exemplo em Python: código de servidor Python (server.py)

Esta seção fornece o código para o servidor Python descrito em Exemplo de Python (HTML5 cliente e servidor Python).

```
"""
Example Python 2.7+/3.3+ Application

This application consists of a HTTP 1.1 server using the HTTP chunked transfer
coding (https://tools.ietf.org/html/rfc2616#section-3.6.1) and a minimal HTML5
user interface that interacts with it.

The goal of this example is to start streaming the speech to the client (the
HTML5 web UI) as soon as the first consumable chunk of speech is returned in
order to start playing the audio as soon as possible.
For use cases where low latency and responsiveness are strong requirements,
this is the recommended approach.

The service documentation contains examples for non-streaming use cases where
waiting for the speech synthesis to complete and fetching the whole audio stream
```

```
at once are an option.

To test the application, run 'python server.py' and then open the URL
displayed in the terminal in a web browser (see index.html for a list of
supported browsers). The address and port for the server can be passed as
parameters to server.py. For more information, run: 'python server.py -h'
"""
from argparse import ArgumentParser
from collections import namedtuple
from contextlib import closing
from io import BytesIO
from json import dumps as json_encode
import os
import sys

if sys.version_info >= (3, 0):
    from http.server import BaseHTTPRequestHandler, HTTPServer
    from socketserver import ThreadingMixIn
    from urllib.parse import parse_qs
else:
    from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer
    from SocketServer import ThreadingMixIn
    from urlparse import parse_qs

from boto3 import Session
from botocore.exceptions import BotoCoreError, ClientError

ResponseStatus = namedtuple("HTTPStatus",
                            ["code", "message"])

ResponseData = namedtuple("ResponseData",
                          ["status", "content_type", "data_stream"])

# Mapping the output format used in the client to the content type for the
# response
AUDIO_FORMATS = {"ogg_vorbis": "audio/ogg",
                 "mp3": "audio/mpeg",
                 "pcm": "audio/wave; codecs=1",
                 "mulaw": "audio/mulaw",
                 "alaw": "audio/alaw"}
CHUNK_SIZE = 1024
HTTP_STATUS = {"OK": ResponseStatus(code=200, message="OK"),
               "BAD_REQUEST": ResponseStatus(code=400, message="Bad request"),
               "NOT_FOUND": ResponseStatus(code=404, message="Not found"),
```

```
                "INTERNAL_SERVER_ERROR": ResponseStatus(code=500, message="Internal
 server error")}
PROTOCOL = "http"
ROUTE_INDEX = "/index.html"
ROUTE_VOICES = "/voices"
ROUTE_READ = "/read"


# Create a client using the credentials and region defined in the adminuser
# section of the AWS credentials and configuration files
session = Session(profile_name="adminuser")
polly = session.client("polly")


class HTTPStatusError(Exception):
    """Exception wrapping a value from http.server.HTTPStatus"""

    def __init__(self, status, description=None):
        """
        Constructs an error instance from a tuple of
        (code, message, description), see http.server.HTTPStatus
        """
        super(HTTPStatusError, self).__init__()
        self.code = status.code
        self.message = status.message
        self.explain = description


class ThreadedHTTPServer(ThreadingMixIn, HTTPServer):
    """An HTTP Server that handle each request in a new thread"""
    daemon_threads = True


class ChunkedHTTPRequestHandler(BaseHTTPRequestHandler):
    """"HTTP 1.1 Chunked encoding request handler"""
    # Use HTTP 1.1 as 1.0 doesn't support chunked encoding
    protocol_version = "HTTP/1.1"

    def query_get(self, queryData, key, default=""):
        """Helper for getting values from a pre-parsed query string"""
        return queryData.get(key, [default])[0]

    def do_GET(self):
        """Handles GET requests"""
```

```
        # Extract values from the query string
        path, _, query_string = self.path.partition('?')
        query = parse_qs(query_string)

        response = None

        print(u"[START]: Received GET for %s with query: %s" % (path, query))

        try:
            # Handle the possible request paths
            if path == ROUTE_INDEX:
                response = self.route_index(path, query)
            elif path == ROUTE_VOICES:
                response = self.route_voices(path, query)
            elif path == ROUTE_READ:
                response = self.route_read(path, query)
            else:
                response = self.route_not_found(path, query)

            self.send_headers(response.status, response.content_type)
            self.stream_data(response.data_stream)

        except HTTPStatusError as err:
            # Respond with an error and log debug
            # information
            if sys.version_info >= (3, 0):
                self.send_error(err.code, err.message, err.explain)
            else:
                self.send_error(err.code, err.message)

            self.log_error(u"%s %s %s - [%d] %s", self.client_address[0],
                           self.command, self.path, err.code, err.explain)

        print("[END]")

    def route_not_found(self, path, query):
        """Handles routing for unexpected paths"""
        raise HTTPStatusError(HTTP_STATUS["NOT_FOUND"], "Page not found")

    def route_index(self, path, query):
        """Handles routing for the application's entry point'"""
        try:
            return ResponseData(status=HTTP_STATUS["OK"], content_type="text_html",
```

```
                            # Open a binary stream for reading the index
                            # HTML file
                            data_stream=open(os.path.join(sys.path[0],
                                                          path[1:]), "rb"))
        except IOError as err:
            # Couldn't open the stream
            raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                                  str(err))

    def route_voices(self, path, query):
        """Handles routing for listing available voices"""
        params = {}
        voices = []

        while True:
            try:
                # Request list of available voices, if a continuation token
                # was returned by the previous call then use it to continue
                # listing
                response = polly.describe_voices(**params)
            except (BotoCoreError, ClientError) as err:
                # The service returned an error
                raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                                      str(err))

            # Collect all the voices
            voices.extend(response.get("Voices", []))

            # If a continuation token was returned continue, stop iterating
            # otherwise
            if "NextToken" in response:
                params = {"NextToken": response["NextToken"]}
            else:
                break

        json_data = json_encode(voices)
        bytes_data = bytes(json_data, "utf-8") if sys.version_info >= (3, 0) \
            else bytes(json_data)

        return ResponseData(status=HTTP_STATUS["OK"],
                            content_type="application/json",
                            # Create a binary stream for the JSON data
                            data_stream=BytesIO(bytes_data))
```

```
    def route_read(self, path, query):
        """Handles routing for reading text (speech synthesis)"""
        # Get the parameters from the query string
        text = self.query_get(query, "text")
        voiceId = self.query_get(query, "voiceId")
        outputFormat = self.query_get(query, "outputFormat")

        # Validate the parameters, set error flag in case of unexpected
        # values
        if len(text) == 0 or len(voiceId) == 0 or \
                outputFormat not in AUDIO_FORMATS:
            raise HTTPStatusError(HTTP_STATUS["BAD_REQUEST"],
                                  "Wrong parameters")
        else:
            try:
                # Request speech synthesis
                response = polly.synthesize_speech(Text=text,
                                                   VoiceId=voiceId,
                                                   OutputFormat=outputFormat,
                                                   Engine="neural")
            except (BotoCoreError, ClientError) as err:
                # The service returned an error
                raise HTTPStatusError(HTTP_STATUS["INTERNAL_SERVER_ERROR"],
                                      str(err))

            return ResponseData(status=HTTP_STATUS["OK"],
                                content_type=AUDIO_FORMATS[outputFormat],
                                # Access the audio stream in the response
                                data_stream=response.get("AudioStream"))

    def send_headers(self, status, content_type):
        """Send out the group of headers for a successful request"""
        # Send HTTP headers
        self.send_response(status.code, status.message)
        self.send_header('Content-type', content_type)
        self.send_header('Transfer-Encoding', 'chunked')
        self.send_header('Connection', 'close')
        self.end_headers()

    def stream_data(self, stream):
        """Consumes a stream in chunks to produce the response's output'"""
        print("Streaming started...")

        if stream:
```

```
            # Note: Closing the stream is important as the service throttles on
            # the number of parallel connections. Here we are using
            # contextlib.closing to ensure the close method of the stream object
            # will be called automatically at the end of the with statement's
            # scope.
            with closing(stream) as managed_stream:
                # Push out the stream's content in chunks
                while True:
                    data = managed_stream.read(CHUNK_SIZE)
                    self.wfile.write(b"%X\r\n%s\r\n" % (len(data), data))

                    # If there's no more data to read, stop streaming
                    if not data:
                        break

                # Ensure any buffered output has been transmitted and close the
                # stream
                self.wfile.flush()

            print("Streaming completed.")
        else:
            # The stream passed in is empty
            self.wfile.write(b"0\r\n\r\n")
            print("Nothing to stream.")

# Define and parse the command line arguments
cli = ArgumentParser(description='Example Python Application')
cli.add_argument(
    "-p", "--port", type=int, metavar="PORT", dest="port", default=8000)
cli.add_argument(
    "--host", type=str, metavar="HOST", dest="host", default="localhost")
arguments = cli.parse_args()

# If the module is invoked directly, initialize the application
if __name__ == '__main__':
    # Create and configure the HTTP server instance
    server = ThreadedHTTPServer((arguments.host, arguments.port),
                                ChunkedHTTPRequestHandler)
    print("Starting server, use <Ctrl-C> to stop...")
    print(u"Open {0}://{1}:{2}{3} in a web browser.".format(PROTOCOL,
                                                           arguments.host,
                                                           arguments.port,
                                                           ROUTE_INDEX))
```

```
    try:
        # Listen for requests indefinitely
        server.serve_forever()
    except KeyboardInterrupt:
        # A request to terminate has been received, stop the server
        print("\nShutting down...")
        server.socket.close()
```

Exemplo para iOS

O exemplo a seguir usa o SDK do iOS para o Amazon Polly ler o texto especificado usando uma voz selecionada de uma lista.

O código exibido aqui abrange as principais tarefas, mas não lida com erros. Para ver o código completo, consulte a Demonstração do Amazon Polly no AWS Mobile SDK for iOS.

Inicializar

```
// Region of Amazon Polly.
let AwsRegion = AWSRegionType.usEast1

// Cognito pool ID. Pool needs to be unauthenticated pool with
// Amazon Polly permissions.
let CognitoIdentityPoolId = "YourCognitoIdentityPoolId"

// Initialize the Amazon Cognito credentials provider.
let credentialProvider = AWSCognitoCredentialsProvider(regionType: AwsRegion,
 identityPoolId: CognitoIdentityPoolId)

// Create an audio player
var audioPlayer = AVPlayer()
```

Obtenha a lista de vozes disponíveis

```
// Use the configuration as default
AWSServiceManager.default().defaultServiceConfiguration = configuration

// Get all the voices (no parameters specified in input) from Amazon Polly
// This creates an async task.
let task = AWSPolly.default().describeVoices(AWSPollyDescribeVoicesInput())
```

```
// When the request is done, asynchronously do the following block
// (we ignore all the errors, but in a real-world scenario they need
// to be handled)
task.continue(successBlock: { (awsTask: AWSTask) -> Any? in
      // awsTask.result is an instance of AWSPollyDescribeVoicesOutput in
      // case of the "describeVoices" method
      let voices = (awsTask.result! as AWSPollyDescribeVoicesOutput).voices

      return nil
})
```

Synthesize Speech

```
// First, Amazon Polly requires an input, which we need to prepare.
// Again, we ignore the errors, however this should be handled in
// real applications. Here we are using the URL Builder Request,
// since in order to make the synthesis quicker we will pass the
// presigned URL to the system audio player.
let input = AWSPollySynthesizeSpeechURLBuilderRequest()

// Text to synthesize
input.text = "Sample text"

// We expect the output in MP3 format
input.outputFormat = AWSPollyOutputFormat.mp3

// Choose the voice ID
input.voiceId = AWSPollyVoiceId.joanna

// Create an task to synthesize speech using the given synthesis input
let builder = AWSPollySynthesizeSpeechURLBuilder.default().getPreSignedURL(input)

// Request the URL for synthesis result
builder.continueOnSuccessWith(block: { (awsTask: AWSTask<NSURL>) -> Any? in
 // The result of getPresignedURL task is NSURL.
 // Again, we ignore the errors in the example.
 let url = awsTask.result!

 // Try playing the data using the system AVAudioPlayer
 self.audioPlayer.replaceCurrentItem(with: AVPlayerItem(url: url as URL))
 self.audioPlayer.play()
```

```
 return nil
})
```

Exemplo para Android

O exemplo a seguir usa o SDK do Android para o Amazon Polly ler o texto especificado usando uma voz selecionada de uma lista.

O código exibido aqui abrange as principais tarefas, mas não lida com erros. Para ver o código completo, consulte a Demonstração do Amazon Polly no AWS Mobile SDK para Android.

Inicializar

```
// Cognito pool ID. Pool needs to be unauthenticated pool with
// Amazon Polly permissions.
String COGNITO_POOL_ID = "YourCognitoIdentityPoolId";

// Region of Amazon Polly.
Regions MY_REGION = Regions.US_EAST_1;

// Initialize the Amazon Cognito credentials provider.
CognitoCachingCredentialsProvider credentialsProvider = new
 CognitoCachingCredentialsProvider(
        getApplicationContext(),
        COGNITO_POOL_ID,
        MY_REGION
);

// Create a client that supports generation of presigned URLs.
AmazonPollyPresigningClient client = new
 AmazonPollyPresigningClient(credentialsProvider);
```

Obtenha a lista de vozes disponíveis

```
// Create describe voices request.
DescribeVoicesRequest describeVoicesRequest = new DescribeVoicesRequest();

// Synchronously ask Amazon Polly to describe available TTS voices.
DescribeVoicesResult describeVoicesResult =
 client.describeVoices(describeVoicesRequest);
```

```
List<Voice> voices = describeVoicesResult.getVoices();
```

Obtenha o URL para fluxo de áudio

```
// Create speech synthesis request.
SynthesizeSpeechPresignRequest synthesizeSpeechPresignRequest =
        new SynthesizeSpeechPresignRequest()
        // Set the text to synthesize.
        .withText("Hello world!")
        // Select voice for synthesis.
        .withVoiceId(voices.get(0).getId()) // "Joanna"
        // Set format to MP3.
        .withOutputFormat(OutputFormat.Mp3);

// Get the presigned URL for synthesized speech audio stream.
URL presignedSynthesizeSpeechUrl =
        client.getPresignedSynthesizeSpeechUrl(synthesizeSpeechPresignRequest);
```

Reproduzir fala sintetizada

```
// Use MediaPlayer: https://developer.android.com/guide/topics/media/mediaplayer.html

// Create a media player to play the synthesized audio stream.
MediaPlayer mediaPlayer = new MediaPlayer();
mediaPlayer.setAudioStreamType(AudioManager.STREAM_MUSIC);

try {
    // Set media player's data source to previously obtained URL.
    mediaPlayer.setDataSource(presignedSynthesizeSpeechUrl.toString());
} catch (IOException e) {
    Log.e(TAG, "Unable to set data source for the media player! " + e.getMessage());
}

// Prepare the MediaPlayer asynchronously (since the data source is a network stream).
mediaPlayer.prepareAsync();

// Set the callback to start the MediaPlayer when it's prepared.
mediaPlayer.setOnPreparedListener(new MediaPlayer.OnPreparedListener() {
    @Override
    public void onPrepared(MediaPlayer mp) {
        mp.start();
```

```
    }
});

// Set the callback to release the MediaPlayer after playback is completed.
mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
    @Override
    public void onCompletion(MediaPlayer mp) {
 mp.release();
    }
});
```

Exemplos de código para Amazon Polly usando AWS SDKs

Os exemplos de código a seguir mostram como usar o Amazon Polly com um kit de desenvolvimento AWS de software (SDK).

Ações são trechos de código de programas maiores e devem ser executadas em contexto. Embora as ações mostrem como chamar perfis de serviço individuais, você pode ver as ações no contexto em seus cenários relacionados.

Cenários são exemplos de código que mostram como realizar tarefas específicas chamando várias funções dentro de um serviço ou combinadas com outros Serviços da AWS.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Exemplos de código

- Exemplos básicos para o uso do Amazon Polly AWS SDKs
  - Ações para o Amazon Polly usando AWS SDKs
    - Use DeleteLexiconcom um AWS SDK ou CLI
    - Use DescribeVoicescom um AWS SDK
    - Use GetLexiconcom um AWS SDK ou CLI
    - Use GetSpeechSynthesisTaskcom um AWS SDK ou CLI
    - Use ListLexiconscom um AWS SDK ou CLI
    - Use ListSpeechSynthesisTaskscom um AWS SDK ou CLI
    - Use PutLexiconcom um AWS SDK ou CLI
    - Use StartSpeechSynthesisTaskcom um AWS SDK ou CLI
    - Use SynthesizeSpeechcom um AWS SDK
- Cenários para o uso do Amazon Polly AWS SDKs
  - Converta texto em fala e volte para texto usando um AWS SDK
  - Crie um aplicativo de sincronização labial com o Amazon Polly usando um AWS SDK
  - Criar uma aplicação que analise o feedback dos clientes e sintetize o áudio
  - Introdução à síntese de conversão de texto em voz

Exemplos básicos para o uso do Amazon Polly AWS SDKs

Os exemplos de código a seguir mostram como usar os conceitos básicos do Amazon Polly com AWS SDKs.

Exemplos

- Ações para o Amazon Polly usando AWS SDKs
  - Use DeleteLexiconcom um AWS SDK ou CLI
  - Use DescribeVoicescom um AWS SDK
  - Use GetLexiconcom um AWS SDK ou CLI
  - Use GetSpeechSynthesisTaskcom um AWS SDK ou CLI
  - Use ListLexiconscom um AWS SDK ou CLI
  - Use ListSpeechSynthesisTaskscom um AWS SDK ou CLI
  - Use PutLexiconcom um AWS SDK ou CLI
  - Use StartSpeechSynthesisTaskcom um AWS SDK ou CLI
  - Use SynthesizeSpeechcom um AWS SDK

Ações para o Amazon Polly usando AWS SDKs

Os exemplos de código a seguir demonstram como realizar ações individuais do Amazon Polly com AWS SDKs. Cada exemplo inclui um link para GitHub, onde você pode encontrar instruções para configurar e executar o código.

Esses trechos chamam a API do Amazon Polly e são trechos de código de programas maiores que devem ser executados no contexto. É possível ver as ações em contexto em Cenários para o uso do Amazon Polly AWS SDKs.

Os exemplos a seguir incluem apenas as ações mais utilizadas. Consulte uma lista completa na Referência de API do Amazon Polly.

Exemplos

- Use DeleteLexiconcom um AWS SDK ou CLI
- Use DescribeVoicescom um AWS SDK
- Use GetLexiconcom um AWS SDK ou CLI
- Use GetSpeechSynthesisTaskcom um AWS SDK ou CLI

- Use ListLexiconscom um AWS SDK ou CLI
- Use ListSpeechSynthesisTaskscom um AWS SDK ou CLI
- Use PutLexiconcom um AWS SDK ou CLI
- Use StartSpeechSynthesisTaskcom um AWS SDK ou CLI
- Use SynthesizeSpeechcom um AWS SDK

Use **`DeleteLexicon`**com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `DeleteLexicon`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Introdução à síntese de conversão de texto em voz

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
using System;
using System.Threading.Tasks;
using Amazon.Polly;
using Amazon.Polly.Model;

/// <summary>
/// Deletes an existing Amazon Polly lexicon using the AWS SDK for .NET.
/// </summary>
public class DeleteLexicon
{
    public static async Task Main()
    {
        string lexiconName = "SampleLexicon";
```

```
            var client = new AmazonPollyClient();

            var success = await DeletePollyLexiconAsync(client, lexiconName);

            if (success)
            {
                Console.WriteLine($"Successfully deleted {lexiconName}.");
            }
            else
            {
                Console.WriteLine($"Could not delete {lexiconName}.");
            }
        }

        /// <summary>
        /// Deletes the named Amazon Polly lexicon.
        /// </summary>
        /// <param name="client">The initialized Amazon Polly client object.</
param>
        /// <param name="lexiconName">The name of the Amazon Polly lexicon to
        /// delete.</param>
        /// <returns>A Boolean value indicating the success of the operation.</
returns>
        public static async Task<bool> DeletePollyLexiconAsync(
            AmazonPollyClient client,
            string lexiconName)
        {
            var deleteLexiconRequest = new DeleteLexiconRequest()
            {
                Name = lexiconName,
            };

            var response = await client.DeleteLexiconAsync(deleteLexiconRequest);

            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
    }
```

- Para obter detalhes da API, consulte DeleteLexicona Referência AWS SDK para .NET da API.

CLI

AWS CLI

Como excluir um léxico

O exemplo de `delete-lexicon` a seguir exclui o léxico especificado.

```
aws polly delete-lexicon \
    --name w3c
```

Este comando não produz saída.

Para obter mais informações, consulte Como usar a DeleteLexicon operação no Amazon Polly Developer Guide.

- Para obter detalhes da API, consulte DeleteLexiconna Referência de AWS CLI Comandos.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
TRY.
    lo_ply->deletelexicon( iv_name ).
    MESSAGE 'Lexicon deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Para obter detalhes da API, consulte a DeleteLexiconreferência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`DescribeVoices`**com um AWS SDK

Os exemplos de código a seguir mostram como usar o `DescribeVoices`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Introdução à síntese de conversão de texto em voz

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
using System;
using System.Threading.Tasks;
using Amazon.Polly;
using Amazon.Polly.Model;

public class DescribeVoices
{
    public static async Task Main()
    {
        var client = new AmazonPollyClient();

        var allVoicesRequest = new DescribeVoicesRequest();
        var enUsVoicesRequest = new DescribeVoicesRequest()
        {
            LanguageCode = "en-US",
        };

        try
```

```
            {
                string nextToken;
                do
                {
                    var allVoicesResponse = await
client.DescribeVoicesAsync(allVoicesRequest);
                    nextToken = allVoicesResponse.NextToken;
                    allVoicesRequest.NextToken = nextToken;

                    Console.WriteLine("\nAll voices: ");
                    allVoicesResponse.Voices.ForEach(voice =>
                    {
                        DisplayVoiceInfo(voice);
                    });
                }
                while (nextToken is not null);

                do
                {
                    var enUsVoicesResponse = await
client.DescribeVoicesAsync(enUsVoicesRequest);
                    nextToken = enUsVoicesResponse.NextToken;
                    enUsVoicesRequest.NextToken = nextToken;

                    Console.WriteLine("\nen-US voices: ");
                    enUsVoicesResponse.Voices.ForEach(voice =>
                    {
                        DisplayVoiceInfo(voice);
                    });
                }
                while (nextToken is not null);
            }
            catch (Exception ex)
            {
                Console.WriteLine("Exception caught: " + ex.Message);
            }
        }

        public static void DisplayVoiceInfo(Voice voice)
        {
            Console.WriteLine($" Name: {voice.Name}\tGender:
{voice.Gender}\tLanguageName: {voice.LanguageName}");
        }
    }
```

- Para obter detalhes da API, consulte DescribeVoicesa Referência AWS SDK para .NET da API.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.DescribeVoicesRequest;
import software.amazon.awssdk.services.polly.model.DescribeVoicesResponse;
import software.amazon.awssdk.services.polly.model.PollyException;
import software.amazon.awssdk.services.polly.model.Voice;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeVoicesSample {
    public static void main(String args[]) {
        PollyClient polly = PollyClient.builder()
                .region(Region.US_WEST_2)
                .build();

        describeVoice(polly);
        polly.close();
```

```
    }

    public static void describeVoice(PollyClient polly) {
        try {
            DescribeVoicesRequest voicesRequest = DescribeVoicesRequest.builder()
                    .languageCode("en-US")
                    .build();

            DescribeVoicesResponse enUsVoicesResult =
 polly.describeVoices(voicesRequest);
            List<Voice> voices = enUsVoicesResult.voices();
            for (Voice myVoice : voices) {
                System.out.println("The ID of the voice is " + myVoice.id());
                System.out.println("The gender of the voice is " +
 myVoice.gender());
            }

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte DescribeVoicesa Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class PollyWrapper:
    """Encapsulates Amazon Polly functions."""
```

```
    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None


    def describe_voices(self):
        """
        Gets metadata about available voices.

        :return: The list of voice metadata.
        """
        try:
            response = self.polly_client.describe_voices()
            self.voice_metadata = response["Voices"]
            logger.info("Got metadata about %s voices.",
len(self.voice_metadata))
        except ClientError:
            logger.exception("Couldn't get voice metadata.")
            raise
        else:
            return self.voice_metadata
```

- Para obter detalhes da API, consulte a DescribeVoicesReferência da API AWS SDK for Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
require 'aws-sdk-polly' # In v2: require 'aws-sdk'

begin
  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/
config
  polly = Aws::Polly::Client.new

  # Get US English voices
  resp = polly.describe_voices(language_code: 'en-US')

  resp.voices.each do |v|
    puts v.name
    puts "  #{v.gender}"
    puts
  end
rescue StandardError => e
  puts 'Could not get voices'
  puts 'Error message:'
  puts e.message
end
```

- Para obter detalhes da API, consulte DescribeVoicesa Referência AWS SDK para Ruby da API.

