Guia do Desenvolvedor do SDK do Cliente Web

Amazon DCV

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon DCV: Guia do Desenvolvedor do SDK do Cliente Web

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais 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 SDK do cliente Web do Amazon DCV? ... 1
Pré-requisitos ... 2
Recursos compatíveis ... 2
Suporte ao navegador ... 2
Convenção de versionamento ... 3
Começar ... 4
Conecte-se a um servidor Amazon DCV e obtenha o primeiro quadro ... 5
Etapa 1: Preparar a página HTML ... 5
Etapa 2: Autenticar, conectar e obter o primeiro quadro ... 6
Bônus: crie automaticamente um formulário de login em HTML ... 9
Trabalhar com os recursos do Amazon DCV ... 10
Entender a função de retorno de chamada featuresUpdate ... 10
Controlar atualizações do atributo ... 11
Usar o SDK de Interface do Usuário da Web do Amazon DCV ... 11
Pré-requisitos ... 12
Etapa 1: Preparar a página HTML ... 13
Etapa 2: Autenticar, conectar e renderizar o componente React do `DCVViewer`. ... 13
Atualizar o AWS-UI para o Cloudscape Design System ... 18
Referência do SDK ... 19
Módulo DCV ... 19
Métodos ... 19
Membros ... 22
Definições de tipo e retorno de chamada ... 27
Classe de conexão ... 70
Métodos ... 19
Classe de autenticação ... 99
Métodos ... 19
Classe de recurso ... 101
Métodos ... 19
SDK de Interface do Usuário da Web do Amazon DCV ... 102
Componentes ... 102
Notas de versão e histórico de documentos ... 110
Notas da versão ... 110
1.8.7 — 31 de outubro de 2024 ... 111

1.8.4 — 1º de outubro de 2024 111
1.5.10: 19 de dezembro de 2023 112
1.5.6 — 9 de novembro de 2023 112
1.4.4 — 29 de junho de 2023 113
1.4.0 — 28 de março de 2023 114
1.3.1 — 9 de dezembro de 2022 115
1.3.0 — 11 de novembro de 2022 116
1.2.1 — 21 de julho de 2022 117
1.2.0 — 29 de junho de 2022 117
1.1.3 — 23 de maio de 2022 118
1.1.2 — 19 de maio de 2022 118
1.1.1 — 23 de março de 2022 119
1.1.0 — 23 de fevereiro de 2022 119
1.0.4 — 20 de dezembro de 2021 120
1.0.3 — 1º de setembro de 2021 120
1.0.2 — 30 de julho de 2021 121
1.0.1 — 31 de maio de 2021 122
1.0.0 — 24 de março de 2021 122
Histórico do documento 122
..... cxxv

O que é o SDK do cliente Web do Amazon DCV?

> **Note**
>
> O Amazon DCV era conhecido anteriormente como NICE DCV.

O Amazon DCV é um protocolo de exibição remota de alto desempenho. Ele permite oferecer desktops remotos e streaming de aplicativos, de forma segura, de qualquer nuvem ou datacenter para qualquer dispositivo, em diferentes condições de rede. Ao usar o Amazon DCV com a Amazon EC2, você pode executar aplicativos com uso intensivo de gráficos remotamente nas instâncias da Amazon. EC2 Depois, é possível fazer streaming dos resultados para computadores cliente mais modestos, o que elimina a necessidade de estações de trabalho dedicadas e dispendiosas.

O Amazon DCV Web Client SDK é uma JavaScript biblioteca que você pode usar para desenvolver seus próprios aplicativos cliente de navegador da Web Amazon DCV. Seus usuários finais podem usar esses aplicativos para se conectar e interagir com uma sessão do Amazon DCV em execução.

Usando o SDK do Cliente Web do Amazon DCV como um componente básico, você pode criar aplicativos Web personalizados que fornecem aos usuários acesso instantâneo a seus desktops ou aplicativos de qualquer lugar, com um desempenho responsivo e fluido que é quase indistinguível de um aplicativo instalado nativamente.

Este guia explica como usar o SDK do Cliente Web do Amazon DCV para criar seus aplicativos cliente do navegador da Web personalizados para interagir com as sessões do Amazon DCV em seus fluxos de trabalho.

Tópicos

- Pré-requisitos
- Recursos compatíveis
- Suporte ao navegador
- Convenção de versionamento

Pré-requisitos

Antes de começar a trabalhar com o SDK do Cliente Web do Amazon DCV, certifique-se de estar familiarizado com o Amazon DCV e com as sessões do Amazon DCV. Para obter mais informações, consulte o Guia do Administrador do Amazon DCV.

O SDK do Cliente Web do Amazon DCV é compatível com o servidor do Amazon DCV versão 2020 e posterior.

Recursos compatíveis

Você pode criar aplicativos cliente do navegador da Web personalizados compatíveis com os seguintes recursos do Amazon DCV:

- Conectar-se aos servidores Amazon DCV do Windows
- Conectar-se aos servidores Amazon DCV do Linux
- Gerenciar modos de streaming
- Transferir arquivos
- Imprimir de sessões
- Copiar e colar
- Reprodução de áudio estéreo 2.0
- Gravação de áudio estéreo 2.0 (em servidores do Windows)
- Tela sensível ao toque
- Caneta (em servidores do Linux, Windows 10 e Windows Server 2019)
- Suporte a vários monitores

Para obter informações sobre esses recursos, consulte Recursos compatíveis no Guia do Usuário do Amazon DCV.

Suporte ao navegador

O Amazon DCV Web Client SDK suporta JavaScript (ES6) e pode ser usado em JavaScript nossos TypeScript aplicativos.

O SDK do Cliente Web do Amazon DCV é compatível com os seguintes navegadores da Web:

Navegador	Versão
Google Chrome	Últimas três versões principais
Mozilla Firefox	Últimas três versões principais
Microsoft Edge	Últimas três versões principais
Apple Safari para macOS	Últimas três versões principais

Convenção de versionamento

A versão do SDK do Cliente Web do Amazon DCV é definida no seguinte formato: `major.minor.patch`. Geralmente, a convenção de versionamento segue o modelo de versionamento semântico. Uma alteração na versão principal, como de `1.x.x` para `2.x.x`, indica que foram introduzidas mudanças importantes que podem exigir alterações no código e uma implantação planejada. Uma alteração em uma versão secundária, como de `1.1.x` para `1.2.x`, é sempre compatível com versões anteriores, mas pode incluir elementos descontinuados.

Conceitos básicos do SDK do cliente web do Amazon DCV

O SDK do cliente web do Amazon DCV é composto por um arquivo principal `dcv.js` e alguns componentes auxiliares. Todos os arquivos são distribuídos em um arquivo compactado que pode ser baixado do site do Amazon DCV.

Para começar a usar o SDK do cliente web do Amazon DCV

1. O arquivo SDK do cliente web do Amazon DCV é assinado digitalmente com uma assinatura GPG segura. Para verificar a assinatura do arquivo, você deve importar a chave NICE GPG. Para fazer isso, abra uma janela de terminal e importe a chave GPG do NICE.

   ```
   $ wget https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
   ```

   ```
   $ gpg --import NICE-GPG-KEY
   ```

2. Baixe o arquivo do SDK do cliente web do Amazon DCV e a assinatura do arquivo do SDK do cliente web do Amazon DCV do site do Amazon DCV.
3. Verifique a assinatura do arquivo do SDK do cliente web do Amazon DCV usando a assinatura.

   ```
   $ gpg --verify
   signature_filename.zip.sign
   archive_filename.zip
   ```

   Por exemplo:

   ```
   $ gpg --verify nice-dcv-web-client-sdk-1.8.7-858.zip.sign nice-dcv-web-client-
   sdk-1.8.7-858.zip
   ```

4. Se a assinatura for verificada com sucesso, extraia o conteúdo do arquivo SDK do cliente web do Amazon DCV e coloque o diretório extraído em seu servidor web. Por exemplo:

   ```
   $ unzip
   archive_filename.zip
   -d /
   path_to
   ```

```
/
server_directory
/
```

> **Important**
>
> - Você deve reter a estrutura de pastas ao implantar o SDK do cliente web do Amazon DCV em seu servidor web.
> - Ao usar o Amazon DCV Web UI SDK, lembre-se de que o componente `DCVViewer` React espera que os arquivos EULA.txt third-party-licenses e.txt desse pacote estejam presentes no caminho da URL do servidor web incorporado. O third-party-licenses arquivo.txt deve ser modificado para incluir também o conteúdo do arquivo correspondente do pacote Amazon DCV Web Client SDK e, possivelmente, qualquer outra informação de licença das bibliotecas usadas pelo aplicativo do usuário consumidor.

Conecte-se a um servidor Amazon DCV e obtenha o primeiro quadro

O tutorial a seguir mostra como preparar sua página HTML para seu cliente da web personalizado, como autenticar e se conectar a um servidor Amazon DCV e como receber o primeiro quadro de conteúdo transmitido da sessão do Amazon DCV.

Tópicos

- Etapa 1: Preparar a página HTML
- Etapa 2: Autenticar, conectar e obter o primeiro quadro
- Bônus: crie automaticamente um formulário de login em HTML

Etapa 1: Preparar a página HTML

Em sua página da web, você deve carregar os JavaScript módulos necessários e adicionar um elemento `<div>` HTML com um elemento válido `id` onde você deseja que o Amazon DCV Web Client SDK desenhe o fluxo de conteúdo do servidor Amazon DCV remoto.

Por exemplo:

```
<!DOCTYPE html>
<html lang="en" style="height: 100%;">
  <head>
    <title>DCV first connection</title>
  </head>
  <body style="height: 100%;">
    <div id="root" style="height: 100%;"></div>
    <div id="dcv-display"></div>
    <script type="module" src="index.js"></script>
  </body>
</html>
```

Etapa 2: Autenticar, conectar e obter o primeiro quadro

Esta seção mostra como concluir o processo de autenticação do usuário, como conectar o servidor Amazon DCV e como receber o primeiro quadro de conteúdo do servidor Amazon DCV.

Primeiro, a partir do arquivo `index.js`, importe o SDK do cliente web do Amazon DCV. Ele pode ser importado como um módulo de Definição de Módulo Universal (UMD), da seguinte forma:

```
import "./dcvjs/dcv.js"
```

Caso contrário, a partir da versão`1.1.0`, ele também pode ser importado como um ECMAScript Módulo (ESM) do pacote correspondente, da seguinte forma:

```
import dcv from "./dcvjs/dcv.js"
```

Defina as variáveis a serem usadas para armazenar o objeto Authentication, o objeto Connection e o URL do servidor Amazon DCV.

```
let auth,
    connection,
    serverUrl;
```

No carregamento do script, registre a versão do SDK do cliente web do Amazon DCV e, no carregamento da página, chame a função `main`.

```
console.log("Using Amazon DCV Web Client SDK version " + dcv.version.versionStr);
```

```
document.addEventListener('DOMContentLoaded', main);
```

A função `main` define o nível do log e inicia o processo de autenticação.

```
function main () {
  console.log("Setting log level to INFO");
  dcv.setLogLevel(dcv.LogLevel.INFO);

  serverUrl = "https://your-dcv-server-url:port/";

  console.log("Starting authentication with", serverUrl);

  auth = dcv.authenticate(
    serverUrl,
    {
      promptCredentials: onPromptCredentials,
      error: onError,
      success: onSuccess
    }
  );
}
```

As funções `promptCredentials`, `error` e `success` são funções obrigatórias de retorno de chamada que devem ser definidas no processo de autenticação.

Se o servidor Amazon DCV solicitar credenciais, a função de retorno de chamada `promptCredentials` receberá o desafio de credencial solicitado do servidor Amazon DCV. Se o servidor Amazon DCV estiver configurado para usar a autenticação do sistema, as credenciais de login deverão ser fornecidas. Os exemplos de código a seguir presumem que o nome de usuário é `my_dcv_user` e que a senha é `my_password`.

Se a autenticação falhar, a função de retorno de chamada `error` receberá um objeto de erro do servidor Amazon DCV.

Se a autenticação for bem-sucedida, a função de retorno de chamada `success` receberá uma matriz de pares que inclui o ID da sessão (`sessionId`) e os tokens de autorização (`authToken`) para cada sessão à qual o usuário `my_dcv_user` pode se conectar no servidor Amazon DCV. O exemplo de código a seguir chama a função connect e se conecta à primeira sessão retornada na matriz.

