Entwicklerhandbuch

Amazon WorkDocs

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

Amazon WorkDocs: Entwicklerhandbuch

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

..... iv
Was ist Amazon WorkDocs? 1
Zugreifen WorkDocs 1
Preisgestaltung 1
Ressourcen 2
Erste Schritte 3
Connect WorkDocs mit IAM-Benutzeranmeldedaten herstellen 3
Stellen Sie eine Verbindung her, WorkDocs indem Sie eine Rolle übernehmen 5
Hochladen eines Dokuments 8
Herunterladen eines Dokuments 10
Einrichten von Benachrichtigungen 10
Erstellen eines Benutzers 13
Benutzern Berechtigungen für eine Ressource gewähren 14
Authentifizierung und Zugriffskontrolle für administrative Anwendungen 16
Entwicklern Berechtigungen für die API gewähren WorkDocs 16
Erteilt Drittentwicklern die Erlaubnis für WorkDocs APIs 17
Benutzern die Erlaubnis erteilen, eine IAM-Rolle anzunehmen 19
Beschränken des Zugriffs auf eine bestimmte Instanz WorkDocs 19
Authentifizierung und Zugriffskontrolle für Benutzeranwendungen 21
Erteilung von Berechtigungen zum Aufrufen von WorkDocs APIs 21
Ordner IDs in API-Aufrufen verwenden 23
Erstellen einer Anwendung 24
Anwendungsbereiche 24
Autorisierung 25
Aufrufen WorkDocs APIs 27
WorkDocs Inhaltsmanager 29
Aufbau eines Content Managers WorkDocs 29
Herunterladen eines Dokuments 30
Hochladen eines Dokuments 31

Hinweis: Neukundenanmeldungen und Kontoerweiterungen sind für Amazon WorkDocs nicht mehr verfügbar. Informationen zu den Migrationsschritten finden Sie hier: So migrieren Sie Daten von WorkDocs.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.

Was ist Amazon WorkDocs?

Amazon WorkDocs ist ein System zum Speichern, Zusammenarbeiten und Teilen von Dokumenten. WorkDocs ist vollständig verwaltet, sicher und für Unternehmen geeignet. Es bietet umfassende administrative Kontrollen sowie Feedback-Funktionen, die zur Steigerung der Benutzerproduktivität beitragen. Dateien werden geschützt und sicher in der Cloud gespeichert. Die Dateien Ihrer Benutzer sind nur für diese sowie für ihre ausgewiesenen Beitragsleistenden und Betrachter sichtbar. Andere Mitglieder Ihrer Organisation haben auf Dateien anderer Benutzer keinen Zugriff, wenn ihnen nicht ausdrücklich Zugriff gewährt wurde.

Benutzer können ihre Dateien für andere Mitglieder Ihrer Organisation zur Zusammenarbeit oder Überprüfung freigeben. Die WorkDocs Client-Anwendungen können verwendet werden, um viele verschiedene Dateitypen anzuzeigen, abhängig vom Internet-Medientyp der Datei. WorkDocs unterstützt alle gängigen Dokument- und Bildformate, und Unterstützung für weitere Medientypen wird ständig hinzugefügt.

Weitere Informationen finden Sie auf Amazon WorkDocs.

Zugreifen WorkDocs

Endbenutzer verwenden die Client-Anwendungen für den Zugriff auf ihre Dateien. Benutzer ohne Administratorrechte müssen niemals die WorkDocs Konsole oder das Administrations-Dashboard verwenden. WorkDocs bietet mehrere verschiedene Client-Anwendungen und Dienstprogramme:

- Eine Webanwendung für die Verwaltung und Anzeige von Dokumenten
- Native Apps für Mobilgeräte für das Prüfen von Dokumenten
- WorkDocs Laufwerk, das verwendet wird, um einen Ordner auf Ihrem Mac oder Windows-Desktop mit Ihren WorkDocs Dateien zu synchronisieren.

Preisgestaltung

Bei WorkDocs gibt es keine Vorabgebühren oder Verpflichtungen. Sie zahlen nur für aktive Benutzerkonten und den von Ihnen genutzten Speicherplatz. Weitere Informationen finden Sie unter Preise.

Ressourcen

Die folgenden verwandten Ressourcen bieten Ihnen nützliche Informationen für die Arbeit mit diesem Service.

- Kurse und Workshops — Links zu rollen- und Spezialkursen sowie zu Übungen zum Selbststudium, mit denen Sie Ihre AWS Fähigkeiten verbessern und praktische Erfahrungen sammeln können.
- AWS Developer Center — Erkunden Sie Tutorials, laden Sie Tools herunter und erfahren Sie mehr über Veranstaltungen für Entwickler. AWS
- AWS Entwicklertools — Links zu Entwicklertools SDKs, IDE-Toolkits und Befehlszeilentools für die Entwicklung und Verwaltung von AWS Anwendungen.
- Ressourcencenter für die ersten Schritte — Erfahren Sie AWS-Konto, wie Sie Ihre erste Anwendung einrichten, der AWS Community beitreten und sie starten.
- Praktische Tutorials — Folgen Sie den step-by-step Tutorials, um Ihre erste Anwendung zu starten. AWS
- AWS Whitepapers — Links zu einer umfassenden Liste von technischen AWS Whitepapers zu Themen wie Architektur, Sicherheit und Wirtschaft, die von Solutions Architects oder anderen technischen Experten verfasst wurden. AWS
- AWS -Support Center — Die zentrale Anlaufstelle für die Erstellung und Verwaltung Ihrer Fälle. AWS -Support Enthält auch Links zu anderen hilfreichen Ressourcen wie Foren, technischen FAQs Informationen, Servicestatus und AWS Trusted Advisor.
- Support— Die wichtigste Webseite mit Informationen über Support einen Support-Kanal mit schnellen Reaktionszeiten one-on-one, der Sie bei der Entwicklung und Ausführung von Anwendungen in der Cloud unterstützt.
- Kontakt – Zentraler Kontaktpunkt für Fragen zu AWS -Abrechnung, Konten, Ereignissen Missbrauch und anderen Problemen.
- AWS Nutzungsbedingungen der Website — Detaillierte Informationen zu unseren Urheberrechten und Marken, zu Ihrem Konto, Ihrer Lizenz und Ihrem Zugriff auf die Website sowie zu anderen Themen.

Erste Schritte

Die folgenden Codefragmente können Ihnen bei den ersten Schritten mit dem WorkDocs SDK helfen.