Rust

SDK para Rust

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
async fn list_voices(client: &Client) -> Result<(), Error> {
    let resp = client.describe_voices().send().await?;
```

```
    println!("Voices:");

    let voices = resp.voices();
    for voice in voices {
        println!("  Name:     {}", voice.name().unwrap_or("No name!"));
        println!(
            "  Language: {}",
            voice.language_name().unwrap_or("No language!")
        );

        println!();
    }

    println!("Found {} voices", voices.len());

    Ok(())
}
```

- Para obter detalhes da API, consulte a DescribeVoicesreferência da API AWS SDK for Rust.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
    TRY.
        " Only pass optional parameters if they have values
        IF iv_engine IS NOT INITIAL AND iv_language IS NOT INITIAL.
          oo_result = lo_ply->describevoices(
            iv_engine = iv_engine
            iv_languagecode = iv_language ).
        ELSEIF iv_engine IS NOT INITIAL.
          oo_result = lo_ply->describevoices(
            iv_engine = iv_engine ).
        ELSEIF iv_language IS NOT INITIAL.
          oo_result = lo_ply->describevoices(
```

```
            iv_languagecode = iv_language ).
        ELSE.
          oo_result = lo_ply->describevoices( ).
        ENDIF.
        MESSAGE 'Retrieved voice metadata.' TYPE 'I'.
      CATCH /aws1/cx_plyinvalidnexttokenex.
        MESSAGE 'The NextToken is invalid.' TYPE 'E'.
      CATCH /aws1/cx_plyservicefailureex.
        MESSAGE 'Service failure occurred.' TYPE 'E'.
    ENDTRY.
```

- Para obter detalhes da API, consulte a DescribeVoicesreferência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`GetLexicon`**com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `GetLexicon`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Introdução à síntese de conversão de texto em voz

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
using System;
```

```
    using System.Threading.Tasks;
    using Amazon.Polly;
    using Amazon.Polly.Model;

    /// <summary>
    /// Retrieves information about a specific Amazon Polly lexicon.
    /// </summary>
    public class GetLexicon
    {
        public static async Task Main(string[] args)
        {
            string lexiconName = "SampleLexicon";

            var client = new AmazonPollyClient();

            await GetPollyLexiconAsync(client, lexiconName);
        }

        public static async Task GetPollyLexiconAsync(AmazonPollyClient client,
string lexiconName)
        {
            var getLexiconRequest = new GetLexiconRequest()
            {
                Name = lexiconName,
            };

            try
            {
                var response = await client.GetLexiconAsync(getLexiconRequest);
                Console.WriteLine($"Lexicon:\n Name: {response.Lexicon.Name}");
                Console.WriteLine($"Content: {response.Lexicon.Content}");
            }
            catch (Exception ex)
            {
                Console.WriteLine("Error: " + ex.Message);
            }
        }
    }
```

- Para obter detalhes da API, consulte GetLexicona Referência AWS SDK para .NET da API.

CLI

AWS CLI

Para recuperar o conteúdo de um léxico

O exemplo de `get-lexicon` a seguir recupera o conteúdo do léxico de pronúncia especificado.

```
aws polly get-lexicon \
    --name w3c
```

Saída:

```
{
    "Lexicon": {
        "Content": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<lexicon version=
\"1.0\" \n      xmlns=    \"http://www.w3.org/2005/01/pronunciation-lexicon
\"\n      xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" \n
 xsi:schemaLocation=\"http://www.w3.org/2005/01/pronunciation-lexicon \n
 http://www.w3.org/TR/2007/CR-pronunciation-    lexicon-20071212/pls.xsd\"\n
   alphabet=\"ipa\" \n      xml:lang=\"en-US\">\n  <lexeme>\n    <grapheme>W3C</
grapheme>\n        <alias>World Wide Web Consortium</alias>\n  </lexeme>\n</
lexicon>\n",
        "Name": "w3c"
    },
    "LexiconAttributes": {
        "Alphabet": "ipa",
        "LanguageCode": "en-US",
        "LastModified": 1603908910.99,
        "LexiconArn": "arn:aws:polly:us-west-2:880185128111:lexicon/w3c",
        "LexemesCount": 1,
        "Size": 492
    }
}
```

Para obter mais informações, consulte Como usar a GetLexicon operação no Amazon Polly Developer Guide.

- Para obter detalhes da API, consulte GetLexiconna Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class PollyWrapper:
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
 resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None


    def get_lexicon(self, name):
        """
        Gets metadata and contents of an existing lexicon.

        :param name: The name of the lexicon to retrieve.
        :return: The retrieved lexicon.
        """
        try:
            response = self.polly_client.get_lexicon(Name=name)
            logger.info("Got lexicon %s.", name)
        except ClientError:
            logger.exception("Couldn't get lexicon %s.", name)
            raise
        else:
            return response
```

- Para obter detalhes da API, consulte a GetLexiconReferência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
TRY.
    oo_result = lo_ply->getlexicon( iv_name ).
    DATA(lo_lexicon) = oo_result->get_lexicon( ).
    IF lo_lexicon IS BOUND.
      DATA(lv_lex_name) = lo_lexicon->get_name( ).
      MESSAGE |Retrieved lexicon: { lv_lex_name }| TYPE 'I'.
    ENDIF.
  CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Para obter detalhes da API, consulte a GetLexiconreferência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`GetSpeechSynthesisTask`**com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `GetSpeechSynthesisTask`.

CLI

AWS CLI

Como obter informações sobre uma tarefa de síntese de fala

O exemplo de `get-speech-synthesis-task` a seguir recupera informações sobre a tarefa de síntese de fala especificada.

```
aws polly get-speech-synthesis-task \
    --task-id 70b61c0f-57ce-4715-a247-cae8729dcce9
```

Saída:

```
{
    "SynthesisTask": {
        "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
        "TaskStatus": "completed",
        "OutputUri": "https://s3.us-west-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
        "CreationTime": 1603911042.689,
        "RequestCharacters": 1311,
        "OutputFormat": "mp3",
        "TextType": "text",
        "VoiceId": "Joanna"
    }
}
```

Para obter mais informações, consulte Criar arquivos de áudio longos no Guia do desenvolvedor do Amazon Polly.

- Para obter detalhes da API, consulte GetSpeechSynthesisTaskna Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class PollyWrapper:
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
 resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None


    def get_speech_synthesis_task(self, task_id):
        """
        Gets metadata about an asynchronous speech synthesis task, such as its
 status.

        :param task_id: The ID of the task to retrieve.
        :return: Metadata about the task.
        """
        try:
            response =
 self.polly_client.get_speech_synthesis_task(TaskId=task_id)
            task = response["SynthesisTask"]
            logger.info("Got synthesis task. Status is %s.", task["TaskStatus"])
        except ClientError:
            logger.exception("Couldn't get synthesis task %s.", task_id)
            raise
        else:
            return task
```

- Para obter detalhes da API, consulte a GetSpeechSynthesisTaskReferência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
TRY.
    oo_result = lo_ply->getspeechsynthesistask( iv_task_id ).
    DATA(lo_task) = oo_result->get_synthesistask( ).
    IF lo_task IS BOUND.
      DATA(lv_status) = lo_task->get_taskstatus( ).
      MESSAGE |Task status: { lv_status }| TYPE 'I'.
    ENDIF.
  CATCH /aws1/cx_plyinvalidtaskidex.
    MESSAGE 'Invalid task ID.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
  CATCH /aws1/cx_plysynthesistsknotf00.
    MESSAGE 'Synthesis task not found.' TYPE 'E'.
ENDTRY.
```

- Para obter detalhes da API, consulte a GetSpeechSynthesisTaskreferência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`ListLexicons`**com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListLexicons`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Introdução à síntese de conversão de texto em voz

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
using System;
using System.Threading.Tasks;
using Amazon.Polly;
using Amazon.Polly.Model;

/// <summary>
/// Lists the Amazon Polly lexicons that have been defined. By default,
/// lists the lexicons that are defined in the same AWS Region as the default
/// user. To view Amazon Polly lexicons that are defined in a different AWS
/// Region, supply it as a parameter to the Amazon Polly constructor.
/// </summary>
public class ListLexicons
{
    public static async Task Main()
    {
        var client = new AmazonPollyClient();
        var request = new ListLexiconsRequest();

        try
        {
            Console.WriteLine("All voices: ");

            do
```

```
                {
                    var response = await client.ListLexiconsAsync(request);
                    request.NextToken = response.NextToken;

                    response.Lexicons.ForEach(lexicon =>
                    {
                        var attributes = lexicon.Attributes;
                        Console.WriteLine($"Name: {lexicon.Name}");
                        Console.WriteLine($"\tAlphabet: {attributes.Alphabet}");
                        Console.WriteLine($"\tLanguageCode:
{attributes.LanguageCode}");
                        Console.WriteLine($"\tLastModified:
{attributes.LastModified}");
                        Console.WriteLine($"\tLexemesCount:
{attributes.LexemesCount}");
                        Console.WriteLine($"\tLexiconArn:
{attributes.LexiconArn}");
                        Console.WriteLine($"\tSize: {attributes.Size}");
                    });
                }
                while (request.NextToken is not null);
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Error: {ex.Message}");
            }
        }
    }
```

- Para obter detalhes da API, consulte ListLexiconsa Referência AWS SDK para .NET da API.

CLI

AWS CLI

Para listar os léxicos

O exemplo de `list-lexicons` a seguir lista os léxicos de pronúncia.

```
aws polly list-lexicons
```

Saída:

```
{
    "Lexicons": [
        {
            "Name": "w3c",
            "Attributes": {
                "Alphabet": "ipa",
                "LanguageCode": "en-US",
                "LastModified": 1603908910.99,
                "LexiconArn": "arn:aws:polly:us-east-2:123456789012:lexicon/w3c",
                "LexemesCount": 1,
                "Size": 492
            }
        }
    ]
}
```

Para obter mais informações, consulte Como usar a ListLexicons operação no Amazon Polly Developer Guide.

- Para obter detalhes da API, consulte ListLexiconsna Referência de AWS CLI Comandos.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.ListLexiconsResponse;
import software.amazon.awssdk.services.polly.model.ListLexiconsRequest;
import software.amazon.awssdk.services.polly.model.LexiconDescription;
import software.amazon.awssdk.services.polly.model.PollyException;
import java.util.List;

/**
```

```
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListLexicons {
    public static void main(String args[]) {
        PollyClient polly = PollyClient.builder()
                .region(Region.US_WEST_2)
                .build();

        listLexicons(polly);
        polly.close();
    }

    public static void listLexicons(PollyClient client) {
        try {
            ListLexiconsRequest listLexiconsRequest =
 ListLexiconsRequest.builder()
                    .build();

            ListLexiconsResponse listLexiconsResult =
 client.listLexicons(listLexiconsRequest);
            List<LexiconDescription> lexiconDescription =
 listLexiconsResult.lexicons();
            for (LexiconDescription lexDescription : lexiconDescription) {
                System.out.println("The name of the Lexicon is " +
 lexDescription.name());
            }

        } catch (PollyException e) {
            System.err.println("Exception caught: " + e);
            System.exit(1);
        }
    }
}
```

- Para obter detalhes da API, consulte ListLexiconsa Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class PollyWrapper:
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
 resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None


    def list_lexicons(self):
        """
        Lists lexicons in the current account.

        :return: The list of lexicons.
        """
        try:
            response = self.polly_client.list_lexicons()
            lexicons = response["Lexicons"]
            logger.info("Got %s lexicons.", len(lexicons))
        except ClientError:
            logger.exception(
                "Couldn't get  %s.",
            )
            raise
        else:
            return lexicons
```

- Para obter detalhes da API, consulte a ListLexiconsReferência da API AWS SDK for Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
require 'aws-sdk-polly' # In v2: require 'aws-sdk'

begin
  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/
config
  polly = Aws::Polly::Client.new

  resp = polly.list_lexicons

  resp.lexicons.each do |l|
    puts l.name
    puts "  Alphabet:#{l.attributes.alphabet}"
    puts "  Language:#{l.attributes.language}"
    puts
  end
rescue StandardError => e
  puts 'Could not get lexicons'
  puts 'Error message:'
  puts e.message
end
```

- Para obter detalhes da API, consulte ListLexiconsa Referência AWS SDK para Ruby da API.

Rust

SDK para Rust

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
async fn show_lexicons(client: &Client) -> Result<(), Error> {
    let resp = client.list_lexicons().send().await?;

    println!("Lexicons:");

    let lexicons = resp.lexicons();

    for lexicon in lexicons {
        println!("  Name:     {}", lexicon.name().unwrap_or_default());
        println!(
            "  Language: {:?}\n",
            lexicon
                .attributes()
                .as_ref()
                .map(|attrib| attrib
                    .language_code
                    .as_ref()
                    .expect("languages must have language codes"))
                .expect("languages must have attributes")
        );
    }

    println!();
    println!("Found {} lexicons.", lexicons.len());
    println!();

    Ok(())
}
```

- Para obter detalhes da API, consulte a ListLexiconsreferência da API AWS SDK for Rust.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
TRY.
    oo_result = lo_ply->listlexicons( ).
    DATA(lt_lexicons) = oo_result->get_lexicons( ).
    DATA(lv_count) = lines( lt_lexicons ).
    MESSAGE |Found { lv_count } lexicons| TYPE 'I'.
  CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'Invalid NextToken.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Para obter detalhes da API, consulte a ListLexiconsreferência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`ListSpeechSynthesisTasks`**com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `ListSpeechSynthesisTasks`.

CLI

AWS CLI

Para listar suas tarefas de síntese de fala

O exemplo `list-speech-synthesis-tasks` a seguir lista suas tarefas de síntese de fala.

```
aws polly list-speech-synthesis-tasks
```

Saída:

```
{
    "SynthesisTasks": [
        {
            "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
            "TaskStatus": "completed",
            "OutputUri": "https://s3.us-west-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
            "CreationTime": 1603911042.689,
            "RequestCharacters": 1311,
            "OutputFormat": "mp3",
            "TextType": "text",
            "VoiceId": "Joanna"
        }
    ]
}
```

Para obter mais informações, consulte Criar arquivos de áudio longos no Guia do desenvolvedor do Amazon Polly.

- Para obter detalhes da API, consulte ListSpeechSynthesisTasksna Referência de AWS CLI Comandos.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
TRY.
    " Only pass optional parameters if they have values
    IF iv_max_results IS NOT INITIAL AND iv_status IS NOT INITIAL.
      oo_result = lo_ply->listspeechsynthesistasks(
        iv_maxresults = iv_max_results
        iv_status = iv_status ).
    ELSEIF iv_max_results IS NOT INITIAL.
      oo_result = lo_ply->listspeechsynthesistasks(
        iv_maxresults = iv_max_results ).
    ELSEIF iv_status IS NOT INITIAL.
      oo_result = lo_ply->listspeechsynthesistasks(
        iv_status = iv_status ).
    ELSE.
      oo_result = lo_ply->listspeechsynthesistasks( ).
    ENDIF.
    DATA(lt_tasks) = oo_result->get_synthesistasks( ).
    DATA(lv_count) = lines( lt_tasks ).
    MESSAGE |Found { lv_count } synthesis tasks| TYPE 'I'.
  CATCH /aws1/cx_plyinvalidnexttokenex.
    MESSAGE 'Invalid NextToken.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
ENDTRY.
```

- Para obter detalhes da API, consulte a ListSpeechSynthesisTasksreferência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`PutLexicon`**com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `PutLexicon`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Introdução à síntese de conversão de texto em voz

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.Polly;
    using Amazon.Polly.Model;

    /// <summary>
    /// Creates a new Amazon Polly lexicon using the AWS SDK for .NET.
    /// </summary>
    public class PutLexicon
    {
        public static async Task Main()
        {
            string lexiconContent = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>"
 +
                "<lexicon version=\"1.0\" xmlns=\"http://www.w3.org/2005/01/
pronunciation-lexicon\" xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" "
 +
                "xsi:schemaLocation=\"http://www.w3.org/2005/01/pronunciation-
lexicon http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd\" " +
                "alphabet=\"ipa\" xml:lang=\"en-US\">" +
                "<lexeme><grapheme>test1</grapheme><alias>test2</alias></lexeme>"
 +
```

```
                "</lexicon>";
            string lexiconName = "SampleLexicon";

            var client = new AmazonPollyClient();
            var putLexiconRequest = new PutLexiconRequest()
            {
                Name = lexiconName,
                Content = lexiconContent,
            };

            try
            {
                var response = await client.PutLexiconAsync(putLexiconRequest);
                if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
                {
                    Console.WriteLine($"Successfully created Lexicon:
{lexiconName}.");
                }
                else
                {
                    Console.WriteLine($"Could not create Lexicon:
{lexiconName}.");
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine("Exception caught: " + ex.Message);
            }
        }
    }
```

- Para obter detalhes da API, consulte PutLexicona Referência AWS SDK para .NET da API.

CLI

AWS CLI

Como armazenar um léxico

O exemplo de `put-lexicon` a seguir armazena o léxico de pronúncia especificado. O `example.pls` arquivo especifica um léxico W3C PLS-compliant .

```
aws polly put-lexicon \
    --name w3c \
    --content file://example.pls
```

Conteúdo de `example.pls`

```
{
    <?xml version="1.0" encoding="UTF-8"?>
    <lexicon version="1.0"
        xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
            http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
        alphabet="ipa"
        xml:lang="en-US">
        <lexeme>
            <grapheme>W3C</grapheme>
            <alias>World Wide Web Consortium</alias>
        </lexeme>
    </lexicon>
}
```

Este comando não produz saída.

Para obter mais informações, consulte Como usar a PutLexicon operação no Amazon Polly Developer Guide.

- Para obter detalhes da API, consulte PutLexiconna Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class PollyWrapper:
```

```
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None


    def create_lexicon(self, name, content):
        """
        Creates a lexicon with the specified content. A lexicon contains custom
        pronunciations.

        :param name: The name of the lexicon.
        :param content: The content of the lexicon.
        """
        try:
            self.polly_client.put_lexicon(Name=name, Content=content)
            logger.info("Created lexicon %s.", name)
        except ClientError:
            logger.exception("Couldn't create lexicon %s.")
            raise
```

- Para obter detalhes da API, consulte a PutLexiconReferência da API AWS SDK for Python (Boto3).

Rust

SDK para Rust

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
async fn make_lexicon(client: &Client, name: &str, from: &str, to: &str) ->
 Result<(), Error> {
    let content = format!("<?xml version=\"1.0\" encoding=\"UTF-8\"?>
    <lexicon version=\"1.0\" xmlns=\"http://www.w3.org/2005/01/pronunciation-
lexicon\" xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
    xsi:schemaLocation=\"http://www.w3.org/2005/01/pronunciation-lexicon http://
www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd\"
    alphabet=\"ipa\" xml:lang=\"en-US\">
    <lexeme><grapheme>{}</grapheme><alias>{}</alias></lexeme>
    </lexicon>", from, to);

    client
        .put_lexicon()
        .name(name)
        .content(content)
        .send()
        .await?;

    println!("Added lexicon");

    Ok(())
}
```

- Para obter detalhes da API, consulte a PutLexiconreferência da API AWS SDK for Rust.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
    TRY.
        lo_ply->putlexicon(
          iv_name = iv_name
          iv_content = iv_content ).
        MESSAGE 'Lexicon created successfully.' TYPE 'I'.
```

```
      CATCH /aws1/cx_plyinvalidlexiconex.
        MESSAGE 'Invalid lexicon.' TYPE 'E'.
      CATCH /aws1/cx_plylexiconsizeexcdex.
        MESSAGE 'Lexicon size exceeded.' TYPE 'E'.
      CATCH /aws1/cx_plymaxlexemelengthe00.
        MESSAGE 'Maximum lexeme length exceeded.' TYPE 'E'.
      CATCH /aws1/cx_plymaxlexiconsnoexc00.
        MESSAGE 'Maximum number of lexicons exceeded.' TYPE 'E'.
      CATCH /aws1/cx_plyservicefailureex.
        MESSAGE 'Service failure occurred.' TYPE 'E'.
      CATCH /aws1/cx_plyunsuppedplsalpha00.
        MESSAGE 'Unsupported PLS alphabet.' TYPE 'E'.
      CATCH /aws1/cx_plyunsuppedplslangu00.
        MESSAGE 'Unsupported PLS language.' TYPE 'E'.
    ENDTRY.
```

- Para obter detalhes da API, consulte a PutLexiconreferência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`StartSpeechSynthesisTask`**com um AWS SDK ou CLI

Os exemplos de código a seguir mostram como usar o `StartSpeechSynthesisTask`.

CLI

AWS CLI

Como sintetizar texto

O exemplo de `start-speech-synthesis-task` a seguir sintetiza o texto em `text_file.txt` e armazena o arquivo MP3 resultante no bucket especificado.

```
aws polly start-speech-synthesis-task \
    --output-format mp3 \
    --output-s3-bucket-name amzn-s3-demo-bucket \
    --text  file://text_file.txt \
    --voice-id Joanna
```

Saída:

```
{
    "SynthesisTask": {
        "TaskId": "70b61c0f-57ce-4715-a247-cae8729dcce9",
        "TaskStatus": "scheduled",
        "OutputUri": "https://s3.us-east-2.amazonaws.com/amzn-s3-demo-
bucket/70b61c0f-57ce-4715-a247-cae8729dcce9.mp3",
        "CreationTime": 1603911042.689,
        "RequestCharacters": 1311,
        "OutputFormat": "mp3",
        "TextType": "text",
        "VoiceId": "Joanna"
    }
}
```

Para obter mais informações, consulte Criar arquivos de áudio longos no Guia do desenvolvedor do Amazon Polly.

- Para obter detalhes da API, consulte StartSpeechSynthesisTaskna Referência de AWS CLI Comandos.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class PollyWrapper:
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
 resource.
        """
```

```
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None


    def do_synthesis_task(
        self,
        text,
        engine,
        voice,
        audio_format,
        s3_bucket,
        lang_code=None,
        include_visemes=False,
        wait_callback=None,
    ):
        """
        Start an asynchronous task to synthesize speech or speech marks, wait for
        the task to complete, retrieve the output from Amazon S3, and return the
        data.

        An asynchronous task is required when the text is too long for near-real
time
        synthesis.

        :param text: The text to synthesize.
        :param engine: The kind of engine used. Can be standard or neural.
        :param voice: The ID of the voice to use.
        :param audio_format: The audio format to return for synthesized speech.
When
                             speech marks are synthesized, the output format is
JSON.
        :param s3_bucket: The name of an existing Amazon S3 bucket that you have
                          write access to. Synthesis output is written to this
bucket.
        :param lang_code: The language code of the voice to use. This has an
effect
                          only when a bilingual voice is selected.
        :param include_visemes: When True, a second request is made to Amazon
Polly
                                to synthesize a list of visemes, using the
specified
                                text and voice. A viseme represents the visual
position
```

```
                                of the face and mouth when saying part of a word.
        :param wait_callback: A callback function that is called periodically
during
                              task processing, to give the caller an opportunity
to
                              take action, such as to display status.
        :return: The audio stream that contains the synthesized speech and a list
                 of visemes that are associated with the speech audio.
        """
        try:
            kwargs = {
                "Engine": engine,
                "OutputFormat": audio_format,
                "OutputS3BucketName": s3_bucket,
                "Text": text,
                "VoiceId": voice,
            }
            if lang_code is not None:
                kwargs["LanguageCode"] = lang_code
            response = self.polly_client.start_speech_synthesis_task(**kwargs)
            speech_task = response["SynthesisTask"]
            logger.info("Started speech synthesis task %s.",
speech_task["TaskId"])

            viseme_task = None
            if include_visemes:
                kwargs["OutputFormat"] = "json"
                kwargs["SpeechMarkTypes"] = ["viseme"]
                response =
self.polly_client.start_speech_synthesis_task(**kwargs)
                viseme_task = response["SynthesisTask"]
                logger.info("Started viseme synthesis task %s.",
viseme_task["TaskId"])
        except ClientError:
            logger.exception("Couldn't start synthesis task.")
            raise
        else:
            bucket = self.s3_resource.Bucket(s3_bucket)
            audio_stream = self._wait_for_task(
                10, speech_task["TaskId"], "speech", wait_callback, bucket
            )

            visemes = None
            if include_visemes:
```

```
                viseme_data = self._wait_for_task(
                    10, viseme_task["TaskId"], "viseme", wait_callback, bucket
                )
                visemes = [
                    json.loads(v) for v in viseme_data.read().decode().split() if
v
                ]

            return audio_stream, visemes
```

- Para obter detalhes da API, consulte a StartSpeechSynthesisTaskReferência da API AWS SDK for Python (Boto3).

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
    TRY.
        " Only pass optional parameters if they have values
        IF iv_lang_code IS NOT INITIAL AND iv_s3_key_prefix IS NOT INITIAL.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_outputs3keyprefix = iv_s3_key_prefix
            iv_text = iv_text
            iv_voiceid = iv_voice_id
            iv_languagecode = iv_lang_code ).
        ELSEIF iv_lang_code IS NOT INITIAL.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
```

```
            iv_text = iv_text
            iv_voiceid = iv_voice_id
            iv_languagecode = iv_lang_code ).
        ELSEIF iv_s3_key_prefix IS NOT INITIAL.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_outputs3keyprefix = iv_s3_key_prefix
            iv_text = iv_text
            iv_voiceid = iv_voice_id ).
        ELSE.
          oo_result = lo_ply->startspeechsynthesistask(
            iv_engine = iv_engine
            iv_outputformat = iv_audio_format
            iv_outputs3bucketname = iv_s3_bucket
            iv_text = iv_text
            iv_voiceid = iv_voice_id ).
        ENDIF.
        MESSAGE 'Speech synthesis task started.' TYPE 'I'.
      CATCH /aws1/cx_plyinvalids3bucketex.
        MESSAGE 'Invalid S3 bucket.' TYPE 'E'.
      CATCH /aws1/cx_plyinvalidssmlex.
        MESSAGE 'Invalid SSML.' TYPE 'E'.
      CATCH /aws1/cx_plylexiconnotfoundex.
        MESSAGE 'Lexicon not found.' TYPE 'E'.
      CATCH /aws1/cx_plyservicefailureex.
        MESSAGE 'Service failure occurred.' TYPE 'E'.
      CATCH /aws1/cx_plytextlengthexcdex.
        MESSAGE 'Text length exceeded maximum.' TYPE 'E'.
    ENDTRY.
```

- Para obter detalhes da API, consulte a StartSpeechSynthesisTaskreferência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Use **`SynthesizeSpeech`**com um AWS SDK

Os exemplos de código a seguir mostram como usar o `SynthesizeSpeech`.