> **Note**
>
> No exemplo de código a seguir, substitua `MY_DCV_USER` por seu próprio nome de usuário e `MY_PASSWORD` por sua própria senha.

```
function onPromptCredentials(auth, challenge) {
  // Let's check if in challege we have a username and password request
  if (challengeHasField(challenge, "username") && challengeHasField(challenge,
 "password")) {
    auth.sendCredentials({username: MY_DCV_USER, password: MY_PASSWORD})
  } else {
    // Challenge is requesting something else...
  }
}

function challengeHasField(challenge, field) {
  return challenge.requiredCredentials.some(credential => credential.name === field);
}

function onError(auth, error) {
  console.log("Error during the authentication: " + error.message);
}

// We connect to the first session returned
function onSuccess(auth, result) {
  let {sessionId, authToken} = {...result[0]};

  connect(sessionId, authToken);
}
```

Conecte-se ao servidor Amazon DCV. O método de retorno de chamada `firstFrame` será chamado quando o primeiro quadro for recebido do servidor Amazon DCV.

```
function connect (sessionId, authToken) {
  console.log(sessionId, authToken);

  dcv.connect({
    url: serverUrl,
    sessionId: sessionId,
    authToken: authToken,
    divId: "dcv-display",
```

```
    callbacks: {
      firstFrame: () => console.log("First frame received")
    }
  }).then(function (conn) {
    console.log("Connection established!");
    connection= conn;
  }).catch(function (error) {
    console.log("Connection failed with error " + error.message);
  });
}
```

Bônus: crie automaticamente um formulário de login em HTML

O objeto `challenge` é retornado quando a função de retorno de chamada `promptCredentials` é chamada. Ele inclui uma propriedade chamada `requiredCredentials` que é uma matriz de objetos - um objeto por credencial que é solicitado pelo servidor Amazon DCV. Cada objeto inclui o nome e o tipo da credencial solicitada. Você pode usar os objetos `challenge` e `requiredCredentials` para criar automaticamente um formulário de login em HTML.

O exemplo de código a seguir mostra como fazer isso.

```
let form,
    fieldSet;

function submitCredentials (e) {
  var credentials = {};
  fieldSet.childNodes.forEach(input => credentials[input.id] = input.value);
  auth.sendCredentials(credentials);
  e.preventDefault();
}

function createLoginForm () {
  var submitButton = document.createElement("button");

  submitButton.type = "submit";
  submitButton.textContent = "Login";

  form = document.createElement("form");
  fieldSet = document.createElement("fieldset");

  form.onsubmit = submitCredentials;
  form.appendChild(fieldSet);
```

```
  form.appendChild(submitButton);

  document.body.appendChild(form);
}

function addInput (name) {
  var type = name === "password" ? "password" : "text";

  var inputField = document.createElement("input");
  inputField.name = name;
  inputField.id = name;
  inputField.placeholder = name;
  inputField.type = type;
  fieldSet.appendChild(inputField);
}

function onPromptCredentials (_, credentialsChallenge) {
  createLoginForm();
  credentialsChallenge.requiredCredentials.forEach(challenge =>
 addInput(challenge.name));
}
```

Trabalhar com os recursos do Amazon DCV

A disponibilidade dos recursos do Amazon DCV depende das permissões configuradas para a sessão do Amazon DCV e dos recursos do navegador da web do cliente.

Os recursos disponíveis em uma sessão do Amazon DCV são gerenciados pelas permissões que foram especificadas para a sessão. Isso significa que mesmo que um recurso seja compatível com o SDK do cliente web do Amazon DCV, o acesso a esse recurso pode ser impedido com base nas permissões definidas pelo administrador da sessão. Para obter mais informações, consulte Configurar a autorização do Amazon DCV no Guia do administrador do Amazon DCV.

Entender a função de retorno de chamada featuresUpdate

Quando a disponibilidade de um recurso em uma sessão do Amazon DCV muda, o SDK do Amazon DCV Web Client notifica você usando a função de retorno de chamada `featuresUpdate` que você especifica no momento do estabelecimento da conexão. Por exemplo:

```
featuresUpdate: function (connection, list) {
  ...
```

```
},
```

A função de retorno de chamada notifica você somente sobre os atributo cuja disponibilidade foi alterada. O parâmetro `list` é uma matriz de strings e inclui somente os nomes dos atributo atualizados. Por exemplo, se a disponibilidade do atributo de entrada de áudio for alterada durante a sessão, o parâmetro incluirá somente `["audio-in"]`. Se, posteriormente, a disponibilidade dos atributo de copiar e colar da área de transferência mudar para a sessão, o parâmetro incluirá somente `["clipboard-copy", "clipboard-paste"]`.

Controlar atualizações do atributo

A função de retorno de chamada `featuresUpdate` notifica você somente sobre os atributos cuja disponibilidade foi alterada. Para saber quais atributos foram atualizados, você deve consultar o atributo usando o método `connection.queryFeature`. Isso pode ser feito a qualquer momento após o recebimento da notificação de alteração. Esse método retorna uma `Promise` que resolve o status atualizado do atributo solicitado. O valor `status` está sempre associado e tem uma propriedade booleana (`true` | `false`) chamada `enabled`. Alguns atributos podem ter propriedades adicionais no valor `status`. Se a disponibilidade do atributo não tiver sido atualizada, ela será rejeitada.

O exemplo de código a seguir mostra como fazer isso.

```
// Connection callback called
function featuresUpdate (_, list) {
  if (list.length > 0) {
    list.forEach((feat) => {
      connection.queryFeature(feat).then(status => console.log(feat, "is",
 status.enabled)));
    });
  }
}
```

Usar o SDK de Interface do Usuário da Web do Amazon DCV

O tutorial a seguir mostra como se autenticar no servidor Amazon DCV, se conectar a ele e renderizar o componente React do `DCVViewer` a partir do SDK da interface de usuário da Web do Amazon DCV.

Tópicos

- Pré-requisitos
- Etapa 1: Preparar a página HTML
- Etapa 2: Autenticar, conectar e renderizar o componente React do DCVViewer.
- Atualizar o AWS-UI para o Cloudscape Design System

Pré-requisitos

Você precisa instalar `React`, `ReactDOM`, `Cloudscape Design Components React`, `Cloudscape Design Global Styles` e `Cloudscape Design Design Tokens`.

```
$ npm i react react-dom @cloudscape-design/components @cloudscape-design/global-styles
 @cloudscape-design/design-tokens
```

Você também precisaria baixar de `Amazon DCV Web Client SDK`. Leia Conceitos básicos do SDK do cliente web do Amazon DCV o step-by-step guia sobre como fazer isso.

Você deve criar um alias para importar o módulo `dcv`, pois ele é uma dependência externa do SDK da interface de usuário da Web do Amazon DCV. Por exemplo, se você estiver usando o webpack para agrupar seu aplicativo da Web, poderá usar a opção resolve.alias da seguinte forma:

```
const path = require('path');

module.exports = {
  //...
  resolve: {
    alias: {
      dcv: path.resolve('path', 'to', 'dcv.js'),
    },
  },
};
```

Se você estiver usando o rollup para empacotamento, poderá instalar o @rollup/plugin-alias e usá-lo assim:

```
import alias from '@rollup/plugin-alias';
const path = require('path');

module.exports = {
```

```
  //...
  plugins: [
    alias({
      entries: [
        { find: 'dcv', replacement: path.resolve('path', 'to', 'dcv.js') },
      ]
    })
  ]
};
```

Etapa 1: Preparar a página HTML

Em sua página da Web, você deve carregar os JavaScript módulos necessários e ter um elemento `<div>` HTML com um elemento válido em `id` que o componente de entrada do seu aplicativo será renderizado.

Por exemplo:

```
<!DOCTYPE html>
<html lang="en" style="height: 100%;">
  <head>
    <title>DCV first connection</title>
  </head>
  <body style="height: 100%;">
    <div id="root" style="height: 100%;"></div>
    <script type="module" src="index.js"></script>
  </body>
</html>
```

Etapa 2: Autenticar, conectar e renderizar o componente React do **`DCVViewer`**.

Esta seção mostra como concluir o processo de autenticação do usuário, como conectar o servidor Amazon DCV e como renderizar o componente React do `DCVViewer`.

Primeiro, a partir do arquivo `index.js`, importe `React`, `ReactDOM` e seu componente de nível superior `App`.

```
import React from "react";
import ReactDOM from 'react-dom';
import App from './App';
```

Renderize o nó de contêiner de nível superior do seu aplicativo.

```
ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById("root")
);
```

No arquivo `App.js`, importe o SDK do cliente web do Amazon DCV como um módulo ESM, o componente React `DCVViewer` do SDK da Interface do Usuário do Amazon DCV, `React` e o pacote `Cloudscape Design Global Styles`.

```
import React from "react";
import dcv from "dcv";
import "@cloudscape-design/global-styles/index.css";
import {DCVViewer} from "./dcv-ui/dcv-ui.js";
```

A seguir está um exemplo que mostra como se autenticar no Servidor Amazon DCV e renderizar o componente React do `DCVViewer` a partir do SDK da Interface da Web do Amazon DCV, desde que a autenticação tenha sido bem-sucedida.

```
const LOG_LEVEL = dcv.LogLevel.INFO;
const SERVER_URL = "https://your-dcv-server-url:port/";
const BASE_URL = "/static/js/dcvjs";

let auth;

function App() {
  const [authenticated, setAuthenticated] = React.useState(false);
  const [sessionId, setSessionId] = React.useState('');
  const [authToken, setAuthToken] = React.useState('');
  const [credentials, setCredentials] = React.useState({});

  const onSuccess = (_, result) => {
    var { sessionId, authToken } = { ...result[0] };

    console.log("Authentication successful.");

    setSessionId(sessionId);
    setAuthToken(authToken);
    setAuthenticated(true);
```

```
    setCredentials({});
  }

  const onPromptCredentials = (_, credentialsChallenge) => {
    let requestedCredentials = {};

    credentialsChallenge.requiredCredentials.forEach(challenge =>
 requestedCredentials[challenge.name] = "");
    setCredentials(requestedCredentials);
  }

  const authenticate = () => {
    dcv.setLogLevel(LOG_LEVEL);

    auth = dcv.authenticate(
      SERVER_URL,
      {
        promptCredentials: onPromptCredentials,
        error: onError,
        success: onSuccess
      }
    );
  }

  const updateCredentials = (e) => {
    const { name, value } = e.target;
    setCredentials({
      ...credentials,
      [name]: value
    });
  }

  const submitCredentials = (e) => {
    auth.sendCredentials(credentials);
    e.preventDefault();
  }

  React.useEffect(() => {
    if (!authenticated) {
      authenticate();
    }
  }, [authenticated]);

  const handleDisconnect = (reason) => {
```

```
    console.log("Disconnected: " + reason.message + " (code: " + reason.code + ")");
    auth.retry();
    setAuthenticated(false);
  }

  return (
    authenticated ?
    <DCVViewer
      dcv={{
        sessionId: sessionId,
        authToken: authToken,
        serverUrl: SERVER_URL,
        baseUrl: BASE_URL,
        onDisconnect: handleDisconnect,
        logLevel: LOG_LEVEL
      }}
      uiConfig={{
        toolbar: {
          visible: true,
          fullscreenButton: true,
          multimonitorButton: true,
        },
      }}
    />
    :
    <div
      style={{
        height: window.innerHeight,
        backgroundColor: "#373737",
        display: 'flex',
        alignItems: 'center',
        justifyContent: 'center',
      }}
    >
      <form>
        <fieldset>
          {Object.keys(credentials).map((cred) => (
            <input
              key={cred}
              name={cred}
              placeholder={cred}
              type={cred === "password" ? "password" : "text"}
              onChange={updateCredentials}
              value={credentials[cred]}
```

```
            />
          ))}
        </fieldset>
        <button
          type="submit"
          onClick={submitCredentials}
        >
          Login
        </button>
      </form>
    </div>
  );
}

const onError = (_, error) => {
  console.log("Error during the authentication: " + error.message);
}

export default App;
```

As funções `promptCredentials`, `error` e `success` são funções obrigatórias de retorno de chamada que devem ser definidas no processo de autenticação.

Se o servidor Amazon DCV solicitar credenciais, a função de retorno de chamada `promptCredentials` receberá o desafio de credencial solicitado do servidor Amazon DCV. Se o servidor Amazon DCV estiver configurado para usar a autenticação do sistema, as credenciais deverão ser fornecidas no formato de um nome de usuário e senha.

Se a autenticação falhar, a função de retorno de chamada `error` receberá um objeto de erro do servidor Amazon DCV.

Se a autenticação for bem-sucedida, a função de retorno de chamada `success` receberá uma matriz de pares que inclui o ID da sessão (`sessionId`) e os tokens de autorização (`authToken`) para cada sessão à qual o usuário pode se conectar no servidor Amazon DCV. O exemplo de código acima atualiza o estado do React para renderizar o componente `DCVViewer` em caso de autenticação bem-sucedida.

Para saber mais sobre as propriedades aceitas por esse componente, consulte a referência do SDK da interface de usuário da Web do Amazon DCV.

Para saber mais sobre certificados autoassinados, consulte Esclarecimentos sobre redirecionamento com certificados autoassinados.

Atualizar o AWS-UI para o Cloudscape Design System

A partir da versão 1.3.0 do SDK, atualizamos nosso componente `DCVViewer` do AWS-UI para sua evolução: Cloudscape Design.

O Cloudscape usa um tema visual diferente do AWS-UI, mas a base de código subjacente permanece a mesma. Portanto, migrar seu aplicativo com base no `DCVViewer` deve ser fácil. Para migrar, substitua os pacotes NPM relacionados ao AWS-UI que você instalou pelos pacotes associados do Cloudscape:

Nome do pacote AWS-UI	Nome do pacote Cloudscape
@awsui/components-react	@cloudscape-design/components
@awsui/global-styles	@cloudscape-design/global-styles
@awsui/collection-hooks	@cloudscape-design/collection-hooks
@awsui/design-tokens	@cloudscape-design/design-tokens

Para obter mais detalhes sobre a migração, consulte a página de documentação da AWS-UI GitHub .

