



Guide du développeur

Amazon WorkDocs



Amazon WorkDocs: Guide du développeur

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

.....	iv
Qu'est-ce qu'Amazon WorkDocs ?	1
Accès à Amazon WorkDocs	1
Tarification	1
Ressources	2
Mise en route	3
Connectez-vous à Amazon WorkDocs avec les informations d'identification utilisateur IAM	3
Se connecter à Amazon WorkDocs en assumant un rôle	5
Chargement d'un document	8
Téléchargement d'un document	10
Configuration des notifications	10
Création d'un utilisateur	13
Octroi d'autorisations aux utilisateurs sur une ressource	14
Authentification et contrôle d'accès pour les applications administratives	16
Accorder aux développeurs des autorisations d'accès à l' WorkDocs API Amazon	16
Accorder aux développeurs tiers l'autorisation d'accéder aux WorkDocs API Amazon	17
Accorder aux utilisateurs l'autorisation d'assumer un rôle IAM	19
Restriction de l'accès à une WorkDocs instance Amazon spécifique	19
Authentification et contrôle d'accès pour les applications utilisateur	21
Octroi des autorisations d'appeler Amazon WorkDocs API	21
Utilisation des ID de dossier dans les appels d'API	23
Création d'une application	24
Périmètres d'application	24
Autorisation	25
Appel d'Amazon WorkDocs API	26
Amazon WorkDocs Gestionnaire de contenu	28
Construction d'Amazon WorkDocs Gestionnaire de contenu	28
Téléchargement d'un document	29
Chargement d'un document	30

Remarque : les inscriptions de nouveaux clients et les mises à niveau de compte ne sont plus disponibles pour Amazon WorkDocs. Découvrez les étapes de migration ici : [Comment migrer des données depuis Amazon WorkDocs](#).

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.

Qu'est-ce qu'Amazon WorkDocs ?

Amazon WorkDocs est un système de stockage, de collaboration et de partage de documents. Amazon WorkDocs est entièrement géré, sécurisé et à l'échelle de l'entreprise. Il fournit des contrôles administratifs stricts, ainsi que des fonctionnalités de feedback qui contribuent à améliorer la productivité des utilisateurs. Vos fichiers sont stockés dans [le cloud](#) en toute sécurité. Les fichiers de vos utilisateurs ne sont visibles que d'eux et de leurs collaborateurs et utilisateurs désignés. Les autres membres de votre organisation n'ont pas accès aux fichiers des autres utilisateurs, sauf si un accès spécifique leur est accordé.

Les utilisateurs peuvent partager leurs fichiers avec d'autres membres de votre organisation à des fins de collaboration ou de vérification. Les applications WorkDocs clientes Amazon peuvent être utilisées pour afficher de nombreux types de fichiers, en fonction du type de média Internet du fichier. Amazon WorkDocs prend en charge tous les formats de document et d'image courants, et la prise en charge d'autres types de supports est constamment ajoutée.

Pour plus d'informations, consultez [Amazon WorkDocs](#).

Accès à Amazon WorkDocs

Les utilisateurs finaux utilisent les applications clientes pour accéder à leurs fichiers. Les utilisateurs non administrateurs n'ont jamais besoin d'utiliser la WorkDocs console Amazon ou le tableau de bord d'administration. Amazon WorkDocs propose plusieurs applications et utilitaires clients différents :

- Une application web utilisée pour la gestion et la consultation de documents.
- Des applications natives pour appareils mobiles, utilisées pour la consultation des documents.
- Amazon WorkDocs Drive est utilisé pour synchroniser un dossier sur votre ordinateur Mac ou Windows avec vos WorkDocs fichiers Amazon.

Tarification

Avec Amazon WorkDocs, il n'y a pas de frais ni d'engagement initiaux. Vous ne payez que pour les comptes utilisateurs actifs et l'espace de stockage que vous utilisez. Pour plus d'informations, consultez la section concernant d'informations, section concernant [la](#) section concernant

Ressources

Les ressources connexes suivantes peuvent s'avérer utiles lors de l'utilisation de ce service.