Exemplos de ações são trechos de código de programas maiores e devem ser executados em contexto. É possível ver essa ação em contexto no seguinte exemplo de código:

- Introdução à síntese de conversão de texto em voz

.NET

SDK para .NET

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
    using System;
    using System.IO;
    using System.Threading.Tasks;
    using Amazon.Polly;
    using Amazon.Polly.Model;

    public class SynthesizeSpeech
    {
        public static async Task Main()
        {
            string outputFileName = "speech.mp3";
            string text = "Twas brillig, and the slithy toves did gyre and gimbol
in the wabe";

            var client = new AmazonPollyClient();
            var response = await PollySynthesizeSpeech(client, text);

            WriteSpeechToStream(response.AudioStream, outputFileName);
        }

        /// <summary>
        /// Calls the Amazon Polly SynthesizeSpeechAsync method to convert text
        /// to speech.
```

```
        /// </summary>
        /// <param name="client">The Amazon Polly client object used to connect
        /// to the Amazon Polly service.</param>
        /// <param name="text">The text to convert to speech.</param>
        /// <returns>A SynthesizeSpeechResponse object that includes an
AudioStream
        /// object with the converted text.</returns>
        private static async Task<SynthesizeSpeechResponse>
PollySynthesizeSpeech(IAmazonPolly client, string text)
        {
            var synthesizeSpeechRequest = new SynthesizeSpeechRequest()
            {
                OutputFormat = OutputFormat.Mp3,
                VoiceId = VoiceId.Joanna,
                Text = text,
            };

            var synthesizeSpeechResponse =
                await client.SynthesizeSpeechAsync(synthesizeSpeechRequest);

            return synthesizeSpeechResponse;
        }

        /// <summary>
        /// Writes the AudioStream returned from the call to
        /// SynthesizeSpeechAsync to a file in MP3 format.
        /// </summary>
        /// <param name="audioStream">The AudioStream returned from the
        /// call to the SynthesizeSpeechAsync method.</param>
        /// <param name="outputFileName">The full path to the file in which to
        /// save the audio stream.</param>
        private static void WriteSpeechToStream(Stream audioStream, string
outputFileName)
        {
            var outputStream = new FileStream(
                outputFileName,
                FileMode.Create,
                FileAccess.Write);
            byte[] buffer = new byte[2 * 1024];
            int readBytes;

            while ((readBytes = audioStream.Read(buffer, 0, 2 * 1024)) > 0)
            {
                outputStream.Write(buffer, 0, readBytes);
```

```
            }

            // Flushes the buffer to avoid losing the last second or so of
            // the synthesized text.
            outputStream.Flush();
            Console.WriteLine($"Saved {outputFileName} to disk.");
        }
    }
```

Sintetize fala a partir de texto usando marcas de fala com o Amazon Polly usando um SDK. AWS

```
    using System;
    using System.Collections.Generic;
    using System.IO;
    using System.Threading.Tasks;
    using Amazon.Polly;
    using Amazon.Polly.Model;

    public class SynthesizeSpeechMarks
    {
        public static async Task Main()
        {
            var client = new AmazonPollyClient();
            string outputFileName = "speechMarks.json";

            var synthesizeSpeechRequest = new SynthesizeSpeechRequest()
            {
                OutputFormat = OutputFormat.Json,
                SpeechMarkTypes = new List<string>
                {
                    SpeechMarkType.Viseme,
                    SpeechMarkType.Word,
                },
                VoiceId = VoiceId.Joanna,
                Text = "This is a sample text to be synthesized.",
            };

            try
            {
```

```
                using (var outputStream = new FileStream(outputFileName,
FileMode.Create, FileAccess.Write))
                {
                    var synthesizeSpeechResponse = await
client.SynthesizeSpeechAsync(synthesizeSpeechRequest);
                    var buffer = new byte[2 * 1024];
                    int readBytes;

                    var inputStream = synthesizeSpeechResponse.AudioStream;
                    while ((readBytes = inputStream.Read(buffer, 0, 2 * 1024)) >
0)
                    {
                        outputStream.Write(buffer, 0, readBytes);
                    }
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Error: {ex.Message}");
            }
        }
    }
```

- Para obter detalhes da API, consulte SynthesizeSpeecha Referência AWS SDK para .NET da API.

Java

SDK para Java 2.x

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
import javazoom.jl.decoder.JavaLayerException;
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.polly.PollyClient;
import software.amazon.awssdk.services.polly.model.DescribeVoicesRequest;
import software.amazon.awssdk.services.polly.model.Voice;
import software.amazon.awssdk.services.polly.model.DescribeVoicesResponse;
import software.amazon.awssdk.services.polly.model.OutputFormat;
import software.amazon.awssdk.services.polly.model.PollyException;
import software.amazon.awssdk.services.polly.model.SynthesizeSpeechRequest;
import software.amazon.awssdk.services.polly.model.SynthesizeSpeechResponse;
import java.io.IOException;
import java.io.InputStream;
import javazoom.jl.player.advanced.AdvancedPlayer;
import javazoom.jl.player.advanced.PlaybackEvent;
import javazoom.jl.player.advanced.PlaybackListener;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class PollyDemo {
    private static final String SAMPLE = "Congratulations. You have successfully
 built this working demo " +
            " of Amazon Polly in Java Version 2. Have fun building voice enabled
 apps with Amazon Polly (that's me!), and always "
            +
            " look at the AWS website for tips and tricks on using Amazon Polly
 and other great services from AWS";

    public static void main(String args[]) {
        PollyClient polly = PollyClient.builder()
                .region(Region.US_WEST_2)
                .build();

        talkPolly(polly);
        polly.close();
    }

    public static void talkPolly(PollyClient polly) {
        try {
```

```
            DescribeVoicesRequest describeVoiceRequest =
DescribeVoicesRequest.builder()
                    .engine("standard")
                    .build();

            DescribeVoicesResponse describeVoicesResult =
polly.describeVoices(describeVoiceRequest);
            Voice voice = describeVoicesResult.voices().stream()
                    .filter(v -> v.name().equals("Joanna"))
                    .findFirst()
                    .orElseThrow(() -> new RuntimeException("Voice not found"));
            InputStream stream = synthesize(polly, SAMPLE, voice,
OutputFormat.MP3);
            AdvancedPlayer player = new AdvancedPlayer(stream,

javazoom.jl.player.FactoryRegistry.systemRegistry().createAudioDevice());
            player.setPlayBackListener(new PlaybackListener() {
                public void playbackStarted(PlaybackEvent evt) {
                    System.out.println("Playback started");
                    System.out.println(SAMPLE);
                }

                public void playbackFinished(PlaybackEvent evt) {
                    System.out.println("Playback finished");
                }
            });

            // play it!
            player.play();

        } catch (PollyException | JavaLayerException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static InputStream synthesize(PollyClient polly, String text, Voice
voice, OutputFormat format)
            throws IOException {
        SynthesizeSpeechRequest synthReq = SynthesizeSpeechRequest.builder()
                .text(text)
                .voiceId(voice.id())
                .outputFormat(format)
                .build();
```

```
        ResponseInputStream<SynthesizeSpeechResponse> synthRes =
 polly.synthesizeSpeech(synthReq);
        return synthRes;
    }
}
```

- Para obter detalhes da API, consulte SynthesizeSpeecha Referência AWS SDK for Java 2.x da API.

Python

SDK para Python (Boto3)

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
class PollyWrapper:
    """Encapsulates Amazon Polly functions."""

    def __init__(self, polly_client, s3_resource):
        """
        :param polly_client: A Boto3 Amazon Polly client.
        :param s3_resource: A Boto3 Amazon Simple Storage Service (Amazon S3)
 resource.
        """
        self.polly_client = polly_client
        self.s3_resource = s3_resource
        self.voice_metadata = None


    def synthesize(
        self, text, engine, voice, audio_format, lang_code=None,
 include_visemes=False
    ):
        """
        Synthesizes speech or speech marks from text, using the specified voice.
```

```
        :param text: The text to synthesize.
        :param engine: The kind of engine used. Can be standard or neural.
        :param voice: The ID of the voice to use.
        :param audio_format: The audio format to return for synthesized speech.
When
                             speech marks are synthesized, the output format is
JSON.
        :param lang_code: The language code of the voice to use. This has an
effect
                          only when a bilingual voice is selected.
        :param include_visemes: When True, a second request is made to Amazon
Polly
                                to synthesize a list of visemes, using the
specified
                                text and voice. A viseme represents the visual
position
                                of the face and mouth when saying part of a word.
        :return: The audio stream that contains the synthesized speech and a list
                 of visemes that are associated with the speech audio.
        """
        try:
            kwargs = {
                "Engine": engine,
                "OutputFormat": audio_format,
                "Text": text,
                "VoiceId": voice,
            }
            if lang_code is not None:
                kwargs["LanguageCode"] = lang_code
            response = self.polly_client.synthesize_speech(**kwargs)
            audio_stream = response["AudioStream"]
            logger.info("Got audio stream spoken by %s.", voice)
            visemes = None
            if include_visemes:
                kwargs["OutputFormat"] = "json"
                kwargs["SpeechMarkTypes"] = ["viseme"]
                response = self.polly_client.synthesize_speech(**kwargs)
                visemes = [
                    json.loads(v)
                    for v in response["AudioStream"].read().decode().split()
                    if v
                ]
                logger.info("Got %s visemes.", len(visemes))
```

```
        except ClientError:
            logger.exception("Couldn't get audio stream.")
            raise
        else:
            return audio_stream, visemes
```

- Para obter detalhes da API, consulte a SynthesizeSpeechReferência da API AWS SDK for Python (Boto3).

Ruby

SDK para Ruby

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
require 'aws-sdk-polly' # In v2: require 'aws-sdk'

begin
  # Get the filename from the command line
  if ARGV.empty?
    puts 'You must supply a filename'
    exit 1
  end

  filename = ARGV[0]

  # Open file and get the contents as a string
  if File.exist?(filename)
    contents = IO.read(filename)
  else
    puts "No such file: #{filename}"
    exit 1
  end
```

```
  # Create an Amazon Polly client using
  # credentials from the shared credentials file ~/.aws/credentials
  # and the configuration (region) from the shared configuration file ~/.aws/
config
  polly = Aws::Polly::Client.new

  resp = polly.synthesize_speech({
                                   output_format: 'mp3',
                                   text: contents,
                                   voice_id: 'Joanna'
                                 })

  # Save output
  # Get just the file name
  #  abc/xyz.txt -> xyx.txt
  name = File.basename(filename)

  # Split up name so we get just the xyz part
  parts = name.split('.')
  first_part = parts[0]
  mp3_file = "#{first_part}.mp3"

  IO.copy_stream(resp.audio_stream, mp3_file)

  puts "Wrote MP3 content to: #{mp3_file}"
rescue StandardError => e
  puts 'Got error:'
  puts 'Error message:'
  puts e.message
end
```

- Para obter detalhes da API, consulte SynthesizeSpeecha Referência AWS SDK para Ruby da API.

Rust

SDK para Rust

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
async fn synthesize(client: &Client, filename: &str) -> Result<(), Error> {
    let content = fs::read_to_string(filename);

    let resp = client
        .synthesize_speech()
        .output_format(OutputFormat::Mp3)
        .text(content.unwrap())
        .voice_id(VoiceId::Joanna)
        .send()
        .await?;

    // Get MP3 data from response and save it
    let mut blob = resp
        .audio_stream
        .collect()
        .await
        .expect("failed to read data");

    let parts: Vec<&str> = filename.split('.').collect();
    let out_file = format!("{}{}", String::from(parts[0]), ".mp3");

    let mut file = tokio::fs::File::create(out_file)
        .await
        .expect("failed to create file");

    file.write_all_buf(&mut blob)
        .await
        .expect("failed to write to file");

    Ok(())
}
```

- Para obter detalhes da API, consulte a SynthesizeSpeechreferência da API AWS SDK for Rust.

SAP ABAP

SDK para SAP ABAP

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no AWS Code Examples Repository.

```
TRY.
    " Only pass optional language code if it has a value
    IF iv_lang_code IS NOT INITIAL.
      oo_result = lo_ply->synthesizespeech(
        iv_engine = iv_engine
        iv_outputformat = iv_output_fmt
        iv_text = iv_text
        iv_voiceid = iv_voice_id
        iv_languagecode = iv_lang_code ).
    ELSE.
      oo_result = lo_ply->synthesizespeech(
        iv_engine = iv_engine
        iv_outputformat = iv_output_fmt
        iv_text = iv_text
        iv_voiceid = iv_voice_id ).
    ENDIF.
    MESSAGE 'Speech synthesized successfully.' TYPE 'I'.
  CATCH /aws1/cx_plyinvalidssmlex.
    MESSAGE 'Invalid SSML.' TYPE 'E'.
  CATCH /aws1/cx_plylexiconnotfoundex.
    MESSAGE 'Lexicon not found.' TYPE 'E'.
  CATCH /aws1/cx_plyservicefailureex.
    MESSAGE 'Service failure occurred.' TYPE 'E'.
  CATCH /aws1/cx_plytextlengthexcdex.
    MESSAGE 'Text length exceeded maximum.' TYPE 'E'.
ENDTRY.
```

- Para obter detalhes da API, consulte a SynthesizeSpeechreferência da API AWS SDK for SAP ABAP.

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Cenários para o uso do Amazon Polly AWS SDKs

Os exemplos de código a seguir mostram como implementar cenários comuns no Amazon Polly com AWS SDKs. Esses casos mostram como realizar tarefas específicas chamando várias funções dentro do Amazon Polly ou combinadas com outros Serviços da AWS. Cada cenário inclui um link para o código-fonte completo, onde podem ser encontradas instruções sobre como configurar e executar o código.

Os cenários têm como alvo um nível intermediário de experiência para ajudar você a compreender ações de serviço em contexto.

Exemplos

- Converta texto em fala e volte para texto usando um AWS SDK
- Crie um aplicativo de sincronização labial com o Amazon Polly usando um AWS SDK
- Criar uma aplicação que analise o feedback dos clientes e sintetize o áudio
- Introdução à síntese de conversão de texto em voz

Converta texto em fala e volte para texto usando um AWS SDK

O exemplo de código a seguir mostra como:

- Use o Amazon Polly para sintetizar um arquivo de entrada de texto simples (UTF-8) em um arquivo de áudio.
- Fazer upload do arquivo de áudio para um bucket do Amazon S3.
- Usar o Amazon Transcribe para converter o arquivo de áudio em texto.
- Exibir o texto.

Rust

SDK para Rust

Use o Amazon Polly para sintetizar um arquivo de entrada de texto simples (UTF-8) em um arquivo de áudio, fazer upload do arquivo de áudio em um bucket do Amazon S3, usar o Amazon Transcribe para converter esse arquivo de áudio em texto e exibir o texto.

Para obter o código-fonte completo e instruções sobre como configurar e executar, veja o exemplo completo em GitHub.

Serviços usados neste exemplo

- Amazon Polly
- Amazon S3
- Amazon Transcribe

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Crie um aplicativo de sincronização labial com o Amazon Polly usando um AWS SDK

O exemplo de código a seguir mostra como criar um aplicativo de sincronização de lábios com o Amazon Polly.

Python

SDK para Python (Boto3)

Mostra como usar o Amazon Polly e o Tkinter para criar um aplicativo de sincronização labial que exibe um rosto animado falando junto com a fala sintetizada pelo Amazon Polly. Lip-sync é realizado solicitando uma lista de fonemas do Amazon Polly que correspondam à fala sintetizada.

- Obter metadados de voz do Amazon Polly e exibi-os em um aplicativo Tkinter.
- Obter áudio de fala sintetizado e respectivas marcas de fala de fonema do Amazon Polly.
- Reproduzir o áudio com movimentos sincronizados da boca em um rosto animado.

- Enviar tarefas de síntese assíncrona para textos longos e recuperar a saída de um bucket do Amazon Simple Storage Service (Amazon S3).

Para obter o código-fonte completo e instruções sobre como configurar e executar, veja o exemplo completo em GitHub.

Serviços usados neste exemplo

- Amazon Polly

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Criar uma aplicação que analise o feedback dos clientes e sintetize o áudio

Os exemplos de código a seguir mostram como criar uma aplicação que analisa os cartões de comentários dos clientes, os traduz do idioma original, determina seus sentimentos e gera um arquivo de áudio do texto traduzido.

.NET

SDK para .NET

Esta aplicação de exemplo analisa e armazena cartões de feedback de clientes. Especificamente, ela atende à necessidade de um hotel fictício na cidade de Nova York. O hotel recebe feedback dos hóspedes em vários idiomas na forma de cartões de comentários físicos. Esse feedback é enviado para a aplicação por meio de um cliente web. Depois de fazer upload da imagem de um cartão de comentário, ocorrem as seguintes etapas:

- O texto é extraído da imagem usando o Amazon Textract.
- O Amazon Comprehend determina o sentimento do texto extraído e o idioma.
- O texto extraído é traduzido para o inglês com o Amazon Translate.
- O Amazon Polly sintetiza um arquivo de áudio do texto extraído.

A aplicação completa pode ser implantada com o AWS CDK. Para obter o código-fonte e as instruções de implantação, consulte o projeto em GitHub.

Serviços usados neste exemplo

- Amazon Comprehend

- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Java

SDK para Java 2.x

Esta aplicação de exemplo analisa e armazena cartões de feedback de clientes. Especificamente, ela atende à necessidade de um hotel fictício na cidade de Nova York. O hotel recebe feedback dos hóspedes em vários idiomas na forma de cartões de comentários físicos. Esse feedback é enviado para a aplicação por meio de um cliente web. Depois de fazer upload da imagem de um cartão de comentário, ocorrem as seguintes etapas:

- O texto é extraído da imagem usando o Amazon Textract.
- O Amazon Comprehend determina o sentimento do texto extraído e o idioma.
- O texto extraído é traduzido para o inglês com o Amazon Translate.
- O Amazon Polly sintetiza um arquivo de áudio do texto extraído.

A aplicação completa pode ser implantada com o AWS CDK. Para obter o código-fonte e as instruções de implantação, consulte o projeto em GitHub.

Serviços usados neste exemplo

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

JavaScript

SDK para JavaScript (v3)

Esta aplicação de exemplo analisa e armazena cartões de feedback de clientes. Especificamente, ela atende à necessidade de um hotel fictício na cidade de Nova York. O hotel recebe feedback dos hóspedes em vários idiomas na forma de cartões de comentários

físicos. Esse feedback é enviado para a aplicação por meio de um cliente web. Depois de fazer upload da imagem de um cartão de comentário, ocorrem as seguintes etapas:

- O texto é extraído da imagem usando o Amazon Textract.
- O Amazon Comprehend determina o sentimento do texto extraído e o idioma.
- O texto extraído é traduzido para o inglês com o Amazon Translate.
- O Amazon Polly sintetiza um arquivo de áudio do texto extraído.

A aplicação completa pode ser implantada com o AWS CDK. Para obter o código-fonte e as instruções de implantação, consulte o projeto em GitHub. Os trechos a seguir mostram como o AWS SDK para JavaScript é usado nas funções do Lambda.

```
import {
  ComprehendClient,
  DetectDominantLanguageCommand,
  DetectSentimentCommand,
} from "@aws-sdk/client-comprehend";

/**
 * Determine the language and sentiment of the extracted text.
 *
 * @param {{ source_text: string}} extractTextOutput
 */
export const handler = async (extractTextOutput) => {
  const comprehendClient = new ComprehendClient({});

  const detectDominantLanguageCommand = new DetectDominantLanguageCommand({
    Text: extractTextOutput.source_text,
  });

  // The source language is required for sentiment analysis and
  // translation in the next step.
  const { Languages } = await comprehendClient.send(
    detectDominantLanguageCommand,
  );

  const languageCode = Languages[0].LanguageCode;

  const detectSentimentCommand = new DetectSentimentCommand({
    Text: extractTextOutput.source_text,
    LanguageCode: languageCode,
  });
```

```
  const { Sentiment } = await comprehendClient.send(detectSentimentCommand);

  return {
    sentiment: Sentiment,
    language_code: languageCode,
  };
};
```

```
import {
  DetectDocumentTextCommand,
  TextractClient,
} from "@aws-sdk/client-textract";

/**
 * Fetch the S3 object from the event and analyze it using Amazon Textract.
 *
 * @param {import("@types/aws-lambda").EventBridgeEvent<"Object Created">}
 eventBridgeS3Event
 */
export const handler = async (eventBridgeS3Event) => {
  const textractClient = new TextractClient();

  const detectDocumentTextCommand = new DetectDocumentTextCommand({
    Document: {
      S3Object: {
        Bucket: eventBridgeS3Event.bucket,
        Name: eventBridgeS3Event.object,
      },
    },
  });

  // Textract returns a list of blocks. A block can be a line, a page, word, etc.
  // Each block also contains geometry of the detected text.
  // For more information on the Block type, see https://docs.aws.amazon.com/
textract/latest/dg/API_Block.html.
  const { Blocks } = await textractClient.send(detectDocumentTextCommand);

  // For the purpose of this example, we are only interested in words.
  const extractedWords = Blocks.filter((b) => b.BlockType === "WORD").map(
    (b) => b.Text,
  );
```

```
  return extractedWords.join(" ");
};
```

```
import { PollyClient, SynthesizeSpeechCommand } from "@aws-sdk/client-polly";
import { S3Client } from "@aws-sdk/client-s3";
import { Upload } from "@aws-sdk/lib-storage";

/**
 * Synthesize an audio file from text.
 *
 * @param {{ bucket: string, translated_text: string, object: string}}
 sourceDestinationConfig
 */
export const handler = async (sourceDestinationConfig) => {
  const pollyClient = new PollyClient({});

  const synthesizeSpeechCommand = new SynthesizeSpeechCommand({
    Engine: "neural",
    Text: sourceDestinationConfig.translated_text,
    VoiceId: "Ruth",
    OutputFormat: "mp3",
  });

  const { AudioStream } = await pollyClient.send(synthesizeSpeechCommand);

  const audioKey = `${sourceDestinationConfig.object}.mp3`;

  // Store the audio file in S3.
  const s3Client = new S3Client();
  const upload = new Upload({
    client: s3Client,
    params: {
      Bucket: sourceDestinationConfig.bucket,
      Key: audioKey,
      Body: AudioStream,
      ContentType: "audio/mp3",
    },
  });

  await upload.done();
  return audioKey;
};
```

```
import {
  TranslateClient,
  TranslateTextCommand,
} from "@aws-sdk/client-translate";

/**
 * Translate the extracted text to English.
 *
 * @param {{ extracted_text: string, source_language_code: string}}
 textAndSourceLanguage
 */
export const handler = async (textAndSourceLanguage) => {
  const translateClient = new TranslateClient({});

  const translateCommand = new TranslateTextCommand({
    SourceLanguageCode: textAndSourceLanguage.source_language_code,
    TargetLanguageCode: "en",
    Text: textAndSourceLanguage.extracted_text,
  });

  const { TranslatedText } = await translateClient.send(translateCommand);

  return { translated_text: TranslatedText };
};
```

Serviços usados neste exemplo

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Ruby

SDK para Ruby

Esta aplicação de exemplo analisa e armazena cartões de feedback de clientes. Especificamente, ela atende à necessidade de um hotel fictício na cidade de Nova York. O hotel recebe feedback dos hóspedes em vários idiomas na forma de cartões de comentários

físicos. Esse feedback é enviado para a aplicação por meio de um cliente web. Depois de fazer upload da imagem de um cartão de comentário, ocorrem as seguintes etapas:

- O texto é extraído da imagem usando o Amazon Textract.
- O Amazon Comprehend determina o sentimento do texto extraído e o idioma.
- O texto extraído é traduzido para o inglês com o Amazon Translate.
- O Amazon Polly sintetiza um arquivo de áudio do texto extraído.

A aplicação completa pode ser implantada com o AWS CDK. Para obter o código-fonte e as instruções de implantação, consulte o projeto em GitHub.

Serviços usados neste exemplo

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Introdução à síntese de conversão de texto em voz

O exemplo de código a seguir mostra como:

- Limpar os recursos

Bash

AWS CLI com script Bash

> **Note**
>
> Tem mais sobre GitHub. Encontre o exemplo completo e saiba como configurar e executar no repositório Sample developer tutorials.