Referência do SDK

Esta seção fornece descrições, sintaxe e exemplos de uso para cada SDK do Cliente Web do Amazon DCV.

Tópicos

- Módulo DCV
- Classe de conexão
- Classe de autenticação
- Classe de recurso
- SDK de Interface do Usuário da Web do Amazon DCV

Módulo DCV

Um módulo que implementa o lado do cliente do protocolo DCV.

Expõe

- Métodos
- Membros
- Definições de tipo e retorno de chamada

Métodos

Lista

- autenticar (AuthParams) → {Autenticação}
- conectar (configuração) → {Promessa. < Conexão >|Promessa. < {código: ConnectionErrorCode, mensagem: string} >}
- setLogHandler(manipulador) → {void}
- setLogLevel(nível) → {vazio}

autenticar (AuthParams) → {Autenticação}

Inicia o processo de autenticação para o endpoint do servidor Amazon DCV especificado.

Parâmetros:

Name	Tipo	Descrição
`url`	string	O nome do host e a porta do servidor Amazon DCV em execução no seguinte formato: `https://dcv_host_address:port` . Por exemplo: `https://my-dcv-server:8443` .
`authenticationToken`	string	O token de autenticação a ser usado para autenticação.
`callbacks`	authenticationCallbacks	Os retornos de chamada que estão disponíveis para serem chamados durante o processo de autenticação.

Retorna:

- O objeto Authentication.

Tipo

Autenticação

conectar (configuração) → {Promessa. < Conexão >|Promessa. < {código: ConnectionErrorCode, mensagem: string} >}

Conecta-se ao endpoint do servidor Amazon DCV especificado. Se a conexão for bem-sucedida, ela retornará um objeto Connection. Se a conexão falhar, ela retornará um objeto de erro.

Parâmetros:

Name	Tipo	Descrição
`config`	ConnectionConfig	O ConnectionConfig objeto.

Retorna:

- Um objeto Connection ou um objeto de erro.

Tipo

Promessa. < Conexão > | Promessa. < {código: ConnectionErrorCode, mensagem: string} >

setLogHandler(manipulador) → {void}

Define uma função personalizada de manipulador de log. Ao substituir o manipulador de log padrão, a posição original da entrada do log será perdida durante a depuração usando o console do navegador.

Parâmetros:

Name	Tipo	Descrição
`handler`	função	A função personalizada do manipulador de logs. A função do manipulador contém level (número), levelName (string), domínio (string) e message (string).

Retorna:

Tipo

nulo

setLogLevel(nível) → {vazio}

Define o nível do log. Isso será necessário apenas se o manipulador de logs padrão for usado.

Parâmetros:

Name	Tipo	Descrição
`level`	LogLevel	O nível de log a ser usado.

Retorna:

Tipo

nulo

Membros

Lista

- (constante) AudioError: AudioErrorCode
- (constante) AuthenticationError: AuthenticationErrorCode
- (constante) ChannelError: ChannelErrorCode
- (constante) ClosingReasonError: ClosingReasonErrorCode
- (constante) ConnectionError: ConnectionErrorCode
- (constante) CustomChannelError: CustomChannelErrorCode
- (constante) DisplayConfigError: DisplayConfigErrorCode
- (constante) FileStorageError: FileStorageErrorCode
- (constante) LogLevel: LogLevel
- (constante) MultiMonitorError: MultiMonitorErrorCode
- (constante) ResolutionError: ResolutionErrorCode
- (constante) TimezoneRedirectionError: TimezoneRedirectionErrorCode
- (constante) TimezoneRedirectionSetting: TimezoneRedirectionSettingCode
- (constante) TimezoneRedirectionStatus: TimezoneRedirectionStatusCode
- versionamento (constante)

- (constante) ScreenshotError: ScreenshotErrorCode
- (constante) WebcamError: WebcamErrorCode

(constante) AudioError: AudioErrorCode

A enumeração dos AudioError códigos.

Tipo:

- AudioErrorCode

(constante) AuthenticationError: AuthenticationErrorCode

A enumeração dos AuthenticationError códigos.

Tipo:

- AuthenticationErrorCode

(constante) ChannelError: ChannelErrorCode

A enumeração dos ChannelError códigos.

Tipo:

- ChannelErrorCode

(constante) ClosingReasonError: ClosingReasonErrorCode

A enumeração dos ClosingReasonError códigos.

Tipo:

- ClosingReasonErrorCode

(constante) ConnectionError: ConnectionErrorCode

A enumeração dos ConnectionError códigos.

Tipo:

- ConnectionErrorCode

(constante) CustomChannelError: CustomChannelErrorCode

A enumeração dos CustomChannelError códigos.

Tipo:

- CustomChannelErrorCode

(constante) DisplayConfigError: DisplayConfigErrorCode

A enumeração dos DisplayConfigError códigos.

Tipo:

- DisplayConfigErrorCode

(constante) FileStorageError: FileStorageErrorCode

A enumeração dos FileStorageError códigos.

Tipo:

- FileStorageErrorCode

(constante) LogLevel: LogLevel

Os níveis de log do SDK disponíveis.

Tipo:

- LogLevel

(constante) MultiMonitorError: MultiMonitorErrorCode

A enumeração dos MultiMonitorError códigos.

Tipo:

- MultiMonitorErrorCode

(constante) ResolutionError: ResolutionErrorCode

A enumeração dos ResolutionError códigos.

Tipo:

- ResolutionErrorCode

(constante) TimezoneRedirectionError: TimezoneRedirectionErrorCode

A enumeração dos TimezoneRedirectionError códigos.

Tipo:

- TimezoneRedirectionErrorCode

(constante) TimezoneRedirectionSetting: TimezoneRedirectionSettingCode

A enumeração dos TimezoneRedirectionSetting códigos.

Tipo:

- TimezoneRedirectionSettingCode

(constante) TimezoneRedirectionStatus: TimezoneRedirectionStatusCode

A enumeração dos TimezoneRedirectionStatus códigos.

Tipo:

- TimezoneRedirectionStatusCode

versionamento (constante)

O versionamento do Amazon DCV com versões principais, secundária, de correção, revisão, estendida e versionStr.

Propriedades:

Name	Tipo	Descrição
`major`	integer	O número da versão principal.
`minor`	integer	O número da versão secundária.
`patch`	integer	O número da versão de correção.
`revision`	integer	O número da revisão.
`extended`	string	A sequência estendida.
`versionStr`	string	Uma concatenação dos números principais, secundários, de correção e de revisão no formulário `major.minor.patch+build.revision` .

(constante) ScreenshotError: ScreenshotErrorCode

A enumeração dos ScreenshotError códigos.

Tipo:

- ScreenshotErrorCode

(constante) WebcamError: WebcamErrorCode

A enumeração dos WebcamError códigos.

Tipo:

- WebcamErrorCode

Definições de tipo e retorno de chamada

Lista

- AudioErrorCode
- authenticationCallbacks
- AuthenticationErrorCode
- authErrorCallback(autenticação, erro)
- authPromptCredentialsRetorno de chamada (autenticação, desafio)
- authSuccessCallback(autenticação, dados de autenticação)
- Canal
- ChannelErrorCode
- clipboardEventCallback(evento)
- ClosingReasonErrorCode
- Colorspace
- connectionCallbacks
- ConnectionConfig
- ConnectionErrorCode
- createDirectory(path)
- CustomChannelErrorCode
- dataChannelCallback(informações)
- deleteFile(path)
- deviceChangeEventRetorno de chamada ()
- disconnectCallback(reason)
- displayAvailabilityCallback(status, DisplayID)
- DisplayConfigErrorCode
- displayLayoutCallback(largura do servidor, altura do servidor, cabeças)
- recurso
- featuresUpdateCallback(Lista de recursos)
- fileDownloadCallback(Recurso de arquivo)
- filePrintedCallback(Recurso de impressão)

- filestorage
- filestorageEnabledCallback(ativado)
- FileStorageErrorCode
- firstFrameCallback(redimensionamento ativado, relativeMouseMode ativado, ID de exibição)
- idleWarningNotificationRetorno de chamada () disconnectionDateTime
- collaboratorListCallback(colaboradores)
- licenseNotificationCallback(notificação)
- list(path)
- LogLevel
- Monitorar
- MultiMonitorErrorCode
- qualityIndicatorStateRetorno de chamada (estado)
- renameDirectory(src, dest)
- renameFile(src, dest)
- ResolutionErrorCode
- retrieveFile(path)
- screenshotCallback(screenshot)
- ScreenshotErrorCode
- serverInfo
- stats
- storeFile(file, dir)
- TimezoneRedirectionErrorCode
- TimezoneRedirectionSettingCode
- TimezoneRedirectionStatusCode
- WebcamErrorCode
- httpExtraSearchParamsCallback(método, url, corpo)
- httpExtraHeadersRetorno de chamada (método, url, corpo)

AudioErrorCode

Os enums de AudioError código disponíveis no módulo DCV

- `SETTING_AUDIO_FAILED`
- `CHANNEL_NOT_AVAILABLE`

Tipo:

- número

authenticationCallbacks

Retornos de chamada de autenticação

Tipo:

- Objeto

Propriedades:

Name	Tipo	Descrição
`promptCredentials`	authPromptCredentialsRetorno de chamada	A função de retorno de chamada a ser chamada quando o usuário é desafiado a fornecer credenciais.
`error`	authErrorCallback	A função de retorno de chamada a ser chamada quando a autenticação falhar.
`success`	authSuccessCallback	A função de retorno de chamada a ser chamada quando a autenticação for bem-sucedida.
`httpExtraSearchParamsCallback`	httpExtraSearchParamsCallback	A função de retorno de chamada a ser chamada no `authenticate` método para injetar parâmetros de consulta personalizados no

Name	Tipo	Descrição
		URL de autenticação antes de iniciar a solicitação. Também pode ser usado no `connect` método para personalizar a URL usada ao estabelecer uma WebSocket conexão com o servidor DCV.

AuthenticationErrorCode

Os enums de AuthenticationError código disponíveis no módulo DCV

- `INVALID_MESSAGE`
- `UNKNOWN_AUTH_MODE`
- `SESSION_NOT_AVAILABLE`
- `NO_SESSIONS`
- `WRONG_CREDENTIALS`
- `SASL_CHALLENGE`
- `SASL_AUTH_MECHANISM`
- `FAILED_COMMUNICATION`
- `AUTHENTICATION_REJECTED`
- `GENERIC_ERROR`
- `WRONG_CREDENTIALS_FORMAT`
- `WRONG_CREDENTIALS_TYPE`
- `UNREQUESTED_CREDENTIALS`
- `MISSING_CREDENTIAL`

Tipo:

- número

authErrorCallback(autenticação, erro)

A função de retorno de chamada a ser chamada quando a autenticação falhar.

Parâmetros:

Name	Tipo	Descrição
`authentication`	Autenticação	O objeto Authentication.
`error`	Objeto	O objeto de erro gerado pelo processo de autenticação. Name: `code` Tipo: AuthenticationErrorCode Descrição: O código do erro. Name: `message` Tipo: string Descrição: A mensagem de erro.

authPromptCredentialsRetorno de chamada (autenticação, desafio)

A função de retorno de chamada a ser chamada quando o usuário é desafiado a fornecer credenciais. O usuário deve responder ao desafio fornecendo as credenciais solicitadas.

Parâmetros:

Name	Tipo	Descrição
`authentication`	Autenticação	O objeto Authentication.
`challenge`	Objeto	O desafio.

Name	Tipo	Descrição
		Name: `requiredCredentials` — Tipo: Array.<Object> — Descrição: Uma matriz de objetos de credenciais solicitados. Name: `name` — Tipo: string — Descrição: O nome da credencial solicitada. Name: `type` — Tipo: string — Descrição: O tipo da credencial solicitada.

authSuccessCallback(autenticação, dados de autenticação)

A função de retorno de chamada a ser chamada quando a autenticação for bem-sucedida.

Parâmetros:

Name	Tipo	Descrição
`authentication`	Autenticação	O objeto Authentication.
`authenticationData`	Array.<Object>	Uma variedade de objetos que incluem tokens de sessão IDs e autenticação do Amazon DCV. Name / Tipo / Descrição `sessior` / string / O ID da sessão do Amazon DCV. `authTok` / string / O token de autenticação para a sessão do Amazon DCV.

Canal

Os canais disponíveis que podem ser especificados.

Tipo:

- "clipboard" | "display" | "input" | "audio" | "filestorage"

ChannelErrorCode

Os enums de ChannelError código disponíveis no módulo DCV

- `ALREADY_OPEN`
- `INITIALIZATION_FAILED`
- `REJECTED`

Tipo:

- número

clipboardEventCallback(evento)

A função de retorno de chamada a ser chamada quando a `clipboardEvent` é gerada.