- [Formations et ateliers](#) : liens vers des formations spécialisées et basées sur les rôles, ainsi que des ateliers d'autoformation pour améliorer vos compétences AWS et acquérir une expérience pratique.
- [Centre pour développeurs AWS](#) : parcourez des didacticiels, téléchargez des outils et découvrez les événements pour les développeurs AWS.
- [Outils pour développeur AWS](#) : liens vers des outils pour développeur, kits SDK, boîtes à outils IDE et outils de ligne de commande pour développer et gérer des applications AWS.
- [Centre de ressources pour la mise en route](#) : découvrez comment configurer votre Compte AWS, rejoindre la communauté AWS et lancer votre première application.
- [Tutoriels pratiques](#) : suivez des step-by-step tutoriels suivez des tutoriels pour lancer votre première application AWS.
- [Livres blancs AWS](#) : liens vers une liste complète des livres blancs techniques AWS couvrant des sujets tels que l'architecture, la sécurité et l'économie, créés par des architectes de solutions AWS ou d'autres experts techniques.
- [AWS SupportCentre](#) – Hub pour la création et la gestion de vos cas AWS Support. Inclut également des liens vers d'autres ressources utiles, telles que des forums, des FAQ techniques, l'état de santé d'un service et AWS Trusted Advisor.
- [AWS Support](#): principale page web d'informations à propos deAWS Support one-on-one, un canal d'assistance technique rapide pour vous aider à développer et à exécuter des applications dans le cloud.
- [Contactez-nous](#) : point de contact central pour toute question relative à la facturation AWS, à votre compte, aux événements, à des abus ou à d'autres problèmes.
- [AWSConditions d'utilisation du site](#) : informations détaillées sur nos droits d'auteur et notre marque, sur votre compte, votre licence et votre accès au site, et sur d'autres sujets.

Mise en route

Les extraits de code suivants peuvent vous aider à commencer à utiliser le SDK Amazon WorkDocs .

Note

Pour plus de sécurité, créez des utilisateurs fédérés plutôt que des utilisateurs IAM dans la mesure du possible.

Exemples

- [Connectez-vous à Amazon WorkDocs avec les informations d'identification utilisateur IAM et interrogez les utilisateurs](#)
- [Se connecter à Amazon WorkDocs en assumant un rôle](#)
- [Chargement d'un document](#)
- [Téléchargement d'un document](#)
- [Configuration des notifications](#)
- [Création d'un utilisateur](#)
- [Octroi d'autorisations aux utilisateurs sur une ressource](#)

Connectez-vous à Amazon WorkDocs avec les informations d'identification utilisateur IAM et interrogez les utilisateurs

Le code suivant montre comment utiliser les informations d'identification d'API d'un utilisateur IAM pour effectuer des appels d'API. Dans ce cas, l'utilisateur de l'API et le WorkDocs site Amazon appartiennent au même AWS compte.

Note

Pour plus de sécurité, créez des utilisateurs fédérés plutôt que des utilisateurs IAM dans la mesure du possible.

Assurez-vous que l'utilisateur IAM a obtenu l'accès à WorkDocs l'API Amazon via une politique IAM appropriée.

L'exemple de code utilise l'[DescribeUsers](#) API pour rechercher des utilisateurs et obtenir des métadonnées pour les utilisateurs. Les métadonnées utilisateur fournissent des informations telles que le prénom, le nom de famille, l'ID utilisateur et l'ID du dossier racine. L'ID du dossier racine est particulièrement utile si vous souhaitez effectuer des opérations de chargement ou de téléchargement de contenu pour le compte de l'utilisateur.

Le code nécessite que vous obteniez un identifiant d' WorkDocs organisation Amazon.

Procédez comme suit pour obtenir un identifiant d' WorkDocs organisation Amazon depuis la AWS console :

Pour obtenir un ID d'organisation

1. Dans le panneau de navigation de la [console AWS Directory Service](#), choisissez Annuaires.
2. Notez la valeur de l'ID de répertoire qui correspond à votre WorkDocs site Amazon. Il s'agit de l'identifiant d'organisation du site.

L'exemple suivant montre comment utiliser les informations d'identification IAM pour effectuer des appels d'API.

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

Se connecter à Amazon WorkDocs en assumant un rôle

Cet exemple utilise le SDK AWS Java pour assumer un rôle et utiliser les informations de sécurité temporaires du rôle pour accéder à Amazon WorkDocs. L'exemple de code utilise l'[DescribeFolderContents](#) API pour répertorier les éléments du dossier d'un utilisateur.