```
#!/bin/bash

# Amazon Polly Getting Started Script
# This script demonstrates how to use Amazon Polly with the AWS CLI

set -euo pipefail

# Set up logging
LOG_FILE="polly-tutorial.log"
SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
WORK_DIR=$(mktemp -d)
trap 'cleanup_temp' EXIT

cleanup_temp() {
    rm -rf "$WORK_DIR"
}

echo "Starting Amazon Polly tutorial at $(date)" > "$LOG_FILE"

# Function to log commands and their output
log_cmd() {
    echo "Running: $1" | tee -a "$LOG_FILE"
    # Use bash array to safely handle arguments
    bash -c "$1" 2>&1 | tee -a "$LOG_FILE" || return $?
}

# Function to check for errors
check_error() {
    if echo "$1" | grep -iq "error"; then
        echo "ERROR detected in output. Exiting script." | tee -a "$LOG_FILE"
        echo "$1" | tee -a "$LOG_FILE"
        return 1
    fi
    return 0
}

# Function to handle errors and cleanup
handle_error() {
    local line_number=$1
    echo "Error occurred at line $line_number. Attempting cleanup..." | tee -a
 "$LOG_FILE"
    cleanup
    exit 1
```

```
}

# Function to clean up resources
cleanup() {
    echo "" | tee -a "$LOG_FILE"
    echo "============================================================" | tee -a
 "$LOG_FILE"
    echo "CLEANUP PROCESS" | tee -a "$LOG_FILE"
    echo "============================================================" | tee -a
 "$LOG_FILE"

    # Delete lexicon if it exists
    if [[ -n "${LEXICON_NAME:-}" ]]; then
        echo "Deleting lexicon: $LEXICON_NAME" | tee -a "$LOG_FILE"
        if aws polly delete-lexicon --name "$LEXICON_NAME" 2>&1 | tee -a
 "$LOG_FILE"; then
            echo "Lexicon deleted successfully." | tee -a "$LOG_FILE"
        else
            echo "Warning: Failed to delete lexicon." | tee -a "$LOG_FILE"
        fi
    fi

    # Remove audio files
    for file in output.mp3 ssml-output.mp3 lexicon-output.mp3 example.pls; do
        if [[ -f "$file" ]]; then
            rm -f "$file"
            echo "Removed $file" | tee -a "$LOG_FILE"
        fi
    done

    echo "Cleanup complete." | tee -a "$LOG_FILE"
}

# Trap errors with line number
trap 'handle_error ${LINENO}' ERR

# Verify AWS CLI is available
if ! command -v aws &> /dev/null; then
    echo "AWS CLI is not installed. Please install it first." | tee -a
 "$LOG_FILE"
    exit 1
fi

# Verify AWS credentials are configured
```

```
if ! aws sts get-caller-identity &> /dev/null; then
    echo "AWS credentials are not configured. Please configure them first." | tee
 -a "$LOG_FILE"
    exit 1
fi

# Step 1: Verify Amazon Polly is available
echo "Step 1: Verifying Amazon Polly availability" | tee -a "$LOG_FILE"
if aws polly describe-voices --query 'Voices[0].Name' --output text &> /dev/null;
 then
    echo "Amazon Polly is available. Proceeding with tutorial." | tee -a
 "$LOG_FILE"
else
    echo "Amazon Polly is not available in your AWS CLI installation or region."
 | tee -a "$LOG_FILE"
    echo "Please update your AWS CLI to the latest version or check your region."
 | tee -a "$LOG_FILE"
    exit 1
fi

# Step 2: List available voices
echo "" | tee -a "$LOG_FILE"
echo "Step 2: Listing available voices" | tee -a "$LOG_FILE"
log_cmd "aws polly describe-voices --language-code en-US --output text --query
 'Voices[0:3].[Id, LanguageCode, Gender]'" || true

# Step 3: Basic text-to-speech conversion
echo "" | tee -a "$LOG_FILE"
echo "Step 3: Converting text to speech" | tee -a "$LOG_FILE"
OUTPUT_FILE="${WORK_DIR}/output.mp3"
POLLY_TEXT="Hello, welcome to Amazon Polly. This is a sample text to speech
 conversion."
log_cmd "aws polly synthesize-speech --output-format mp3 --voice-id Joanna --text
 '$POLLY_TEXT' '$OUTPUT_FILE'" || true

if [[ -f "$OUTPUT_FILE" ]]; then
    echo "Successfully created output.mp3 file." | tee -a "$LOG_FILE"
    echo "You can play this file with your preferred audio player." | tee -a
 "$LOG_FILE"
    cp "$OUTPUT_FILE" output.mp3
else
    echo "Failed to create output.mp3 file." | tee -a "$LOG_FILE"
    exit 1
fi
```

```
# Step 4: Using SSML for enhanced speech
echo "" | tee -a "$LOG_FILE"
echo "Step 4: Using SSML for enhanced speech" | tee -a "$LOG_FILE"
SSML_OUTPUT="${WORK_DIR}/ssml-output.mp3"
SSML_TEXT='<speak>Hello! <break time="1s"/> This is a sample of <emphasis>SSML
 enhanced speech</emphasis>.</speak>'
log_cmd "aws polly synthesize-speech --output-format mp3 --voice-id Matthew --
text-type ssml --text '$SSML_TEXT' '$SSML_OUTPUT'" || true

if [[ -f "$SSML_OUTPUT" ]]; then
    echo "Successfully created ssml-output.mp3 file." | tee -a "$LOG_FILE"
    echo "You can play this file with your preferred audio player." | tee -a
 "$LOG_FILE"
    cp "$SSML_OUTPUT" ssml-output.mp3
else
    echo "Failed to create ssml-output.mp3 file." | tee -a "$LOG_FILE"
    exit 1
fi

# Step 5: Working with lexicons
echo "" | tee -a "$LOG_FILE"
echo "Step 5: Working with lexicons" | tee -a "$LOG_FILE"

# Generate a random identifier for the lexicon (max 20 chars, alphanumeric only)
LEXICON_NAME="example$(openssl rand -hex 6 | cut -c 1-10)"
echo "Using lexicon name: $LEXICON_NAME" | tee -a "$LOG_FILE"

# Create a lexicon file
echo "Creating lexicon file..." | tee -a "$LOG_FILE"
LEXICON_FILE="${WORK_DIR}/example.pls"
cat > "$LEXICON_FILE" << 'EOF'
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
      xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
        http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20071212/pls.xsd"
      alphabet="ipa"
      xml:lang="en-US">
  <lexeme>
    <grapheme>AWS</grapheme>
    <alias>Amazon Web Services</alias>
  </lexeme>
```

```
</lexicon>
EOF

# Upload the lexicon
echo "Uploading lexicon..." | tee -a "$LOG_FILE"
LEXICON_ARN=$(aws polly put-lexicon --name "$LEXICON_NAME" --content
 file://"$LEXICON_FILE" --query 'LexiconArn' --output text 2>&1) || true
if [[ -n "$LEXICON_ARN" && "$LEXICON_ARN" != "" && ! "$LEXICON_ARN" =~ error ]];
 then
    echo "Lexicon uploaded with ARN: $LEXICON_ARN" | tee -a "$LOG_FILE"
    aws polly tag-resource --resource-arn "$LEXICON_ARN" --tags
 Key=project,Value=doc-smith Key=tutorial,Value=amazon-polly-gs 2>&1 | tee -a
 "$LOG_FILE" || true
else
    echo "Lexicon uploaded." | tee -a "$LOG_FILE"
fi

# List available lexicons
echo "Listing available lexicons..." | tee -a "$LOG_FILE"
log_cmd "aws polly list-lexicons --output text --query 'Lexicons[*].[Name]'" ||
 true

# Get details about the lexicon
echo "Getting details about the lexicon..." | tee -a "$LOG_FILE"
log_cmd "aws polly get-lexicon --name '$LEXICON_NAME' --output text --query
 'Lexicon.Name'" || true

# Use the lexicon when synthesizing speech
echo "Using the lexicon for speech synthesis..." | tee -a "$LOG_FILE"
LEXICON_OUTPUT="${WORK_DIR}/lexicon-output.mp3"
LEXICON_TEXT="I work with AWS every day."
log_cmd "aws polly synthesize-speech --output-format mp3 --voice-id Joanna --
lexicon-names '$LEXICON_NAME' --text '$LEXICON_TEXT' '$LEXICON_OUTPUT'" || true

if [[ -f "$LEXICON_OUTPUT" ]]; then
    echo "Successfully created lexicon-output.mp3 file." | tee -a "$LOG_FILE"
    echo "You can play this file with your preferred audio player." | tee -a
 "$LOG_FILE"
    cp "$LEXICON_OUTPUT" lexicon-output.mp3
else
    echo "Failed to create lexicon-output.mp3 file." | tee -a "$LOG_FILE"
    exit 1
fi
```

```
# Summary of created resources
echo "" | tee -a "$LOG_FILE"
echo "============================================================" | tee -a
 "$LOG_FILE"
echo "TUTORIAL SUMMARY" | tee -a "$LOG_FILE"
echo "============================================================" | tee -a
 "$LOG_FILE"
echo "Created resources:" | tee -a "$LOG_FILE"
echo "1. Lexicon: $LEXICON_NAME" | tee -a "$LOG_FILE"
echo "2. Audio files:" | tee -a "$LOG_FILE"
echo "   - output.mp3" | tee -a "$LOG_FILE"
echo "   - ssml-output.mp3" | tee -a "$LOG_FILE"
echo "   - lexicon-output.mp3" | tee -a "$LOG_FILE"
echo "" | tee -a "$LOG_FILE"

# Cleanup with auto-confirmation
echo "" | tee -a "$LOG_FILE"
echo "============================================================" | tee -a
 "$LOG_FILE"
echo "CLEANUP CONFIRMATION" | tee -a "$LOG_FILE"
echo "============================================================" | tee -a
 "$LOG_FILE"
echo "Cleaning up all created resources..." | tee -a "$LOG_FILE"
cleanup

echo "" | tee -a "$LOG_FILE"
echo "Tutorial completed successfully!" | tee -a "$LOG_FILE"
echo "Log file: $LOG_FILE" | tee -a "$LOG_FILE"
```

- Consulte detalhes da API nos tópicos a seguir na Referência de comandos da AWS CLI .
  - DeleteLexicon
  - DescribeVoices
  - GetLexicon
  - Help
  - ListLexicons
  - PutLexicon
  - SynthesizeSpeech

Para obter uma lista completa dos guias do desenvolvedor do AWS SDK e exemplos de código, consulteUsando o Amazon Polly com um SDK AWS. Este tópico também inclui informações sobre como começar e detalhes sobre versões anteriores do SDK.

Segurança no Amazon Polly

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Third-party auditores testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade que se aplicam ao Amazon Polly, consulte AWS Serviços no escopo do programa de conformidade AWS Serviços no escopo do programa .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade dos dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon Polly. Os tópicos a seguir mostram como configurar o Amazon Polly para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos do Amazon Polly.

Tópicos

- Proteção de dados no Amazon Polly
- Gerenciamento de identidade e acesso no Amazon Polly
- Registrar em log e monitorar no Amazon Polly
- Validação de conformidade para o Amazon Polly
- Resiliência no Amazon Polly
- Segurança da infraestrutura no Amazon Polly
- Práticas recomendadas de segurança para o Amazon Polly
- Usar o Amazon Polly com endpoints da VPC de interface

Proteção de dados no Amazon Polly

O Amazon Polly está em conformidade com o modelo de responsabilidade AWS compartilhada, que inclui regulamentações e diretrizes para proteção de dados. AWS é responsável por proteger a infraestrutura global que executa todos os AWS serviços. AWS mantém o controle sobre os dados hospedados nessa infraestrutura, incluindo os controles de configuração de segurança para lidar com o conteúdo do cliente e os dados pessoais. AWS clientes e parceiros da APN, atuando como controladores ou processadores de dados, são responsáveis por todos os dados pessoais que colocam na AWS nuvem.

Para fins de proteção de dados, recomendamos que você proteja as credenciais da AWS conta e configure usuários individuais com AWS Identity and Access Management (IAM), para que cada usuário receba somente as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- Use soluções AWS de criptografia, juntamente com todos os controles de segurança padrão nos AWS serviços.

É altamente recomendável que você nunca coloque informações de identificação confidenciais, como números de conta dos seus clientes, em campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com o Amazon Polly ou outros AWS serviços usando o console, a API ou AWS os AWS CLI SDKs. Todos os dados que você insere no Amazon Polly ou em outros serviços podem ser selecionados para inclusão em logs de diagnóstico. Ao fornecer um URL para um servidor externo, não inclua informações de credenciais no URL para validar a solicitação a esse servidor.

Para mais informações sobre proteção de dados, consulte a publicação Modelo de responsabilidade compartilhada da AWS e do GDPR no Blog de segurança da AWS .

Criptografia em repouso

A saída da síntese de voz do Amazon Polly pode ser salva em seu próprio sistema. Também é possível chamar o Amazon Polly, criptografar o arquivo com qualquer chave de criptografia de sua escolha e armazená-lo no Amazon Simple Storage Service (Amazon S3) ou em outro armazenamento seguro. A operação the section called “SynthesizeSpeech” do Amazon Polly é sem

estado e não está associada a uma identidade do cliente. Não é possível recuperá-la do Amazon Polly posteriormente.

Criptografia em trânsito

Todos os envios de texto são protegidos por TLS durante o trânsito. O Amazon Polly não retém o conteúdo dos envios de texto.

Privacidade do tráfego entre redes

O acesso ao Amazon Polly é feito por meio do AWS console, da CLI ou dos SDKs. As comunicações utilizam criptografia de sessão TLS (Transport Layer Security) para confidencialidade e assinaturas digitais para autenticação e integridade.

Gerenciamento de identidade e acesso no Amazon Polly

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado e autorizado para utilizar os recursos do Amazon Polly. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- Público
- Autenticação com identidades
- Gerenciar o acesso usando políticas
- Como o Amazon Polly funciona com o IAM
- Identity-based exemplos de políticas para Amazon Polly
- Permissões de API do Amazon Polly: referência de ações, permissões e recursos
- Solução de problemas de identidade e acesso da Amazon Polly

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte Solução de problemas de identidade e acesso da Amazon Polly).

- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte Como o Amazon Polly funciona com o IAM)
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte Identity-based exemplos de políticas para Amazon Polly)

Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte Como fazer login em sua Conta da AWS no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte AWS Signature Version 4 para solicitações de API no Guia do usuário do IAM.

Conta da AWS usuário-raiz

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em Tarefas que exigem credenciais de usuário-raiz no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários humanos usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para saber mais, consulte O que é o IAM Identity Center? no Guia do usuário do Centro de Identidade do AWS IAM .

Usuários e grupos do IAM

Um usuário do IAM é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias no Guia do usuário do IAM.

Um grupo do IAM especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte Casos de uso de usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Uma perfil do IAM é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função mudando de um usuário para uma função do IAM (console) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte Métodos para assumir um perfil no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Identity-based políticas

Identity-based políticas são documentos de políticas de permissões JSON que você anexa a uma identidade (usuário, grupo ou função). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Identity-based as políticas podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Resource-based políticas

Resource-based políticas são documentos de política JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário especificar uma entidade principal em uma política baseada em recursos.

Resource-based políticas são políticas embutidas que estão localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de Controle de Serviços (SCPs): as SCPs especificam o número máximo de permissões para uma organização ou uma unidade organizacional no AWS Organizations. Para saber mais, consulte Políticas de controle de serviço no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs): definem o número máximo de permissões disponíveis para recursos em suas contas. Consulte mais informações em Resource control policies (RCPs) no Guia do usuário do AWS Organizations .

- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como o Amazon Polly funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao Amazon Polly, entenda que atributos do IAM estão disponíveis para uso com o Amazon Polly.

Recursos do IAM que você pode usar com o Amazon Polly

Recurso do IAM	Suporte ao Amazon Polly
Identity-based políticas	Sim
Resource-based políticas	Não
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Não
ACLs	Não
ABAC (tags em políticas)	Não
Credenciais temporárias	Sim
Sessões de acesso direto (FAS) para o Amazon Polly	Sim
Perfis de serviço	Não

Recurso do IAM	Suporte ao Amazon Polly
Service-linked funções	Não

Para ter uma visão de alto nível de como o Amazon Polly e AWS outros serviços funcionam com a maioria dos recursos do IAM, AWS consulte os serviços que funcionam com o IAM no Guia do usuário do IAM.

Identity-based políticas para Amazon Polly

Compatível com políticas baseadas em identidade: sim

Identity-based políticas são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como um usuário do IAM, um grupo de usuários ou uma função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais atributos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de política JSON do IAM no Guia do usuário do IAM.

Identity-based exemplos de políticas para Amazon Polly

Para visualizar exemplos de políticas baseadas em identidade do Amazon Polly, consulte Identity-based exemplos de políticas para Amazon Polly.

Resource-based políticas dentro do Amazon Polly

Compatibilidade com políticas baseadas em recursos: não

Resource-based políticas são documentos de política JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificado pode executar nesse atributo e em que condições. É necessário especificar uma

entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, é possível especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Ações das políticas para o Amazon Polly

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de ações das políticas do Amazon Polly, consulte Ações definidas pelo Amazon Polly na Referência de autorização do serviço.

As ações das política no Amazon Polly usam o seguinte prefixo antes da ação:

```
polly
```

Para especificar várias ações em uma única declaração, separe-as com vírgulas.

```
"Action": [
      "polly:action1",
      "polly:action2"
         ]
```

Para visualizar exemplos de políticas baseadas em identidade do Amazon Polly, consulte Identity-based exemplos de políticas para Amazon Polly.

Recursos de políticas para o Amazon Polly

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de recursos do Amazon Polly e seus ARNs, consulte Recursos definidos pelo Amazon Polly na Referência de autorização do serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo Amazon Polly.

Para visualizar exemplos de políticas baseadas em identidade do Amazon Polly, consulte Identity-based exemplos de políticas para Amazon Polly.

Chaves de condição de política para o Amazon Polly

Compatível com chaves de condição de política específicas de serviço: não

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia do usuário do IAM.

Para ver uma lista de chaves de condição do Amazon Polly, consulte Chaves de condição do Amazon Polly na Referência de autorização do serviço. Para saber com quais ações e recursos é possível usar uma chave de condição, consulte Ações definidas pelo Amazon Polly.

Para visualizar exemplos de políticas baseadas em identidade do Amazon Polly, consulte Identity-based exemplos de políticas para Amazon Polly.

ACLs no Amazon Polly

Compatível com ACLs: não

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou perfis da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com o Amazon Polly

Oferece compatibilidade com ABAC (tags em políticas): não

Attribute-based controle de acesso (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados de tags. Você pode anexar tags a entidades e AWS recursos do IAM e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para saber mais sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso por atributo (ABAC) no Guia do usuário do IAM.

Usar credenciais temporárias com o Amazon Polly

Compatível com credenciais temporárias: sim

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para ter mais informações, consulte Credenciais de segurança temporárias no IAM e Serviços da Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Cross-service sessões de acesso direto (FAS) para Amazon Polly

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer

solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

Perfis de serviço para o Amazon Polly

Compatível com perfis de serviço: não

O perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para saber mais, consulte Criar um perfil para delegar permissões a um AWS service (Serviço da AWS) no Guia do Usuário do IAM.

> **Warning**
>
> A alteração das permissões de um perfil de serviço pode interromper a funcionalidade do Amazon Polly. Edite perfis de serviço somente quando o Amazon Polly fornecer orientação para isso.

Service-linked funções para Amazon Polly

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir a função de realizar uma ação em seu nome. Service-linked as funções aparecem no seu Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte Serviços da AWS que funcionam com o IAM. Encontre um serviço na tabela que inclua um `Yes` na coluna de Service-linked função. Escolha o link Sim para visualizar a documentação do perfil vinculado ao serviço desse serviço.

Perfis do IAM no Amazon Polly

Você pode associar uma política de permissões baseada em identidade a um perfil do IAM para conceder permissões entre contas. Por exemplo, o administrador da conta A pode criar uma função para conceder permissões entre contas a outra AWS conta (por exemplo, conta B) ou a um AWS serviço da seguinte forma:

1. Um administrador da Conta A cria um perfil do IAM e anexa uma política de permissões ao perfil que concede permissões em recursos da Conta A.
2. Um administrador da conta A anexa uma política de confiança ao perfil identificando a conta B como a entidade principal, que pode assumir a função.
3. O administrador da conta B pode então delegar permissões para assumir a função a qualquer usuário na conta B. Isso permite que os usuários da conta B criem ou acessem recursos na conta A. O principal na política de confiança também pode ser um diretor de AWS serviço se você quiser conceder a um AWS serviço permissões para assumir a função.

Para obter mais informações sobre o uso do IAM para delegar permissões, consulte Gerenciamento de acesso no Guia do usuário do IAM.

Veja a seguir um exemplo de política que concede permissões para colocar e obter léxicos, bem como para listar os léxicos disponíveis no momento.

O Amazon Polly oferece suporte a Identity-based políticas para ações em nível de recursos. Em alguns casos, o recurso pode ser limitado por um ARN. Isso é válido para as operações `SynthesizeSpeech`, `StartSpeechSynthesisTask`, `PutLexicon`, `GetLexicon` e `DeleteLexicon`. Nesses casos, o valor `Resource` é indicado pelo ARN. Por exemplo: `arn:aws:polly:`*`us-east-2`*`:`*`account-id`*`:lexicon/*` como o valor de `Resource` especifica as permissões em todos os léxicos próprios dentro da região `us-east-2`.

No entanto, nem todas as operações usam ARNs. Esse é o caso das operações `DescribeVoices`, `ListLexicons`, `GetSpeechSynthesisTasks`, e `ListSpeechSynthesisTasks`.

Para obter mais informações sobre usuários, grupos, funções e permissões, consulte Identidades (usuários, grupos e funções) no Guia do usuário do IAM.

Identity-based exemplos de políticas para Amazon Polly

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do Amazon Polly. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte Criar políticas do IAM (console) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pelo Amazon Polly, incluindo o formato dos ARNs para cada um dos tipos de recursos, consulte Ações, recursos e chaves de condição para o Amazon Polly na Referência de autorização do serviço.

Tópicos

- Práticas recomendadas de política
- Usar o console do Amazon Polly
- Permitir que os usuários visualizem suas próprias permissões
- AWS políticas gerenciadas (predefinidas) para o Amazon Polly
- Atualizações do Amazon Polly para AWS políticas gerenciadas
- Customer-managed exemplos de políticas

Práticas recomendadas de política

Identity-based as políticas determinam se alguém pode criar, acessar ou excluir recursos do Amazon Polly em sua conta. Essas ações podem incorrer em custos para sua Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation.

Para saber mais, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.

- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usar o console do Amazon Polly

Para acessar o console da Amazon Polly, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos do Amazon Polly em seu. Conta da AWS Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Para garantir que usuários e funções ainda possam usar o console do Amazon Polly, anexe também o Amazon `ConsoleAccess` Polly `ReadOnly` AWS ou a política gerenciada às entidades. Para saber mais, consulte Adicionar permissões a um usuário no Guia do usuário do IAM.

Para usar o console do Amazon Polly, conceda permissões para todas as APIs do Amazon Polly. Não há permissões adicionais necessárias. Para obter a funcionalidade completa do console, é possível usar a seguinte política:

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS políticas gerenciadas (predefinidas) para o Amazon Polly

AWS aborda muitos casos de uso comuns fornecendo políticas autônomas do IAM que são criadas e administradas pela AWS. Essas políticas AWS gerenciadas concedem as permissões necessárias para casos de uso comuns, para que você possa evitar a necessidade de investigar quais permissões são necessárias. Para obter mais informações, consulte Políticas gerenciadas pela AWS no Guia do usuário do IAM.

As seguintes políticas AWS gerenciadas, que você pode associar aos usuários em sua conta, são específicas do Amazon Polly:

- AmazonPollyReadOnlyAccess— Concede acesso somente para leitura aos recursos, permite listar léxicos, buscar léxicos, listar vozes disponíveis e sintetizar fala (inclusive aplicar léxicos à fala sintetizada).

  Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AmazonPollyReadOnlyAccess no AWS Managed Policy Reference Guide.
- AmazonPollyFullAccess— Concede acesso total aos recursos e a todas as operações suportadas.

  Para visualizar mais detalhes sobre a política, inclusive a versão mais recente do documento de política JSON, consulte AmazonPollyFullAccess no AWS Managed Policy Reference Guide.

> **Note**
>
> É possível analisar essas políticas de permissões fazendo login no console do IAM e pesquisando políticas específicas.

Você também pode criar as próprias políticas do IAM personalizadas a fim de conceder permissões para ações e recursos do Amazon Polly. Você pode anexar essas políticas personalizadas a usuários ou grupos do IAM que exijam essas permissões.

Atualizações do Amazon Polly para AWS políticas gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do Amazon Polly desde que esse serviço começou a monitorar essas mudanças. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página Histórico de documentos do Amazon Polly.

Atualizações do Amazon Polly para AWS políticas gerenciadas

Política	Alteração	Data
AmazonPollyReadOnlyAccess	Política gerenciada atualizada - Foi adicionada a permissão para oferecer suporte `polly:StartSpeechSynthesisStream` aos recursos de síntese de voz de streaming bidirecional.	19 de março de 2026

Customer-managed exemplos de políticas

Nesta seção, você pode encontrar exemplos de políticas de usuário que concedem permissões para várias ações do Amazon Polly. Essas políticas funcionam quando você usa AWS SDKs ou o. AWS CLI Quando você estiver usando o console, conceda permissões para todas as APIs do Amazon Polly.

> **Note**
>
> Todos os exemplos usam a região us-east-2 e contêm IDs de conta fictícios.

Exemplos

- Exemplo 1: permitir todas as ações do Amazon Polly
- Exemplo 2: Permitir todas as ações do Amazon Polly, exceto DeleteLexicon
- Exemplo 3: Permitir DeleteLexicon
- Exemplo 4: permitir a exclusão de léxico de uma região especificada
- Exemplo 5: Permitir DeleteLexicon um léxico especificado

Exemplo 1: permitir todas as ações do Amazon Polly

Depois de se cadastrar (consulte Conceitos básicos do Amazon Polly), crie um usuário administrador para gerenciar a conta, incluindo a criação de usuários e o gerenciamento das permissões deles.

Você pode criar um usuário que tenha permissões para todas as ações do Amazon Polly. Pense nesse usuário como um administrador específico do serviço para trabalhar com o Amazon Polly. Você pode anexar a política de permissões a seguir para este usuário.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [{
      "Sid": "AllowAllPollyActions",
      "Effect": "Allow",
      "Action": [
         "polly:*"],
      "Resource": "*"
      }
   ]
}
```

Exemplo 2: Permitir todas as ações do Amazon Polly, exceto DeleteLexicon

A política de permissões a seguir concede ao usuário permissões para executar todas as ações, exceto `DeleteLexicon`, com as permissões para excluir explicitamente negadas em todas as regiões.