Parâmetros:

Name	Tipo	Descrição
`event`	Objeto	Informações sobre o evento da área de transferência. Name: `name` Tipo: estabelecido | copiar | colar | | | dataSilert Atributo: Descrição: Sempre presente. O nome do evento.

Name	Tipo	Descrição
		Name | Tipo | Atribut | Descrição | | autoC one newDa ailable autoPa Done | Remot or | paste/ lableD | | | `clipb Data` | Object | string | | Os dados na área de transferê ncia. | | `autoC` | boolea | <opcic > | Indica se a cópia automátic a da área de transferê ncia da sessão para a |

Name	Tipo	Descrição
		Name | Tipo | Atribut | Descrição área de transferência do cliente local está ativada. `maxDa ze` | número | <opcic> | A quantidade máxima de dados que podem ser colocados na área de transferência. `error` | string | <opcic> | Informações de erro, se aplicável.

ClosingReasonErrorCode

Os enums de ClosingReasonError código disponíveis no módulo DCV

- `TRANSPORT_ERROR`
- `NO_ERROR`
- `GENERIC_ERROR`
- `INTERNAL_SERVER_ERROR`
- `PROTOCOL_ERROR`
- `AUTHORIZATION_DENIED`
- `AUTHORIZATION_REVOKED`
- `ACCESS_REJECTED`
- `IDLE_TIMEOUT_EXPIRED`
- `DISCONNECT_BY_OWNER`
- `DISCONNECT_BY_USER`
- `EVICTED`
- `EXTERNAL_PROTOCOL_CONNECTION_EVICTED`
- `DISCONNECTION_REQUESTED`

Tipo:

- número

Colorspace

Os espaços de cores disponíveis que podem ser especificados.

Tipo:

- “RGB” | “YUV_ REC6 01" | “YUV_ 09" REC7

connectionCallbacks

Os retornos de chamada que estão disponíveis para serem chamados no caso de um erro de conexão.

Tipo:

- Objeto

Propriedades:

Name	Tipo	Descrição
`disconnect`	disconnectCallback	A função de retorno de chamada a ser chamada quando a conexão terminar.
`displayLayout`	displayLayoutCallback	A função de retorno de chamada a ser chamada quando o layout ou a resolução da tela são alterados.
`displayAvailability`	displayAvailabilityCallback	A função de retorno de chamada a ser chamada quando a disponibilidade de uma tela mudar.
`firstFrame`	firstFrameCallback	A função de retorno de chamada a ser chamada quando o primeiro quadro for recebido do servidor Amazon DCV.
`filePrinted`	filePrintedCallback	A função de retorno de chamada a ser chamada quando um arquivo for impresso no servidor Amazon DCV.
`fileDownload`	fileDownloadCallback	A função de retorno de chamada a ser chamada quando um arquivo baixado no servidor Amazon DCV.
`dataChannel`	dataChannelCallback	A função de retorno de chamada a ser chamada

Name	Tipo	Descrição
		quando o servidor Amazon DCV enviar uma notificação sobre a disponibilidade de um canal de dados.
`licenseNotification`	licenseNotificationCallback	A função de retorno de chamada a ser chamada quando o servidor Amazon DCV enviar uma notificação sobre o estado da licença.
`idleWarningNotification`	idleWarningNotificationRetorno de chamada	A função de retorno de chamada a ser chamada quando o servidor Amazon DCV envia um aviso de tempo limite de inatividade.
`collaboratorList`	collaboratorListCallback	A função de retorno de chamada a ser chamada quando o servidor Amazon DCV envia a lista de colaboradores (desde o SDK do Cliente Web do Amazon DCV versão 1.1.0).
`qualityIndicatorState`	qualityIndicatorStateRetorno de chamada	A função de retorno de chamada a ser chamada quando o indicador de qualidade da conexão mudar de estado.
`filestorageEnabled`	filestorageEnabledCallback	A função de retorno de chamada a ser chamada quando o armazenamento de arquivos estiver ativado ou desativado.

Name	Tipo	Descrição
`featuresUpdate`	featuresUpdateCallback	A função de retorno de chamada a ser chamada quando o status de um atributo for alterado.
`clipboardEvent`	clipboardEventCallback	A função de retorno de chamada a ser chamada quando a `clipboardEvent` é gerada.
`deviceChangeEvent`	deviceChangeEventRetorno de chamada	A função de retorno de chamada a ser chamada quando um evento `deviceChange` for acionado.
`screenshot`	screenshotCallback	A função de retorno de chamada a ser chamada quando a `screenshot` for gerada.
`httpExtraSearchParamsCallback`	httpExtraSearchParamsCallback	A função de retorno de chamada a ser chamada para personalizar a URL ao estabelecer uma WebSocket conexão com o servidor Amazon DCV. Observe que esse retorno de chamada também pode ser usado com o `authenticate` método para anexar dinamicamente os parâmetros de consulta ao URL de autenticação antes que o SDK envie a solicitação.

Name	Tipo	Descrição
`httpExtraHeadersCallback`	[httpExtraHeadersRetorno de chamada]	A função de retorno de chamada a ser chamada para adicionar cabeçalhos personalizados à solicitação HTTP durante o estabelecimento da conexão.

ConnectionConfig

A configuração para uma conexão do Amazon DCV.

Tipo:

- Objeto

Propriedades:

Name	Tipo	Descrição
`url`	string	O nome do host e a porta do servidor Amazon DCV em execução no seguinte formato: `https://dcv_host_address:port` . Por exemplo: `https://my-dcv-server:8443` .
`sessionId`	string	O ID da sessão do Amazon DCV.
`authToken`	string	O token de autenticação a ser usado ao se conectar ao servidor.

Name	Tipo	Descrição
`baseUrl`	string	O URL absoluto ou relativo do qual carregar arquivos do SDK.
`resourceBaseUrl`	string	O URL absoluto ou relativo a partir do qual acessar os recursos do DCV.
`enabledChannels`	Array.<Channel>	Indica a lista de canais que podem ser ativados. Se não for especificado ou se uma matriz vazia for fornecida, o padrão será todos os canais disponíveis.
`losslessColorspace`	Colorspace	Indica o espaço de cores que será usado. Se não for especificado, o padrão será “RGB”.
`divId`	string	O ID do objeto `div` no HTML DOM em que o SDK deve criar a tela com o fluxo remoto.
`volumeLevel`	integer	O nível de volume preferido. O intervalo válido é de 0 a 100.
`clipboardAutoSync`	boolean	Indica se a cópia automátic a da área de transferência da sessão do Amazon DCV para a área de transferência do cliente local está ativada para navegadores da web compatíveis.

Name	Tipo	Descrição
`dynamicAudioTuning`	boolean	Indica se o áudio deve ser ajustado dinamicamente com base nas configurações de áudio do servidor Amazon DCV quando uma conexão é estabelecida.
`clientHiDpiScaling`	boolean	Indica se a tela deve ser escalada com base no DPI do cliente.
`highColorAccuracy`	boolean	Indica se deve ser usada a alta precisão de cores, se disponível. O padrão é `false`, se não for especificado.
`enableWebCodecs`	Booleano	Indica se WebCodecs deve ser usado, se disponível. O padrão é `false`, se não for especificado.
`observers`	connectionCallbacks	O retorno de chamada funciona para chamar eventos relacionados à conexão.
`callbacks`	connectionCallbacks	O mesmo que a proprieda de `observers` , mas cada retorno de chamada inclui o objeto Conexão como primeiro parâmetro.

ConnectionErrorCode

Os enums de ConnectionError código disponíveis no módulo DCV

- `ALREADY_OPEN`

- `INVALID_CONFIG`
- `INITIALIZATION_FAILED`
- `REJECTED`
- `MAIN_CHANNEL_ALREADY_OPEN`
- `GENERIC_ERROR` (desde o DCV Server 2021.0)
- `INTERNAL_SERVER_ERROR` (desde o DCV Server 2021.0)
- `AUTHENTICATION_FAILED` (desde o DCV Server 2021.0)
- `PROTOCOL_ERROR` (desde o DCV Server 2021.0)
- `INVALID_SESSION_ID` (desde o DCV Server 2021.0)
- `INVALID_CONNECTION_ID` (desde o DCV Server 2021.0)
- `CONNECTION_LIMIT_REACHED` (desde o DCV Server 2021.0)
- `SERVER_UNREACHABLE` (desde o DCV Server 2022.1)
- `GATEWAY_BUSY`
- `UNSUPPORTED_CREDENTIAL` (desde o DCV Server 2022.2)
- `TRANSPORT_ERROR`

Tipo:

- número

createDirectory(path)

Parâmetros:

Name	Tipo	Descrição
`path`	string	O caminho absoluto no servidor no qual queremos criar um diretório. Também deve incluir o nome do diretório de destino.

CustomChannelErrorCode

Os enums de CustomChannelError código disponíveis no módulo DCV

- `TRANSPORT_ERROR`

Tipo:

- número

dataChannelCallback(informações)

A função de retorno de chamada a ser chamada quando o servidor Amazon DCV enviar uma notificação sobre a disponibilidade de um canal de dados.

Parâmetros:

<table>
<tr><th>Name</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td>info</td><td>Objeto</td><td>Informações sobre o canal de dados.
<table>
<tr><th>Name</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td>name</td><td>string</td><td>O nome do canal de dados.</td></tr>
<tr><td>token</td><td>string</td><td>O token de autenticação do canal de dados.</td></tr>
</table></td></tr>
</table>

deleteFile(path)

Parâmetros:

Name	Tipo	Descrição
`path`	string	O caminho absoluto no servidor que identifica o arquivo que queremos excluir.

deviceChangeEventRetorno de chamada ()

A função de retorno de chamada a ser chamada quando um evento `deviceChange` for acionado.

disconnectCallback(reason)

A função de retorno de chamada a ser chamada quando a conexão terminar.

Parâmetros:

Name	Tipo	Descrição
`reason`	Objeto	O motivo da desconexão. Name / Tipo / Descrição `code` / número / O código do motivo. `message` / string / A mensagem do motivo.

displayAvailabilityCallback(status, DisplayID)

A função de retorno de chamada a ser chamada quando a disponibilidade de uma tela mudar.

Parâmetros:

<table>
<tr><th>Name</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td>status</td><td>Objeto</td><td>O status do monitor.
<table>
<tr><th>Name</th><th>Tipo</th><th>Descrição</th></tr>
<tr><td>enablec</td><td>boolean</td><td>Indica se o monitor está ativado.</td></tr>
<tr><td>closed</td><td>boolean</td><td>Indica se o monitor está fechado.</td></tr>
</table></td></tr>
<tr><td>displayId</td><td>número</td><td>O identificador do monitor.</td></tr>
</table>

DisplayConfigErrorCode

Os enums de DisplayConfigError código disponíveis no módulo DCV

- `INVALID_ARGUMENT`
- `UNSUPPORTED_OPERATION`
- `NO_CHANNEL`

Tipo:

- número

displayLayoutCallback(largura do servidor, altura do servidor, cabeças)

A função de retorno de chamada a ser chamada quando o layout ou a resolução da tela são alterados.

Parâmetros:

Name	Tipo	Descrição
`serverWidth`	número	A largura (em pixels) do monitor principal.
`serverHeight`	número	A altura (em pixels) do monitor principal.
`heads`	Array.<Monitor>	Os títulos da exibição compatíveis com o servidor Amazon DCV.

recurso

Os valores do atributo.

- `display` - Indica a disponibilidade de um streaming de vídeo com um único monitor.
- `display-multi` - Indica a disponibilidade de um streaming de vídeo com vários monitores.
- `high-color-accuracy` - Indica a disponibilidade de alta precisão de cores (desde o SDK do cliente web do Amazon DCV versão 1.1.0).
- `mouse` - Indica a disponibilidade da funcionalidade do mouse.
- `keyboard` - Indica a disponibilidade da funcionalidade do teclado.
- `keyboard-sas` - Indica a disponibilidade da funcionalidade de sequência SAS (Control + Alt + Delete).
- `relative-mouse` - Indica a disponibilidade do modo relativo do mouse.
- `clipboard-copy` - Indica a disponibilidade da funcionalidade de cópia da área de transferência do servidor Amazon DCV para o cliente.
- `clipboard-paste` - Indica a disponibilidade da funcionalidade de colagem da área de transferência do servidor Amazon DCV para o cliente.
- `audio-in` - Indica a disponibilidade da funcionalidade de entrada de áudio usando o microfone.

- `audio-out` - Indica a disponibilidade da funcionalidade de reprodução de áudio.
- `webcam` - Indica a disponibilidade da funcionalidade de streaming da webcam.
- `file-download` - Indica a disponibilidade da funcionalidade de download do arquivo do servidor Amazon DCV para o cliente.
- `file-upload` - Indica a disponibilidade da funcionalidade de upload do arquivo do servidor Amazon DCV para o cliente.
- `timezone-redirection` - Indica a disponibilidade da funcionalidade de redirecionamento de fuso horário (desde o SDK do cliente web do Amazon DCV versão 1.3.0).

Tipo:

- string

featuresUpdateCallback(Lista de recursos)

A função de retorno de chamada a ser chamada quando o status de um atributo for alterado.