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

Chargement d'un document

Note

Vous devez être un développeur de logiciels pour effectuer les étapes décrites dans cette section. Pour plus d'informations sur l'utilisation WorkDocs d'Amazon pour charger des fichiers, consultez la section [Chargement de fichiers](#) dans le guide de WorkDocs l'utilisateur Amazon.

Suivez la procédure ci-dessous pour charger un document sur Amazon WorkDocs.

Pour charger un document

1. Créez une instance `AmazonWorkDocsClient` comme suit :

Si vous utilisez les informations d'identification de l'utilisateur IAM, reportez-vous à [Connectez-vous à Amazon WorkDocs avec les informations d'identification utilisateur IAM et interrogez les utilisateurs](#). Si vous assumez un rôle IAM, reportez-vous à [Se connecter à Amazon WorkDocs en assumant un rôle](#) pour plus d'informations.

Note

Pour plus de sécurité, créez des utilisateurs fédérés plutôt que des utilisateurs IAM dans la mesure du possible.

```
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. Obtenez l'URL signée pour le chargement comme suit :

```
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. Chargez le document à l'aide de l'URL signée comme suit :

```
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. Complétez le processus de chargement en modifiant le statut du document en ACTIVE comme suit :

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

Téléchargement d'un document

Note

Vous devez être un développeur de logiciels pour effectuer les étapes décrites dans cette section. Pour plus d'informations sur l'utilisation WorkDocs d'Amazon pour télécharger des fichiers, consultez la section [Téléchargement de fichiers](#) dans le guide de WorkDocs l'utilisateur Amazon.

Pour télécharger un document depuis Amazon WorkDocs, obtenez l'URL du téléchargement comme suit, puis utilisez les actions d'API fournies par votre plateforme de développement pour télécharger le fichier à l'aide de l'URL.

```
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());
```

Configuration des notifications

Pour configurer les notifications, procédez comme suit :

1. Définissez les autorisations d'utilisateur ou de rôle IAM pour permettre à l'appelant d'accéder aux API de gestion des abonnements aux notifications.
2. Appelez les API d'abonnement aux notifications pour activer ou désactiver la publication de messages SNS sur votre terminal.

Note

Pour plus de sécurité, créez des utilisateurs fédérés plutôt que des utilisateurs IAM dans la mesure du possible.

Pour définir les autorisations des utilisateurs IAM

- Utilisez la console IAM pour définir les autorisations suivantes pour l'utilisateur :

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

Pour activer les notifications

L'activation des notifications vous permet d'appeler [CreateNotificationSubscription](#) après vous être abonné aux notifications.

1. Ouvrez la WorkDocs console Amazon à l'[adresse https://console.aws.amazon.com/zocalo/](https://console.aws.amazon.com/zocalo/).
2. Sur la page Gérer vos WorkDocs sites, sélectionnez le répertoire souhaité et choisissez Actions, puis Gérer les notifications.
3. Sur la page Manage Notifications (Gérer les notifications), sélectionnez Enable Notifications (Activer les notifications).
4. Entrez l'ARN de l'utilisateur ou du rôle que vous souhaitez autoriser à recevoir des notifications de votre WorkDocs site Amazon.

Pour plus d'informations sur l'autorisation WorkDocs d'Amazon à utiliser les notifications, consultez [Utilisation de l' WorkDocs API Amazon avec le SDK AWS pour Python et AWS Lambda](#). Une fois que vous avez activé les notifications, vous et votre utilisateur pouvez vous y abonner.

Pour vous abonner aux WorkDocs notifications

1. Préparez votre terminal pour traiter les messages Amazon SNS. Pour plus d'informations, consultez [Fanout to HTTP/S endpoints dans le manuel](#) Amazon Simple Notification Service Developer Guide.

Important

SNS envoie un message de confirmation à votre point de terminaison configuré. Vous devez confirmer ce message pour recevoir des notifications. De même, si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-2 pour accéder à AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour en savoir plus sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez [Federal Information Processing Standard \(FIPS\) 140-2](#) (Normes de traitement de l'information fédérale).

2. Procédez comme suit :
 - Obtenir un identifiant d'organisation
 1. Dans le volet de navigation [de la console AWS Directory Service](#), sélectionnez Répertoires.
 2. L'ID de répertoire correspondant à votre WorkDocs site Amazon sert également d'identifiant d'organisation pour ce site.
 - Créez la demande d'abonnement comme suit :