Exemplo 3: Permitir DeleteLexicon

A política de permissões a seguir concede ao usuário permissões para excluir qualquer léxico que você possua, independentemente do projeto ou da região em que está localizado.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
      "Sid": "AllowDeleteLexicon",
      "Effect": "Allow",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "*"
      }
   ]
}
```

Exemplo 4: permitir a exclusão de léxico de uma região especificada

A política de permissões a seguir concede ao usuário permissões para excluir qualquer léxico em qualquer projeto que você possui e que esteja localizado em uma única região (neste caso, us-east-2).

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
      "Sid": "AllowDeleteSpecifiedRegion",
      "Effect": "Allow",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "arn:aws:polly:us-east-2:111122223333:lexicon/*"
      }
   ]
}
```

Exemplo 5: Permitir DeleteLexicon um léxico especificado

A política de permissões a seguir concede as permissões de usuário para excluir um léxico específico que você possui (neste caso, myLexicon) em uma região específica (neste caso, us-east-2).

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [{
      "Sid": "AllowDeleteForSpecifiedLexicon",
      "Effect": "Allow",
      "Action": [
         "polly:DeleteLexicon"],
      "Resource": "arn:aws:polly:us-east-2:111122223333:lexicon/myLexicon"
      }
   ]
}
```

Permissões de API do Amazon Polly: referência de ações, permissões e recursos

Ao configurar uma política de permissões para anexar a uma identidade do IAM (políticas baseadas em identidade), você pode usar a lista a seguir como referência. A inclui cada operação da API Amazon Polly, as ações correspondentes para as quais você pode conceder permissões para realizar a ação e o AWS recurso para o qual você pode conceder as permissões. Você especifica as ações no campo `Action` da política e o valor do recurso no campo `Resource` da política.

Você pode usar chaves AWS de condição abrangentes em suas políticas do Amazon Polly para expressar condições. Para obter uma lista completa AWS de chaves gerais, consulte as chaves disponíveis no Guia do usuário do IAM.

> **Note**
>
> Para especificar uma ação, use o prefixo `polly` seguido do nome da operação da API (por exemplo, `polly:GetLexicon`).

O Amazon Polly oferece suporte a Identity-based políticas para ações em nível de recursos. Portanto, o valor `Resource` é indicado pelo ARN. Por exemplo: `arn:aws:polly:`*`us-east-2`*`:`*`account-id`*`:lexicon/*` como o valor de `Resource` especifica as permissões em todos os léxicos próprios dentro da região `us-east-2`.

Como o Amazon Polly não é compatível com permissões para ações no nível de recurso, a maioria das políticas especifica um asterisco (*) como valor de `Resource`. No entanto, se for necessário limitar as permissões para uma região específica, esse caractere curinga será substituído pelo ARN adequado: `arn:aws:polly:`*`region`*`:`*`account-id`*`:lexicon/*`.

API do Amazon Polly e permissões necessárias para ações

Operação da API: DeleteLexicon

Permissões obrigatórias (ação de API): `polly:DeleteLexicon`

Recursos: `arn:aws:polly:`*`region`*`:`*`account-id`*`:lexicon/`*`LexiconName`*

Operação da API: DescribeVoices

Permissões obrigatórias (ação de API): `polly:DescribeVoices`

Recursos: `arn:aws:polly:region:account-id:lexicon/voice-name`

Operação da API: GetLexicon

Permissões obrigatórias (ação de API): `polly:GetLexicon`

Recursos: `arn:aws:polly:region:account-id:lexicon/voice-name`

Operação da API: ListLexicons

Permissões obrigatórias (ação de API): `polly:ListLexicons`

Recursos: `arn:aws:polly:region:account-id:lexicon/*`

Operação da API: PutLexicon

Permissões obrigatórias (ação de API): `polly:ListLexicons`

Recursos: *

Operação da API: SynthesizeSpeech

Permissões obrigatórias (ação de API): `polly:SynthesizeSpeech`

Recursos: *

Solução de problemas de identidade e acesso da Amazon Polly

Use as informações a seguir para ajudar a diagnosticar e corrigir problemas comuns que você possa encontrar ao trabalhar com o Amazon Polly e o IAM.

Tópicos

- Não tenho autorização para executar uma ação no Amazon Polly
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha Conta da AWS para acessar meus recursos do Amazon Polly

Não tenho autorização para executar uma ação no Amazon Polly

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, suas políticas deverão ser atualizadas para permitir que você realize a ação.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para visualizar detalhes sobre um atributo *`my-example-widget`* fictício, mas não tem as permissões `polly:`*`GetWidget`* fictícias.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 polly:GetWidget on resource: my-example-widget
```

Nesse caso, a política do usuário `mateojackson` deve ser atualizada para permitir o acesso ao recurso *`my-example-widget`* usando a ação `polly:`*`GetWidget`*.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Se receber uma mensagem de erro informando que você não tem autorização para executar a ação `iam:PassRole`, suas políticas devem ser atualizadas para permitir a transmissão de um perfil ao Amazon Polly.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O erro exemplificado a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no Amazon Polly. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha Conta da AWS para acessar meus recursos do Amazon Polly

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para serviços que oferecem compatibilidade com políticas baseadas em recursos ou listas de controle de acesso (ACLs), é possível usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se o Amazon Polly é compatível com esses recursos, consulte Como o Amazon Polly funciona com o IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como fornecer acesso Contas da AWS a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Registrar em log e monitorar no Amazon Polly

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e o desempenho dos aplicativos do Amazon Polly. Para monitorar as chamadas da API Amazon Polly, você pode usar. AWS CloudTrail Para monitorar o status de seus trabalhos, use o Amazon CloudWatch Logs.

- Amazon CloudWatch Alarms — Usando CloudWatch alarmes, você observa uma única métrica durante um período de tempo especificado por você. Se a métrica exceder um determinado limite, uma notificação será enviada para um tópico ou AWS Auto Scaling política do Amazon Simple Notification Service. CloudWatchos alarmes não invocam ações quando uma métrica está em um estado específico. O estado deve ter sido alterado e mantido por uma quantidade especificada de períodos. Para obter mais informações, consulte Integrar o CloudWatch ao Amazon Polly.

- CloudTrail logs — CloudTrail fornece um registro das ações realizadas por um usuário, função ou AWS serviço no Amazon Polly. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita ao Amazon Polly. Você também pode determinar o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita, e detalhes adicionais. Para obter mais informações, consulte Registrar chamadas de API do Amazon Polly em log com o AWS CloudTrail.

Validação de conformidade para o Amazon Polly

Third-party os auditores avaliam a segurança e a conformidade do Amazon Polly como parte de AWS vários programas de conformidade. Isso inclui SOC, PCI, FedRAMP, HIPAA e outros.

Para obter uma lista de AWS serviços no escopo de programas de conformidade específicos, consulte AWS Serviços no escopo do programa de conformidade AWS . Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para saber mais, consulte Baixar relatórios no AWS Artifact.

Sua responsabilidade de conformidade ao usar o Amazon Polly é determinada pela confidencialidade dos seus dados, pelos objetivos de conformidade da sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Guias de início rápido de segurança e compatibilidade: esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em compatibilidade e segurança na AWS.
- Documento técnico sobre arquitetura para segurança e conformidade com a HIPAA — Este whitepaper descreve como as empresas podem usar para criar aplicativos. AWS HIPAA-compliant
- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- Avaliação de recursos com regras no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub CSPM— Esse AWS serviço fornece uma visão abrangente do seu estado de segurança interno, AWS que ajuda você a verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

Resiliência no Amazon Polly

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte Infraestrutura AWS global.

Segurança da infraestrutura no Amazon Polly

Como um serviço gerenciado, o Amazon Polly é protegido pelos procedimentos AWS globais de segurança de rede descritos no whitepaper Amazon Web Services: Visão geral dos processos de segurança.

Você usa chamadas de API AWS publicadas para acessar o Amazon Polly pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.0 ou posterior. Recomendamos usar o TLS 1.2 ou posterior. Os clientes também devem oferecer suporte a pacotes de criptografia com sigilo direto perfeito (PFS), como Ephemeral (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Diffie-Hellman A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Práticas recomendadas de segurança para o Amazon Polly

Sua confiança, privacidade e a segurança do seu conteúdo são nossas maiores prioridades. Implementamos controles técnicos e físicos que são responsáveis e avançados para impedir o acesso não autorizado ao seu conteúdo, bem como a sua divulgação, e garantir que o nosso uso esteja de acordo com os compromissos que firmamos com você. Para obter mais informações, consulte AWS Perguntas frequentes sobre a privacidade de dados.

O Amazon Polly não retém o conteúdo dos envios de texto.

Para ter uma visão ampla da AWS segurança, incluindo conformidade, testes de penetração, boletins e recursos, visite o site AWS Cloud Security.

Usar o Amazon Polly com endpoints da VPC de interface

Se você usa a Amazon Virtual Private Cloud (Amazon VPC) para hospedar seus AWS recursos, você pode estabelecer uma conexão privada entre sua VPC e o Amazon Polly. Você pode usar essa conexão para sintetizar a fala com o Amazon Polly sem passar pela internet pública.

O Amazon VPC é um AWS serviço que você pode usar para lançar AWS recursos em uma rede virtual que você define. Com a VPC, você tem controle sobre as configurações de rede, como o intervalo de endereços IP, sub-redes, tabelas de rotas e gateways de rede. Para conectar a VPC ao Amazon Polly, você define um Endpoint da VPC de interface. Esse tipo de endpoint permite que você conecte a VPC aos serviços da Serviços da AWS. O endpoint fornece uma conectividade confiável e escalável ao Amazon Polly sem a necessidade de um gateway da Internet, de uma instância de conversão de endereço de rede (NAT) ou de uma conexão VPN. Para obter mais informações, consulte O que é a Amazon VPC? no Guia do usuário da Amazon VPC.

Os endpoints VPC da Interface são alimentados por AWS PrivateLink uma AWS tecnologia que permite a comunicação privada entre o Serviços da AWS uso de uma interface de rede elástica com endereços IP privados. Para obter mais informações, consulte Novo - AWS PrivateLink para Serviços da AWS.

As etapas a seguir são para usuários da Amazon VPC. Para obter mais informações, consulte Conceitos básicos no Guia do usuário da Amazon VPC.

Disponibilidade

Os endpoints da VPC têm suporte em todas as regiões em que o Amazon Polly é compatível. Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte Infraestrutura AWS global.

Os endpoints do FIPS estão disponíveis nas seguintes regiões:

- Leste dos EUA (Norte da Virgínia) (us-east-1)
- Leste dos EUA (Ohio) (us-east-2)
- Oeste dos EUA (Norte da Califórnia) (us-west-1)

- Oeste dos EUA (Oregon) (us-west-2)
- AWS GovCloud (US-West) (us-gov-west-1)
- Canadá (Central) (ca-central-1)

Os endpoints do FIPS são do formato `com.amazonaws.REGION.polly-fips`.

Criar uma política de endpoint da VPC para o Amazon Polly

Para começar a usar o Amazon Polly com sua VPC, crie um endpoint da VPC de interface. O serviço a ser escolhido é com.amazonaws. ***Region***.polly. Não é necessário alterar as configurações do Amazon Polly. Para obter mais informações, consulte Criar um endpoint de interface no Guia do usuário da Amazon VPC.

Como testar a conexão entre a VPC e o Amazon Polly

Depois de criar o endpoint, você pode testar a conexão.

Como testar a conexão entre sua VPC e seu endpoint do Amazon Polly

1. Conecte-se a uma instância do Amazon EC2 que reside na VPC. Para obter mais informações sobre a conexão, consulte Como se conectar à sua instância do Linux ou Conexão com a sua instância do Windows na documentação do Amazon Polly.
2. Na instância, use `aws polly describe-voices` da AWS CLI para listar as vozes disponíveis do Amazon Polly.

Se a resposta ao comando incluir a lista de vozes disponíveis no Amazon Polly, o comando terá sido bem-sucedido e o endpoint da VPC estará funcionando.

Controlar o acesso ao endpoint da VPC ao Amazon Polly

Uma política de endpoint da VPC é uma política de recursos do IAM que você anexa a um endpoint quando cria ou modifica o endpoint. Se você não anexar uma política quando criar um endpoint, anexaremos uma política padrão que permita o acesso total ao serviço. Uma política de endpoint não substitui políticas de usuário do IAM nem políticas de serviço específicas. É uma política separada para controlar o acesso do endpoint ao serviço especificado.

Políticas de endpoint devem ser gravadas em formato JSON.

Para obter mais informações, consulte Controlar o acesso a serviços com VPC endpoints no Guia do usuário da Amazon VPC.

Veja a seguir um exemplo de política de endpoint para o Amazon Polly. Esta política permite que os usuários se conectem ao Amazon Polly por meio da VPC para descrever vozes e sintetizar a fala com o Amazon Polly, e impede que outras ações do Amazon Polly sejam executadas.

```
{
  "Statement": [
    {
      "Sid": "SynthesisAndDescribeVoicesOnly",
      "Principal": "*",
      "Action": [
        "polly:DescribeVoices",
        "polly:SynthesizeSpeech"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Como alterar a política de endpoint da VPC para o Amazon Polly

1. Abra o console da Amazon VPC em https://console.aws.amazon.com/vpc.
2. No painel de navegação, escolha Endpoints.
3. Se você ainda não tiver criado o endpoint para o Amazon Polly, selecione Criar endpoint. Em seguida, selecione com.amazonaws. ***Region***.polly e escolha Criar endpoint.
4. Selecione o com.amazonaws. ***Region***.polly endpoint e escolha a guia Política na metade inferior da tela.
5. Selecione Edit policy (Editar política) e faça as alterações na política.

Compatibilidade com chaves de contexto da VPC

O Amazon Polly oferece suporte às chaves de contexto `aws:SourceVpc` e `aws:SourceVpce` que podem limitar o acesso a VPCs ou a endpoints da VPC específicos. Essas chaves funcionam somente quando o usuário está usando VPC endpoints. Para obter mais informações, consulte Chaves disponíveis para alguns serviços no Guia do usuário do IAM.

Registrar chamadas de API do Amazon Polly em log com o AWS CloudTrail

O Amazon Polly é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações tomadas por um usuário, uma função ou um serviço da AWS no Amazon Polly. O CloudTrail captura todas as chamadas de API para o Amazon Polly como eventos. As chamadas capturadas incluem as chamadas do console do Amazon Polly e as chamadas de código para as operações da API do Amazon Polly. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o Amazon Polly. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Histórico de eventos. Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita para o Amazon Polly, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre o CloudTrail, incluindo como configurá-lo e ativá-lo, consulte o AWS CloudTrail Guia do usuário do.

Informações sobre o Amazon Polly no CloudTrail

O CloudTrail é habilitado em sua conta AWS ao criá-la. Quando a atividade do evento compatível ocorrer no Amazon Polly, ela será registrada em um evento do CloudTrail juntamente com outros eventos de serviços da AWS no Histórico de eventos. É possível visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte Visualização de eventos com o histórico de eventos do CloudTrail.

Para obter um registro contínuo de eventos da conta da AWS, incluindo eventos do Amazon Polly, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando uma trilha é criada no console, a mesma é aplicada a todas as regiões da AWS. A trilha registra logs de eventos de todas as Regiões na AWS divisória e entrega os arquivos do log para o bucket Amazon S3 especificado. Além disso, é possível configurar outros AWS serviços para melhor analisar e agir de acordo com dados coletados do evento nos logs CloudTrail. Para obter mais informações, consulte:

- Visão Geral para Criar uma Trilha
- Serviços e integrações com suporte no CloudTrail
- Configurando Notificações Amazon SNS para CloudTrail

- Recebendo Arquivos de Log CloudTrail de Várias Regiões e Recebendo Arquivos de Log CloudTrail de Várias Contas

O Amazon Polly é compatível com as seguintes ações como eventos nos arquivos de log do CloudTrail:

- DeleteLexicon
- DescribeVoices
- GetLexicon
- GetSpeechSynthesisTask
- ListLexicons
- ListSpeechSynthesisTasks
- PutLexicon
- StartSpeechSynthesisTask
- SynthesizeSpeech

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário-raiz ou de usuário do AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para mais informações, consulte Elemento userIdentity CloudTrail.

Exemplo: entradas de arquivo de log do Amazon Polly

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket Amazon S3 especificado. Os arquivos de log CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte, e inclui informações sobre a ação solicitada, data e hora da ação, parâmetros da solicitação e assim por diante. Os arquivos de log do

CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública, portanto, não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `SynthesizeSpeech`.


```
{
"Records": [
        {
"awsRegion": "us-east-2",
            "eventID": "19bd70f7-5e60-4cdc-9825-936c552278ae",
            "eventName": "SynthesizeSpeech",
            "eventSource": "polly.amazonaws.com",
            "eventTime": "2016-11-02T03:49:39Z",
            "eventType": "AwsApiCall",
            "eventVersion": "1.05",
            "recipientAccountId": "123456789012",
            "requestID": "414288c2-a1af-11e6-b17f-d7cfc06cb461",
            "requestParameters": {
"lexiconNames": [
                    "SampleLexicon"
                ],
                "engine": "neural",
                "outputFormat": "mp3",
                "sampleRate": "22050",
                "text": "**********",
                "textType": "text",
                "voiceId": "Kendra"
            },
            "responseElements": null,
            "sourceIPAddress": "1.2.3.4",
            "userAgent": "Amazon CLI/Polly 1.10 API 2016-06-10",
            "userIdentity": {
"accessKeyId": "EXAMPLE_KEY_ID",
                "accountId": "123456789012",
                "arn": "arn:aws:iam::123456789012:user/Alice",
                "principalId": "EX_PRINCIPAL_ID",
                "type": "IAMUser",
                "userName": "Alice"
            }
        }

    ]
```

```
}
```

Integrar o CloudWatch ao Amazon Polly

Ao interagir com o Amazon Polly, ele envia as seguintes métricas e dimensões para o CloudWatch a cada minuto. Use os procedimentos a seguir para visualizar as métricas do Amazon Polly.

É possível monitorar o Amazon Polly usando o CloudWatch, que coleta e processa dados brutos do Amazon Polly e os transforma em métricas legíveis quase em tempo real. Essas estatísticas são registradas por um período de duas semanas, para que você possa acessar `historical information` e obter uma perspectiva melhor sobre como seu serviço ou sua aplicação web estão se saindo. Por padrão, dados métricos do Amazon Polly são enviados para o CloudWatch em intervalos de 1 minuto. Para obter mais informações, consulte O que é o Amazon CloudWatch? no Guia do usuário do Amazon CloudWatch.

Obter métricas do CloudWatch (Console)

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Métricas.
3. No painel Métricas do CloudWatch por categoria, na categoria de métricas para o Amazon Polly, selecione uma categoria de métricas e, em seguida, no painel superior, role para baixo para visualizar a lista completa de métricas.

Obter métricas do CloudWatch na AWS CLI

O código a seguir exibe métricas disponíveis para o Amazon Polly.

```
aws cloudwatch list-metrics --namespace "AWS/Polly"
```

O comando anterior apresenta uma lista de métricas do Amazon Polly semelhante à lista a seguir. O elemento `MetricName` identifica qual é a métrica.