> **Note**
>
> Um die Sicherheit zu erhöhen, sollten Sie nach Möglichkeit Verbundbenutzer anstelle von IAM-Benutzern erstellen.

Beispiele

- Connect WorkDocs mit IAM-Benutzeranmeldedaten her und fragen Sie nach Benutzern ab
- Stellen Sie eine Verbindung her, WorkDocs indem Sie eine Rolle übernehmen
- Hochladen eines Dokuments
- Herunterladen eines Dokuments
- Einrichten von Benachrichtigungen
- Erstellen eines Benutzers
- Benutzern Berechtigungen für eine Ressource gewähren

Connect WorkDocs mit IAM-Benutzeranmeldedaten her und fragen Sie nach Benutzern ab

Der folgende Code zeigt, wie Sie die API-Anmeldeinformationen eines IAM-Benutzers für API-Aufrufe verwenden. In diesem Fall gehören der API-Benutzer und die WorkDocs Site zu demselben AWS Konto.

> **Note**
>
> Um die Sicherheit zu erhöhen, sollten Sie nach Möglichkeit Verbundbenutzer anstelle von IAM-Benutzern erstellen.

Stellen Sie sicher, dass dem IAM-Benutzer über eine entsprechende WorkDocs IAM-Richtlinie API-Zugriff gewährt wurde.

Das Codebeispiel verwendet die DescribeUsersAPI, um nach Benutzern zu suchen und Metadaten für Benutzer abzurufen. Benutzermetadaten enthalten Details wie Vorname, Nachname, Benutzer-ID und Stammordner-ID. Die Stammordner-ID ist besonders hilfreich, wenn Sie im Namen des Benutzers Vorgänge zum Hoch- oder Herunterladen von Inhalten ausführen möchten.

Der Code erfordert, dass Sie eine WorkDocs Organisations-ID erhalten.

Gehen Sie wie folgt vor, um eine WorkDocs Organisations-ID von der AWS Konsole abzurufen:

So rufen Sie eine Organisations-ID ab

1. Wählen Sie im Navigationsbereich der AWS Directory Service -Konsole Directories aus.
2. Notieren Sie sich den Verzeichnis-ID-Wert, der Ihrer WorkDocs Site entspricht. Das ist die Organisations-ID für die Site.

Das folgende Beispiel zeigt, wie Sie IAM-Anmeldeinformationen für API-Aufrufe verwenden.

```
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.workdocs.AmazonWorkDocs;
import com.amazonaws.services.workdocs.AmazonWorkDocsClient;
import com.amazonaws.services.workdocs.model.DescribeUsersRequest;
import com.amazonaws.services.workdocs.model.DescribeUsersResult;
import com.amazonaws.services.workdocs.model.User;

public class GetUserDemo {

  public static void main(String[] args) throws Exception {
    AWSCredentials longTermCredentials =
        new BasicAWSCredentials("accessKey", "secretKey");
    AWSStaticCredentialsProvider staticCredentialProvider =
        new AWSStaticCredentialsProvider(longTermCredentials);

    AmazonWorkDocs workDocs =
        AmazonWorkDocsClient.builder().withCredentials(staticCredentialProvider)
            .withRegion(Regions.US_WEST_2).build();
```

```
    List<User> wdUsers = new ArrayList<>();
    DescribeUsersRequest request = new DescribeUsersRequest();

    // The OrganizationId used here is an example and it should be replaced
    // with the OrganizationId of your WorkDocs site.
    request.setOrganizationId("d-123456789c");
    request.setQuery("joe");

    String marker = null;
    do {
      request.setMarker(marker);
      DescribeUsersResult result = workDocs.describeUsers(request);
      wdUsers.addAll(result.getUsers());
      marker = result.getMarker();
    } while (marker != null);

    System.out.println("List of users matching the query string: joe ");

 for (User wdUser : wdUsers) {
      System.out.printf("Firstname:%s | Lastname:%s | Email:%s | root-folder-id:%s\n",
          wdUser.getGivenName(), wdUser.getSurname(), wdUser.getEmailAddress(),
          wdUser.getRootFolderId());
    }
  }
}
```

Stellen Sie eine Verbindung her, WorkDocs indem Sie eine Rolle übernehmen

In diesem Beispiel wird das AWS Java-SDK verwendet, um eine Rolle anzunehmen und die temporären Sicherheitsanmeldedaten der Rolle für den Zugriff WorkDocs zu verwenden. Das Codebeispiel verwendet die DescribeFolderContentsAPI, um die Elemente in einem Benutzerordner aufzulisten.

```
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.auth.BasicSessionCredentials;
```

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.securitytoken.AWSSecurityTokenService;
import com.amazonaws.services.securitytoken.AWSSecurityTokenServiceClientBuilder;
import com.amazonaws.services.securitytoken.model.AssumeRoleRequest;
import com.amazonaws.services.securitytoken.model.AssumeRoleResult;
import com.amazonaws.services.workdocs.AmazonWorkDocs;
import com.amazonaws.services.workdocs.AmazonWorkDocsClient;
import com.amazonaws.services.workdocs.model.DescribeFolderContentsRequest;
import com.amazonaws.services.workdocs.model.DescribeFolderContentsResult;
import com.amazonaws.services.workdocs.model.DocumentMetadata;
import com.amazonaws.services.workdocs.model.FolderMetadata;

public class AssumeRoleDemo {
  private static final String DEMO_ROLE_ARN = "arn:aws:iam::111122223333:role/workdocs-
readonly-role";
  private static AmazonWorkDocs workDocs;

  public static void main(String[] args) throws Exception {

    AWSCredentials longTermCredentials =
        new BasicAWSCredentials("accessKey", "secretKey");

    // Use developer’s long-term credentials to call the AWS Security Token Service
 (STS)
 // AssumeRole API, specifying the ARN for the role workdocs-readonly-role in
 // 3rd party AWS account.

