



Entwicklerhandbuch

# Amazon WorkDocs



# Amazon WorkDocs: Entwicklerhandbuch

Copyright © 2024 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
Zugriff auf Amazon WorkDocs .....	1
Preise .....	1
Ressourcen .....	2
Erste Schritte .....	3
Herstellen einer Verbindung zu Amazon WorkDocs mit IAM-Benutzeranmeldeinformationen .....	3
Herstellen einer Verbindung zu Amazon WorkDocs durch Übernahme einer Rolle .....	5
Hochladen eines Dokuments .....	8
Herunterladen eines Dokuments .....	10
Einrichten von Benachrichtigungen .....	10
Erstellen eines Benutzers .....	13
Erteilen von Berechtigungen für Benutzer an eine Ressource .....	14
Authentifizierung und Zugriffskontrolle für administrative Anwendungen .....	16
Erteilung von Entwicklerberechtigungen für die WorkDocs Amazon-API .....	16
Erteilung der Genehmigung von Drittanbietern für die WorkDocs Amazon-APIs .....	17
Benutzern die Erlaubnis erteilen, eine IAM-Rolle zu übernehmen .....	19
Beschränken des Zugriff auf eine bestimmte WorkDocs Amazon-Instance .....	20
Authentifizierung und Zugriffskontrolle für Benutzeranwendungen .....	21
Berechtigungen zum Aufrufen von Amazon erteilen WorkDocs -APIs .....	21
Ordner-IDs in API-Aufrufen .....	23
Erstellen einer Anwendung .....	24
Anwendungsbereiche .....	24
Autorisierung .....	25
Aufrufen von Amazon WorkDocs -APIs .....	27
Amazon WorkDocs Content Manager .....	29
Erstellen von Amazon WorkDocs Content Manager .....	29
Herunterladen eines Dokuments .....	30
Hochladen eines Dokuments .....	31

Hinweis: Neukundenanmeldungen und Kontoerweiterungen sind für Amazon WorkDocs nicht mehr verfügbar. Erfahren Sie hier mehr über Migrationsschritte: [So migrieren Sie Daten von Amazon 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. Amazon WorkDocs ist vollständig verwaltet, sicher und auf Unternehmensebene skalierbar. Es bietet strenge administrative Kontrollen sowie Feedback-Funktionen, die zur Verbesserung 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 Amazon-Client-Anwendungen können verwendet werden, um viele verschiedene Dateitypen anzuzeigen, abhängig vom Internet-Medientyp der Datei. Amazon WorkDocs unterstützt alle gängigen Dokument- und Bildformate, und die Unterstützung für weitere Medientypen wird ständig hinzugefügt.

Weitere Informationen finden Sie unter [Amazon WorkDocs](#).

## Zugriff auf Amazon WorkDocs

Endbenutzer verwenden die Client-Anwendungen für den Zugriff auf ihre Dateien. Benutzer ohne Administratorrechte müssen niemals die WorkDocs Amazon-Konsole oder das Administrator-Dashboard verwenden. Amazon WorkDocs bietet verschiedene Client-Anwendungen und Utilities an:

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

## Preise

Bei Amazon WorkDocs fallen keine Vorabgebühren oder Verpflichtungen an. Sie zahlen nur für aktive Benutzerkonten und den Speicherplatz, den Sie nutzen. 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 rollenbasierten und speziellen Kursen sowie Übungen im Selbststudium zur Verbesserung Ihrer AWS-Kompetenzen und Erweiterung Ihrer praktischen Erfahrung.
- [AWS-Entwicklerzentrum](#) – Entdecken Sie Tutorials, laden Sie Tools herunter und erfahren Sie mehr über Veranstaltungen für AWS-Entwickler.
- [AWS-Entwickler-Tools](#) – Links zu Entwickler-Tools, SDKs, IDE-Toolkits und Befehlszeilen-Tools für die Entwicklung und Verwaltung von AWS-Anwendungen.
- [Ressourcenzentrum für die ersten Schritte](#) – Erfahren Sie, wie Sie Ihr AWS-Konto einrichten, der AWS-Community beitreten und Ihre erste Anwendung starten.
- [Praktische step-by-step Tutorials](#) — Schritt-zum Starten Ihrer ersten Anwendung auf AWS.
- [AWS Whitepaper](#) – Links zu einer umfangreichen Liste technischer AWS-Whitepaper zu Themen wie Architektur, Sicherheit und Wirtschaftlichkeit. Diese Whitepaper wurden von AWS-Lösungsarchitekten und anderen technischen Experten verfasst.
- [AWS Support-Center](#) – Hub für die Erstellung und Verwaltung Ihrer AWS Support-Fälle. Stellt darüber hinaus Links zu weiteren nützlichen Ressourcen bereit, beispielsweise Foren, häufig gestellten technischen Fragen, Status der Service-Integrität und AWS Trusted Advisor.
- [AWS Support](#)— Primäre Website für Informationen zu AWS Support one-on-one, einem reaktionsschnellen Support-Channel, der Sie bei der Erstellung und Ausführung von Anwendungen in der Cloud unterstützt.
- [Kontakt](#) – Zentraler Kontaktpunkt für Fragen zu AWS-Abrechnung, Konten, Ereignissen Missbrauch und anderen Problemen.
- [Nutzungsbedingungen für die AWS-Website](#) – Detaillierte Informationen zu unseren Copyright- und Markenbestimmungen, Ihrem Konto, den Lizenzen und anderen Themen.

# Erste Schritte

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

## Note

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

## Beispiele

- [Herstellen einer Verbindung zu Amazon WorkDocs mit IAM-Benutzeranmeldeinformationen und Abfragen nach Benutzern](#)
- [Herstellen einer Verbindung zu Amazon WorkDocs durch Übernahme einer Rolle](#)
- [Hochladen eines Dokuments](#)
- [Herunterladen eines Dokuments](#)
- [Einrichten von Benachrichtigungen](#)
- [Erstellen eines Benutzers](#)
- [Erteilen von Berechtigungen für Benutzer an eine Ressource](#)

## Herstellen einer Verbindung zu Amazon WorkDocs mit IAM-Benutzeranmeldeinformationen und Abfragen nach Benutzern

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 Amazon- WorkDocs Website zum selben AWS Konto.

## Note

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

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

Das Codebeispiel verwendet die [DescribeUsers](#) -API, 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 Upload- oder Download-Vorgänge für Inhalte durchführen möchten.

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

Gehen Sie wie folgt vor, um eine Amazon- 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 Ihrem Amazon- WorkDocs Standort entspricht. Das ist die Organisations-ID für den Standort.

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

```
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());
}
}
}
```

## Herstellen einer Verbindung zu Amazon WorkDocs durch Übernahme einer Rolle

In diesem Beispiel wird das AWS Java SDK verwendet, um eine Rolle zu übernehmen, und die temporären Sicherheitsanmeldeinformationen der Rolle für den Zugriff auf Amazon WorkDocs. Das Codebeispiel verwendet die [DescribeFolderContents](#) -API, um die Elemente im Ordner eines Benutzers 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 ein Softwareentwickler sein, um die Schritte in diesem Abschnitt ausführen zu können. Informationen zur Verwendung von Amazon WorkDocs zum Hochladen von Dateien finden Sie unter [Hochladen von Dateien](#) im Amazon- WorkDocs Benutzerhandbuch.

Gehen Sie wie folgt vor, um ein Dokument in Amazon hochzuladen WorkDocs.

So laden Sie ein Dokument hoch

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

Wenn Sie IAM-Benutzeranmeldeinformationen verwenden, finden Sie weitere Informationen unter [Herstellen einer Verbindung zu Amazon WorkDocs mit IAM-Benutzeranmeldeinformationen und Abfragen nach Benutzern](#). Wenn Sie eine IAM-Rolle übernehmen, finden Sie weitere Informationen unter [Herstellen einer Verbindung zu Amazon WorkDocs durch Übernahme einer Rolle](#).