```
{
    "Metrics": [
        {
            "Namespace": "AWS/Polly",
            "Dimensions": [
                {
                    "Name": "Operation",
```

```
                "Value": "SynthesizeSpeech"
            }
        ],
        "MetricName": "ResponseLatency"
    },
    {
        "Namespace": "AWS/Polly",
        "Dimensions": [
            {
                "Name": "Operation",
                "Value": "SynthesizeSpeech"
            }
        ],
        "MetricName": "RequestCharacters"
    }
```

Para obter mais informações, consulte GetMetricStatistics na Referência de APIs do Amazon CloudWatch.

Métricas do Amazon Polly

O Amazon Polly produz as seguintes métricas para cada solicitação. Essas métricas são agregadas e, em intervalos de um minuto, enviadas para o CloudWatch, onde estão disponíveis.

Métrica	Descrição
`RequestCharacters`	Número de caracteres nas solicitações. Somente caracteres faturáveis e não inclui tags SSML. Dimensão válida: operação Estatísticas válidas: mínima, máxima, média, SampleCount, soma. Unidade: Contagem
`ResponseLatency`	A latência entre o momento da solicitação e o início da resposta de streaming. Dimensões válidas: operação

Métrica	Descrição
	Estatísticas válidas: mínima, máxima, média, SampleCount. Unidade: microssegundos
2XXCount	Código de nível HTTP 200 retornado após uma resposta bem-sucedida. Dimensões válidas: operação Estatísticas válidas: média, SampleCount, soma Unidade: Contagem
4XXCount	Código de erro de nível HTTP 400 retornado após um erro. Para cada resposta bem-sucedida, um zero (0) é emitido. Dimensões válidas: operação Estatísticas válidas: média, SampleCount, soma Unidade: Contagem
5XXCount	Código de erro de nível HTTP 500 retornado após um erro. Para cada resposta bem-sucedida, um zero (0) é emitido. Dimensões válidas: operação Estatísticas válidas: média, SampleCount, soma Unidade: Contagem

Dimensões para métricas do Amazon Polly

As métricas do Amazon Polly usam o namespace AWS/Polly e fornecem métricas para as dimensões a seguir.

Dimensão	Descrição
`Operation`	As métricas são agrupadas pelo método de API ao qual elas se referem. Os valores possíveis são `SynthesizeSpeech` , `PutLexicon` , `DescribeVoices` , etc.

Referência da API do Amazon Polly

O Amazon Polly é um serviço da web que facilita a síntese de fala com base em texto.

O serviço Amazon Polly fornece operações de API para sintetizar voz de alta qualidade com base em texto simples e linguagem SSML (Speech Synthesis Markup Language), além de gerenciar léxicos de pronúncias que permitem que você obtenha os melhores resultados para o domínio da sua aplicação.

Chamadas de API autenticadas devem ser assinadas usando o processo de assinatura do Signature versão 4. Para obter mais informações, consulte Assinatura de solicitações de AWS API no Referência geral da Amazon Web Services.

Tópicos

- Ações
- Tipos de dados
- Tipos de erro comuns
- Parâmetros gerais

Ações

As ações a seguir são compatíveis:

- DeleteLexicon
- DescribeVoices
- GetLexicon
- GetSpeechSynthesisTask
- ListLexicons
- ListSpeechSynthesisTasks
- PutLexicon
- StartSpeechSynthesisStream
- StartSpeechSynthesisTask
- SynthesizeSpeech

DeleteLexicon

Exclui o léxico de pronúncia armazenado em uma Região da AWS. Um léxico que foi excluído não está disponível para síntese de fala, nem é possível recuperá-lo usando o ou. `GetLexicon` `ListLexicon` APIs

Para obter mais informações, consulte Gerenciamento de léxicos.

Sintaxe da Solicitação

```
DELETE /v1/lexicons/LexiconName HTTP/1.1
```

Parâmetros da Solicitação de URI

A solicitação usa os seguintes parâmetros de URI:

LexiconName

Nome da conexão a ser excluída. Deve ser um léxico existente na região.

Padrão: `[0-9A-Za-z]{1,20}`

Obrigatório: Sim

Corpo da Solicitação

Essa solicitação não tem corpo.

Sintaxe da resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço reenviará uma resposta 200 HTTP com um corpo HTTP vazio.

Erros

LexiconNotFoundException

O Amazon Polly não encontrou o léxico especificado. Isso pode ser causado por um léxico inexistente, pelo nome escrito incorretamente ou pela especificação de um léxico que está em uma região diferente.

Verifique se o léxico existe, se está na região (consulte ListLexicons) e se o nome dele está escrito corretamente. Tente novamente.

Código de status HTTP: 404

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Código de status HTTP: 500

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS Interface de linha de comando V2
- AWS SDK para .NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

DescribeVoices

Retorna a lista de vozes disponíveis para uso ao solicitar uma síntese de fala. Cada voz fala um idioma específico, é masculina ou feminina e é identificada por um ID, que é a versão ASCII do nome da voz.

Ao sintetizar a fala (`SynthesizeSpeech`), você fornece o ID de voz para a voz desejada na lista de vozes exibidas em `DescribeVoices`.

Por exemplo, você deseja que seu aplicativo leitor de notícias leia notícias em um idioma específico, mas ofereça ao usuário a opção de escolher a voz. Usando a operação `DescribeVoices`, você pode fornecer ao usuário uma lista de vozes disponíveis para seleção.

Opcionalmente, você pode especificar um código de idioma para filtrar as vozes disponíveis. Por exemplo, se você especificar `en-US`, a operação retornará uma lista de todas as vozes disponíveis em inglês dos EUA.

Essa operação exige permissões para executar a ação `polly:DescribeVoices`.

Sintaxe da Solicitação

```
GET /v1/voices?
Engine=Engine&IncludeAdditionalLanguageCodes=IncludeAdditionalLanguageCodes&LanguageCode=Langua
 HTTP/1.1
```

Parâmetros da Solicitação de URI

A solicitação usa os seguintes parâmetros de URI:

Engine

Especifica o mecanismo (`standard`, `neural`, `long-form` ou `generative`) usado pelo Amazon Polly ao processar o texto de entrada para síntese de fala.

Valores válidos: `standard | neural | long-form | generative`

IncludeAdditionalLanguageCodes

Valor booleano que indica se deve retornar alguma voz bilíngue que use o idioma especificado como idioma adicional. Por exemplo, se você solicitar todos os idiomas que usam inglês dos EUA

(en-US) e houver uma voz em italiano que fale italiano (it-IT) e inglês dos EUA, essa voz será incluída se você especificar, `yes` mas não se você especificar no.

LanguageCode

A tag de identificação do idioma (código ISO 639 para o nome do idioma-código ISO 3166 para o país) para filtrar a lista de vozes retornadas. Se você não especificar esse parâmetro opcional todas as vozes disponíveis serão retornadas.

Valores válidos: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG`

NextToken

Um token de paginação opaco retornou da operação `DescribeVoices` anterior. Se presente, indica de onde continuar a listagem.

Restrições de tamanho: tamanho mínimo 0. Comprimento máximo de 4.096.

Corpo da Solicitação

Essa solicitação não tem corpo.

Sintaxe da resposta

```
HTTP/1.1 200
Content-type: application/json

{
   "NextToken": "string",
   "Voices": [
      {
         "AdditionalLanguageCodes": [ "string" ],
         "Gender": "string",
         "Id": "string",
         "LanguageCode": "string",
         "LanguageName": "string",
         "Name": "string",
         "SupportedEngines": [ "string" ]
```

```
        }
    ]
}
```

Elementos de resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

NextToken

O token de paginação a ser usado na próxima solicitação para continuar a listagem de vozes. `NextToken` retorna apenas quando a resposta está truncada.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Comprimento máximo de 4.096.

Voices

Lista de vozes com suas propriedades.

Tipo: matriz de objetos Voice

Erros

InvalidNextTokenException

O NextToken é inválido. Verifique se ele está escrito corretamente e tente de novo.

Código de status HTTP: 400

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Código de status HTTP: 500

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS Interface de linha de comando V2
- AWS SDK para .NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

GetLexicon

Retorna o conteúdo do léxico de pronúncia especificado armazenado em uma Região da AWS. Para obter mais informações, consulte Gerenciamento de léxicos.

Sintaxe da Solicitação

```
GET /v1/lexicons/LexiconName HTTP/1.1
```

Parâmetros da Solicitação de URI

A solicitação usa os seguintes parâmetros de URI:

LexiconName

Nome do léxico.

Padrão: `[0-9A-Za-z]{1,20}`

Obrigatório: Sim

Corpo da Solicitação

Essa solicitação não tem corpo.

Sintaxe da resposta

```
HTTP/1.1 200
Content-type: application/json

{
   "Lexicon": {
      "Content": "string",
      "Name": "string"
   },
   "LexiconAttributes": {
      "Alphabet": "string",
      "LanguageCode": "string",
      "LastModified": number,
      "LexemesCount": number,
```

```
      "LexiconArn": "string",
      "Size": number
   }
}
```

Elementos de resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

Lexicon

Objeto de léxico que fornece o nome e o conteúdo da string do léxico.

Tipo: objeto Lexicon

LexiconAttributes

Metadados do léxico, incluindo alfabeto fonético usado, código do idioma, ARN do léxico, número de lexemas definidos nele e tamanho em bytes.

Tipo: objeto LexiconAttributes

Erros

LexiconNotFoundException

O Amazon Polly não encontrou o léxico especificado. Isso pode ser causado por um léxico inexistente, pelo nome escrito incorretamente ou pela especificação de um léxico que está em uma região diferente.

Verifique se o léxico existe, se está na região (consulte ListLexicons) e se o nome dele está escrito corretamente. Tente novamente.

Código de status HTTP: 404

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Código de status HTTP: 500

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS Interface de linha de comando V2
- AWS SDK para .NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

GetSpeechSynthesisTask

Recupera um SpeechSynthesisTask objeto específico com base em seu TaskID. Esse objeto contém informações sobre determinada tarefa de síntese de fala, incluindo o status da tarefa e um link para o bucket do S3 contendo a saída da tarefa.

Sintaxe da Solicitação

```
GET /v1/synthesisTasks/TaskId HTTP/1.1
```

Parâmetros da Solicitação de URI

A solicitação usa os seguintes parâmetros de URI:

TaskId

O identificador gerado pelo Amazon Polly para uma tarefa de síntese de fala.

Padrão: `^[a-zA-Z0-9_-]{1,100}$`

Obrigatório: Sim

Corpo da Solicitação

Essa solicitação não tem corpo.

Sintaxe da resposta

```
HTTP/1.1 200
Content-type: application/json

{
   "SynthesisTask": {
      "CreationTime": number,
      "Engine": "string",
      "LanguageCode": "string",
      "LexiconNames": [ "string" ],
      "OutputFormat": "string",
      "OutputUri": "string",
      "RequestCharacters": number,
```

```
      "SampleRate": "string",
      "SnsTopicArn": "string",
      "SpeechMarkTypes": [ "string" ],
      "TaskId": "string",
      "TaskStatus": "string",
      "TaskStatusReason": "string",
      "TextType": "string",
      "VoiceId": "string"
   }
}
```

Elementos de resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

SynthesisTask

SynthesisTask objeto que fornece informações da tarefa solicitada, incluindo formato de saída, hora de criação, status da tarefa e assim por diante.

Tipo: objeto SynthesisTask

Erros

InvalidTaskIdException

O ID da tarefa fornecido não é válido. Forneça um ID de tarefa válido e tente novamente.

Código de status HTTP: 400

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Código de status HTTP: 500

SynthesisTaskNotFoundException

A tarefa de Síntese de Fala com o ID de tarefa solicitado não foi encontrada.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS Interface de linha de comando V2
- AWS SDK para .NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

ListLexicons

Retorna uma lista de léxicos de pronúncia armazenados em uma Região da AWS. Para obter mais informações, consulte Gerenciamento de léxicos.

Sintaxe da Solicitação

```
GET /v1/lexicons?NextToken=NextToken HTTP/1.1
```

Parâmetros da Solicitação de URI

A solicitação usa os seguintes parâmetros de URI:

NextToken

Um token de paginação opaco retornado da operação `ListLexicons` anterior. Se presente, indica de onde continuar a lista de léxicos.

Restrições de tamanho: tamanho mínimo 0. Comprimento máximo de 4.096.

Corpo da Solicitação

Essa solicitação não tem corpo.

Sintaxe da resposta

```
HTTP/1.1 200
Content-type: application/json

{
   "Lexicons": [
      {
         "Attributes": {
            "Alphabet": "string",
            "LanguageCode": "string",
            "LastModified": number,
            "LexemesCount": number,
            "LexiconArn": "string",
            "Size": number
         },
         "Name": "string"
      }
```

```
    ],
    "NextToken": "string"
}
```

Elementos de resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

Lexicons

Uma lista de nomes e atributos do léxico.

Tipo: matriz de objetos LexiconDescription

NextToken

O token de paginação a ser usado na próxima solicitação para continuar a listagem de léxicos. Retornará `NextToken` somente se a resposta estiver truncada.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Comprimento máximo de 4.096.

Erros

InvalidNextTokenException

O NextToken é inválido. Verifique se ele está escrito corretamente e tente de novo.

Código de status HTTP: 400

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Código de status HTTP: 500

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS Interface de linha de comando V2
- AWS SDK para .NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

ListSpeechSynthesisTasks

Retorna uma lista de SpeechSynthesisTask objetos ordenados pela data de criação. Essa operação pode filtrar as tarefas pelo status. Por exemplo, permitindo que os usuários listem somente as tarefas concluídas.

Sintaxe da Solicitação

```
GET /v1/synthesisTasks?MaxResults=MaxResults&NextToken=NextToken&Status=Status HTTP/1.1
```

Parâmetros da Solicitação de URI

A solicitação usa os seguintes parâmetros de URI:

MaxResults

Número máximo de tarefas de síntese de voz retornadas em uma operação de lista.

Faixa válida: valor mínimo de 1. Valor máximo de 100.

NextToken

O token de paginação a ser usado na próxima solicitação para continuar a listagem das tarefas de síntese de fala.

Restrições de tamanho: tamanho mínimo 0. Comprimento máximo de 4.096.

Status

Status das tarefas de síntese de fala retornadas em uma operação de Lista

Valores válidos: `scheduled | inProgress | completed | failed`

Corpo da Solicitação

Essa solicitação não tem corpo.

Sintaxe da resposta

```
HTTP/1.1 200
Content-type: application/json
```

```
{
   "NextToken": "string",
   "SynthesisTasks": [
      {
         "CreationTime": number,
         "Engine": "string",
         "LanguageCode": "string",
         "LexiconNames": [ "string" ],
         "OutputFormat": "string",
         "OutputUri": "string",
         "RequestCharacters": number,
         "SampleRate": "string",
         "SnsTopicArn": "string",
         "SpeechMarkTypes": [ "string" ],
         "TaskId": "string",
         "TaskStatus": "string",
         "TaskStatusReason": "string",
         "TextType": "string",
         "VoiceId": "string"
      }
   ]
}
```

Elementos de resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

NextToken

Um token de paginação opaco retornado da operação de Lista anterior nessa solicitação. Se presente, indica onde continuar a listagem.

Tipo: string

Restrições de tamanho: tamanho mínimo 0. Comprimento máximo de 4.096.

SynthesisTasks

Lista de SynthesisTask objetos que fornece informações da tarefa especificada na solicitação de lista, incluindo formato de saída, hora de criação, status da tarefa e assim por diante.

Tipo: matriz de objetos SynthesisTask

Erros

InvalidNextTokenException

O NextToken é inválido. Verifique se ele está escrito corretamente e tente de novo.

Código de status HTTP: 400

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Código de status HTTP: 500

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS Interface de linha de comando V2
- AWS SDK para .NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

PutLexicon

Armazena um léxico de pronúncia em uma Região da AWS. Se um léxico com o mesmo nome já existir na região, ele será substituído pelo novo léxico. As operações do léxico têm consistência eventual, portanto, pode levar algum tempo até que o léxico esteja disponível para a operação. SynthesizeSpeech

Para obter mais informações, consulte Gerenciamento de léxicos.

Sintaxe da Solicitação

```
PUT /v1/lexicons/LexiconName HTTP/1.1
Content-type: application/json

{
   "Content": "string"
}
```

Parâmetros da Solicitação de URI

A solicitação usa os seguintes parâmetros de URI:

LexiconName

Nome do léxico. O nome deve seguir o formato expresso regular [0-9A-Za-z]{1,20}. Ou seja, o nome é uma sequência alfanumérica com distinção entre maiúsculas e minúsculas de até 20 caracteres.

Padrão: `[0-9A-Za-z]{1,20}`

Obrigatório: Sim

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

Content

Conteúdo do léxico PLS como dados de string.

Tipo: sequência

Obrigatório: sim

Sintaxe da resposta

```
HTTP/1.1 200
```

Elementos de Resposta

Se a ação for bem-sucedida, o serviço reenviará uma resposta 200 HTTP com um corpo HTTP vazio.

Erros

InvalidLexiconException

O Amazon Polly não encontrou o léxico especificado. Verifique se o nome do léxico está escrito corretamente e tente de novo.

Código de status HTTP: 400

LexiconSizeExceededException

O tamanho máximo do léxico especificado seria excedido por essa operação.

Código de status HTTP: 400

MaxLexemeLengthExceededException

O tamanho máximo do lexema seria excedido por essa operação.

Código de status HTTP: 400

MaxLexiconsNumberExceededException

O número máximo de léxicos seria excedido por essa operação.

Código de status HTTP: 400

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Código de status HTTP: 500

UnsupportedPlsAlphabetException

O alfabeto especificado pelo léxico não é um alfabeto compatível. Os valores válidos são `x-sampa` e `ipa`.

Código de status HTTP: 400

UnsupportedPlsLanguageException

O idioma especificado no léxico não é compatível. Para ver uma lista dos idiomas compatíveis, consulte Atributos do léxico.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS Interface de linha de comando V2
- AWS SDK para .NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

StartSpeechSynthesisStream

Sintetiza entrada UTF-8, texto simples ou SSML em uma conexão de streaming bidirecional. Especifique os parâmetros de síntese nos cabeçalhos HTTP/2, envie texto incrementalmente como eventos no fluxo de entrada e receba áudio sintetizado assim que estiver disponível.

Essa operação serve como uma contrapartida bidirecional para: `SynthesizeSpeech`

- SynthesizeSpeech

Sintaxe da Solicitação

```
POST /v1/synthesisStream HTTP/1.1
x-amzn-Engine: Engine
x-amzn-LanguageCode: LanguageCode
x-amzn-LexiconNames: LexiconNames
x-amzn-OutputFormat: OutputFormat
x-amzn-SampleRate: SampleRate
x-amzn-VoiceId: VoiceId
Content-type: application/json

{
   "CloseStreamEvent": {
   },
   "TextEvent": {
      "FlushStreamConfiguration": {
         "Force": boolean
      },
      "Text": "string",
      "TextType": "string"
   }
}
```

Parâmetros da Solicitação de URI

A solicitação usa os seguintes parâmetros de URI:

Engine

Especifica o mecanismo que o Amazon Polly deve usar ao processar texto de entrada para síntese de fala. Atualmente, somente o `generative` motor é suportado. Se você especificar uma voz que o mecanismo selecionado não suporta, o Amazon Polly retornará um erro.

Valores válidos: `standard | neural | long-form | generative`

Obrigatório: Sim

LanguageCode

Um parâmetro opcional que define o código do idioma para a solicitação de síntese de fala. Especifique esse parâmetro somente ao usar uma voz bilíngue. Se uma voz bilíngue for usada e nenhum código de idioma for especificado, o Amazon Polly usará o idioma padrão da voz bilíngue.

Valores válidos: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG`

LexiconNames

Os nomes de um ou mais léxicos de pronúncia para o serviço aplicar durante a síntese. O Amazon Polly aplica léxicos somente quando o idioma do léxico corresponde ao idioma da voz.

Membros da matriz: número máximo de 5 itens.

Padrão: `[0-9A-Za-z]{1,20}`

OutputFormat

O formato de áudio para a fala sintetizada. Atualmente, o Amazon Polly não oferece suporte a marcas de fala JSON.

Valores válidos: `json | mp3 | ogg_opus | ogg_vorbis | pcm | mulaw | alaw`

Obrigatório: Sim

SampleRate

A frequência de áudio, especificada em Hz.

VoiceId

A voz a ser usada em síntese. Para obter uma lista das vozes disponíveis IDs, use a DescribeVoicesoperação.

Valores válidos: `Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal | Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu | Jitka | Sabrina | Jasmine | Jihye | Ambre | Beatrice | Florian | Lennart | Lorenzo | Tiffany`

Obrigatório: Sim

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

CloseStreamEvent

Um evento que indica o fim do fluxo de entrada.

Tipo: objeto CloseStreamEvent

Obrigatório: não

TextEvent

Um evento de texto contendo conteúdo a ser sintetizado.

Tipo: objeto TextEvent

Obrigatório: não

Sintaxe da resposta

```
HTTP/1.1 200
Content-type: application/json

{
   "AudioEvent": {
      "AudioChunk": blob
   },
   "ServiceFailureException": {
   },
   "ServiceQuotaExceededException": {
   },
   "StreamClosedEvent": {
      "RequestCharacters": number
   },
   "ThrottlingException": {
   },
   "ValidationException": {
   }
}
```

Elementos de resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

AudioEvent

Um evento de áudio contendo fala sintetizada.

Tipo: objeto AudioEvent

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Tipo: Exceção

Código de status HTTP: 500

ServiceQuotaExceededException

Uma exceção indicando que uma cota de serviço seria excedida.

Tipo: Exceção

Código de status HTTP: 402

StreamClosedEvent

Um evento, com informações resumidas, indicando que o stream foi encerrado.

Tipo: objeto StreamClosedEvent

ThrottlingException

Uma exceção indicando que a solicitação foi limitada.

Tipo: Exceção

Código de status HTTP: 400

ValidationException

Uma exceção indicando falha na validação da entrada.

Tipo: Exceção

Código de status HTTP: 400

Erros

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Código de status HTTP: 500

ServiceQuotaExceededException

A solicitação faria com que uma cota de serviço fosse excedida.

quotaCode

O código da cota que identifica a cota específica.

serviceCode

O código do serviço que identifica o serviço de origem.

Código de status HTTP: 402

ThrottlingException

A solicitação foi negada devido à limitação da solicitação.

throttlingReasons

Uma lista de motivos que explicam por que a solicitação foi suspensa.

Código de status HTTP: 400

ValidationException

A entrada não atende às restrições especificadas pelo serviço da .

fields

Os campos que causaram o erro de validação.

reason

O motivo pelo qual a solicitação falhou na validação.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS Interface de linha de comando V2
- AWS SDK para .NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

StartSpeechSynthesisTask

Permite a criação de uma tarefa de síntese assíncrona, iniciando uma nova `SpeechSynthesisTask`. Essa operação requer todas as informações padrão necessárias para a síntese de fala, além do nome de um bucket do Amazon S3 para que o serviço armazene a saída da tarefa de síntese e dois parâmetros opcionais (`OutputS3KeyPrefix` e `SnsTopicArn`). Depois que a tarefa de síntese for criada, a operação retornará um objeto de `SpeechSynthesisTask` que incluirá um identificador a tarefa e o status atual. O objeto `SpeechSynthesisTask` fica disponível por 72 horas após o início da tarefa de síntese assíncrona.

Sintaxe da Solicitação

```
POST /v1/synthesisTasks HTTP/1.1
Content-type: application/json

{
   "Engine": "string",
   "LanguageCode": "string",
   "LexiconNames": [ "string" ],
   "OutputFormat": "string",
   "OutputS3BucketName": "string",
   "OutputS3KeyPrefix": "string",
   "SampleRate": "string",
   "SnsTopicArn": "string",
   "SpeechMarkTypes": [ "string" ],
   "Text": "string",
   "TextType": "string",
   "VoiceId": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

Engine

Especifica o mecanismo (`standard`, `neural`, `long-form` ou `generative`) que o Amazon Polly usará ao processar texto de entrada para a síntese de fala. Usar uma voz que não seja compatível com o mecanismo selecionado resultará em erro.

Tipo: string

Valores válidos: `standard | neural | long-form | generative`

Obrigatório: não

LanguageCode

Código de idioma opcional para a solicitação de síntese de fala. É necessário apenas se estiver usando uma voz bilíngue, como Aditi, que pode ser usada para inglês indiano (en-IN) ou hindi (hi-IN).

Se uma voz bilíngue for usada e nenhum código de idioma for especificado, o Amazon Polly usará o idioma padrão da voz bilíngue. O idioma padrão para qualquer voz é aquele retornado pela DescribeVoicesoperação do `LanguageCode` parâmetro. Por exemplo, se nenhum código de idioma for especificado, Aditi usará inglês indiano em vez de hindi.

Tipo: string

Valores válidos: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG`

Obrigatório: não

LexiconNames

Lista de um ou mais nomes de léxico de pronúncia que você deseja que o serviço aplique durante a síntese. O léxico só será aplicado ao texto se o idioma especificado nele for o mesmo que o da voz.

Tipo: matriz de strings

Membros da matriz: número máximo de 5 itens.

Padrão: `[0-9A-Za-z]{1,20}`

Obrigatório: não

OutputFormat

O formato no qual a saída retornada será codificada. Para transmissão de áudio, serão mp3, ogg_vorbis, ogg_opus, mu-law, a-law ou pcm. Marcas de fala serão JSON.

Tipo: string

Valores válidos: `json | mp3 | ogg_opus | ogg_vorbis | pcm | mulaw | alaw`

Obrigatório: Sim

OutputS3BucketName

Nome do bucket do Amazon S3 no qual o arquivo de saída será salvo.

Tipo: sequência

Padrão: `^[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]$`

Obrigatório: Sim

OutputS3KeyPrefix

Prefixo das chaves do Amazon S3 para o arquivo de fala de saída.

Tipo: sequência

Padrão: `^[0-9a-zA-Z\/\!\-_\.\*\'\(\):;\$@=+\,\?&]{0,800}$`

Obrigatório: não

SampleRate

A frequência de áudio especificada em Hz.

Os valores válidos para MP3 e ogg_vorbis são 8000, 16000, 22050 e 24000. O valor padrão para vozes padrão é 22050. O valor padrão para vozes neurais é 24000. O valor padrão para vozes em formato longo é “24.000”. O valor padrão para vozes generativas é “24000”.

Os valores válidos para pcm são 8000 e 16000. O valor padrão é 16000.

O valor válido para ogg_opus é “48000".

O valor válido para mu-law e a-law é “8000".

Tipo: string

Obrigatório: não

SnsTopicArn

O ARN do tópico do SNS é usado opcionalmente para fornecer notificação de status para uma tarefa de síntese de fala.

Tipo: sequência

Padrão: `^arn:aws(-(cn|iso(-b)?|us-gov))?:sns:[a-z0-9_-]{1,50}:\d{12}:[a-zA-Z0-9_-]{1,251}([a-zA-Z0-9_-]{0,5}|\.fifo)$`

Obrigatório: não

SpeechMarkTypes

Tipo de marcas de fala retornadas para o texto de entrada.

Tipo: matriz de strings

Membros da matriz: número máximo de 4 itens.

Valores válidos: `sentence | ssml | viseme | word`

Obrigatório: não

Text

Texto de entrada a ser sintetizado. Se você especificar ssml como TextType, siga o formato SSML para o texto de entrada.

Tipo: String

Obrigatório: Sim

TextType

Especifica se o texto de entrada é simples ou SSML. O valor padrão é texto sem formatação.

Tipo: string

Valores válidos: `ssml | text`

Obrigatório: não

VoiceId

ID de voz a ser usado para a síntese.