```
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());
```

Notifications SNS

Le message comprend les informations suivantes :

- `organizationId`— L'identifiant de l'organisation.
- `parentEntityType`— Le type du parent (`Document` | `DocumentVersion` | `Folder`).
- `parentEntityId`— L'identifiant du parent.
- `entityType`— Le type de l'entité (`Document` | `DocumentVersion` | `Folder`).
- `entityId`— L'ID de l'entité.
- `action` — L'action, qui peut prendre l'une des valeurs suivantes :
 - `delete_document`
 - `move_document`
 - `recycle_document`
 - `rename_document`
 - `revoke_share_document`
 - `share_document`
 - `upload_document_version`

Pour désactiver les notifications

1. Ouvrez la WorkDocs console Amazon à l'[adresse https://console.aws.amazon.com/zocalo/](https://console.aws.amazon.com/zocalo/).
2. Sur la page Gérer vos WorkDocs sites, sélectionnez le répertoire souhaité et choisissez Actions, puis Gérer les notifications.
3. Sur la page Manage Notifications (Gérer les notifications), sélectionnez l'ARN pour lequel vous voulez désactiver les notifications, puis choisissez Disable Notifications (Désactiver les notifications).

Création d'un utilisateur

L'exemple suivant montre comment créer un utilisateur sur Amazon WorkDocs.

Note

Notez qu'il ne s'agit pas d'une opération valide pour une configuration Connected AD. Pour créer un utilisateur dans la configuration Connected AD, celui-ci doit déjà être présent dans l'annuaire de l'entreprise. Ensuite, vous devez appeler l'[ActivateUser](#) API pour activer l'utilisateur dans Amazon WorkDocs.

L'exemple suivant montre comment créer un utilisateur avec un quota de stockage de 1 gigaoctet.

```
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);
```

Procédez comme suit pour obtenir un identifiant d' WorkDocs organisation Amazon depuis la AWS console :

Pour obtenir un ID d'organisation

1. Dans le panneau de navigation de la [console AWS Directory Service](#), choisissez Annuaires.
2. Notez la valeur de l'ID de répertoire qui correspond à votre WorkDocs site Amazon. Il s'agit de l'identifiant d'organisation du site.

Octroi d'autorisations aux utilisateurs sur une ressource

L'exemple suivant montre comment utiliser l'[AddResourcePermissionsAPI](#) pour accorder CONTRIBUTOR des autorisations USER à une ressource. Vous pouvez également utiliser l'API pour accorder des autorisations à un utilisateur ou à un groupe sur un dossier ou un document.