    AWSSecurityTokenService stsClient =
        AWSSecurityTokenServiceClientBuilder.standard()
            .withCredentials(new AWSStaticCredentialsProvider(longTermCredentials))
            .withRegion(Regions.DEFAULT_REGION.getName()).build();;

    // If you are accessing a 3rd party account, set ExternalId
    // on assumeRequest using the withExternalId() function.
    AssumeRoleRequest assumeRequest =
        new AssumeRoleRequest().withRoleArn(DEMO_ROLE_ARN).withDurationSeconds(3600)
            .withRoleSessionName("demo");

    AssumeRoleResult assumeResult = stsClient.assumeRole(assumeRequest);

    // AssumeRole returns temporary security credentials for the
 // workdocs-readonly-role

    BasicSessionCredentials temporaryCredentials =
```

```
        new BasicSessionCredentials(assumeResult.getCredentials().getAccessKeyId(),
 assumeResult
            .getCredentials().getSecretAccessKey(),
 assumeResult.getCredentials().getSessionToken());

    // Build WorkDocs client using the temporary credentials.
    workDocs =
        AmazonWorkDocsClient.builder()
            .withCredentials(new AWSStaticCredentialsProvider(temporaryCredentials))
            .withRegion(Regions.US_WEST_2).build();


    // Invoke WorkDocs service calls using the temporary security credentials
    // obtained for workdocs-readonly-role. In this case a call has been made
 // to get metadata of Folders and Documents present in a user’s root folder.

    describeFolder("root-folder-id");
  }

  private static void describeFolder(String folderId) {
    DescribeFolderContentsRequest request = new DescribeFolderContentsRequest();
    request.setFolderId(folderId);
    request.setLimit(2);
    List<DocumentMetadata> documents = new ArrayList<>();
    List<FolderMetadata> folders = new ArrayList<>();

    String marker = null;

    do {
      request.setMarker(marker);
      DescribeFolderContentsResult result = workDocs.describeFolderContents(request);
      documents.addAll(result.getDocuments());
      folders.addAll(result.getFolders());
      marker = result.getMarker();
    } while (marker != null);

    for (FolderMetadata folder : folders)
      System.out.println("Folder:" + folder.getName());
    for (DocumentMetadata document : documents)
      System.out.println("Document:" + document.getLatestVersionMetadata().getName());
  }
}
```

Hochladen eines Dokuments

> **Note**
>
> Sie müssen Softwareentwickler sein, um die Schritte in diesem Abschnitt ausführen zu können. Informationen zur Verwendung WorkDocs zum Hochladen von Dateien finden Sie im WorkDocs Benutzerhandbuch unter Hochladen von Dateien.

Führen Sie die folgenden Schritte aus, um ein Dokument zu WorkDocs hochzuladen.

So laden Sie ein Dokument hoch

1. Erstellen Sie folgendermaßen eine Instance von `AmazonWorkDocsClient`:

   Wenn Sie IAM-Benutzeranmeldeinformationen verwenden, finden Sie weitere Informationen unter. Connect WorkDocs mit IAM-Benutzeranmeldedaten her und fragen Sie nach Benutzern ab Wenn Sie eine IAM-Rolle annehmen, finden Sie Stellen Sie eine Verbindung her, WorkDocs indem Sie eine Rolle übernehmen weitere Informationen unter.

   > **Note**
   >
   > Um die Sicherheit zu erhöhen, sollten Sie nach Möglichkeit Verbundbenutzer anstelle von IAM-Benutzern erstellen.

   ```
   AWSCredentials longTermCredentials =
     new BasicAWSCredentials("accessKey", "secretKey");
   AWSStaticCredentialsProvider staticCredentialProvider =
     new AWSStaticCredentialsProvider(longTermCredentials);

   // Use the region specific to your WorkDocs site.
   AmazonWorkDocs amazonWorkDocsClient =
     AmazonWorkDocsClient.builder().withCredentials(staticCredentialProvider)
       .withRegion(Regions.US_WEST_2).build();
   ```

2. Fordern Sie folgendermaßen die signierte URL für den Upload an:

   ```
   InitiateDocumentVersionUploadRequest request = new
    InitiateDocumentVersionUploadRequest();
   ```

```
request.setParentFolderId("parent-folder-id");
request.setName("my-document-name");
request.setContentType("application/octet-stream");
InitiateDocumentVersionUploadResult result =
 amazonWorkDocsClient.initiateDocumentVersionUpload(request);
UploadMetadata uploadMetadata = result.getUploadMetadata();
String documentId = result.getMetadata().getId();
String documentVersionId = result.getMetadata().getLatestVersionMetadata().getId();
String uploadUrl = uploadMetadata.getUploadUrl();
```

3. Laden Sie das Dokument folgendermaßen mit der signierten URL hoch:

```
URL url = new URL(uploadUrl);
HttpURLConnection connection = (HttpURLConnection) url.openConnection();
connection.setDoOutput(true);
connection.setRequestMethod("PUT");
// Content-Type supplied here should match with the Content-Type set
// in the InitiateDocumentVersionUpload request.
connection.setRequestProperty("Content-Type","application/octet-stream");
connection.setRequestProperty("x-amz-server-side-encryption", "AES256");
File file = new File("/path/to/file.txt");
FileInputStream fileInputStream = new FileInputStream(file);
OutputStream outputStream = connection.getOutputStream();
com.amazonaws.util.IOUtils.copy(fileInputStream, outputStream);
connection.getResponseCode();
```

4. Schließen Sie den Upload-Vorgang ab, indem Sie den Dokumentstatus folgendermaßen in `ACTIVE` ändern:

```
UpdateDocumentVersionRequest request = new UpdateDocumentVersionRequest();
request.setDocumentId("document-id");
request.setVersionId("document-version-id");
request.setVersionStatus(DocumentVersionStatus.ACTIVE);
amazonWorkDocsClient.updateDocumentVersion(request);
```

Herunterladen eines Dokuments

> **Note**
>
> Sie müssen Softwareentwickler sein, um die Schritte in diesem Abschnitt ausführen zu können. Informationen zur Verwendung WorkDocs zum Herunterladen von Dateien finden Sie im WorkDocs Benutzerhandbuch unter Herunterladen von Dateien.

Um ein Dokument von herunterzuladen WorkDocs, rufen Sie wie folgt eine URL für den Download ab und verwenden Sie dann die API-Aktionen, die von Ihrer Entwicklungsplattform bereitgestellt werden, um die Datei mithilfe der URL herunterzuladen.