Parâmetros:

Name	Tipo	Descrição
`featuresList`	Matriz.<feature>	Uma série de atributos que foram alterados.

fileDownloadCallback(Recurso de arquivo)

A função de retorno de chamada a ser chamada quando um arquivo baixado no servidor Amazon DCV.

Parâmetros:

Name	Tipo	Descrição
`fileResource`	Objeto	Informações sobre o arquivo que está pronto para ser baixado.

Name	Tipo	Descrição

Name	Tipo	Descrição
`id`	string	O identificador do arquivo.
`url`	string	O URL a ser usado para baixar o arquivo.
`domain`	string	O domínio do recurso.
`token`	string	O token de autenticação a ser usado para baixar o arquivo. O token também

Name	Tipo	Descrição
		Name | Tipo | Descrição está incluído no URL.

filePrintedCallback(Recurso de impressão)

A função de retorno de chamada a ser chamada quando um arquivo for impresso no servidor Amazon DCV.

Parâmetros:

Name	Tipo	Descrição
`printResource`	Objeto	Informações sobre o arquivo impresso. Name | Tipo | Descrição `id` | string | O identificador do arquivo impresso. `url` | string | O URL a ser usado para baixar o

Name	Tipo	Descrição
		Name / Tipo / Descrição arquivo impresso. `domain` — string — O domínio do recurso. Nesse caso, `printer`. `token` — string — O token de autenticação a ser usado para baixar o arquivo impresso. O token também está incluído no URL.

filestorage

Objeto que permite explorar e executar ações no sistema de arquivos.

Tipo:

- Objeto

Propriedades:

Name	Tipo	Descrição
`list`	list	Função que permite listar os itens (arquivos e diretório s) presentes no caminho fornecido no servidor.
`createDirectory`	createDirectory	Função que permite criar um diretório no caminho especific ado no servidor.
`retrieveFile`	retrieveFile	Função que permite baixar localmente um arquivo no caminho especificado no servidor.
`deleteFile`	deleteFile	Função que permite excluir um arquivo no caminho especificado no servidor.
`renameFile`	renameFile	Função que permite renomear um arquivo do caminho de origem especificado para o caminho de destino especific ado.
`renameDirectory`	renameDirectory	Função que permite renomear um diretório do caminho de origem especificado para o

Name	Tipo	Descrição
		caminho de destino especificado.
`storeFile`	storeFile	Função que permite fazer o upload de um arquivo local para o caminho fornecido no servidor.

filestorageEnabledCallback(ativado)

A função de retorno de chamada a ser chamada quando o armazenamento de arquivos estiver ativado. Canal lento somente no Internet Explorer 11.

Parâmetros:

Name	Tipo	Descrição
`enabled`	boolean	Indica se o armazenamento de arquivos está habilitado.

FileStorageErrorCode

Os enums de FileStorageError código disponíveis no módulo DCV

- `CANCELLED`
- `ABORTED`
- `INVALID_ARGUMENT`
- `NOT_IMPLEMENTED`
- `ERROR`
- `ALREADY_EXIST`
- `NOT_FOUND`

Tipo:

- número

firstFrameCallback(redimensionamento ativado, relativeMouseMode ativado, ID de exibição)

A função de retorno de chamada a ser chamada quando o primeiro quadro for recebido do servidor Amazon DCV. Emitido para cada monitor.

Parâmetros:

Name	Tipo	Descrição
`resizeEnabled`	boolean	Indica se o servidor é compatível com o redimensionamento do layout de exibição do cliente.
`relativeMouseModeEnabled`	boolean	Indica se o servidor é compatível com o modo relativo do mouse.
`displayId`	número	O identificador do monitor.

idleWarningNotificationRetorno de chamada () disconnectionDateTime

A função de retorno de chamada a ser chamada quando o servidor Amazon DCV envia um aviso de tempo limite de inatividade.

Parâmetros:

Name	Tipo	Descrição
`disconnectionDateTime`	Data	A data e a hora da desconexão.

collaboratorListCallback(colaboradores)

A função de retorno de chamada a ser chamada quando o servidor Amazon DCV envia a lista de colaboradores.

Parâmetros:

Name	Tipo	Descrição
`collaborators`	Array.<Object>	Uma lista de objetos contendo informações sobre colaboradores. Name: `usernam`; Tipo: string; Descrição: O nome de usuário do colaborador. Name: `owner`; Tipo: boolean; Descrição: Indica se o colaborador é o proprietário da sessão. Name: `connectnId`; Tipo: número; Descrição: Indica o ID atribuído pelo servidor à conexão.

licenseNotificationCallback(notificação)

A função de retorno de chamada a ser chamada quando o servidor Amazon DCV enviar uma notificação sobre o estado da licença.

Parâmetros:

Name	Tipo	Descrição
`notification`	Objeto	A notificação. Name / Tipo / Descrição `product` / string / O produto do DCV. `status` / string / O status da licença. `message` / string / Uma mensagem. `leftDay` / número / O número de dias antes que a licença expire. `isDemo` / boolean / Indica se a licença é uma licença

Name	Tipo	Descrição
		Name / Tipo / Descrição de demonstração. numUnlinsed / número / O número de conexões não licenciadas. licensiMode / string / O modelo de licença. documentationUrl / string / O URL da documentação.

list(path)

Parâmetros:

Name	Tipo	Descrição
`path`	string	O caminho absoluto no servidor do qual queremos listar o conteúdo.

LogLevel

Os níveis de log do SDK disponíveis.

Tipo:

- TRACE | DEBUG | INFO | WARN | ERROR | SILENT

Monitorar

Tipo:

- Objeto

Propriedades:

Name	Tipo	Descrição
`name`	string	O nome do título da exibição.
`rect`	Objeto	Informações sobre o título da exibição. Name: x; Tipo: número; Descrição: A coordenada inicial x do título da exibição. Name: y; Tipo: número; Descrição: A coordenada inicial

Name	Tipo	Descrição
		<table><tr><th>Name</th><th>Tipo</th><th>Descrição</th></tr><tr><td></td><td></td><td>y do título da exibição.</td></tr><tr><td>width</td><td>número</td><td>A largura (em pixels) do título da exibição.</td></tr><tr><td>height</td><td>número</td><td>A altura (em pixels) do título da exibição.</td></tr></table>
primary	boolean	Indica se o título da exibição é o título da exibição principal . Isso é determinado pelo sistema operacional remoto, se disponível.
dpi	número	O DPI do título da exibição.

MultiMonitorErrorCode

Os enums de MultiMonitorError código disponíveis no módulo DCV

- `NO_DISPLAY_CHANNEL`
- `MAX_DISPLAY_NUMBER_REACHED`
- `INVALID_ARGUMENT`
- `DISPLAY_NOT_OPENED_BY_SERVER`
- `REQUEST_TIMEOUT`
- `GENERIC_ERROR`
- `NO_ERROR`

Tipo:

- número

qualityIndicatorStateRetorno de chamada (estado)

A função de retorno de chamada a ser chamada quando o indicador de qualidade da conexão mudar de estado.

Parâmetros:

Name	Tipo	Descrição
`state`	Array.<Object>	Informações sobre a qualidade da conexão. Name | Tipo | Descrição `name` | string | O nome do indicador. `status` | NORMAL | WARNIN | Uma descrição do status.

Name	Tipo	Descrição
		Name | Tipo | Descrição | CRITICA changec boolean Indica se o status foi alterado.

renameDirectory(src, dest)

Parâmetros:

Name	Tipo	Descrição
`src`	string	O caminho de origem absoluto no servidor que identifica o diretório que queremos renomear.
`dest`	string	O caminho de destino absoluto no servidor que especifica o caminho de destino e o nome do diretório.

renameFile(src, dest)

Parâmetros:

Name	Tipo	Descrição
`src`	string	O caminho de origem absoluto no servidor que identifica

Name	Tipo	Descrição
		o arquivo que queremos renomear.
`dest`	string	O caminho de destino absoluto no servidor que especifica o caminho de destino e o nome do arquivo.

ResolutionErrorCode

Os enums de ResolutionError código disponíveis no módulo DCV

- `INVALID_ARGUMENT`
- `NO_CHANNEL`
- `NOT_IMPLEMENTED`

Tipo:

- número

retrieveFile(path)

Parâmetros:

Name	Tipo	Descrição
`path`	string	O caminho absoluto no servidor que identifica o arquivo que queremos baixar localmente.

screenshotCallback(screenshot)

A função de retorno de chamada a ser chamada quando a screenshot for gerada.

Parâmetros:

Name	Tipo	Descrição
`screenshot`	byte[]	Buffer da captura de tela no formato PNG ou `null` se a recuperação da captura de tela falhar.

ScreenshotErrorCode

Os enums de ScreenshotError código disponíveis no módulo DCV

- `NO_CHANNEL`
- `GENERIC_ERROR`

Tipo:

- número

serverInfo

Tipo:

- Objeto

Propriedades:

Name	Tipo	Descrição
`name`	string	O nome do software.
`version`	Objeto	O número da versão do software.

Name	Tipo	Descrição
		<table><tr><th>Name</th><th>Tipo</th><th>Descrição</th></tr><tr><td>major</td><td>número</td><td>O número da versão principal.</td></tr><tr><td>minor</td><td>número</td><td>O número da versão secundária.</td></tr><tr><td>revisic</td><td>número</td><td>O número da versão da revisão.</td></tr></table>
os	string	O sistema operacional.
arch	string	A arquitetura.
hostname	string	O nome do host.

stats

Tipo:

- Objeto

Propriedades:

Name	Tipo	Descrição
`fps`	número	Os quadros atuais por segundo.
`traffic`	número	O tráfego atual em bit/s.
`peakTraffic`	número	O pico de tráfego bit/s desde que a conexão foi estabelec ida.
`latency`	número	A latência atual em ms.
`currentChannels`	número	O número de canais que foram abertos desde que a conexão foi estabelecida.
`openedChannels`	número	O número de canais abertos atualmente.
`channelErrors`	número	O número de canais que relataram um erro.

storeFile(file, dir)

Parâmetros:

Name	Tipo	Descrição
`file`	Arquivo	O objeto do arquivo (para obter mais informações, consulte https://developer. mozilla). org/en-US/docs/Web /API/File) queremos fazer o upload para o servidor.

Name	Tipo	Descrição
`dir`	string	O caminho absoluto no servidor no qual queremos fazer o upload do arquivo.

TimezoneRedirectionErrorCode

Os enums de TimezoneRedirectionError código disponíveis no módulo DCV

- `INVALID_ARGUMENT`
- `NO_CHANNEL`
- `USER_CANNOT_CHANGE`

Tipo:

- número

TimezoneRedirectionSettingCode

Os enums de TimezoneRedirectionSetting código disponíveis no módulo DCV

- `ALWAYS_OFF`
- `ALWAYS_ON`
- `CLIENT_DECIDES`

Tipo:

- número

TimezoneRedirectionStatusCode

Os enums de TimezoneRedirectionStatus código disponíveis no módulo DCV

- `SUCCESS`
- `PERMISSION_ERROR`

- `GENERIC_ERROR`

Tipo:

- número

WebcamErrorCode

Os enums de WebcamError código disponíveis no módulo DCV

- `SETTING_WEBCAM_FAILED`
- `CHANNEL_NOT_AVAILABLE`

Tipo:

- número

httpExtraSearchParamsCallback(método, url, corpo)

A função de retorno de chamada a ser chamada para injetar parâmetros de consulta personalizados URLs durante a autenticação e o estabelecimento da conexão. Isso permite cenários avançados de integração, incluindo a capacidade de acrescentar parâmetros de consulta personalizados e adicionar valores assinados do AWS Signature Version 4 (SigV4) para proteger e autorizar conexões por meio de sistemas externos.

Esse retorno de chamada também é usado para personalizar a URL usada ao estabelecer uma WebSocket conexão com o servidor Amazon DCV.

Parâmetros:

Name	Tipo	Descrição
`method`	string	O método HTTP que está sendo usado para a solicitaç ão.
`url`	string	O URL que será usado para a solicitação.

Name	Tipo	Descrição
`body`	string	O conteúdo do corpo da solicitação.

Retorna:

`URLSearchParams`objeto contendo os parâmetros de consulta personalizados a serem anexados ao URL.

Tipo

URLSearchParâmetros

httpExtraHeadersRetorno de chamada (método, url, corpo)

A função de retorno de chamada a ser chamada para inserir cabeçalhos personalizados (como`Authorization`) na solicitação HTTP durante o estabelecimento da conexão.

Parâmetros:

Name	Tipo	Descrição
`method`	string	O método HTTP que está sendo usado para a solicitaç ão.
`url`	string	O URL que será usado para a solicitação.
`body`	string	O conteúdo do corpo da solicitação.

Retorna:

Objeto contendo pares de valores-chave que representam os cabeçalhos personalizados a serem adicionados à solicitação HTTP.

Tipo

Objeto

Classe de conexão

A Classe de conexão obtida chamando o método `connect` do modulo `dcv`. Para ver um exemplo de como usá-lo, consulte a seção Introdução.