```
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);
```

Authentification et contrôle d'accès pour les applications administratives

Les API WorkDocs administratives Amazon sont authentifiées et autorisées par le biais des politiques IAM. Les administrateurs IAM peuvent créer une politique IAM qui permet au développeur d'accéder à l'API.

Voici quelques exemples :

Tâches

- [Accorder aux développeurs des autorisations d'accès à l' WorkDocs API Amazon](#)
- [Accorder aux développeurs tiers l'autorisation d'accéder aux WorkDocs API Amazon](#)
- [Accorder aux utilisateurs l'autorisation d'assumer un rôle IAM](#)
- [Restriction de l'accès à une WorkDocs instance Amazon spécifique](#)

Accorder aux développeurs des autorisations d'accès à l' WorkDocs API Amazon

Note

Pour plus de sécurité, créez des utilisateurs fédérés plutôt que des utilisateurs IAM dans la mesure du possible.

Si vous êtes administrateur IAM, vous pouvez accorder l'accès à WorkDocs l'API Amazon à un utilisateur IAM à partir du même AWS compte. Pour ce faire, créez une politique d'autorisation de l' WorkDocs API Amazon et associez-la à l'utilisateur IAM. La politique d'API suivante accorde une autorisation en lecture seule aux différentes Describe API.

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

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

Accorder aux développeurs tiers l'autorisation d'accéder aux WorkDocs API Amazon

Vous pouvez accorder l'accès à des développeurs tiers ou à des utilisateurs qui utilisent un autre compte AWS. Pour ce faire, créez un rôle IAM et associez des politiques d'autorisation de l'WorkDocs API Amazon.

Cette forme d'accès est requise dans les cas suivants :

- Le développeur appartient à la même organisation, mais son AWS compte est différent du WorkDocs AWS compte Amazon.
- Lorsqu'une entreprise souhaite accorder l'accès à WorkDocs l'API Amazon à des développeurs d'applications tiers.

Dans ces deux scénarios, deux AWS comptes sont concernés : un AWS compte de développeur et un compte différent hébergeant un WorkDocs site Amazon.

Le développeur devra fournir les informations suivantes afin que l'administrateur du compte puisse créer le rôle IAM :

- Votre ID de compte AWS
- Un External ID unique que votre client utilise pour vous identifier. Pour plus d'informations, consultez [Procédure d'utilisation d'un ID externe lorsque vous accordez l'accès à vos ressources AWS à un tiers](#).
- Liste des WorkDocs API Amazon auxquelles votre application doit accéder. Le contrôle des politiques basé sur l'IAM fournit un contrôle granulaire, la possibilité de définir des politiques d'autorisation ou de refus au niveau de chaque API. Pour obtenir la liste des WorkDocs API Amazon, consultez [Amazon WorkDocs API Reference](#).

La procédure suivante décrit les étapes de la configuration d'IAM pour un accès inter-compte.

Pour configurer IAM pour l'accès entre comptes

1. Créez une politique d'autorisation des WorkDocs API Amazon, appelez-la `WorkDocsAPIReadOnly` politique.
2. Créez un nouveau rôle dans la console IAM du AWS compte hébergeant le WorkDocs site Amazon :
 - a. Connectez-vous à la AWS Management Console, puis ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
 - b. Dans le volet de navigation de la console, cliquez sur Roles, puis sur Create New Role.
 - c. Pour Role name (Nom du rôle), entrez un nom de rôle permettant d'identifier l'objectif du rôle : par exemple `workdocs_app_role`. Les noms de rôle de votre compte AWS doivent être uniques. Une fois le nom entré, cliquez sur Next Step.
 - d. Sur la page Select Role Type, sélectionnez la section Role for Cross-Account Access , puis choisissez le type de rôle que vous souhaitez créer :
 - Sélectionnez Provide access between AWS accounts you own (Fournir un accès entre des comptes qui vous appartiennent) si vous êtes l'administrateur du compte utilisateur et du compte de ressources, ou si les deux comptes appartiennent à la même organisation. C'est également l'option à sélectionner lorsque les utilisateurs, le rôle et les ressources auxquels l'accès est requis se trouvent tous dans le même compte.
 - Sélectionnez Fournir un accès entre votre AWS compte et un AWS compte tiers si vous êtes l'administrateur du compte propriétaire du WorkDocs site Amazon et que vous souhaitez accorder des autorisations aux utilisateurs depuis un compte de développeur d'applications. Cette option requiert la spécification d'un ID externe (que le tiers doit vous fournir) afin de fournir un contrôle supplémentaire sur les circonstances dans lesquelles le tiers peut utiliser le rôle pour accéder à vos ressources. Pour plus d'informations, consultez [Procédure d'utilisation d'un ID externe lorsque vous accordez l'accès à vos ressources AWS à un tiers](#).
 - e. Sur la page suivante, spécifiez l'ID de compte AWS auquel vous voulez accorder l'accès à vos ressources et entrez aussi External ID (ID externe) en cas d'accès tiers.
 - f. Cliquez sur Next Step (Étape suivante) pour attacher une stratégie.
3. Sur la page Attach Policy, recherchez la politique d'autorisation de l' WorkDocs API Amazon qui a été créée précédemment, cochez la case à côté de la politique et cliquez sur Étape suivante.

4. Vérifiez les détails, copiez l'ARN du rôle à titre de référence ultérieure et cliquez sur **Create Role** (Créer un rôle) pour terminer la création du rôle.
5. Partagez l'ARN du rôle avec le développeur. Voici un exemple de l'ARN du rôle :

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

Accorder aux utilisateurs l'autorisation d'assumer un rôle IAM

Un développeur disposant d'un AWS compte administratif peut autoriser un utilisateur à assumer un rôle IAM. Pour ce faire, vous devez créer une nouvelle politique et l'associer à cet utilisateur.

La politique doit inclure une déclaration ayant un `Allow` effet sur `sts:AssumeRole`, ainsi que l'Amazon Resource Name (ARN) du rôle dans un `Resource` élément, comme illustré dans l'exemple suivant. Les utilisateurs disposant de la politique, via l'appartenance à un groupe ou par la politique directement attachée, peuvent endosser le rôle spécifié.

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

Restriction de l'accès à une WorkDocs instance Amazon spécifique

Si vous avez plusieurs WorkDocs sites Amazon sur un AWS compte et que vous souhaitez accorder un accès API à un site spécifique, vous pouvez définir un `Condition` élément. L'élément `Condition` vous permet de spécifier des conditions lorsqu'une stratégie est appliquée.

L'exemple suivant montre un élément de condition :

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