```
GetDocumentVersionRequest request = new GetDocumentVersionRequest();
request.setDocumentId("document-id");
request.setVersionId("document-version-id");
request.setFields("SOURCE");
GetDocumentVersionResult result = amazonWorkDocsClient.getDocumentVersion(request);
String downloadUrl =
 result.getMetadata().getSource().get(DocumentSourceType.ORIGINAL.name());
```

Einrichten von Benachrichtigungen

Gehen Sie wie folgt vor, um Benachrichtigungen einzurichten:

1. Richten Sie IAM-Benutzer- oder Rollenberechtigungen ein, um Anrufern Zugriff auf die Abonnementverwaltung für Benachrichtigungen zu gewähren. APIs
2. Rufen Sie das Benachrichtigungsabonnement APIs auf, um die Veröffentlichung von SNS-Nachrichten auf Ihrem Endpunkt zu aktivieren oder zu deaktivieren.

> **Note**
>
> Um die Sicherheit zu erhöhen, sollten Sie nach Möglichkeit Verbundbenutzer anstelle von IAM-Benutzern erstellen.

So legen Sie IAM-Benutzerberechtigungen fest

- Verwenden Sie die IAM-Konsole, um die folgenden Berechtigungen für den Benutzer festzulegen:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Action": [
            "workdocs:CreateNotificationSubscription",
            "workdocs:DeleteNotificationSubscription",
            "workdocs:DescribeNotificationSubscriptions"
            ],
        "Resource": "*"
        }
    ]
}
```

So aktivieren Sie Benachrichtigungen

Wenn Sie Benachrichtigungen aktivieren, können Sie anrufen, CreateNotificationSubscriptionnachdem Sie Benachrichtigungen abonniert haben.

1. Öffnen Sie die WorkDocs Konsole unter https://console.aws.amazon.com/zocalo/.
2. Wählen Sie auf der Seite „Ihre WorkDocs Websites verwalten“ das gewünschte Verzeichnis aus und klicken Sie auf Aktionen und dann auf Benachrichtigungen verwalten.
3. Wählen Sie auf der Seite Benachrichtigungen verwalten die Option Enable Notifications (Benachrichtigungen aktivieren).
4. Geben Sie den ARN für den Benutzer oder die Rolle ein, dem Sie den Empfang von Benachrichtigungen von Ihrer WorkDocs Site erlauben möchten.

Informationen WorkDocs zur Aktivierung der Verwendung von Benachrichtigungen finden Sie unter Verwenden der WorkDocs Amazon-API mit dem AWS-SDK für Python und AWS Lambda. Sobald Sie Benachrichtigungen aktiviert haben, können Sie und Ihr Benutzer sie abonnieren.

Um WorkDocs Benachrichtigungen zu abonnieren

1. Bereiten Sie Ihren Endpunkt für die Verarbeitung von Amazon SNS SNS-Nachrichten vor. Weitere Informationen finden Sie unter Fanout to HTTP/S Endpoints im Amazon Simple Notification Service Developer Guide.

   > **Important**
   >
   > SNS sendet eine Bestätigungsnachricht an Ihren konfigurierten Endpunkt. Sie müssen diese Nachricht bestätigen, um Benachrichtigungen zu erhalten. Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie außerdem einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

2. Gehen Sie wie folgt vor:

   - Holen Sie sich eine Organisations-ID
     1. Wählen Sie im Navigationsbereich der AWS Directory Service Service-Konsole die Option Verzeichnisse aus.
     2. Die Verzeichnis-ID, die Ihrer WorkDocs Amazon-Website entspricht, dient auch als Organisations-ID für diese Site.
   - Erstellen Sie folgendermaßen die Abonnementanforderung:

```
CreateNotificationSubscriptionRequest request = new
 CreateNotificationSubscriptionRequest();
request.setOrganizationId("d-1234567890");
request.setProtocol(SubscriptionProtocolType.Https);
request.setEndpoint("https://my-webhook-service.com/webhook");
request.setSubscriptionType(SubscriptionType.ALL);
CreateNotificationSubscriptionResult result =
 amazonWorkDocsClient.createNotificationSubscription(request);
System.out.println("WorkDocs notifications subscription-id: "
 result.getSubscription().getSubscriptionId());
```

SNS-Benachrichtigungen

Die Nachricht enthält die folgenden Informationen:

- `organizationId`— Die ID der Organisation.
- `parentEntityType`— Der Typ des übergeordneten Elements (`Document`| `DocumentVersion` |`Folder`).
- `parentEntityId`— Die ID des Elternteils.
- `entityType`— Der Typ der Entität (`Document`| `DocumentVersion` |`Folder`).
- `entityId`— Die ID der Entität.
- action — Die Aktion, die einer der folgenden Werte sein kann:
  - `delete_document`
  - `move_document`
  - `recycle_document`
  - `rename_document`
  - `revoke_share_document`
  - `share_document`
  - `upload_document_version`

So deaktivieren Sie Benachrichtigungen

1. Öffnen Sie die WorkDocs Konsole unter https://console.aws.amazon.com/zocalo/.
2. Wählen Sie auf der Seite „Ihre WorkDocs Websites verwalten“ das gewünschte Verzeichnis aus und klicken Sie auf Aktionen und dann auf Benachrichtigungen verwalten.
3. Wählen Sie auf der Seite Benachrichtigungen verwalten den ARN aus, für den Sie Benachrichtigungen deaktivieren möchten, und wählen Sie Benachrichtigungen deaktivieren.

Erstellen eines Benutzers

Das folgende Beispiel zeigt, wie Sie einen Benutzer in WorkDocs erstellen.

> **Note**
>
> Dies ist für eine Connected AD-Konfiguration keine gültige Operation. Um einen Benutzer in der Connected AD-Konfiguration zu erstellen, muss der Benutzer bereits im Unternehmensverzeichnis vorhanden sein. Anschließend müssen Sie die ActivateUserAPI aufrufen, um den Benutzer in zu aktivieren WorkDocs.

Das folgende Beispiel zeigt, wie Sie einen Benutzer mit einem Speicherkontingent von 1 Gigabyte erstellen.

```
CreateUserRequest request = new CreateUserRequest();
    request.setGivenName("GivenName");
    request.setOrganizationId("d-12345678c4");
    // Passwords should:
    //   Be between 8 and 64 characters
    //   Contain three of the four below:
    //   A Lowercase Character
    //   An Uppercase Character
    //   A Number
    //   A Special Character
    request.setPassword("Badpa$$w0rd");
    request.setSurname("surname");
    request.setUsername("UserName");
    StorageRuleType storageRule = new StorageRuleType();
    storageRule.setStorageType(StorageType.QUOTA);
    storageRule.setStorageAllocatedInBytes(new Long(1048576l));
    request.setStorageRule(storageRule);
    CreateUserResult result = workDocsClient.createUser(request);
```

Gehen Sie wie folgt vor, um eine WorkDocs Organisations-ID von der AWS Konsole abzurufen:

So rufen Sie eine Organisations-ID ab

1. Wählen Sie im Navigationsbereich der AWS Directory Service -Konsole Directories aus.
2. Notieren Sie sich den Verzeichnis-ID-Wert, der Ihrer WorkDocs Site entspricht. Das ist die Organisations-ID für die Site.