Expõe

- Métodos

Métodos

Lista

- attachDisplay (win, displayConf) → {Promise. <number>|Promessa. < {código: MultiMonitorErrorCode, mensagem: string} >}
- captureClipboardEvents(ativado, win, displayID) → {void}
- detachDisplay(displayId) → {void}
- disconnect() → {void}
- disconnectCollaborator(connectionId) → {void}
- enableDisplayQualityAtualizações (ativar) → {void}
- enableHighPixelDensidade (ativar) → {vazio}
- enableTimezoneRedirection(habilitar) → {Promise|Promessa. < {código: TimezoneRedirectionErrorCode, mensagem: string} >}
- enterRelativeMouseModo () → {void}
- getConnectedDevices() → {Promessa. <Matriz. < MediaDeviceInfo >>|Promessa. < {message: string} >}
- getFileExplorer() → {Promessa. < armazenamento de arquivos >|Promessa. < {código: ChannelErrorCode, mensagem: string} >}
- getServerInfo() → {Informações do servidor}
- getScreenshot () → {Promise|Promessa. < {código: ScreenshotErrorCode, mensagem: string} >}
- getStats() → {stats}

- latchModifierKey(chave, localização, isDown) → {boolean}
- OpenChannel (nome, authToken, retornos de chamada, namespace) → {Promise|Promise. < {código: ChannelErrorCode, mensagem: string} >}
- queryFeature(featureName) → {Promise.<{enabled: boolean, remote?: string, autoCopy?: boolean, autoPaste?: boolean, serviceStatus?: string, available?: boolean}>|Promise.<{message: string}>}
- registerKeyboardShortcuts(atalhos) → {void}
- requestDisplayConfig(highColorAccuracy) → {Promessa|Promessa. < {código: DisplayConfigErrorCode, mensagem: string} >}
- requestDisplayLayout(layout) → {Promise|Promessa. < {código: ResolutionErrorCode, mensagem: string} >}
- Solicitar resolução (largura, altura) → {Promise|Promise. < {código: ResolutionErrorCode, mensagem: string} >}
- sendKeyboardEvent(evento) → {boolean}
- sendKeyboardShortcut(atalho) → {void}
- setDisplayQuality(min, maxopt) → {vazio}
- setDisplayScale(ScaleRatio, DisplayID) → {Promise|Promise. < {código: ResolutionErrorCode, mensagem: string} >} (OBSOLETO)
- setKeyboardQuirks(peculiaridades) → {vazio}
- setMaxDisplayResolução (maxWidth, maxHeight) → {void}
- SetMicrophone (enable) → {Promise|Promise. < {código: AudioErrorCode, mensagem: string} >}
- setMinDisplayResolução (minWidth, minHeight) → {void}
- setUploadBandwidth(valor) → {número}
- setVolume(volume) → {void}
- setMicrophone (habilitar, ID do dispositivo) → {Promise|Promise. < {código: AudioErrorCode, mensagem: string} >}
- setWebcam (habilitar, ID do dispositivo) → {Promise|Promise. < {código: WebcamErrorCode, mensagem: string} >}
- syncClipboards() → {boolean}

attachDisplay (win, displayConf) → {Promise. <number>|Promessa. < {código: MultiMonitorErrorCode, mensagem: string} >}

Anexa uma tela específica em uma janela. Você não pode anexar a tela principal. Se bem-sucedida, a função retornará `displayId`.

Parâmetros:

Name	Tipo	Descrição
`win`	Objeto	A janela à qual a exibição deve ser anexada.
`displayConf`	Objeto	A configuração da exibição. Name: `displ` Tipo: númer Atribut: <opcic> Descrição: O ID da exibição. Name: `displ vName` Tipo: Atribut: Descrição: O nome do título da div. da exibição.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa. <number>| Promessa. < {código: MultiMonitorErrorCode, mensagem: string} >

captureClipboardEvents(ativado, win, displayID) → {void}

Inicia ou para de ouvir eventos de copiar e colar. No caso de pranchetas interativas (sempre no caso de colar), precisamos começar a ouvir os copy/paste eventos. Pode ser útil iniciar e parar de ouvir somente quando necessário, por exemplo, quando um modal é exibido.

Parâmetros:

Name	Tipo	Atributos	Descrição
`enabled`	boolean		Para começar a ouvir eventos, especifiq ue `true`. Para parar de ouvir eventos, especifique `false`.
`win`	Objeto	<opcional>	A janela na qual se deve ouvir os eventos. Se omitido, a janela padrão será usada.
`displayId`	número	<opcional>	O ID do monitor que deve ouvir os eventos. Se omitido, o monitor padrão da janela será usada.

Retorna:

Tipo

nulo

detachDisplay(displayId) → {void}

Desconecta um monitor específico. O monitor principal não pode ser desconectado.

Parâmetros:

Name	Tipo	Descrição
`displayId`	número	O ID da tela a ser desconectado.

Retorna:

Tipo

nulo

disconnect() → {void}

Desconecta-se do servidor Amazon DCV e fecha a conexão.

Retorna:

Tipo

nulo

disconnectCollaborator(connectionId) → {void}

Solicita a desconexão do colaborador conectado com o ID de conexão fornecido (desde o SDK do Cliente Web do Amazon DCV versão 1.1.0).

Parâmetros:

Name	Tipo	Descrição
`connectionId`	boolean	O ID da conexão que será desconectada.

Retorna:

Tipo

nulo

enableDisplayQualityAtualizações (ativar) → {void}

Ativa ou desativa as atualizações de qualidade de exibição para áreas de streaming que não recebem atualizações. Desativar as atualizações da qualidade de exibição reduz o uso da largura de banda, mas também diminui a qualidade da tela.

Parâmetros:

Name	Tipo	Descrição
`enable`	boolean	Para ativar as atualizações da qualidade de exibição, especifique `true`. Para desativar as atualizações da qualidade de exibição, especifique `false`.

Retorna:

Tipo

nulo

enableHighPixelDensidade (ativar) → {vazio}

Ativa ou desativa a alta densidade de pixels no cliente.

Parâmetros:

Name	Tipo	Descrição
`enable`	booleano	Se a alta densidade de pixels deve ou não ser ativada.

Retorna:

Tipo

nulo

enableTimezoneRedirection(habilitar) → {Promise|Promessa. < {código: TimezoneRedirectionErrorCode, mensagem: string} >}

Ativa ou desativa o redirecionamento de fuso horário. Depois de ativado, o cliente solicita que o servidor faça com que o fuso horário da área de trabalho do servidor corresponda ao fuso horário do cliente.

Parâmetros:

Name	Tipo	Descrição
`enable`	boolean	Para ativar o redirecionamento de fuso horário, especifiq ue `true`. Para desativar o redirecionamento de fuso horário, especifique `false`.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa. <number>| Promessa. < {código: TimezoneRedirectionErrorCode, mensagem: string} >

enterRelativeMouseModo () → {void}

Ativa o modo relativo do mouse.

Retorna:

Tipo

nulo

getConnectedDevices() → {Promessa. <Matriz. < MediaDeviceInfo >>|Promessa. < {message: string} >}

Solicita uma lista dos dispositivos de mídia conectados ao computador cliente.

Retorna:

Se for bem-sucedido, ele retornará uma promessa que é resolvida em uma matriz de MediaDeviceInfo objetos. Para obter mais informações, consulte https://developer.mozilla. org/en-US/docs/Web/API/MediaDeviceInfo. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa. <Matriz. < MediaDeviceInfo >> | Promessa. < {message: string} >

getFileExplorer() → {Promessa. < armazenamento de arquivos >|Promessa. < {código: ChannelErrorCode, mensagem: string} >}

Obtém um objeto para gerenciar o armazenamento de arquivos do servidor Amazon DCV.

Retorna:

Promise. Resolve para o objeto explorador de arquivos, se preenchido, ou para um objeto de erro, se rejeitado.

Tipo

Promessa. < filestorage > | Promessa. < {código: ChannelErrorCode, mensagem: string} >

getServerInfo() → {Informações do servidor}

Obtém informações sobre o servidor Amazon DCV.

Retorna:

Informações sobre o software do servidor.

Tipo

serverInfo

getScreenshot () → {Promise|Promessa. < {código: ScreenshotErrorCode, mensagem: string} >}

Recupera a captura de tela da área de trabalho remota no formato PNG. A captura de tela será retornada no observador screenshotCallback. Em vez disso, `null` será retornado em caso de falhas.

Retorna:

Promessa que será resolvida se a solicitação for processada. Se rejeitado, recebemos um objeto de erro.

Tipo

Promessa | Promessa. < {código: ScreenshotErrorCode, mensagem: string} >

getStats() → {stats}

Obtém estatísticas sobre o servidor Amazon DCV.

Retorna:

Informações sobre as estatísticas de streaming.

Tipo

stats

latchModifierKey(chave, localização, isDown) → {boolean}

Envia um único evento `keydown` ou `keyup` do teclado para um modificador permitido.

Parâmetros:

Name	Tipo	Descrição
`key`	Controle | Alt | AltGraph | Meta | OS | Shift	A tecla para enviar.
`location`	KeyboardEvent.localização	O local da tecla. Para obter mais informações, consulte https://developer.mozilla. org/en-US/docs/Web/API/KeyboardEvent/location.
`isDown`	boolean	Se o evento chave a ser injetado for um keydown (`true`) ou um keyup (`false`).

Retorna:

Se a combinação solicitada for válida, a função retornará `true`, caso contrário, retornará `false`.

Tipo

boolean

OpenChannel (nome, authToken, retornos de chamada, namespace) → {Promise| Promise. < {código: ChannelErrorCode, mensagem: string} >}

Abre um canal de dados personalizado na conexão se ela tiver sido criada no Servidor Amazon DCV.

Parâmetros:

Name	Tipo	Descrição
`name`	string	O nome do canal.

Name	Tipo	Descrição
`authToken`	string	O token de autenticação a ser usado ao se conectar ao servidor.
`callbacks`	Objeto	Os retornos de chamada onMessage e onClose funcionam para chamar.
`namespace`	string	O namespace do modelo. Disponível desde o SDK do cliente web do Amazon DCV 1.2.0 e o Servidor Amazon DCV 2022.1.

Retorna:

Promise. Se rejeitado, recebemos um objeto de erro.

Tipo

Promessa | Promessa. < {código: ChannelErrorCode, mensagem: string} >

queryFeature(featureName) → {Promise.<{enabled: boolean, remote?: string, autoCopy?: boolean, autoPaste?: boolean, serviceStatus?: string, available?: boolean}>|Promise.<{message: string}>}

Consulta o status de um recurso específico do servidor Amazon DCV.

Parâmetros:

Name	Tipo	Descrição
`featureName`	feature	O nome do atributo a ser consultado.

Retorna:

Promise. Se resolvida, a função retorna um objeto `status` que sempre contém uma propriedade `enabled` e, possivelmente, também outras propriedades. Se rejeitada, a função retornará um objeto `error`.

Tipo

{Promise.<{enabled: boolean, remote?: string, autoCopy?: boolean, autoPaste?: boolean, serviceStatus?: string, available?: boolean}> | Promise.<{message: string}>

registerKeyboardShortcuts(atalhos) → {void}

Registra os atalhos de teclado.

Parâmetros:

Name	Tipo	Descrição
`shortcuts`	Array.<Object>	A matriz de chaves e mapeamentos a serem registrados. Name: `sequenc` Tipo: Array.<Object> Descrição: O atalho do teclado para se registrar.

Name	Tipo	Descrição
		Name | Tipo | Descrição
		N | T | Descrição
		k | K v e | O valor da tecla pressionada pelo usuário. Para obter mais informações, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/key. | KeyboardEvent

Name	Tipo	Descrição
		Name | Tipo | Descrição
		Name | Tipo | Descrição
		l | Keyboard.getLayoutMap() | Array de chaves a serem enviadas. A localização da tecla no teclado. Para obter mais informações, consulte https://developer.mozilla.org/en-US/docs/Web/

Name	Tipo	Descrição
		Name: `output` — Tipo: Array.<Object> — Descrição: A ação que deve ser executada pelo atalho. Name: `keyboardEvent` — Tipo: KeyboardEvent — Descrição: O valor da tecla pressionada pelo usuário. Para obter mais informações, consulte API/KeyboardEvent/location.

Name	Tipo	Descrição
		Name | Tipo | Descrição N | T | Descrição https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/key. l | K v li | Array de matrizes de chaves a serem enviadas. A localização da tecla no teclado. Para

Name	Tipo	Descrição
		Name Tipo Descrição
		N T Descrição obter mais informações, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/location.

Retorna:

Tipo

nulo

requestDisplayConfig(highColorAccuracy) → {Promessa|Promessa. < {código: DisplayConfigErrorCode, mensagem: string} >}

Solicita uma configuração de exibição atualizada do servidor Amazon DCV. Disponível desde o SDK do cliente web do Amazon DCV 1.1.0 e o Servidor Amazon DCV 2022.0.

Parâmetros:

Name	Tipo	Descrição
`highColorAccuracy`	boolean	Se a alta precisão de cores deve ou não ser solicitada.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa | Promessa. < {código: DisplayConfigErrorCode, mensagem: string} >

requestDisplayLayout(layout) → {Promise|Promessa. < {código: ResolutionErrorCode, mensagem: string} >}

Solicita um layout de exibição atualizado para a conexão.