### Note

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

```
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);
URLConnection connection = (URLConnection) 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 ein Softwareentwickler sein, um die Schritte in diesem Abschnitt ausführen zu können. Informationen zur Verwendung von Amazon WorkDocs zum Herunterladen von Dateien finden Sie unter [Herunterladen von Dateien](#) im Amazon- WorkDocs Benutzerhandbuch.

Um ein Dokument von Amazon herunterzuladen WorkDocs, rufen Sie wie folgt eine URL für den Download ab und verwenden Sie dann die von Ihrer Entwicklungsplattform bereitgestellten API-Aktionen, 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. Legen Sie IAM-Benutzer- oder Rollenberechtigungen fest, um Anrufern Zugriff auf APIs zur Verwaltung von Benachrichtigungsabonnements zu gewähren.
2. Rufen Sie die APIs für Benachrichtigungsabonnements auf, um die Veröffentlichung von SNS-Nachrichten auf Ihrem Endpunkt zu aktivieren oder zu deaktivieren.

### Note

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

## 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

Durch Aktivieren von Benachrichtigungen können Sie aufrufen, [CreateNotificationSubscription](#) nachdem Sie Benachrichtigungen abonniert haben.

1. Öffnen Sie die Amazon- WorkDocs Konsole unter <https://console.aws.amazon.com/zocalo/>.
2. Wählen Sie auf der Seite Ihre WorkDocs Standorte verwalten das gewünschte Verzeichnis aus und wählen Sie Aktionen und dann Benachrichtigungen verwalten aus.
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 bzw. der Sie den Empfang von Benachrichtigungen von Ihrer Amazon- WorkDocs Website erlauben möchten.

Informationen WorkDocs zum Aktivieren von Amazon für die Verwendung von Benachrichtigungen finden Sie unter [Verwenden der Amazon WorkDocs -API mit dem AWS SDK for Python und AWS Lambda](#) . Sobald Sie Benachrichtigungen aktiviert haben, können Sie und Ihr Benutzer sie abonnieren.

## So abonnieren Sie WorkDocs Benachrichtigungen

1. Bereiten Sie Ihren Endpunkt für die Verarbeitung von Amazon SNS-Nachrichten vor. Weitere Informationen finden Sie unter [Fanout zu HTTP/S-Endpunkten](#) im Amazon Simple Notification Service-Entwicklerhandbuch.

### Important

SNS sendet eine Bestätigungsnachricht an Ihren konfigurierten Endpunkt. Sie müssen diese Nachricht bestätigen, um Benachrichtigungen zu erhalten. Wenn Sie außerdem FIPS 140-2-validierte kryptografische Module benötigen, wenn Sie über eine Befehlszeilenschnittstelle oder eine API auf AWS zugreifen, verwenden Sie 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:
  - Organisations-ID abrufen
    1. Wählen Sie im Navigationsbereich der [AWS Directory Service-Konsole](#) Verzeichnisse aus.
    2. Die Verzeichnis-ID, die Ihrem Amazon- WorkDocs Standort entspricht, dient auch als Organisations-ID für diesen Standort.
  - 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 (Document | DocumentVersion | Folder).
- `parentEntityId` – Die ID des übergeordneten Elements.
- `entityType` – Der Typ der Entität (Document | DocumentVersion | Folder).
- `entityId` – Die ID der Entität.
- Aktion – 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 Amazon- WorkDocs Konsole unter <https://console.aws.amazon.com/zocalo/>.
2. Wählen Sie auf der Seite Ihre WorkDocs Standorte verwalten das gewünschte Verzeichnis aus und wählen Sie Aktionen und dann Benachrichtigungen verwalten aus.
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 Amazon erstellen WorkDocs.

### Note

Dies ist für eine Connected AD-Konfiguration keine gültige Operation. Um einen Benutzer in der Konfiguration Connected AD zu erstellen, muss der Benutzer bereits im Unternehmensverzeichnis vorhanden sein. Anschließend müssen Sie einen Aufruf an die [ActivateUser](#) -API senden, um den Benutzer in Amazon 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(10485761));
    request.setStorageRule(storageRule);
    CreateUserResult result = workDocsClient.createUser(request);
```

Gehen Sie wie folgt vor, um eine Amazon- 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 Ihrem Amazon- WorkDocs Standort entspricht. Das ist die Organisations-ID für den Standort.