Benutzern Berechtigungen für eine Ressource gewähren

Das folgende Beispiel zeigt, wie die AddResourcePermissionsAPI verwendet wird, um einer `USER` Ressource `CONTRIBUTOR` Berechtigungen zu erteilen. Sie können die API auch verwenden, um einem Benutzer oder einer Gruppe Berechtigungen für einen Ordner oder ein Dokument zu erteilen.

```
AddResourcePermissionsRequest request = new AddResourcePermissionsRequest();
    request.setResourceId("resource-id");
    Collection<SharePrincipal> principals = new ArrayList<>();;
    SharePrincipal principal = new SharePrincipal();
```

```
    principal.setId("user-id");
    principal.setType(PrincipalType.USER);
    principal.setRole(RoleType.CONTRIBUTOR);
    principals.add(principal);
    request.setPrincipals(principals);
    AddResourcePermissionsResult result =
 workDocsClient.addResourcePermissions(request);
```

Authentifizierung und Zugriffskontrolle für administrative Anwendungen

WorkDocs Administratoren APIs werden über IAM-Richtlinien authentifiziert und autorisiert. IAM-Administratoren können eine IAM-Richtlinie erstellen und sie einer IAM-Rolle oder einem IAM-Benutzer zuordnen, die vom Entwickler für den Zugriff auf die API verwendet werden können.

Nachstehend folgen einige Beispiele:

Aufgaben

- Entwicklern Berechtigungen für die API gewähren WorkDocs
- Erteilt Drittentwicklern die Erlaubnis für WorkDocs APIs
- Benutzern die Erlaubnis erteilen, eine IAM-Rolle anzunehmen
- Beschränken des Zugriffs auf eine bestimmte Instanz WorkDocs

Entwicklern Berechtigungen für die API gewähren WorkDocs

> **Note**
>
> Um die Sicherheit zu erhöhen, sollten Sie nach Möglichkeit Verbundbenutzer anstelle von IAM-Benutzern erstellen.

Wenn Sie ein IAM-Administrator sind, können Sie einem IAM-Benutzer von demselben Konto aus WorkDocs API-Zugriff gewähren. AWS Erstellen Sie dazu eine WorkDocs API-Berechtigungsrichtlinie und fügen Sie sie dem IAM-Benutzer hinzu. Die folgende API-Richtlinie gewährt den verschiedenen Benutzern nur Leseberechtigungen. `Describe` APIs

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "WorkDocsAPIReadOnly",
            "Effect": "Allow",
            "Action": [
                "workdocs:Get*",
```

```
      "workdocs:Describe*"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Erteilt Drittentwicklern die Erlaubnis für WorkDocs APIs

Sie können Drittentwicklern oder Benutzern, die ein anderes AWS Konto verwenden, Zugriff gewähren. Erstellen Sie dazu eine IAM-Rolle und fügen Sie WorkDocs API-Zulassungsrichtlinien hinzu.

Diese Art von Zugriff ist in den folgenden Szenarien erforderlich:

- Der Entwickler gehört derselben Organisation an, aber das AWS Konto des Entwicklers unterscheidet sich vom WorkDocs AWS Konto.
- Wenn ein Unternehmen Anwendungsentwicklern von Drittanbietern WorkDocs API-Zugriff gewähren möchte.

In beiden Szenarien sind zwei AWS Konten involviert, ein AWS Entwicklerkonto und ein anderes Konto, das eine WorkDocs Website hostet.

Der Entwickler muss die folgenden Informationen bereitstellen, damit der Kontoadministrator die IAM-Rolle erstellen kann:

- Ihre AWS Konto-ID
- Eine eindeutige `External ID`, anhand der Ihr Kunde Sie identifiziert. Weitere Informationen finden Sie unter So verwenden Sie eine externe ID, wenn Sie Dritten Zugriff auf Ihre AWS Ressourcen gewähren.
- Eine Liste WorkDocs APIs Ihrer Anwendung, auf die Sie Zugriff benötigen. Die IAM-basierte Richtliniensteuerung bietet eine detaillierte Steuerung, d. h. die Möglichkeit, Richtlinien zum Zulassen oder Verweigern auf individueller API-Ebene zu definieren. Eine Liste von finden Sie WorkDocs APIs unter WorkDocs API-Referenz.

Die folgenden Schritte beschreiben, wie IAM für kontoübergreifenden Zugriff konfiguriert wird.

So konfigurieren Sie IAM für den kontoübergreifenden Zugriff

1. Erstellen Sie eine WorkDocs API-Berechtigungsrichtlinie, nennen Sie sie `WorkDocsAPIReadOnly` Richtlinie.
2. Erstellen Sie eine neue Rolle in der IAM-Konsole des AWS Kontos, das die WorkDocs Site hostet:

   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM-Konsole unter. https://console.aws.amazon.com/iam/

   b. Klicken Sie im Navigationsbereich der Konsole auf Roles (Rollen) und danach auf Create new role (Neue Rolle erstellen).

   c. Geben Sie für Role Name (Rollennamen) einen Rollennamen ein, damit Sie den Zweck dieser Rolle leichter erkennen können, z. B. `workdocs_app_role`. Rollennamen müssen in Ihrem AWS Konto eindeutig sein. Klicken Sie nach der Eingabe des Namens auf Next Step (Nächster Schritt).

   d. Wählen Sie auf der Seite Select Role Type (Rollentyp auswählen) den Bereich Role for Cross-Account Access (Rolle für kontoübergreifenden Zugriff). Wählen Sie dann die Art von Rolle aus, die Sie erstellen möchten:

      - Wählen Sie Zugriff zwischen AWS Konten bereitstellen, die Sie besitzen, wenn Sie der Administrator sowohl des Benutzerkontos als auch des Ressourcenkontos sind oder beide Konten demselben Unternehmen gehören. Dies ist auch die auszuwählende Option, wenn sich die Benutzer, die Rolle und die Ressource, auf die zugegriffen werden sollen, alle in demselben Konto befinden.
      - Wählen Sie Zugriff zwischen Ihrem AWS Konto und einem AWS Drittanbieterkonto bereitstellen aus, wenn Sie der Administrator des Kontos sind, dem die WorkDocs Website gehört, und Sie Benutzern über ein Anwendungsentwicklerkonto Berechtigungen gewähren möchten. Bei dieser Option müssen Sie eine externe ID angeben (die Ihnen vom Dritten bereitgestellt werden muss), um zusätzliche Kontrolle über die Umstände zu gewähren, unter denen der Dritte die Rolle verwenden kann, um auf Ihre Ressourcen zuzugreifen. Weitere Informationen finden Sie unter Verwenden einer externen ID, um Dritten Zugriff auf Ihre AWS-Ressourcen zu gewähren.

   e. Geben Sie auf der nächsten Seite die AWS Konto-ID an, der Sie Zugriff auf Ihre Ressourcen gewähren möchten, und geben Sie bei Zugriff durch Dritte auch die externe ID ein.

   f. Klicken Sie auf Next Step (Nächster Schritt), um eine Richtlinie anzufügen.