Parâmetros:

Name	Tipo	Descrição
`layout`	Array.<Monitor>	O solicitado é exibido no layout.

Retorna:

Promise. Se rejeitado, recebemos um objeto de erro.

Tipo

Promessa | Promessa. < {código: ResolutionErrorCode, mensagem: string} >

Solicitar resolução (largura, altura) → {Promise|Promise. < {código: ResolutionErrorCode, mensagem: string} >}

Solicita uma resolução de exibição atualizada do servidor Amazon DCV.

Parâmetros:

Name	Tipo	Descrição
`width`	número	A largura a ser solicitada em pixels. O valor mínimo permitido é `0`.
`height`	número	A altura a ser solicitada em pixels. O valor mínimo permitido é `0`.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa | Promessa. < {código: ResolutionErrorCode, mensagem: string} >

sendKeyboardEvent(evento) → {boolean}

Envia um evento de atalho de teclado. Para obter mais informações sobre eventos de teclado, consulte https://developer.mozilla. org/en-US/docs/Web/API/KeyboardEvent. Os eventos de teclado válidos incluem: `keydown`, `keypress` e `keyup`. Para obter mais informações sobre esses eventos, consulte https://developer.mozilla. org/en-US/docs/Web/API/KeyboardEvent#events.

Parâmetros:

Name	Tipo	Descrição
`event`	KeyboardEvent	O evento de teclado a ser enviado.

Retorna:

Se o evento não for válido, a função retornará `false`. Se o evento for válido, a função retornará `true`.

Tipo

boolean

sendKeyboardShortcut(atalho) → {void}

Envia um atalho de teclado. Use essa função para enviar uma sequência completa de `keydown` ou `keyup`. Por exemplo, enviar Ctrl + Alt + Del envia os eventos `keydown` para todas as teclas seguidas pelos eventos `keyup`. Use essa função mesmo se quiser enviar uma única chave.

Parâmetros:

Name	Tipo	Descrição
`shortcut`	Array.<Object>	A matriz de chaves a serem enviadas. Name: `key` Tipo: Keyboard vent.chave Descrição: O valor da tecla pressionada pelo

Name	Tipo	Descrição
		Name / Tipo / Descrição
		usuário. Para obter mais informações, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/key.
		`locatic` / Keyboarc vent.localização / A matriz de chaves a serem enviadas. A localização da tecla no teclado.

Name	Tipo	Descrição
		Name: Tipo: Descrição: Para obter mais informações, consulte https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/location.

Retorna:

Tipo

nulo

setDisplayQuality(min, maxopt) → {vazio}

Define a qualidade da imagem a ser usada para a conexão. O intervalo válido é `0` a `100`, sendo `1` a menor qualidade de imagem e `100` a mais alta qualidade de imagem. Especifique `0` para reter o valor atual.

Parâmetros:

Name	Tipo	Atributos	Descrição
`min`	número		A qualidade mínima da imagem.
`max`	número	<opcional>	A qualidade máxima da imagem.

Retorna:

Tipo

nulo

setDisplayScale(ScaleRatio, DisplayID) → {Promise|Promise. < {código: ResolutionErrorCode, mensagem: string} >} (OBSOLETO)

Obsoleto desde a versão 1.3.0. Não há mais necessidade de definir a escala de exibição. As coordenadas do mouse serão gerenciadas internamente de forma automática.

Notifica o Amazon DCV de que a tela está escalada no lado do cliente. Use isso para notificar o servidor de que ele precisa escalar os eventos do mouse para corresponder à proporção de exibição do cliente.

Parâmetros:

Name	Tipo	Descrição
`scaleRatio`	flutuação	A taxa de escala a ser usada. Deve ser um número estritamente positivo.
`displayId`	número	O ID da tela a ser escalada.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa | Promessa. < {código: ResolutionErrorCode, mensagem: string} >

setKeyboardQuirks(peculiaridades) → {vazio}

Define peculiaridades do teclado para o computador cliente.

Parâmetros:

Name	Tipo	Descrição
`quirks`	Objeto	As peculiaridades do teclado a serem ativadas ou desativadas. Name: `macOptiToAlt` Tipo: boolean Descrição: Para mapear a tecla Option como Alt no macOS, especifique `true`. Caso contrário, especifique `false`.

Name	Tipo	Descrição
		<table><tr><th>Name</th><th>Tipo</th><th>Descrição</th></tr><tr><td>`macCommdToContl`</td><td>boolean</td><td>Para mapear a tecla Command como Ctrl no macOS, especifique `true`. Caso contrário, especifique `false`.</td></tr></table>

Retorna:

Tipo

nulo

setMaxDisplayResolução (maxWidth, maxHeight) → {void}

Define a resolução máxima da tela a ser usada para a conexão.

Parâmetros:

Name	Tipo	Descrição
`maxWidth`	número	A largura máxima da tela em pixels. O valor mínimo permitido é `0`.
`maxHeight`	número	A altura máxima da tela em pixels. O valor mínimo permitido é `0`.

Retorna:

Tipo

nulo

SetMicrophone (enable) → {Promise|Promise. < {código: AudioErrorCode, mensagem: string} >}

Ativa ou desativa o microfone.

Parâmetros:

Name	Tipo	Descrição
`enable`	boolean	Para ativar o microfone , especifique `true`. Para desativar o microfone, especifique `false`.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa | Promessa. < {código: AudioErrorCode, mensagem: string} >

setMinDisplayResolução (minWidth, minHeight) → {void}

Define a resolução mínima da tela a ser usada para a conexão. Alguns aplicativos podem exigir uma resolução mínima da tela. Se a resolução mínima exigida for maior que a resolução máxima compatível com o cliente, será usada uma estratégia de redimensionamento. Use essa função com cuidado. A estratégia de redimensionamento pode fazer com que o sistema de entrada do mouse e do teclado fique menos preciso.

Parâmetros:

Name	Tipo	Descrição
`minWidth`	número	A largura mínima da tela em pixels. O valor mínimo permitido é `0`.
`minHeight`	número	A largura mínima da tela em pixels. O valor mínimo permitido é `0`.

Retorna:

Tipo

nulo

setUploadBandwidth(valor) → {número}

Define a largura de banda máxima a ser usada para carregar arquivos no servidor Amazon DCV.

Parâmetros:

Name	Tipo	Descrição
`value`	número	O limite máxima da largura de banda em Kbps. O intervalo válido é de 1.024 kbps a 102.400 kbps.

Retorna:

- O limite de largura de banda definido. `null` se o atributo de armazenamento de arquivos estiver desativado no servidor.

Tipo

número

setVolume(volume) → {void}

Define o nível de volume a ser usado para áudio. O intervalo válido é de 0 a 100, sendo 0 o volume mais baixo e 100 sendo o volume mais alto.

Parâmetros:

Name	Tipo	Descrição
`volume`	número	O nível de volume a ser usado.

Retorna:

Tipo

nulo

setMicrophone (habilitar, ID do dispositivo) → {Promise|Promise. < {código: AudioErrorCode, mensagem: string} >}

[Experimental - pode mudar no futuro] Ativa ou desativa o microfone.

Parâmetros:

Name	Tipo	Descrição
`enable`	boolean	Para ativar o microfone , especifique `true`. Para desativar o microfone, especifique `false`.
`deviceId`	string	O ID do dispositivo do microfone. Se nenhum deviceId for fornecido, o deviceId `default` será usado.

Retorna:

Promise. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa | Promessa. < {código: AudioErrorCode, mensagem: string} >

setWebcam (habilitar, ID do dispositivo) → {Promise|Promise. < {código: WebcamErrorCode, mensagem: string} >}

Ativa ou desativa a webcam.

Parâmetros:

Name	Tipo	Descrição
`enable`	boolean	Para ativar a webcam, especifique `true`. Para

Name	Tipo	Descrição
		desativar a webcam, especifiq ue `false`.
`deviceId`	string	O ID do dispositivo da webcam.

Retorna:

Prometa que, se for bem-sucedida, será resolvida para o ID do dispositivo da attached/detached webcam. Se rejeitada, a promessa retornará um objeto de erro.

Tipo

Promessa. <string>| Promessa. < {código: WebcamErrorCode, mensagem: string} >

syncClipboards() → {boolean}

Sincroniza a área de transferência do cliente local com a área de transferência remota do servidor Amazon DCV. A cópia automática deve ser compatível com o navegador.

Retorna:

Se as áreas de transferência tiverem sido sincronizadas, a função retornará `true`. Se as áreas de transferência não tiverem sido sincronizadas ou se o navegador não for compatível com a cópia automática, a função retornará `false`.

Tipo

boolean

Classe de autenticação

A classe de autenticação deve ser usada para obter um token de autenticação chamando o `authenticate`método do módulo `dcv`. Para ver um exemplo de como usá-lo, consulte a seção Introdução.

Expõe

- Métodos

Métodos

Lista

- retry() → {void}
- sendCredentials(credentials) → {void}

retry() → {void}

Recupera o processo de autenticação.

Retorna:

Tipo

nulo

sendCredentials(credentials) → {void}

Envia as credenciais de autenticação fornecidas pelo cliente para o servidor Amazon DCV.

Parâmetros:

Name	Tipo	Descrição
`credentials`	Objeto	O objeto que contém as credenciais fornecidas. As credenciais devem ter o mesmo nome e ser do mesmo tipo especificado no desafio.

Retorna:

Tipo

nulo

Classe de recurso

A classe de recurso pode buscar ou descartar o arquivo correspondente que acabou de ser impresso ou baixado. Ao realizar essas ações, as funções do observador correspondentes `filePrinted` e `fileDownload`, respectivamente, seriam invocadas com o objeto de recurso como único argumento. Esse recurso pode ser aceito ou recusado para buscar ou descartar o arquivo ao qual eles fazem referência.

Expõe

- Métodos

Métodos

Lista

- accept(urlParameters) → {void}
- decline() → {void}

accept(urlParameters) → {void}

Baixa localmente o recurso.

Parâmetros:

Name	Tipo	Descrição
`urlParameters`	Objeto	O objeto opcional que contém os key/value pares dos parâmetros de pesquisa de URL passados para a solicitaç ão de busca do recurso.

Retorna:

Tipo

nulo

decline() → {void}

Descarta o recurso.

Retorna:

Tipo

nulo

SDK de Interface do Usuário da Web do Amazon DCV

Uma biblioteca de componentes JavaScript React, atualmente exportando um único componente React chamado, `DCVViewer` que se conecta ao Amazon DCV Server e renderiza a barra de ferramentas para interagir com o stream remoto.

Expõe

- Componentes

Componentes

Lista

- DCVViewer

DCVViewer

O componente React que renderiza a barra de ferramentas com todas as suas funcionalidades úteis para interagir com o fluxo remoto.

Propriedades:

Lista

- dcv
- uiConfig

dcv

Name	Tipo	Obrigatório	Descrição
`dcv`	Objeto	Sim	O objeto que define as propriedades necessárias para estabelecer a conexão com o Servidor Amazon DCV, definindo o nível do log e a URL de onde carregar os ativos do SDK do Cliente Web do Amazon DCV e acessar os recursos do DCV.

Nam	Tipo	Obri io	Descrição
`ses`	Strir	Sim	O ID da sessão do Amazon DCV.
`aut`	Strir	Sim	O token de autentica ção a ser

Name	Tipo	Obrigatório	Descrição
			Nam Tipo Obri io Descrição usado ao se conectar ao servidor. ser Strir Sim O nome do host e a porta do servidor Amazon DCV em execução no seguinte formato: https:// d cv_host_a ddress:po rt. Por exemplo: https: //:8443 -

Name	Tipo	Obrigatório	Descrição
			Nam Tipo Obri io Descrição dcv-server. bas Strir Sim O URL absoluto ou relativo do qual carregar arquivos do SDK. res ase Strir Não (pad "") O URL absoluto ou relativo a partir do qual acessar os recursos do DCV. onD ect func Não (pad A função

Name	Tipo	Obrigatório	Descrição
			Nam Tipo Obri io Descrição ( ) => {} de retorno de chamada é invocada ao se desconectar do servidor Amazon DCV e a conexão é fechada. `log` Logl Não (pad `Log` `INF` O nível de `l.` log a ser usado no visualizador.

uiConfig

Name	Tipo	Obrigatório	Descrição
`uiConfig`	Objeto	Não (padrão: `{}`)	O objeto que define as propriedades para configurar se a barra de ferramentas está visível e se os botões de tela cheia e de vários monitores devem ser exibidos nela. Nam Tipo Obri io Descrição too Obje Não (pad {}) O objeto que define as opções de configura ção da barra de ferrament as.

Name	Tipo	Obrigatório	Descrição
			Nam Tipo Obri io Descrição
			Obrigatório
			Boolean (opção: para) definir se deseja mostrar ou ocultar a barra de ferramenas.
			Boolean (opção: para) definir se deseja mostrar ou ocultar o botão de tela cheia

Name	Tipo	Obrigatório	Descrição
			Nam Tipo Obri io Descrição io na barra de ferramen as. (opção para) definir se deseja mostrar ou ocultar o botão de vários monitore na barra de ferramen as.