Tipo: string

Valores válidos: `Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal | Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu | Jitka | Sabrina | Jasmine | Jihye | Ambre | Beatrice | Florian | Lennart | Lorenzo | Tiffany`

Obrigatório: sim

Sintaxe da resposta

```
HTTP/1.1 200
Content-type: application/json

{
   "SynthesisTask": {
      "CreationTime": number,
      "Engine": "string",
      "LanguageCode": "string",
      "LexiconNames": [ "string" ],
      "OutputFormat": "string",
      "OutputUri": "string",
      "RequestCharacters": number,
      "SampleRate": "string",
      "SnsTopicArn": "string",
```

```
      "SpeechMarkTypes": [ "string" ],
      "TaskId": "string",
      "TaskStatus": "string",
      "TaskStatusReason": "string",
      "TextType": "string",
      "VoiceId": "string"
   }
}
```

Elementos de resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

Os dados a seguir são retornados no formato JSON pelo serviço.

SynthesisTask

SynthesisTask objeto que fornece informações e atributos sobre uma tarefa de síntese de fala recém-enviada.

Tipo: objeto SynthesisTask

Erros

EngineNotSupportedException

Esse mecanismo não é compatível com a voz que você designou. Escolha uma nova voz compatível com o mecanismo ou troque-o e reinicie a operação.

Código de status HTTP: 400

InvalidS3BucketException

O nome do bucket do Amazon S3 fornecido é inválido. Verifique a entrada com os requisitos de nomenclatura do bucket do S3 e tente novamente.

Código de status HTTP: 400

InvalidS3KeyException

O prefixo das chaves do Amazon S3 fornecido é inválido. Forneça um nome de chave de objeto S3 válido.

Código de status HTTP: 400

InvalidSampleRateException

A taxa de amostragem especificada é inválida.

Código de status HTTP: 400

InvalidSnsTopicArnException

O ARN do tópico SNS fornecido é inválido. Forneça um ARN de tópico de SNS válido e tente novamente.

Código de status HTTP: 400

InvalidSsmlException

O SSML fornecido é inválido. Verifique a sintaxe do SSML, a ortografia e valores das tags tente novamente.

Código de status HTTP: 400

LanguageNotSupportedException

O idioma especificado não é compatível com o Amazon Polly nessa capacidade.

Código de status HTTP: 400

LexiconNotFoundException

O Amazon Polly não encontrou o léxico especificado. Isso pode ser causado por um léxico inexistente, pelo nome escrito incorretamente ou pela especificação de um léxico que está em uma região diferente.

Verifique se o léxico existe, se está na região (consulte ListLexicons) e se o nome dele está escrito corretamente. Tente novamente.

Código de status HTTP: 404

MarksNotSupportedForFormatException

As marcas de fala não são compatíveis com o `OutputFormat` selecionado. As marcas de fala estão disponíveis apenas para conteúdo em formato `json`.

Código de status HTTP: 400

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Código de status HTTP: 500

SsmlMarksNotSupportedForTextTypeException

As marcas de fala SSML não são compatíveis com a entrada de texto sem formatação.

Código de status HTTP: 400

TextLengthExceededException

O valor do parâmetro Texto é maior que os limites aceitos. Para a API do `SynthesizeSpeech`, o limite para texto de entrada é de no máximo 6 mil caracteres, dos quais não mais do que 3 mil podem ser faturados. Para a API `StartSpeechSynthesisTask`, o máximo é de 200 mil caracteres, dos quais não mais do que 100 mil podem ser faturados. Tags SSML não são contadas como caracteres cobrados.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em uma linguagem específica AWS SDKs, consulte o seguinte:

- AWS Interface de linha de comando V2
- AWS SDK para .NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

SynthesizeSpeech

Sintetiza entrada UTF-8, texto sem formatação ou SSML em um fluxo de bytes. A entrada SSML deve ser válida e bem formada. Alguns alfabetos podem não estar disponíveis com todas as vozes (por exemplo, o cirílico pode não ser lido pelas vozes em inglês), a menos que o mapeamento de fonemas seja usado. Para obter mais informações, consulte Como funciona.

Sintaxe da Solicitação

```
POST /v1/speech HTTP/1.1
Content-type: application/json

{
   "Engine": "string",
   "LanguageCode": "string",
   "LexiconNames": [ "string" ],
   "OutputFormat": "string",
   "SampleRate": "string",
   "SpeechMarkTypes": [ "string" ],
   "Text": "string",
   "TextType": "string",
   "VoiceId": "string"
}
```

Parâmetros da solicitação de URI

A solicitação não usa nenhum parâmetro de URI.

Corpo da Solicitação

A solicitação aceita os dados a seguir no formato JSON.

Engine

Especifica o mecanismo (`standard`, `neural`, `long-form` ou `generative`) que o Amazon Polly usará ao processar texto de entrada para a síntese de fala. Forneça um mecanismo compatível com a voz selecionada. Se você não fornecer um mecanismo, será selecionado o mecanismo padrão. Se a voz escolhida não for compatível com o mecanismo padrão, ocorrerá um erro. Para obter informações sobre as vozes do Amazon Polly e quais estão disponíveis para cada mecanismo, consulte Vozes disponíveis.

Tipo: string

Valores válidos: `standard | neural | long-form | generative`

Obrigatório: não

LanguageCode

Código de idioma opcional para a solicitação Synthesize Speech. É necessário apenas se estiver usando uma voz bilíngue, como Aditi, que pode ser usada para inglês indiano (en-IN) ou hindi (hi-IN).

Se uma voz bilíngue for usada e nenhum código de idioma for especificado, o Amazon Polly usará o idioma padrão da voz bilíngue. O idioma padrão para qualquer voz é aquele retornado pela DescribeVoicesoperação do `LanguageCode` parâmetro. Por exemplo, se nenhum código de idioma for especificado, Aditi usará inglês indiano em vez de hindi.

Tipo: string

Valores válidos: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG`

Obrigatório: não

LexiconNames

Lista de um ou mais nomes de léxico de pronúncia que você deseja que o serviço aplique durante a síntese. O léxico só será aplicado ao texto se o idioma especificado nele for o mesmo que o da voz. Para obter informações sobre como armazenar léxicos, consulte. PutLexicon

Tipo: matriz de strings

Membros da matriz: número máximo de 5 itens.

Padrão: `[0-9A-Za-z]{1,20}`

Obrigatório: não

OutputFormat

O formato no qual a saída retornada será codificada. Para transmissão de áudio, serão mp3, ogg_vorbis, ogg_opus, mu-law, a-law ou pcm. Marcas de fala serão JSON.

Quando o pcm é usado, o conteúdo retornado está audio/pcm em um formato little-endian assinado de 16 bits, 1 canal (mono).

Tipo: string

Valores válidos: `json | mp3 | ogg_opus | ogg_vorbis | pcm | mulaw | alaw`

Obrigatório: Sim

SampleRate

A frequência de áudio especificada em Hz.

Os valores válidos para MP3 e ogg_vorbis são “8000”, “16000”, “22050”, “24000”, “44100” e “48000”. O valor padrão para vozes padrão é 22050. O valor padrão para vozes neurais é 24000. O valor padrão para vozes em formato longo é “24.000”. O valor padrão para vozes generativas é “24000”.

Os valores válidos para pcm são 8000 e 16000. O valor padrão é 16000.

O valor válido para ogg_opus é “48000".

O valor válido para mu-law e a-law é “8000".

Tipo: string

Obrigatório: não

SpeechMarkTypes

Tipo de marcas de fala retornadas para o texto de entrada.

Tipo: matriz de strings

Membros da matriz: número máximo de 4 itens.

Valores válidos: `sentence | ssml | viseme | word`

Obrigatório: não

Text

Insira texto para sintetizar. Se você especificar `ssml` como `TextType`, siga o formato SSML para o texto de entrada.

Tipo: String

Obrigatório: Sim

TextType

Especifica se o texto de entrada é simples ou SSML. O valor padrão é texto sem formatação. Para obter mais informações, consulte Using SSML.

Tipo: string

Valores válidos: `ssml | text`

Obrigatório: não

VoiceId

ID de voz a ser usado para a síntese. Você pode obter uma lista das vozes disponíveis IDs chamando a DescribeVoicesoperação.

Tipo: string

Valores válidos: `Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal | Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu | Jitka | Sabrina | Jasmine | Jihye | Ambre | Beatrice | Florian | Lennart | Lorenzo | Tiffany`

Obrigatório: sim

Sintaxe da resposta

```
HTTP/1.1 200
```

```
Content-Type: ContentType
x-amzn-RequestCharacters: RequestCharacters

AudioStream
```

Elementos de resposta

Se a ação for bem-sucedida, o serviço retornará uma resposta HTTP 200.

A resposta retorna os cabeçalhos HTTP a seguir.

ContentType

Especifica o tipo de fluxo de áudio. Isso deve refletir o `OutputFormat` parâmetro em sua solicitação.

- Se você solicitar `mp3` como o `OutputFormat`, o `ContentType` retornado será áudio/mpeg.
- Se você solicitar `ogg_vorbis` como o `OutputFormat`, o `ContentType` retornado será audio/ogg.
- Se você solicitar `ogg_opus` como o `OutputFormat`, o `ContentType` retornado será audio/ogg.
- Se você solicitar `pcm` como o`OutputFormat`, o `ContentType` retornado estará audio/pcm em um formato little-endian assinado de 16 bits, 1 canal (mono).
- Se você solicitar `mu-law` como o`OutputFormat`, o `ContentType` retornado é áudio/mulaw.
- Se você solicitar `a-law` como o`OutputFormat`, o `ContentType` retornado é áudio/alaw.
- Se você solicitar `json` como o`OutputFormat`, o `ContentType` retornado é application/x-json-stream.

RequestCharacters

Número de caracteres sintetizados.

A resposta retorna as informações a seguir como corpo HTTP.

AudioStream

Fluxo contendo a fala sintetizada.

Erros

EngineNotSupportedException

Esse mecanismo não é compatível com a voz que você designou. Escolha uma nova voz compatível com o mecanismo ou troque-o e reinicie a operação.

Código de status HTTP: 400

InvalidSampleRateException

A taxa de amostragem especificada é inválida.

Código de status HTTP: 400

InvalidSsmlException

O SSML fornecido é inválido. Verifique a sintaxe do SSML, a ortografia e valores das tags tente novamente.

Código de status HTTP: 400

LanguageNotSupportedException

O idioma especificado não é compatível com o Amazon Polly nessa capacidade.

Código de status HTTP: 400

LexiconNotFoundException

O Amazon Polly não encontrou o léxico especificado. Isso pode ser causado por um léxico inexistente, pelo nome escrito incorretamente ou pela especificação de um léxico que está em uma região diferente.

Verifique se o léxico existe, se está na região (consulte ListLexicons) e se o nome dele está escrito corretamente. Tente novamente.

Código de status HTTP: 404

MarksNotSupportedForFormatException

As marcas de fala não são compatíveis com o `OutputFormat` selecionado. As marcas de fala estão disponíveis apenas para conteúdo em formato `json`.

Código de status HTTP: 400

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Código de status HTTP: 500

SsmlMarksNotSupportedForTextTypeException

As marcas de fala SSML não são compatíveis com a entrada de texto sem formatação.

Código de status HTTP: 400

TextLengthExceededException

O valor do parâmetro Texto é maior que os limites aceitos. Para a API do `SynthesizeSpeech`, o limite para texto de entrada é de no máximo 6 mil caracteres, dos quais não mais do que 3 mil podem ser faturados. Para a API `StartSpeechSynthesisTask`, o máximo é de 200 mil caracteres, dos quais não mais do que 100 mil podem ser faturados. Tags SSML não são contadas como caracteres cobrados.

Código de Status HTTP: 400

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS Interface de linha de comando V2
- AWS SDK para .NET V4
- AWS SDK para C++
- AWS SDK para Go v2
- AWS SDK para Java V2
- AWS SDK para JavaScript V3
- AWS SDK para Kotlin
- AWS SDK para PHP V3
- AWS SDK para Python
- AWS SDK para Ruby V3

Tipos de dados

Os seguintes tipos de dados são compatíveis:

- AudioEvent
- CloseStreamEvent
- FlushStreamConfiguration
- Lexicon
- LexiconAttributes
- LexiconDescription
- StartSpeechSynthesisStreamActionStream
- StartSpeechSynthesisStreamEventStream
- StreamClosedEvent
- SynthesisTask
- TextEvent
- ThrottlingReason
- ValidationExceptionField
- Voice

AudioEvent

Contém uma parte dos dados de áudio sintetizados.

Conteúdo

AudioChunk

Uma parte dos dados de áudio sintetizados codificados no formato especificado pelo parâmetro `OutputFormat`

Tipo: Objeto de dados binários codificado em Base64

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em uma linguagem específica AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

CloseStreamEvent

Indica o fim do fluxo de entrada. Depois de enviar esse evento, o stream de entrada será fechado e todo o áudio será retornado.

Conteúdo

Os membros dessa estrutura de exceção dependem do contexto.

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

FlushStreamConfiguration

Configuração que controla quando os dados de áudio sintetizados são enviados no fluxo de saída.

Conteúdo

Force

Especifica se o mecanismo de síntese deve ser forçado a gravar imediatamente dados de áudio em buffer no fluxo de saída.

Tipo: booliano

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Lexicon

Fornece o nome e o conteúdo do léxico em formato de string. Para mais informações, consulte Especificação do léxico de pronúncia (PLS) Versão 1.0 no site do W3C.

Conteúdo

Content

Conteúdo do léxico em formato de string. O conteúdo de um léxico deve estar no formato PLS.

Tipo: sequência

Obrigatório: Não

Name

Nome do léxico.

Tipo: sequência

Padrão: : `[0-9A-Za-z]{1,20}`

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos SDKs específicos de linguagem AWS, consulte:

- AWS SDK para C++
- AWS SDK for Java V2
- AWS SDK para Ruby V3

LexiconAttributes

Contém metadados que descrevem o léxico, como o número de lexemas, o código do idioma e assim por diante. Para mais informações, consulte Gerenciamento de léxicos.

Conteúdo

Alphabet

Alfabeto fonético usado no léxico. Os valores válidos são `ipa` e `x-sampa`.

Tipo: sequência

Obrigatório: não

LanguageCode

Código de idioma ao qual o léxico se aplica. Um léxico com um código de idioma como “en” seria aplicado a todos os idiomas ingleses (en-GB, en-US, en-AUS, en-WLS e assim por diante).

Tipo: sequência

Valores válidos: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG`

Obrigatório: não

LastModified

Quando o léxico da data foi modificado pela última vez (um valor de carimbo de data/hora).

Tipo: carimbo de data/hora

Obrigatório: não

LexemesCount

Número de lexemas no léxico.

Tipo: inteiro

Obrigatório: não

LexiconArn

O Nome do recurso da Amazon (ARN) da função.

Tipo: sequência

Obrigatório: não

Size

O tamanho total do léxico, em caracteres.

Tipo: inteiro

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

LexiconDescription

Descreve o conteúdo do léxico.

Conteúdo

Attributes

Fornece metadados do léxico.

Tipo: objeto LexiconAttributes

Obrigatório: Não

Name

Nome do léxico.

Tipo: sequência

Padrão: : `[0-9A-Za-z]{1,20}`

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em um dos SDKs específicos de linguagem AWS, consulte:

- AWS SDK para C++
- AWS SDK for Java V2
- AWS SDK para Ruby V3

StartSpeechSynthesisStreamActionStream

Fluxo de eventos de entrada para enviar eventos de entrada e controle para gerenciar a síntese bidirecional da fala.

Conteúdo

CloseStreamEvent

Um evento que indica o fim do fluxo de entrada.

Tipo: objeto CloseStreamEvent

Obrigatório: não

TextEvent

Um evento de texto contendo conteúdo a ser sintetizado.

Tipo: objeto TextEvent

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em uma linguagem específica AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

StartSpeechSynthesisStreamEventStream

Fluxo de eventos de saída que contém dados de áudio sintetizados e eventos de status do fluxo.

Conteúdo

AudioEvent

Um evento de áudio contendo fala sintetizada.

Tipo: objeto AudioEvent

Obrigatório: não

ServiceFailureException

Uma condição desconhecida causou uma falha no serviço.

Tipo: Exceção

Código de status HTTP: 500

Obrigatório: não

ServiceQuotaExceededException

Uma exceção indicando que uma cota de serviço seria excedida.

Tipo: Exceção

Código de status HTTP: 402

Obrigatório: não

StreamClosedEvent

Um evento, com informações resumidas, indicando que o stream foi encerrado.

Tipo: objeto StreamClosedEvent

Obrigatório: não

ThrottlingException

Uma exceção indicando que a solicitação foi limitada.

Tipo: Exceção

Código de status HTTP: 400

Obrigatório: não

ValidationException

Uma exceção indicando falha na validação da entrada.

Tipo: Exceção

Código de status HTTP: 400

Obrigatório: Não

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

StreamClosedEvent

Indica que o fluxo de síntese está fechado e fornece informações resumidas.

Conteúdo

RequestCharacters

O número total de caracteres sintetizados durante a sessão de streaming.

Tipo: inteiro

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

SynthesisTask

SynthesisTask objeto que fornece informações sobre uma tarefa de síntese de fala.

Conteúdo

CreationTime

Carimbo de data/hora em que a tarefa de síntese foi iniciada.

Tipo: carimbo de data/hora

Obrigatório: não

Engine

Especifica o mecanismo (`standard`, `neural`, `long-form` ou `generative`) que o Amazon Polly usará ao processar texto de entrada para a síntese de fala. Usar uma voz que não seja compatível com o mecanismo selecionado resultará em erro.

Tipo: string

Valores válidos: `standard | neural | long-form | generative`

Obrigatório: não

LanguageCode

Código de idioma opcional para uma tarefa de síntese. É necessário apenas se estiver usando uma voz bilíngue, como Aditi, que pode ser usada para inglês indiano (en-IN) ou hindi (hi-IN).

Se uma voz bilíngue for usada e nenhum código de idioma for especificado, o Amazon Polly usará o idioma padrão da voz bilíngue. O idioma padrão para qualquer voz é aquele retornado pela DescribeVoicesoperação do `LanguageCode` parâmetro. Por exemplo, se nenhum código de idioma for especificado, Aditi usará inglês indiano em vez de hindi.

Tipo: string

Valores válidos: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG`

Obrigatório: não

LexiconNames

Lista de um ou mais nomes de léxico de pronúncia que você deseja que o serviço aplique durante a síntese. O léxico só será aplicado ao texto se o idioma especificado nele for o mesmo que o da voz.

Tipo: matriz de strings

Membros da matriz: número máximo de 5 itens.

Padrão: `[0-9A-Za-z]{1,20}`

Obrigatório: não

OutputFormat

O formato no qual a saída retornada será codificada. Para transmissão de áudio, serão mp3, ogg_vorbis, ogg_opus, mu-law, a-law ou pcm. Marcas de fala serão JSON.

Tipo: string

Valores válidos: `json | mp3 | ogg_opus | ogg_vorbis | pcm | mulaw | alaw`

Obrigatório: não

OutputUri

Caminho para o arquivo de fala de saída.

Tipo: string

Obrigatório: não

RequestCharacters

Número de caracteres faturáveis sintetizados.

Tipo: inteiro

Obrigatório: não

SampleRate

A frequência de áudio especificada em Hz.

Os valores válidos para MP3 e ogg_vorbis são 8000, 16000, 22050 e 24000. O valor padrão para vozes padrão é 22050. O valor padrão para vozes neurais é 24000. O valor padrão para vozes em formato longo é “24.000”. O valor padrão para vozes generativas é “24000”.

Os valores válidos para pcm são 8000 e 16000. O valor padrão é 16000.

O valor válido para ogg_opus é “48000".

O valor válido para mu-law e a-law é “8000".

Tipo: string

Obrigatório: não

SnsTopicArn

O ARN do tópico do SNS é usado opcionalmente para fornecer notificação de status para uma tarefa de síntese de fala.

Tipo: sequência

Padrão: `^arn:aws(-(cn|iso(-b)?|us-gov))?:sns:[a-z0-9_-]{1,50}:\d{12}:[a-zA-Z0-9_-]{1,251}([a-zA-Z0-9_-]{0,5}|\.fifo)$`

Obrigatório: não

SpeechMarkTypes

Tipo de marcas de fala retornadas para o texto de entrada.

Tipo: matriz de strings

Membros da matriz: número máximo de 4 itens.

Valores válidos: `sentence | ssml | viseme | word`

Obrigatório: não

TaskId

O identificador gerado pelo Amazon Polly para a tarefa de síntese de fala.

Tipo: sequência

Padrão: `^[a-zA-Z0-9_-]{1,100}$`

Obrigatório: não

TaskStatus

Status atual da tarefa de síntese de fala individual.

Tipo: string

Valores válidos: `scheduled | inProgress | completed | failed`

Obrigatório: não

TaskStatusReason

Motivo do status atual de uma tarefa de síntese de fala específica, incluindo erros se a tarefa falhar.

Tipo: string

Obrigatório: não

TextType

Especifica se o texto de entrada é simples ou SSML. O valor padrão é texto sem formatação.

Tipo: string

Valores válidos: `ssml | text`

Obrigatório: não

VoiceId

ID de voz a ser usado para a síntese.

Tipo: string

Valores válidos: `Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique | Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria`

```
| Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal |
Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi
| Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie
| Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu | Jitka | Sabrina
| Jasmine | Jihye | Ambre | Beatrice | Florian | Lennart | Lorenzo |
Tiffany
```

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

TextEvent

Contém conteúdo de texto para ser sintetizado em fala.

Conteúdo

Text

O conteúdo do texto a ser sintetizado. Se você especificar `ssml` como `TextType`, siga o formato SSML para o texto de entrada.

Tipo: String

Obrigatório: Sim

FlushStreamConfiguration

Configuração para controlar quando o áudio sintetizado é liberado para o fluxo de saída.

Tipo: objeto FlushStreamConfiguration

Obrigatório: não

TextType

Especifica se o texto de entrada é simples ou SSML. Padrão: texto sem formatação.

Tipo: string

Valores válidos: `ssml | text`

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

ThrottlingReason

Fornece informações sobre um motivo específico de limitação.

Conteúdo

reason

O código do motivo que explica por que a solicitação foi limitada.

Tipo: string

Obrigatório: não

resource

O recurso que causou a limitação.

Tipo: string

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

ValidationExceptionField

Informações sobre um campo que falhou na validação.

Conteúdo

message

Uma mensagem descrevendo por que o campo falhou na validação.

Tipo: String

Obrigatório: Sim

name

O nome do campo que falhou na validação.

Tipo: String

Obrigatório: Sim

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Voice

Descrição da voz.

Conteúdo

AdditionalLanguageCodes

Códigos adicionais para idiomas disponíveis para a voz especificada, além do idioma padrão.

Por exemplo, o idioma padrão para Aditi é o inglês indiano (en-IN) porque ele foi usado pela primeira vez para esse idioma. Como Aditi é bilíngue e fluente em inglês indiano e hindi, esse parâmetro mostraria o código `hi-IN`.

Tipo: matriz de strings

Valores válidos: `arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG`

Obrigatório: não

Gender

Gênero da voz.

Tipo: sequências

Valores válidos: `Female | Male`

Obrigatório: não

Id

ID de voz atribuída ao Amazon Polly. Esse é o ID que você especifica ao chamar a operação `SynthesizeSpeech`.

Tipo: sequência

Valores válidos: `Aditi | Amy | Astrid | Bianca | Brian | Camila | Carla | Carmen | Celine | Chantal | Conchita | Cristiano | Dora | Emma | Enrique`

| Ewa | Filiz | Gabrielle | Geraint | Giorgio | Gwyneth | Hans | Ines | Ivy | Jacek | Jan | Joanna | Joey | Justin | Karl | Kendra | Kevin | Kimberly | Lea | Liv | Lotte | Lucia | Lupe | Mads | Maja | Marlene | Mathieu | Matthew | Maxim | Mia | Miguel | Mizuki | Naja | Nicole | Olivia | Penelope | Raveena | Ricardo | Ruben | Russell | Salli | Seoyeon | Takumi | Tatyana | Vicki | Vitoria | Zeina | Zhiyu | Aria | Ayanda | Arlet | Hannah | Arthur | Daniel | Liam | Pedro | Kajal | Hiujin | Laura | Elin | Ida | Suvi | Ola | Hala | Andres | Sergio | Remi | Adriano | Thiago | Ruth | Stephen | Kazuha | Tomoko | Niamh | Sofie | Lisa | Isabelle | Zayd | Danielle | Gregory | Burcu | Jitka | Sabrina | Jasmine | Jihye | Ambre | Beatrice | Florian | Lennart | Lorenzo | Tiffany

Obrigatório: não

LanguageCode

Código de idioma da voz.

Tipo: sequência

Valores válidos: arb | cmn-CN | cy-GB | da-DK | de-DE | en-AU | en-GB | en-GB-WLS | en-IN | en-US | es-ES | es-MX | es-US | fr-CA | fr-FR | is-IS | it-IT | ja-JP | hi-IN | ko-KR | nb-NO | nl-NL | pl-PL | pt-BR | pt-PT | ro-RO | ru-RU | sv-SE | tr-TR | en-NZ | en-ZA | ca-ES | de-AT | yue-CN | ar-AE | fi-FI | en-IE | nl-BE | fr-BE | cs-CZ | de-CH | en-SG

Obrigatório: não

LanguageName

O nome legível do idioma em inglês.