3. Suchen Sie auf der Seite „Richtlinie anhängen“ nach der WorkDocs API-Berechtigungsrichtlinie, die zuvor erstellt wurde, aktivieren Sie das Kästchen neben der Richtlinie und klicken Sie auf Nächster Schritt.
4. Überprüfen Sie die Details, kopieren Sie den ARN der Rolle zur späteren Bezugnahme und klicken Sie auf Create Role (Rolle erstellen), um das Erstellen der Rolle abzuschließen.
5. Geben Sie den ARN der Rolle für den Developer frei. Es folgt ein Beispiel für den ARN einer Rolle:

```
arn:aws:iam::AWS-ACCOUNT-ID:role/workdocs_app_role
```

Benutzern die Erlaubnis erteilen, eine IAM-Rolle anzunehmen

Ein Entwickler mit einem AWS Administratorkonto kann einem Benutzer die Übernahme einer IAM-Rolle ermöglichen. Dazu erstellen Sie eine neue Richtlinie und fügen sie diesem Benutzer hinzu.

Die Richtlinie muss eine Erklärung mit den `Allow` Auswirkungen auf die `sts:AssumeRole` Aktion sowie den Amazon-Ressourcennamen (ARN) der Rolle in einem `Resource` Element enthalten, wie im folgenden Beispiel gezeigt. Benutzer, die die Richtlinie entweder durch Gruppenmitgliedschaft oder direkte Verbindung erhalten, können zur angegebenen Rolle wechseln.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::<aws_account_id>:role/workdocs_app_role"
  }
}
```

Beschränken des Zugriffs auf eine bestimmte Instanz WorkDocs

Wenn Sie mehrere WorkDocs Websites in einem AWS Konto haben und API-Zugriff auf eine bestimmte Site gewähren möchten, können Sie ein `Condition` Element definieren. Mit dem Element `Condition` können Sie angeben, unter welchen Bedingungen eine Richtlinie wirksam ist.

Das folgende Beispiel zeigt ein Bedingungselement:

```
    "Condition":
```

```
    {
                "StringEquals": {
                    "Resource.OrganizationId": "d-123456789c5"
                }
    }
```

Wenn die obige Bedingung in einer Richtlinie enthalten ist, können Benutzer nur mit der ID von auf die WorkDocs Instanz zugreifend `-123456789c5`. WorkDocs Die Instanz-ID wird manchmal als Organisations- oder Verzeichnis-ID bezeichnet. Weitere Informationen finden Sie unter Beschränken des Zugriffs auf eine bestimmte Instanz WorkDocs .

Gehen Sie wie folgt vor, um eine WorkDocs Organisations-ID von der AWS Konsole abzurufen:

So rufen Sie eine Organisations-ID ab

1. Wählen Sie im Navigationsbereich der AWS Directory Service -Konsole Directories aus.
2. Notieren Sie sich den Verzeichnis-ID-Wert, der Ihrer WorkDocs Site entspricht. Das ist die Organisations-ID für die Site.

Authentifizierung und Zugriffskontrolle für Benutzeranwendungen

WorkDocs Anwendungen auf Benutzerebene werden über die WorkDocs Konsole registriert und verwaltet. Entwickler sollten ihre Anwendungen auf der `My Applications` Seite in der WorkDocs Konsole registrieren, die IDs für jede Anwendung eindeutig ist. Bei der Registrierung sollten Entwickler die Weiterleitung angeben, URIs an die sie Zugriffstoken sowie die Anwendungsbereiche erhalten sollen.

Derzeit können Anwendungen nur auf WorkDocs Websites innerhalb desselben AWS Kontos zugreifen, in dem sie registriert sind.

Inhalt

- Erteilung von Berechtigungen zum Aufrufen von WorkDocs APIs
- Ordner IDs in API-Aufrufen verwenden
- Erstellen einer Anwendung
- Anwendungsbereiche
- Autorisierung
- Aufrufen WorkDocs APIs

Erteilung von Berechtigungen zum Aufrufen von WorkDocs APIs

Benutzer der Befehlszeilenschnittstelle müssen über vollständige Berechtigungen für WorkDocs und verfügen AWS Directory Service. Ohne die Berechtigungen geben alle API-Aufrufe UnauthorizedResourceAccessExceptionNachrichten zurück. Die folgende Richtlinie gewährt volle Berechtigungen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
          "workdocs:*",
          "ds:*",
          "ec2:CreateVpc",
          "ec2:CreateSubnet",
```

```
            "ec2:CreateNetworkInterface",
            "ec2:CreateTags",
            "ec2:CreateSecurityGroup",
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeAvailabilityZones",
            "ec2:AuthorizeSecurityGroupEgress",
            "ec2:AuthorizeSecurityGroupIngress",
            "ec2:DeleteSecurityGroup",
            "ec2:DeleteNetworkInterface",
            "ec2:RevokeSecurityGroupEgress",
            "ec2:RevokeSecurityGroupIngress"
            ],
        "Effect": "Allow",
        "Resource": "*"
      }
    ]
}
```

Verwenden Sie diese Richtlinie, wenn Sie nur Leseberechtigungen gewähren möchten.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
          "workdocs:Describe*",
          "ds:DescribeDirectories",
          "ec2:DescribeVpcs",
          "ec2:DescribeSubnets"
          ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

In der Richtlinie gewährt die erste Aktion Zugriff auf alle Operationen. WorkDocs `Describe` Die `DescribeDirectories` Aktion ruft Informationen über Ihre AWS Directory Service Verzeichnisse ab. Die EC2 Amazon-Operationen WorkDocs ermöglichen es, eine Liste Ihrer VPCs und Subnetze zu erhalten.