Notas de versão e histórico de documentos do SDK do cliente web do Amazon DCV

Esta página traz as notas da versão e o histórico de documentos do SDK do cliente web do Amazon DCV.

Tópicos

- Notas de versão do SDK do cliente web do Amazon DCV
- Histórico do documento

Notas de versão do SDK do cliente web do Amazon DCV

Esta seção fornece notas de versão do SDK do cliente web do Amazon DCV por data de versão.

Tópicos

- 1.8.7 — 31 de outubro de 2024
- 1.8.4 — 1º de outubro de 2024
- 1.5.10: 19 de dezembro de 2023
- 1.5.6 — 9 de novembro de 2023
- 1.4.4 — 29 de junho de 2023
- 1.4.0 — 28 de março de 2023
- 1.3.1 — 9 de dezembro de 2022
- 1.3.0 — 11 de novembro de 2022
- 1.2.1 — 21 de julho de 2022
- 1.2.0 — 29 de junho de 2022
- 1.1.3 — 23 de maio de 2022
- 1.1.2 — 19 de maio de 2022
- 1.1.1 — 23 de março de 2022
- 1.1.0 — 23 de fevereiro de 2022
- 1.0.4 — 20 de dezembro de 2021
- 1.0.3 — 1º de setembro de 2021

- 1.0.2 — 30 de julho de 2021
- 1.0.1 — 31 de maio de 2021
- 1.0.0 — 24 de março de 2021

1.8.7 — 31 de outubro de 2024

Números de compilação	Alterações e correções de bugs	
• Versão semântica: 1.8.7 • Compilação: 858	• Renderização corrigida no Firefox 130 e versões mais recentes	

1.8.4 — 1º de outubro de 2024

Números de compilação	Novos atributos	Alterações e correções de bugs
• Versão semântic: 1.8.4 • Compilaç o: 840	Os seguintes recursos foram adicionados: • Renomeado como “SDK do cliente web do Amazon DCV” • Nova enableHighPixel densidade de API adicionada para telas de alto dpi •	• Manuseio aprimorado da webcam • Manuseio aprimorado da reprodução de áudio • WebCodecs Manuseio aprimorado •

Números de compilação	Novos atributos	Alterações e correções de bugs
	Foi adicionada uma API experimental setMicrophone para selecionar o microfone em navegadores compatíveis • Foram adicionados novos erros de conexão GATEWAY_BUSY, UNSUPPORTED_CREDENTIAL e TRANSPORT_ERROR • Foram adicionados novos motivos de encerramento EXTERNAL_PROTOCOL_CONNECTION_EVICTED e DISCONNECTION_REQUESTED	Conexão e desconexão aprimorados do microfone e da webcam • Capacidade aprimorada de arrastar janelas remotas em vários monitores • As permissões de upload e download de armazenamento de arquivos agora são propagadas corretamente • Pequenas correções na renderização

1.5.10: 19 de dezembro de 2023

Versão	Notas de versão
• Versão semântica: 1.5.10 • Compilação: 684	Alterações e correções de bugs • Corrigir erros de decodificação de fluxo

1.5.6 — 9 de novembro de 2023

Versão	Notas de versão

Versão	Notas de versão
• Versão semântica: 1.5.6 • Compilação: 659	Alterações e correções de bugs • Melhorias de desempenho na decodificação e renderização do streaming • Removido o suporte para Internet Explorer 11

1.4.4 — 29 de junho de 2023

Versão	Notas de versão
• Versão semântica: 1.4.4 • Compilação: 573	Alterações e correções de bugs • O componente de interface do usuário do visualizador agora usa a API `navigator.keyboard.lock` em navegadores compatíveis com ela para lidar com teclas especiais em tela cheia. • Corrigido um problema que poderia apresentar cores erradas no Chrome 114 ou mais recente. • WebCodecs Detecção aprimorada. • Corrigido um problema com o estado do botão do mouse ao entrar na janela. • Corrigido um problema que poderia fazer com que as teclas modificadoras permanecessem pressionadas no macOS. •

Versão	Notas de versão
	Robustez de áudio aprimorada para condições de rede degradadas. • Corrigido um vazamento de memória. • Logs aprimorados para incluir tempo e nível.

1.4.0 — 28 de março de 2023

Versão	Notas de versão
• Versão semântica: 1.4.0 • Compilação: 476	Novos atributos • Foi adicionado um novo método `uploadFiles` ao objeto `FileStorage` para carregar vários arquivos. • O componente de interface do usuário do visualizador agora é compatível com arrastar e soltar para iniciar o upload do arquivo. • A API WebCodecs do navegador agora é usada também para áudio e webcam. Alterações e correções de bugs • Corrigidos os vazamentos de memória relacionados a conexões repetidas da mesma página. • `setUploadBandwidth` agora permite valores de até 1 Gbps.

Versão	Notas de versão
	• Renderização otimizada dos componentes da interface do usuário. • Corrigida a compatibilidade com cursores animados no Windows. • Corrigido um problema com a compatibi lidade da área de transferência quando dados de texto e imagem estão presentes na mesma operação. • Maior robustez da API da Webcam: as configurações não podem ser alteradas enquanto uma solicitação já está em andamento. `webcam.setEnabled` agora acompanha o ID do dispositivo quando a solicitação está em andamento e retorna uma Promessa. O componente de interface do usuário do visualizador exibe a notificação em caso de erro.

1.3.1 — 9 de dezembro de 2022

Versão	Notas de versão
• Versão semântica: 1.3.1 • Compilação: 413	Alterações e correções de bugs • Corrigido um problema que poderia fazer com que a interface do usuário de redirecio namento de fuso horário perdesse a sincronização com o servidor. •

Versão	Notas de versão
	Corrigido um vazamento de memória após várias reconexões. • Corrigido um problema que fazia com que uma página ficasse em branco na desconexã o. • Corrigido um bug que fazia com que os avisos do console no decodificador de áudio fechassem.

1.3.0 — 11 de novembro de 2022

Versão	Notas de versão
• Versão semântica: 1.3.0 • Compilação: 407	Novos recursos • Adotou o Cloudscape (ponenthttps://clo udscape.design) for the UI Viewer com. • Adicionada compatibilidade para redirecio namento de fuso horário. Alterações e correções de bugs • Corrigida a falta de atualização na área de transferência assíncrona quando o visualiza dor do DCV está focado. •

Versão	Notas de versão
	A função `setDisplayScale` não é mais necessária ao escalar a tela no lado do cliente. • O componente `DCVViewer` agora chama `disconnect()` automaticamente quando é desmontado.

1.2.1 — 21 de julho de 2022

Versão	Notas de versão
• Versão semântica: 1.2.1 • Compilação: 358	Alterações e correções de bugs • Corrigido um problema que resultava em uma falha na conexão com o servidor Amazon DCV 2019.1 e versões anteriores.

1.2.0 — 29 de junho de 2022

Versão	Notas de versão
• Versão semântica: 1.2.0 • Compilação: 352	Alterações e correções de bugs • Corrigido o erro de falha quando os quadros recebidos são maiores do que a resolução máxima suportada (4.096 x 2.160). • Os objetos de atributo (transmitidos como argumentos para os observadores `fileDownload` e `filePrinted`) agora

Versão	Notas de versão
	têm os métodos `accept` e `decline` que podem ser chamados no objeto para baixar e descartar o recurso, respectivamente. • Pequena correção de bug na sincronização automática da área de transferência ao desconectar.

1.1.3 — 23 de maio de 2022

Versão	Notas de versão
• Versão semântica: 1.1.3 • Compilação: 329	Alterações e correções de bugs • Corrigido um problema que impedia a conexão bem-sucedida ao especificar a web-url-path opção.

1.1.2 — 19 de maio de 2022

Versão	Notas de versão
• Versão semântica: 1.1.2 • Compilação: 322	Alterações e correções de bugs • Corrigido um problema que poderia fazer com que a entrada não funcionasse corretamente após a conexão. • Corrigidas as coordenadas do mouse quando a proporção da escala é maior que 1.

1.1.1 — 23 de março de 2022

Versão	Notas de versão
• Versão semântica: 1.1.1 • Compilação: 309	Alterações e correções de bugs • Relate `Transport Error` quando a comunicação com o servidor atingir o tempo limite. • Corrigido um erro de decodificação recorrent e ao transmitir grandes resoluções.

1.1.0 — 23 de fevereiro de 2022

Versão	Notas de versão
• Versão semântica: 1.1.0 • Compilação: 295	Novos recursos • Inicie a biblioteca do SDK da interface do usuário da Web do Amazon DCV com o componente do React`DCVViewer` . • Exporte o SDK do cliente web do Amazon DCV como módulos UMD e ES. • Adicionada compatibilidade de alta precisão de cores. • Adicionada a capacidade de listar e interagir com clientes conectados a uma sessão. Adicionadas notificações para conexão e desconexão.

Versão	Notas de versão
	Alterações e correções de bugs • Compatibilidade aprimorada de decodific ação de webcodecs. • Várias melhorias no teclado. • Corrigido um bug que estava impedindo a abertura de uma segunda tela quando a área de transferência estava desativada.

1.0.4 — 20 de dezembro de 2021

Versão	Notas de versão
• Versão semântica: 1.0.4 • Compilação: 249	Novos atributos • Compatibilidade para abrir várias conexões na mesma página. • Compatibilidade para carregar o SDK a partir de uma CDN.

1.0.3 — 1º de setembro de 2021

Versão	Notas de versão
• Versão semântica: 1.0.3 • Compilação: 202	Novos atributos • Suporte experimental para WebCodecs . Isso está desativado por padrão e deve

Versão	Notas de versão
	ser ativado por meio do objeto `ConnectionConfig` usando a nova propriedade `enableWebCodecs` . • Área de transferência: adicionada compatibilidade para tipos de dados `image/png` em navegadores baseados em Chromium. • Adicionado observador/retorno de chamada para obter a captura de tela do servidor como uma imagem PNG (requer o servidor Amazon DCV 2021.2). Alterações e correções de bugs • Manipulação aprimorada dos modificadores de teclado.

1.0.2 — 30 de julho de 2021

Versão	Notas de versão
• Versão semântica: 1.0.2 • Compilação: 167	• Corrigimos a detecção de pressão para eventos com caneta. • Compatibilidade aprimorada para layout de teclado coreano no Chrome.

1.0.1 — 31 de maio de 2021

Versão	Notas de versão
• Versão semântica: 1.0.1 • Compilação: 141	• Corrigida a propagação de erros de conexão e motivos de fechamento • Corrigida a atualização do progresso do fragmento de armazenamento de arquivos • Manuseio aprimorado da webcam • Processamento aprimorado de entrada de áudio

1.0.0 — 24 de março de 2021

Versão	Notas de versão
• Versão semântica: 1.0.0 • Compilação: 81	Versão inicial do SDK do cliente web do Amazon DCV.

Histórico do documento

A tabela a seguir descreve a documentação desta versão do SDK do cliente web do Amazon DCV.

Alteração	Descrição	Data
Versão 1.8.4 do SDK do cliente web do Amazon DCV	O SDK do cliente da web do Amazon DCV 1.8.4 já está	1º de outubro de 2024

Alteração	Descrição	Data
	disponível. Para obter mais informações, consulte SDK v.1.8.4.	
Versão 1.5.6 do SDK do cliente web do Amazon DCV	O SDK do cliente da web do Amazon DCV 1.5.6 já está disponível. Para obter mais informações, consulte SDK v.1.5.6.	9 de novembro de 2023
Versão 1.4.4 do SDK do cliente web do Amazon DCV	O SDK do cliente da web do Amazon DCV 1.4.4 já está disponível. Para obter mais informações, consulte SDK v.1.4.4.	29 de junho de 2023
Versão 1.4.0 do SDK do cliente web do Amazon DCV	O SDK do cliente da web do Amazon DCV 1.4.0 já está disponível. Para obter mais informações, consulte SDK v.1.4.0.	28 de março de 2023
Versão 1.3.1 do SDK do cliente web do Amazon DCV	O SDK do cliente da web do Amazon DCV 1.3.1 já está disponível. Para obter mais informações, consulte SDK v.1.3.1.	9 de dezembro de 2022
Versão 1.3.0 do SDK do cliente web do Amazon DCV	O SDK do cliente da web do Amazon DCV 1.3.0 já está disponível. Para obter mais informações, consulte SDK v.1.3.0.	11 de novembro de 2022

Alteração	Descrição	Data
Versão 1.2.0 do SDK do cliente web do Amazon DCV	O SDK do cliente da web do Amazon DCV 1.2.0 já está disponível. Para obter mais informações, consulte SDK v.1.2.0.	29 de junho de 2022
Versão 1.1.0 do SDK do cliente web do Amazon DCV	O SDK do cliente da web do Amazon DCV 1.1.0 já está disponível. Para obter mais informações, consulte SDK v.1.1.0.	23 de fevereiro de 2022
Versão 1.0.1 do SDK do cliente web do Amazon DCV	Corrigidos alguns erros de digitação. Corrigidos pequenos bugs, consulte SDK v.1.0.1.	31 de maio de 2021
Lançamento inicial	Primeira publicação deste conteúdo.	24 de março de 2021

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.