## Erteilen von Berechtigungen für Benutzer an eine Ressource

Das folgende Beispiel zeigt, wie Sie die [AddResourcePermissions](#)-API verwenden, um einem CONTRIBUTOR Berechtigungen für USER eine Ressource 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

Die WorkDocs administrativen APIs von Amazon werden durch IAM-Richtlinien authentifiziert und autorisiert. IAM-Administratoren können eine IAM-Richtlinie erstellen und sie einer IAM-Rolle oder einem IAM-Benutzer anfügen, die vom Entwickler verwendet werden kann, um auf die API zuzugreifen.

Nachstehend folgen einige Beispiele:

## Aufgaben

- [Erteilung von Entwicklerberechtigungen für die WorkDocs Amazon-API](#)
- [Erteilung der Genehmigung von Drittanbietern für die WorkDocs Amazon-APIs](#)
- [Benutzern die Erlaubnis erteilen, eine IAM-Rolle zu übernehmen](#)
- [Beschränken des Zugriff auf eine bestimmte WorkDocs Amazon-Instance](#)

## Erteilung von Entwicklerberechtigungen für die WorkDocs Amazon-API

### 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 AWS Konto aus WorkDocs Amazon-API-Zugriff gewähren. Erstellen Sie dazu eine WorkDocs Amazon-API-Berechtigungsrichtlinie und hängen Sie sie an den IAM-Benutzer an. Die folgende API-Richtlinie gewährt den verschiedenen `Describe` APIs nur Leserechte.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WorkDocsAPIReadOnly",
```