```
}
```

La condition ci-dessus étant inscrite dans une politique, les utilisateurs ne peuvent accéder à l'instance Amazon WorkDocs qu'avec l'ID ded-123456789c5. L'ID d'instance Amazon WorkDocs est parfois appelé ID d'organisation ou ID de répertoire. Pour plus d'informations, veuillez consulter [Restriction de l'accès à une instance Amazon WorkDocs spécifique](#).

Pour obtenir un identifiant d'organisation Amazon à partir de la console AWS, procédez comme suit :

Pour obtenir un ID d'organisation

1. Dans le panneau de navigation de la console [AWS Directory Service](#), choisissez **Annuaire**.
2. Notez la valeur de l'ID de répertoire qui correspond à votre site Amazon WorkDocs. Il s'agit de l'identifiant d'organisation du site.

Authentification et contrôle d'accès pour les applications utilisateur

Amazon WorkDocs les applications de niveau utilisateur sont enregistrées et gérées via Amazon WorkDocs console Les développeurs doivent enregistrer leurs applications sur le My Applications page sur Amazon WorkDocs , qui fournit des ID uniques pour chaque application. Pendant l'inscription, les développeurs doivent spécifier les URI de redirection où ils recevront les jetons d'accès, ainsi que les paramètres d'application.

Actuellement, les applications ne peuvent accéder qu'à Amazon WorkDocs sites au sein du même AWS compte sur lequel ils sont enregistrés.

Table des matières

- [Octroi des autorisations d'appeler Amazon WorkDocs API](#)
- [Utilisation des ID de dossier dans les appels d'API](#)
- [Création d'une application](#)
- [Périmètres d'application](#)
- [Autorisation](#)
- [Appel d'Amazon WorkDocs API](#)

Octroi des autorisations d'appeler Amazon WorkDocs API

Les utilisateurs de l'interface de ligne de commande doivent disposer des autorisations complètes sur Amazon WorkDocs et AWS Directory Service. Sans les autorisations, tous les appels d'API renvoient `UnauthorizedResourceAccessException` Messages. La stratégie suivante accorde des autorisations complètes.

```
{
  "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": "*"
}
]
}
```

Si vous voulez accorder des autorisations en lecture seule, utilisez cette stratégie.

```

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

Dans la politique, la première action accorde l'accès à tous les WorkDocs `Describe`. Le `DescribeDirectories` action obtient des informations sur votre AWS Directory Service annuaires. Les opérations Amazon EC2 permettent à Amazon WorkDocs pour obtenir une liste de vos VPC et sous-réseaux.

Utilisation des ID de dossier dans les appels d'API

Chaque fois qu'un appel d'API accède à un dossier, vous devez utiliser l'ID du dossier, et non le nom du dossier. Par exemple, si vous passez `client.get_folder(FolderId='MyDocs')`, l'appel d'API renvoie un `UnauthorizedResourceAccessException` message et le message 404 suivant.

```
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.
```

Pour éviter cela, utilisez l'ID figurant dans l'URL du dossier.

<site.workdocs/index.html#/folder/abc123def456ghi789jkl1789mno4be7024df198736472dd50ca970eb22796082e3d489577>.