Ordner IDs in API-Aufrufen verwenden

Immer wenn ein API-Aufruf auf einen Ordner zugreift, müssen Sie die Ordner-ID verwenden, nicht den Ordnernamen. Wenn Sie den Vorgang beispielsweise bestehen`client.get_folder(FolderId='MyDocs')`, gibt der API-Aufruf eine UnauthorizedResourceAccessExceptionNachricht und die folgende 404-Nachricht zurück.

```
client.get_folder(FolderId='MyDocs')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\Users\user-name\AppData\Local\Programs\Python\Python36-32\lib\site-packages
\botocore\client.py", line 253, in _api_call
    return self._make_api_call(operation_name, kwargs)
  File "C:\Users\user-name\AppData\Local\Programs\Python\Python36-32\lib\site-packages
\botocore\client.py", line 557, in _make_api_call
    raise error_class(parsed_response, operation_name)
botocore.errorfactory.UnauthorizedResourceAccessException: An error occurred
 (UnauthorizedResourceAccessException) when calling the GetFolder operation:
Principal [arn:aws:iam::395162986870:user/Aman] is not allowed to execute
 [workdocs:GetFolder] on the resource.
```

Um dies zu vermeiden, verwenden Sie die ID in der URL des Ordners.

`site.workdocs/index.html#/folder/abc123def456ghi789jkl789mno4be7024df198736472dd50ca970eb22796082e3d489577`.

Wenn Sie diese ID übergeben, wird ein korrektes Ergebnis zurückgegeben.

```
client.get_folder(FolderId='abc123def456ghi789jkl789mno4be7024df198736472dd50ca970eb22796082e3d
{'ResponseMetadata': {'RequestId': 'f8341d4e-4047-11e7-9e70-afa8d465756c',
 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'f234564e-1234-56e7-89e7-
a10fa45t789c', 'cache-control': 'private, no-cache, no-store, max-age=0',
 'content-type': 'application/json', 'content-length': '733', 'date':
 'Wed, 24 May 2017 06:12:30 GMT'}, 'RetryAttempts': 0}, 'Metadata': {'Id':
 'abc123def456ghi789jkl789mno4be7024df198736472dd50ca970eb22796082e3d489577', 'Name':
 'sentences', 'CreatorId':
 'S-1-5-21-2125721135-1643952666-3011040551-2105&d-906724f1ce', 'ParentFolderId':
 '0a811a922403ae8e1d3c180f4975f38f94372c3d6a2656c50851c7fb76677363',
 'CreatedTimestamp': datetime.datetime(2017, 5, 23, 12, 59, 13, 8000,
 tzinfo=tzlocal()), 'ModifiedTimestamp': datetime.datetime(2017, 5, 23, 13,
 13, 9, 565000, tzinfo=tzlocal()), 'ResourceState': 'ACTIVE', 'Signature':
 'b7f54963d60ae1d6b9ded476f5d20511'}}
```

Erstellen einer Anwendung

Als WorkDocs Administrator erstellen Sie Ihre Anwendung mithilfe der folgenden Schritte.

So erstellen Sie eine Anwendung

1. Öffnen Sie die WorkDocs Konsole unter https://console.aws.amazon.com/zocalo/.
2. Wählen Sie Meine Anwendungen, Eine Anwendung erstellen.
3. Geben Sie die folgenden Werte ein:

   Anwendungsname

   Name für die Anwendung.

   E-Mail

   Die der Anwendung zuzuordnende E-Mail-Adresse.

   Anwendungsbeschreibung

   Beschreibung für die Anwendung.

   Umleiten URIs

   Der Standort, an den Sie WorkDocs den Verkehr umleiten möchten.

   Anwendungsbereiche

   Der für Ihre Anwendung gewünschte Bereich, entweder Lese- oder Schreibzugriff. Weitere Details finden Sie unter Anwendungsbereiche.
4. Wählen Sie Erstellen aus.

Anwendungsbereiche

WorkDocs unterstützt die folgenden Anwendungsbereiche:

- Content Read (`workdocs.content.read`), wodurch Ihre Anwendung auf Folgendes WorkDocs APIs zugreifen kann:
  - Get*

  - Describe*
- Content Write (`workdocs.content.write`), wodurch Ihre Anwendung auf Folgendes zugreifen kann WorkDocs APIs:
  - Geben Sie einen Namen für den Benutzer ein und klicken Sie dann auf*
  - Update*
  - Delete*
  - Initiate*
  - Abort*
  - Add*
  - Remove*

Autorisierung

Nachdem die Registrierung der Anwendung abgeschlossen ist, kann eine Anwendung im Namen eines beliebigen WorkDocs Benutzers eine Autorisierung beantragen. Zu diesem Zweck sollte die Anwendung den WorkDocs OAuth Endpunkt aufrufen `https://auth.amazonworkdocs.com/oauth` und die folgenden Abfrageparameter bereitstellen:

- [Erforderlich] `app_id` — Anwendungs-ID, die generiert wird, wenn eine Anwendung registriert wird.
- [Erforderlich] `auth_type` — Der OAuth Typ für die Anforderung. Der unterstützte Wert ist `ImplicitGrant`.
- [Erforderlich] `redirect_uri` — Die Umleitungs-URI, die für eine Anwendung registriert wurde, um ein Zugriffstoken zu erhalten.
- [Optional] `scopes` — Eine durch Kommas getrennte Liste von Bereichen. Wenn keine Angabe gemacht wird, wird die während der Registrierung ausgewählte Liste von Bereichen verwendet.
- [Optional] `state` — Eine Zeichenfolge, die zusammen mit einem Zugriffstoken zurückgegeben wird.

> **Note**
>
> Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt.

Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Ein Beispiel für eine GET-Anfrage zur Initiierung des OAuth Flows zum Abrufen eines Zugriffstokens:

```
GET https://auth.amazonworkdocs.com/oauth?app_id=my-app-
id&auth_type=ImplicitGrant&redirect_uri=https://myapp.com/
callback&scopes=workdocs.content.read&state=xyz
```

Während des OAuth Autorisierungsablaufs findet Folgendes statt:

1. Der Anwendungsbenutzer wird aufgefordert, den Namen der WorkDocs Site einzugeben.
2. Der Benutzer wird zur WorkDocs Authentifizierungsseite weitergeleitet, um seine Anmeldeinformationen einzugeben.
3. Nach einer erfolgreichen Authentifizierung wird dem Benutzer der Zustimmungsbildschirm angezeigt. Hier kann der Benutzer der Anwendung die Autorisierung zum Zugriff auf WorkDocs entweder gewähren oder verweigern.
4. Nachdem der Benutzer auf dem Zustimmungsbildschirm auf `Accept` klickt, wird sein Browser zusammen mit dem Zugriffstoken und den Regionsangaben als Abfrageparameter zur Rückruf-URL der Anwendung umgeleitet.