```
        "Effect": "Allow",
        "Action": [
            "workdocs:Get*",
            "workdocs:Describe*"
        ],
        "Resource": [
            "*"
        ]
    }
]
```

## Erteilung der Genehmigung von Drittanbietern für die WorkDocs Amazon-APIs

Sie können den Zugriff externen Entwicklern oder Benutzern gewähren, die ein anderes AWS-Konto verwenden. Erstellen Sie dazu eine IAM-Rolle und fügen Sie WorkDocs Amazon-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 Amazon-Konto.
- Wenn ein Unternehmen externen Anwendungsentwicklern Zugriff auf die WorkDocs Amazon-API gewähren möchte.

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

Der Entwickler muss die folgenden Informationen angeben, 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 [Verwenden einer externen ID, um Dritten Zugriff auf Ihre AWS-Ressourcen zu gewähren](#).
- Eine Liste der WorkDocs Amazon-APIs, auf die Ihre Anwendung Zugriff benötigt. Die IAM-basierte Richtliniensteuerung bietet eine granulare Kontrolle und die Möglichkeit, Richtlinien für Zulassen

oder Verweigern auf der einzelnen API-Ebene zu definieren. Eine Liste der WorkDocs Amazon-APIs finden Sie in der [WorkDocs Amazon-API-Referenz](#).

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

So konfigurieren Sie IAM für konZugriff

1. Erstellen Sie eine WorkDocs Amazon-API-Berechtigungsrichtlinie, nennen Sie sie `workDocsAPIReadOnly` Richtlinie.
2. Erstellen Sie eine neue Rolle in der IAM-Konsole des AWS Kontos, das die WorkDocs Amazon-Website hostet:
  - a. Melden Sie sich bei der AWS Management Console an 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 innerhalb Ihres AWS-Kontos 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 gewähren, die Sie besitzen, wenn Sie 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 Drittanbieter-Konto gewähren aus, wenn Sie der Administrator des Kontos sind, dem die WorkDocs Amazon-Website gehört, und Benutzern von einem Anwendungsentwicklerkonto aus 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 ID des AWS-Kontos an, dem Sie Zugriff auf Ihre Ressourcen gewähren möchten. Geben Sie im Falle des Zugriffs durch Dritte auch eine External ID (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 Amazon WorkDocs API-Berechtigungsrichtlinie, die zuvor erstellt wurde, markieren 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 zu übernehmen

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

Die Richtlinie muss eine Aussage enthalten, die sich auf `diests:AssumeRole` AktionAllow auswirkt, und den Amazon-Ressourcennamen (ARN) der Rolle in einem `Resource` Element, wie im folgenden Beispiel dargestellt. Benutzer, die diese Richtlinie erhalten, entweder über eine Gruppenmitgliedschaft oder direkt angehängt, 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 Zugriff auf eine bestimmte WorkDocs Amazon-Instance

Wenn Sie mehrere WorkDocs Amazon-Websites in einem AWS Konto haben und API-Zugriff auf eine bestimmte Website 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 Bedingungelement:

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

Wenn die obige Bedingung in einer Richtlinie erfüllt ist, können Benutzer nur mit der ID von auf die WorkDocs Amazon-Instance zugreifend-123456789c5. Die WorkDocs Amazon-Instanz-ID wird manchmal als Organisations-ID oder Verzeichnis-ID bezeichnet. Weitere Informationen finden Sie unter [Beschränken des Zugriff auf eine bestimmte WorkDocs Amazon-Instance](#).

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

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 Amazon-Website entspricht. Das ist die Organisations-ID für die Site.



# Authentifizierung und Zugriffskontrolle für Benutzeranwendungen

Amazon WorkDocs -Anwendungen auf Benutzerebene werden über Amazon registriert und verwaltet WorkDocs console. Entwickler sollten ihre Anwendungen auf der My Applications Seite auf Amazon WorkDocs -Konsole, die eindeutige IDs für jede Anwendung bereitstellt. Während der Registrierung sollten Developer Umleitungs-URLs angeben, über die sie Zugriffs-Token und Anwendungsbereiche erhalten können.

Derzeit können Anwendungen nur auf Amazon zugreifen WorkDocs Websites innerhalb derselben AWS Konto, in dem sie registriert sind.

## Inhalt

- [Berechtigungen zum Aufrufen von Amazon erteilen WorkDocs -APIs](#)
- [Ordner-IDs in API-Aufrufen](#)
- [Erstellen einer Anwendung](#)
- [Anwendungsbereiche](#)
- [Autorisierung](#)
- [Aufrufen von Amazon WorkDocs -APIs](#)

## Berechtigungen zum Aufrufen von Amazon erteilen WorkDocs -APIs

Benutzer der Befehlszeilenschnittstelle müssen über vollständige Berechtigungen für Amazon verfügen WorkDocs und AWS Directory Service. Ohne die Berechtigungen kehren alle API-Aufrufe zurück UnauthorizedResourceAccessException messages. Die folgende Richtlinie gewährt vollständige 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 Leseberechtigungen erteilen 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 das gesamte Amazon WorkDocs Describe verwenden. Die `DescribeDirectories` action erhält Informationen über Ihre AWS

Directory Service-Verzeichnisse. Die Amazon EC2 EC2-Operationen ermöglichen Amazon WorkDocs , um eine Liste Ihrer VPCs und Subnetze zu erhalten.

## Ordner-IDs in API-Aufrufen

Immer wenn ein API-Aufruf auf einen Ordner zugreift, müssen Sie die Ordner-ID verwenden, nicht den Ordnernamen. Zum Beispiel, wenn Sie `bestehenclient.get_folder(FolderId='MyDocs')` gibt der API-Aufruf eine `UnauthorizedResourceAccessException` und die folgende 404-Nachricht.

```
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/abc123def456ghi789jkl1789mno4be7024df198736472dd50ca970eb22796082e3d489577`.