Tipo: sequência

Obrigatório: não

Name

Nome da voz (por exemplo, Salli, Kendra, etc.). Isso fornece um nome de voz legível por humanos que você pode exibir no aplicativo.

Tipo: sequência

Obrigatório: não

SupportedEngines

Especifica quais mecanismos (`standard`, `neural`, `long-form` ou `generative`) são compatíveis com determinada voz.

Tipo: matriz de strings

Valores válidos: `standard | neural | long-form | generative`

Obrigatório: não

Consulte também

Para obter mais informações sobre como usar essa API em uma das linguagens específicas AWS SDKs, consulte o seguinte:

- AWS SDK para C++
- AWS SDK para Java V2
- AWS SDK para Ruby V3

Tipos de erro comuns

Esta seção lista os tipos de erro comuns que esse AWS serviço pode retornar. Nem todos os serviços retornam todos os tipos de erro listados aqui. Para saber os erros específicos de uma ação de API para esse serviço, consulte o tópico sobre a ação de API em questão.

AccessDeniedException

Você não tem permissão para realizar esta ação. Verifique se sua política do IAM inclui as permissões necessárias.

Código de status HTTP: 403

ExpiredTokenException

O token de segurança incluído na solicitação expirou. Solicite um novo token de segurança e tente novamente.

Código de status HTTP: 403

IncompleteSignature

A assinatura da solicitação não está em conformidade com os AWS padrões. Verifique se você está usando AWS credenciais válidas e se sua solicitação está formatada corretamente. Se você estiver usando um SDK, verifique se ele está atualizado.

Código de status HTTP: 403

InternalFailure

A solicitação não pode ser processada no momento devido a um problema interno do servidor. Tente novamente mais tarde. Se o problema persistir, entre em contato com o AWS Support.

Código de status HTTP: 500

MalformedHttpRequestException

O corpo da solicitação não pode ser processado. Isso geralmente acontece quando o corpo da solicitação não pode ser descompactado usando o algoritmo de codificação de conteúdo especificado. Verifique se o cabeçalho da codificação do conteúdo corresponde ao formato de compactação usado.

Código de status HTTP: 400

NotAuthorized

Você não tem permissões para realizar essa ação. Verifique se sua política do IAM inclui as permissões necessárias.

Código de status HTTP: 401

OptInRequired

Sua AWS conta precisa de uma assinatura para esse serviço. Verifique se você ativou o serviço em sua conta.

Código de status HTTP: 403

RequestAbortedException

A solicitação foi cancelada antes que uma resposta pudesse ser retornada. Isso normalmente acontece quando o cliente fecha a conexão.

Código de status HTTP: 400

RequestEntityTooLargeException

A entidade de solicitação é muito grande. Reduza o tamanho do corpo da solicitação e tente novamente.

Código de status HTTP: 413

RequestTimeoutException

O tempo limite da solicitação atingiu o tempo limite. O servidor não recebeu a solicitação completa dentro do prazo esperado. Tente novamente.

Código de status HTTP: 408

ServiceUnavailable

O serviço está temporariamente indisponível. Tente novamente mais tarde.

Código de status HTTP: 503

ThrottlingException

Sua taxa de solicitação é muito alta. As solicitações de repetição AWS SDKs automática que recebem essa exceção. Reduza a frequência de solicitações.

Código de status HTTP: 400

UnknownOperationException

A ação ou operação não é reconhecida. Verifique se o nome da ação está escrito corretamente e se é compatível com a versão da API que você está usando.

Código de status HTTP: 404

UnrecognizedClientException

O certificado X.509 ou ID da chave de AWS acesso que você forneceu não existe em nossos registros. Verifique se você está usando credenciais válidas e se elas não expiraram.

Código de status HTTP: 403

ValidationError

A entrada não atende ao formato ou às restrições exigidas. Verifique se todos os parâmetros necessários estão incluídos e se os valores são válidos.

Código de status HTTP: 400

Parâmetros gerais

A lista a seguir contém os parâmetros que todas as ações usam para assinar solicitações do Signature versão 4 com uma string de consulta. Todos os parâmetros específicos de uma ação são listados no tópico para a ação. Para obter mais informações sobre o Signature versão 4, consulte Solicitações de AWS API de assinatura no Guia do usuário do IAM.

X-Amz-Algorithm

O algoritmo de hash que foi usado para criar a assinatura da solicitação.

Condição: especifique esse parâmetro quando incluir as informações de autenticação em uma string de consulta em vez de no cabeçalho da autorização HTTP.

Tipo: string

Valores válidos: `AWS4-HMAC-SHA256`

Obrigatório: condicional

X-Amz-Credential

O valor de escopo da credencial, uma string que inclui a sua chave de acesso, a data, a região visada, o serviço que está sendo solicitado e uma sequência de encerramento ("aws4_request"). O valor é expresso no seguinte formato: chave_acesso/AAAAMMDD/região/serviço/aws4_request.

Para obter mais informações, consulte Criar uma solicitação de AWS API assinada no Guia do usuário do IAM.

Condição: especifique esse parâmetro quando incluir as informações de autenticação em uma string de consulta em vez de no cabeçalho da autorização HTTP.

Tipo: string

Obrigatório: Condicional

X-Amz-Date

A data usada para criar a assinatura. O formato deve ser o formato básico ISO 8601 (AAAAMMDD'T'HHMMSS'Z'). Por exemplo, a data e hora a seguir é um X-Amz-Date valor válido:`20120325T120000Z`.

Condição: X-Amz-Date é opcional para todas as solicitações; ela pode ser usada para substituir a data usada para assinar solicitações. Se o cabeçalho da data for especificado no formato básico ISO 8601, não X-Amz-Date é necessário. Quando X-Amz-Date usado, ele sempre substitui o valor do cabeçalho da data. Para obter mais informações, consulte Elementos de uma assinatura de solicitação de AWS API no Guia do usuário do IAM.

Tipo: string

Obrigatório: Condicional

X-Amz-Security-Token

O token de segurança temporário obtido por meio de uma chamada para AWS Security Token Service (AWS STS). Para obter uma lista de serviços que oferecem suporte a credenciais de segurança temporárias do AWS STS, consulte Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Condição: se você estiver usando credenciais de segurança temporárias do AWS STS, deverá incluir o token de segurança.

Tipo: string

Obrigatório: Condicional

X-Amz-Signature

Especifica a assinatura com codificação hexadecimal que foi calculada com base na string a ser assinada e na chave de assinatura derivada.

Condição: especifique esse parâmetro quando incluir as informações de autenticação em uma string de consulta em vez de no cabeçalho da autorização HTTP.

Tipo: string

Obrigatório: Condicional

X-Amz-SignedHeaders

Especifica todos os cabeçalhos HTTP que foram incluídos como parte da solicitação canônica. Para obter mais informações sobre a especificação de cabeçalhos assinados, consulte Criar uma solicitação de AWS API assinada no Guia do usuário do IAM.

Condição: especifique esse parâmetro quando incluir as informações de autenticação em uma string de consulta em vez de no cabeçalho da autorização HTTP.

Tipo: string

Obrigatório: Condicional

Histórico da documentação do Amazon Polly

A tabela a seguir descreve as mudanças importantes em cada versão do Guia do desenvolvedor do Amazon Polly. Para receber notificações sobre atualizações dessa documentação, é possível inscrever-se em um feed RSS.

- Última atualização da documentação: 28 de maio de 2026

Alteração	Descrição	Data
Nova região adicionada para vozes neurais	O Amazon Polly agora está disponível na região Ásia-Pacífico (Tailândia). AWS Esta região suporta vozes neurais TTS (NTTS). Para obter mais informações, consulte Vozes neurais.	28 de maio de 2026
Nova região adicionada para vozes generativas	As vozes generativas e o streaming bidirecional do Amazon Polly agora estão disponíveis em outra AWS região: Europa (Zurique). Para obter mais informações, consulte Vozes generativas.	28 de maio de 2026
Streaming bidirecional disponível em novas regiões	O streaming bidirecional do Amazon Polly agora está disponível em duas AWS regiões adicionais: Europa (Londres) e Canadá (Central) . Para obter mais informações, consulte Vozes generativas.	20 de abril de 2026
Política gerenciada atualizada	O Amazon Polly atualizou a política gerenciada `AmazonPollyReadOnl`	19 de março de 2026

	yAccess para incluir a permissão para a síntese de voz por streaming bidirecional. Para obter informações, consulte as atualizações do Amazon Polly para políticas AWS gerenciadas.	
Adição de novas vozes para a conversão generativa de texto em fala	O Amazon Polly agora fornece dez vozes generativas adicionais: Brian, Aria, Jasmine, Tiffany, Ambre, Florian, Sabrina, Lennart, Beatrice, Lorenzo. Consulte Generative voices para conferir uma lista de vozes de TTS generativa.	19 de março de 2026
Novas regiões adicionadas para vozes generativas	As vozes generativas do Amazon Polly agora estão disponíveis em duas AWS regiões adicionais: Europa (Londres) e Canadá (Central). Para obter mais informações, consulte Vozes generativas.	19 de março de 2026
Novas regiões adicionadas para vozes generativas	As vozes generativas do Amazon Polly agora estão disponíveis em três AWS regiões adicionais: Ásia-Pacífico (Tóquio), Ásia-Pacífico (Seul) e Ásia-Pacífico (Cingapura). Para obter mais informações, consulte Vozes generativas.	18 de novembro de 2025

Adição de novas vozes para a conversão generativa de texto em fala	O Amazon Polly agora fornece seis vozes generativas adicionais: Seoyeon, Camila, Hannah, Niamh, Laura e Lisa. Consulte Generative voices para conferir uma lista de vozes de TTS generativa.	14 de novembro de 2025
Nova região adicionada para vozes neurais	O Amazon Polly agora está disponível na região da Europa (Zurique). AWS Esta região suporta vozes neurais TTS (NTTS). Para obter mais informações, consulte Vozes neurais.	20 de outubro de 2025
Adição de novas vozes para a conversão generativa de texto em fala	O Amazon Polly agora fornece sete vozes generativas adicionais: Salli, Isabelle, Céline, Liam, Gabrielle, Ola e Ewa. Consulte Generative voices para conferir uma lista de vozes de TTS generativa.	26 de agosto de 2025
Nova região adicionada para vozes neurais e padrão	Amazon Polly já está disponível na região da AWS Ásia-Pacífico (Malásia) . Esta região é compatíve l com vozes padrão e TTS neurais (NTTS). Consulte mais informações em Vozes neurais e Vozes padrão.	27 de março de 2025

Adição de nova voz para a conversão de texto em fala neural	O Amazon Polly agora oferece mais uma voz coreana: Jihye. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	26 de março de 2025
Nova região adicionada para vozes neurais e padrão	O Amazon Polly agora está disponível na região da AWS Europa (Espanha). Esta região é compatível com vozes padrão e TTS neurais (NTTS). Consulte mais informações em Vozes neurais e Vozes padrão.	18 de fevereiro de 2025
Adição de nova voz para a conversão de texto em fala neural	O Amazon Polly agora fornece uma voz adicional em inglês (singapurense): Jasmine. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	11 de fevereiro de 2025
Adição de novas vozes para a conversão generativa de texto em fala	O Amazon Polly agora fornece sete novas vozes generativ as: Pedro, Andrés, Sergio, Daniel, Kajal, Rémi e Bianca. Consulte Generative voices para conferir uma lista de vozes de TTS generativa.	21 de novembro de 2024

Adição de novas vozes em formato longo	O Amazon Polly agora fornece mais vozes de formato longo. Adição de uma voz em formato longo em inglês (EUA) e duas em espanhol (Espanha): Patrick, Alba e Raúl. Veja Long-formas vozes para obter uma lista de todas as vozes de formato longo.	14 de novembro de 2024
Adição de novas vozes para a conversão generativa de texto em fala	O Amazon Polly agora fornece seis novas vozes generativas: Ayanda, Léa, Lucia, Mía, Lupe e Vicki. Consulte Generative voices para conferir uma lista de vozes de TTS generativa.	6 de novembro de 2024
Adição de novas vozes para a conversão generativa de texto em fala	O Amazon Polly agora fornece quatro novas vozes generativ as: Olivia, Danielle, Joanna e Stephen. Consulte Generative voices para conferir uma lista de vozes de TTS generativa.	10 de outubro de 2024
Adição de novos idiomas para a conversão de texto em fala	O Amazon Polly agora fornece dois novos idiomas de TTS: tcheco (cs-CZ) e alemão (suíço) (de-CH). Consulte Supported languages para ver uma lista de idiomas.	26 de setembro de 2024
Novas vozes adicionadas para NTTS	O Amazon Polly agora fornece duas novas vozes de NTTS: Jitka e Sabrina. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	27 de agosto de 2024

Adição de novo mecanismo de voz generativa	O Amazon Polly agora oferece um mecanismo de voz generativa desenvolvido para conteúdo mais longo, com três vozes em inglês em uma variante generativa: Amy, Matthew e Ruth. Consulte Generative voices para ter mais informações.	28 de março de 2024
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz de NTTS turca Burcu. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	14 de fevereiro de 2024
Novo mecanismo de voz em formato longo adicionado	O Amazon Polly agora oferece um mecanismo de voz em formato longo desenvolvido para conteúdo mais longo, com três vozes en-US: Danielle, Gregory e Ruth. Consulte Long-formas vozes para obter mais informações.	16 de novembro de 2023
Novas vozes adicionadas para NTTS	O Amazon Polly agora fornece duas novas vozes para NTTS em inglês dos EUA: Danielle e Gregory. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	5 de outubro de 2023
Amazon Polly para Windows	O plug-in Amazon Polly SAPI não será mais compatível.	26 de setembro de 2023

Orientação de cota atualizada para o Amazon Polly	Guia atualizado de cotas do Amazon Polly. Adicionados exemplos e esclarecimentos. Consulte Cotas no Amazon Polly para ver as atualizações.	17 de agosto de 2023
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz NTTS do Golfo Árabe, Zayd. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	16 de agosto de 2023
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz NTTS em francês belga, Isabelle. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	1º de agosto de 2023
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz NTTS belga holandesa (flamenga) de Lisa. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	7 de junho de 2023
Novas vozes adicionadas para NTTS	O Amazon Polly agora fornece duas novas vozes para NTTS: inglês irlandês (Niamh) e dinamarquês (Sofie). Consulte Vozes neurais para obter uma lista de vozes para NTTS.	30 de maio de 2023
Orientação do IAM atualizada para o Amazon Polly	Guia atualizado para alinhamento com as práticas recomendadas do IAM. Para saber mais, consulte Práticas recomendadas de segurança no IAM.	19 de abril de 2023

WordPress update	O WordPress plug-in Amazon Polly não será mais suportado.	6 de abril de 2023
Nova região adicionada	O Amazon Polly agora está disponível na região da AWS Ásia-Pacífico (Osaka). Esta região é compatível com TTS neurais (NTTS). Para obter mais informações, consulte Compatibilidade de recursos e regiões para uma lista de regiões que oferecem compatibilidade para NTTS.	5 de abril de 2023
Novas vozes adicionadas para NTTS	O Amazon Polly agora fornece duas novas vozes japonesas para NTTS: Kazuha e Tomoko. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	7 de fevereiro de 2023
Novas vozes adicionadas para NTTS	O Amazon Polly agora fornece duas novas vozes para NTTS em inglês dos EUA: Stephen e Ruth. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	31 de janeiro de 2023
Novas vozes adicionadas para NTTS	O Amazon Polly agora fornece novas vozes para NTTS para: português do Brasil (Thiago), espanhol castelhano (Sergio), francês (Rémi), italiano (Adriano) e espanhol mexicano (Andrés). Consulte Vozes neurais para obter uma lista de vozes para NTTS.	24 de janeiro de 2023

Novas vozes adicionadas para NTTS	O Amazon Polly agora fornece vozes para NTTS para árabe (Hala) e polonês (Ola). Consulte Vozes neurais para obter uma lista de vozes para NTTS.	17 de novembro de 2022
AWS PrivateLink Suporte de lançamento	O Amazon Polly agora fornece AWS PrivateLink suporte. Consulte Usar o Amazon Polly com endpoints da VPC para saber mais.	9 de novembro de 2022
Novas vozes e idiomas adicionados para NTTS	O Amazon Polly agora fornece vozes para NTTS para finlandês (Suvi), norueguês (Ida) e sueco (Elin). Consulte Vozes neurais para obter uma lista de vozes para NTTS.	8 de novembro de 2022
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz holandesa para NTTS Laura. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	2 de novembro de 2022
Nova região adicionada	O Amazon Polly agora está disponível na região da AWS Europa (Paris). Esta região é compatível com TTS neurais (NTTS). Para obter mais informações, consulte Compatibilidade de recursos e regiões para uma lista de regiões que oferecem compatibilidade para NTTS.	22 de setembro de 2022

Nova voz e idioma adicionados para NTTS	O Amazon Polly agora fornece a voz cantonesa NTTS Hiujin. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	20 de setembro de 2022
Nova região adicionada	O Amazon Polly agora está disponível na região da AWS Ásia-Pacífico (Mumbai). Esta região é compatível com TTS neurais (NTTS). Para obter mais informações, consulte Compatibilidade de recursos e regiões para uma lista de regiões que oferecem compatibilidade para NTTS.	1º de setembro de 2022
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz em mandarim Zhiyu como uma voz NTTS. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	23 de agosto de 2022
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz em hindi NTTS Kajal. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	27 de julho de 2022

Novas vozes adicionadas para NTTS	O Amazon Polly agora fornece vozes para NTTS para espanhol dos EUA (Pedro), alemão (Daniel), francês canadense (Liam) e inglês britânico (Arthur). Consulte Vozes neurais para obter uma lista de vozes para NTTS.	28 de junho de 2022
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz em português brasileiro Vitória como uma voz NTTS. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	27 de abril de 2022
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz em português europeu Inês como uma voz NTTS. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	26 de abril de 2022
Nova voz e idioma adicionados para NTTS	O Amazon Polly agora fornece o idioma alemão austríaco e a voz NTTS Hannah. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	19 de abril de 2022
Novas vozes e idiomas adicionados para NTTS	O Amazon Polly agora fornece a voz espanhola mexicana Mia como uma voz NTTS. Um novo idioma, catalão, foi adicionado junto com a voz NTTS Arlet. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	22 de março de 2022

Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz japonesa Takumi como uma voz NTTS. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	6 de dezembro de 2021
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz francesa Léa como uma voz NTTS. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	18 de novembro de 2021
Novas vozes adicionadas para NTTS	O Amazon Polly agora fornece a voz italiana Bianca e a voz em espanhol europeu Lucia como vozes para NTTS. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	8 de novembro de 2021
Nova voz adicionada para NTTS	O Amazon Polly agora fornece uma nova voz em inglês sul-africano, Ayanda. A voz está disponível somente como voz NTTS. Consulte Vozes neurais para obter uma lista de vozes para NTTS.	1º de setembro de 2021

Nova região adicionada	O Amazon Polly agora está disponível na região da AWS África (Cidade do Cabo). Esta região é compatível com TTS neurais (NTTS). Para obter mais informações, consulte Compatibilidade de recursos e regiões para uma lista de regiões que oferecem compatibilidade para NTTS.	1º de setembro de 2021
Novo idioma e voz adicionados	O Amazon Polly agora é compatível com o inglês neozelandês (en-NZ). Uma nova voz para NTTS, Aria, fala inglês neozelandês e uma seleção de palavras maori.	24 de agosto de 2021
Novo atributo	O Amazon Polly torna o estilo de conversação a versão padrão das vozes neurais de Matthew e Joanna. Removemos as referências ao estilo de conversação.	28 de junho de 2021
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz alemã Vicki como uma voz NTTS.	15 de junho de 2021

Nova voz adicionada	Uma nova voz feminina, Gabrielle, foi adicionad a à localidade francesa canadense (fr-CA). Essa voz é de alta qualidade e está disponível somente como voz NTTS. Como todas as vozes neurais, ela só está disponível em determinadas regiões. Para obter uma lista de regiões, consulte Compatibilidade de recursos e regiões.	1º de junho de 2021
Nova voz adicionada para NTTS	O Amazon Polly agora fornece a voz coreana Seoyeon como uma voz NTTS.	11 de maio de 2021
Nova região adicionada para NTTS	O Amazon Polly agora oferece suporte a TTS neurais (NTTS) na região do Canadá (Central). AWS Para obter mais informações, consulte Compatibilidade de recursos e regiões para NTTS.	17 de março de 2021
Nova voz disponível para o estilo de locutor	Além das vozes de Matthew e Joanna para o estilo de fala de locutor, o Amazon Polly agora oferece uma opção adicional para esse estilo de fala. Usando o mecanismo neural, você pode usar a voz de Amy em inglês britânico para o estilo de locutor. Para obter mais informações, consulte Estilos de fala NTTS.	10 de novembro de 2020

Novas regiões adicionadas para NTTS	Além das regiões existentes para NTTS (us-east-1, us-west-2, eu-west-1 e ap-southeast-2), as vozes neurais agora são compatíveis em quatro regiões adicionais: ap-northeast-1 (Tóquio), ap-southeast-1 (Singapura), eu-central-1 (Frankfurt) e eu-west-2 (Londres). Para obter mais informações, consulte Compatibilidade de recursos e regiões para NTTS.	3 de setembro de 2020
Nova voz adicionada	Além das vozes infantis de Ivy e Justin, uma nova voz masculina, Kevin, foi adicionada ao inglês dos EUA (en-US). Essa nova voz é de alta qualidade e está disponível somente como voz NTTS. Como todas as vozes neurais, ela só é compatível em quatro regiões: us-east-1 (Virgínia do Norte), us-west-2 (Oregon), eu-west-1 (Irlanda) e ap-southeast-2 (Sydney). Para obter mais informações, consulte Vozes NTTS.	16 de junho de 2020

Nova voz disponível para o estilo de locutor	Além das vozes de Matthew e Joanna para o estilo de fala de locutor, o Amazon Polly agora oferece uma opção adicional para esse estilo de fala. Usando o mecanismo neural, você pode usar a voz de Lupe em espanhol (EUA) para o estilo de locutor. Para obter mais informações, consulte Estilos de fala NTTS.	16 de abril de 2020
Novo atributo	Além do estilo de fala de locutor, o Amazon Polly agora fornece um segundo estilo NTTS para ajudar você a sintetizar ainda melhor passagens de texto para fala. O estilo de conversação usa o sistema neural para gerar fala em um estilo de conversação mais amigável e expressivo que pode ser usado em muitos casos de uso. Para obter mais informações, consulte Estilos de fala NTTS.	25 de novembro de 2019
Novas vozes adicionadas	Duas novas vozes foram adicionadas: Camila (feminina Portuguese-Brazil) e Lupe (feminina). Spanish-US	23 de outubro de 2019

Novo recurso adicionado	Adição do plug-in Amazon Polly para Windows para incorporar toda a gama de vozes do Amazon Polly aos aplicativos do Windows. SAPI-compliant	26 de setembro de 2019
Novo recurso principal	Além das vozes padrão de conversão de texto em fala (TTS) compatíveis desde o lançamento, o Amazon Polly agora oferece um sistema de TTS neural (NTTS) aprimorado , com vozes de qualidade ainda mais alta, fornecendo vozes de conversão de texto em fala mais naturais e o mais semelhante possível a vozes humanas. Para obter mais informações, consulte Neural Text-to-Speech.	30 de julho de 2019
Novas vozes adicionadas	Novo vozes adicionadas: Lúcia (feminina, espanhol) e Bianca (feminina, italiano).	2 de agosto de 2018
Novo idioma adicionado	Novo idioma adicionado: espanhol mexicano (es-MX). Este idioma usa a voz feminina de Mia.	2 de agosto de 2018
Novo idioma adicionado	Novo idioma adicionado: hindi (hi-IN). Essa voz usa a voz feminina de Aditi, que também é usada para o inglês indiano, o que torna a Aditi a primeira voz bilíngue do Amazon Polly.	2 de agosto de 2018

Novo recurso adicionado	Adição da Síntese de fala de passagens de texto longas (até 100.000 caracteres cobrados).	17 de julho de 2018
Novo recurso SSML adicionado	Adição da Duração máxima para fala sintetizada.	17 de julho de 2018
Nova voz adicionada	Nova voz adicionada: Léa (feminina, francês).	5 de junho de 2018
Expansão de região	Expansão do serviço do Amazon Polly a todas as regiões comerciais.	4 de junho de 2018
Novo idioma adicionado	Novo idioma adicionado: coreano (ko-KR).	4 de junho de 2018
Recurso expandido	O recurso Amazon Polly WordPress Plugin, incluindo a adição de recursos do Amazon Translate.	4 de junho de 2018
Novas vozes adicionadas	Duas novas vozes adicionadas: Aditi (feminina, inglês indiano) e Seoyeon (feminina, coreano).	15 de novembro de 2017
Novo atributo	Adição do novo recurso Marcas de fala, bem como uma expansão dos recursos SSML.	19 de abril de 2017
Novo guia	Esta é a primeira versão do Guia do desenvolvedor do Amazon Polly.	30 de novembro de 2016

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.