Ein Beispiel für eine GET-Anfrage von WorkDocs:

```
GET https://myapp.com/callback?acessToken=accesstoken&region=us-east-1&state=xyz
```

Zusätzlich zum Zugriffstoken gibt der WorkDocs OAuth Dienst auch `region` als Abfrageparameter für die ausgewählte WorkDocs Site zurück. Externe Anwendungen sollten den `region` Parameter verwenden, um den WorkDocs Dienstendpunkt zu bestimmen.

Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-2.

Aufrufen WorkDocs APIs

Nachdem das Zugriffstoken empfangen wurde, kann Ihre Anwendung API-Aufrufe an WorkDocs-Services vornehmen.

> **Important**

Dieses Beispiel zeigt, wie eine curl-GET-Anfrage verwendet wird, um die Metadaten eines Dokuments abzurufen.

```
Curl "https://workdocs.us-east-1.amazonaws.com/api/v1/documents/{document-id}" -H
 "Accept: application/json" -H "Authentication: Bearer accesstoken"
```

Eine JavaScript Beispielfunktion zur Beschreibung der Stammordner eines Benutzers:

```
function printRootFolders(accessToken, siteRegion) {
    var workdocs = new AWS.WorkDocs({region: siteRegion});
    workdocs.makeUnauthenticatedRequest("describeRootFolders", {AuthenticationToken:
 accessToken}, function (err, folders) {
        if (err) console.log(err);
        else console.log(folders);
    });
}
```

Ein Java-basierter API-Beispielaufruf wird nachstehend beschrieben:

```
AWSCredentialsProvider credentialsProvider = new AWSCredentialsProvider() {
  @Override
  public void refresh() {}

  @Override
  public AWSCredentials getCredentials() {
    new AnonymousAWSCredentials();
  }
};

// Set the correct region obtained during OAuth flow.
workDocs =
    AmazonWorkDocsClient.builder().withCredentials(credentialsProvider)
```

```
        .withRegion(Regions.US_EAST_1).build();

DescribeRootFoldersRequest request = new DescribeRootFoldersRequest();
request.setAuthenticationToken("access-token-obtained-through-workdocs-oauth");
DescribeRootFoldersResult result = workDocs.describeRootFolders(request);

for (FolderMetadata folder : result.getFolders()) {
  System.out.printf("Folder name=%s, Id=%s \n", folder.getName(), folder.getId());
}
```

WorkDocs Inhaltsmanager

WorkDocs Content Manager ist ein nützliches Tool auf hoher Ebene, das Inhalte hochlädt oder von einer WorkDocs Site herunterlädt.

Themen

- Aufbau eines Content Managers WorkDocs
- Herunterladen eines Dokuments
- Hochladen eines Dokuments

Aufbau eines Content Managers WorkDocs

Sie können WorkDocs Content Manager für Verwaltungs- und Benutzeranwendungen verwenden.

Für Benutzeranwendungen muss ein Entwickler den WorkDocs Content Manager mit anonymen AWS Anmeldeinformationen und einem Authentifizierungstoken erstellen.

Für Verwaltungsanwendungen muss der WorkDocs Client mit AWS Identity and Access Management (IAM-) Anmeldeinformationen initialisiert werden. Darüber hinaus muss das Authentifizierungstoken in nachfolgenden API-Aufrufen weggelassen.

Der folgende Code zeigt, wie WorkDocs Content Manager für Benutzeranwendungen mit Java oder C# initialisiert wird.

Java:

```
AWSStaticCredentialsProvider credentialsProvider = new AWSStaticCredentialsProvider(new
 AnonymousAWSCredentials());

AmazonWorkDocs client =
 AmazonWorkDocsClient.builder().withCredentials(credentialsProvider).withRegion("region").build

ContentManager contentManager =
 ContentManagerBuilder.standard().withWorkDocsClient(client).withAuthenticationToken("token").b
```

C#:

```
AmazonWorkDocsClient client = new AmazonWorkDocsClient(new AnonymousAWSCredentials(),
 "region");
```

```
ContentManagerParams params = new ContentManagerParams
{
WorkDocsClient = client,
AuthenticationToken = "token"
};
IContentManager workDocsContentManager = new ContentManager(param);
```

Herunterladen eines Dokuments

Entwickler können WorkDocs Content Manager verwenden, um eine bestimmte Version oder die neueste Version eines Dokuments von WorkDocs herunterzuladen. Die folgenden Beispiele veranschaulichen das Herunterladen einer bestimmten Version eines Dokuments unter Verwendung von Java und C#.

> **Note**
>
> Um die neueste Version eines Dokuments herunterzuladen, geben Sie beim Erstellen der `VersionId`-Anfrage nicht die `GetDocumentStream` ein.

Java

```
ContentManager contentManager =
 ContentManagerBuilder.standard().withWorkDocsClient(client).withAuthenticationToken("auth-
token").build();

// Download document.
GetDocumentStreamRequest request = new GetDocumentStreamRequest();
request.setDocumentId("document-id");
request.setVersionId("version-id");

// stream contains the content of the document version.
InputStream stream = contentManager.getDocumentStream(request).getStream();
```

C#

```
ContentManager contentManager =
 ContentManagerBuilder.standard().withWorkDocsClient(client).withAuthenticationToken("auth-
token").build();
```

```
// Download document.
GetDocumentStreamRequest request = new GetDocumentStreamRequest();
request.setDocumentId("document-id");
request.setVersionId("version-id");

// stream contains the content of the document version.
InputStream stream = contentManager.getDocumentStream(request).getStream();
```

Hochladen eines Dokuments

WorkDocs Content Manager bietet eine API zum Hochladen von Inhalten auf eine Site. WorkDocs Die folgenden Beispiele veranschaulichen das Hochladen eines Dokuments mit Java und C#.

Java

```
File file = new File("file-path");
InputStream stream = new FileInputStream(file);
UploadDocumentStreamRequest request = new UploadDocumentStreamRequest();
request.setParentFolderId("destination-folder-id");
request.setContentType("content-type");
request.setStream(stream);
request.setDocumentName("document-name");
contentManager.uploadDocumentStream(request);
```

C#

```
var stream = new FileStream("file-path", FileMode.Open);

UploadDocumentStreamRequest uploadDocumentStreamRequest = new
 UploadDocumentStreamRequest()
{
ParentFolderId = "destination-id",
DocumentName = "document-name",
ContentType = "content-type",
Stream = stream
};

workDocsContentManager.UploadDocumentStreamAsync(uploadDocumentStreamRequest).Wait();
```