Die Übergabe dieser ID führt zu einem korrekten Ergebnis.

```
client.get_folder(FolderId='abc123def456ghi789jkl1789mno4be7024df198736472dd50ca970eb22796082e3d489577')
{'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': 'abc123def456ghi789jkl1789mno4be7024df198736472dd50ca970eb22796082e3d489577', '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 Amazon WorkDocs administrator, erstellen Sie Ihre Anwendung mit den folgenden Schritten.

So erstellen Sie eine Anwendung

1. Öffne den Amazon WorkDocs -Konsole bei <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.

URIs umleiten

Der Speicherort, den Amazon haben soll WorkDocs zum Umleiten des Datenverkehrs zu.

Anwendungsbereiche

Der für Ihre Anwendung gewünschte Bereich, entweder Lese- oder Schreibzugriff. Weitere Details finden Sie unter [Anwendungsbereiche](#).

4. Wählen Sie Create (Erstellen) aus.

## Anwendungsbereiche

Amazon WorkDocs unterstützt die folgenden Anwendungsbereiche:

- Gelesener Inhalt (`workdocs.content.read`), der Ihrer Anwendung Zugriff auf die folgenden Amazon gewährt WorkDocs APIs:
  - `Get*`
  - `Describe*`
- Schreiben von Inhalten (`workdocs.content.write`), der Ihrer Anwendung Zugriff auf die folgenden Amazon gewährt 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 Auftrag eines beliebigen Amazon eine Autorisierung anfordern. WorkDocs nutzer. Dazu sollte die Anwendung Amazon,, besuchen WorkDocs OAuth-Endpunkt,<https://auth.amazonworkdocs.com/oauth>, und stellen die folgenden Abfrageparameter bereit:

- [Erforderlich]`app_id`— Anwendungs-ID, die beim Registrieren einer Anwendung generiert wird.
- [Erforderlich]`auth_type`— OAuth-Typ für die Anforderung. Der unterstützte Wert ist `ImplicitGrant`.
- [Erforderlich]`redirect_uri`— Umleitungs-URI für eine Anwendung zum Empfangen eines Zugriffstoken.
- [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](#).

Eine GET-Beispielanforderung, mit der der OAuth-Ablauf zum Anfordern eines Zugriffstokens initiiert wird:

```
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
```

Folgendes findet während des OAuth-Autorisierungsablaufs statt:

1. Der Anwendungsbenutzer wird aufgefordert, Amazon zu betreten WorkDocs Site-Name.
2. Der Benutzer wird an Amazon umgeleitet WorkDocs -Authentifizierungsseite zum Eingeben ihrer Anmeldeinformationen.
3. Nach einer erfolgreichen Authentifizierung wird dem Benutzer der Zustimmungsbildschirm angezeigt. Hier kann der Benutzer der Anwendung die Autorisierung zum Zugriff auf Amazon entweder gewähren oder verweigern. WorkDocs.
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.

Eine GET-Beispielanforderung aus Amazon WorkDocs:

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

Neben dem Zugriffstoken ist der Amazon WorkDocs OAuth-Dienst kehrt ebenfalls `regionals` Abfrageparameter für das ausgewählte Amazon WorkDocs Seite. Externe Anwendungen sollten das `region` Parameter zur Bestimmung des Amazonas WorkDocs Service-Endpunkt.

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 von Amazon WorkDocs -APIs

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

### Important

Dieses Beispiel zeigt, wie eine curl-GET-Anforderung 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"
```

Ein -Beispiel JavaScript verwenden, um die Stammordner eines Benutzers zu beschreiben:

```
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());
}
```



# Amazon WorkDocs Content Manager

Amazon WorkDocs Content Manager ist ein übergeordnetes Utility-Tool, das Inhalte von Amazon hochlädt oder herunterlädt WorkDocs Seite.

Themen

- [Erstellen von Amazon WorkDocs Content Manager](#)
- [Herunterladen eines Dokuments](#)
- [Hochladen eines Dokuments](#)

## Erstellen von Amazon WorkDocs Content Manager

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

Für Benutzeranwendungen muss ein Entwickler Amazon erstellen WorkDocs Content Manager mit AnonymAWSAnmeldeinformationen und ein Authentifizierungstoken.

Für administrative Anwendungen bietet Amazon WorkDocs client muss initialisiert werden mitAWS Identity and Access Management(IAM) -Anmeldeinformationen. Darüber hinaus muss das Authentifizierungstoken in nachfolgenden API-Aufrufen weggelassen.

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

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(params);
```

## Herunterladen eines Dokuments

Entwickler können Amazon verwenden WorkDocs Content Manager zum Herunterladen einer bestimmten Version oder der neuesten Version eines Dokuments von Amazon WorkDocs. 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

Amazon WorkDocs Content Manager bietet eine API zum Hochladen von Inhalten auf Amazon WorkDocs Seite. 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();
```