La transmission de cet ID renvoie un résultat correct.

```
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'}}}
```

Création d'une application

En tant qu'Amazon WorkDocs administrateur, créez votre application à l'aide des étapes suivantes.

Pour créer une application

1. Ouvrez Amazon WorkDocs Console chez <https://console.aws.amazon.com/zocalo/>.
2. Choisissez My Applications (Mes applications), Create application (Créer une application).
3. Entrez les valeurs suivantes :

Nom de l'application

Nom de l'application.

E-mail

Adresse e-mail à associer à l'application.

Description de l'application

Description de l'application.

URI de redirection

Emplacement que vous voulez Amazon WorkDocs pour que redirige le trafic vers.

Application Scopes (Périmètres d'application)

Le périmètre, en lecture ou écriture, que votre application doit avoir. Pour en savoir plus, consultez [Périmètres d'application](#).

4. Sélectionnez Create (Créer).

Périmètres d'application

Amazon WorkDocs prend en charge les périmètres d'application suivants :

- Contenu lu (`workdocs.content.read`), qui fournit à votre application l'accès aux Amazon suivants WorkDocs API :
 - Faites*
 - Describe*

- Contenu Write (`workdocs.content.write`), qui fournit à votre application l'accès aux Amazon suivants WorkDocs API :
 - Créer*
 - Mise à jour*
 - Supprimer*
 - Initiate*
 - Abort*
 - Addition*
 - Supprimez*

Autorisation

Une fois son inscription terminée, une application peut demander une autorisation au nom de n'importe quel Amazon WorkDocs utilisateur. Pour ce faire, l'application doit visiter le site Amazon WorkDocs point de terminaison OAuth, `https://auth.amazonworkdocs.com/oauth`, et fournir les paramètres de requête suivants :

- [Obligatoire]`app_id`: ID application généré quand une application est inscrite.
- [Obligatoire]`auth_type`: le type OAuth de la demande. La valeur prise en charge est `ImplicitGrant`.
- [Obligatoire]`redirect_uri`: URI de redirection inscrit pour qu'une application reçoive un jeton d'accès.
- [Facultatif]`scopes`: Liste de périmètres séparés par des virgules. En l'absence de spécification, la liste des paramètres sélectionnés pendant l'inscription sera utilisée.
- [Facultatif]`state`: une chaîne retournée en même temps qu'un jeton d'accès.

Note

Si vous avez besoin de modules cryptographiques validés FIPS 140-2 lorsque vous accédez à AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour en savoir plus sur les points de terminaison FIPS disponibles, consultez [Norme de traitement de l'information fédérale \(Federal Information Processing Standard \(FIPS\)\) 140-2](#).

Exemple de demande GET pour initier le flux OAuth et obtenir un jeton d'accès :

```
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
```

Le flux d'exécution suivant se produit pendant l'autorisation OAuth :

1. L'utilisateur de l'application est invité à accéder à Amazon WorkDocs Nom du site.
2. L'utilisateur est redirigé vers Amazon WorkDocs page d'authentification pour entrer leurs informations d'identification.
3. Après une authentification réussie, un écran de consentement s'affiche, qui permet à l'utilisateur d'accéder à Amazon ou de lui refuser l'autorisation d'accéder à Amazon. WorkDocs.
4. Une fois que l'utilisateur choisit Accept sur l'écran de consentement, son navigateur est redirigé vers l'URL de rappel de l'application avec le jeton d'accès et les informations de région comme paramètres de la requête.

Exemple de demande GET d'Amazon WorkDocs :

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

En plus du jeton d'accès, Amazon WorkDocs Le service OAuth revient également `region` en tant que paramètre de requête pour l'Amazon sélectionné WorkDocs site. Les applications externes doivent utiliser le `region` paramètre pour déterminer l'Amazon WorkDocs point de terminaison du service

Si vous avez besoin de modules cryptographiques validés FIPS 140-2 lorsque vous accédez à AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour en savoir plus sur les points de terminaison FIPS disponibles, consultez [Norme de traitement de l'information fédérale \(Federal Information Processing Standard \(FIPS\)\) 140-2](#).

Appel d'Amazon WorkDocs API

Après avoir obtenu le jeton d'accès, votre application peut faire des appels d'API en direction d'Amazon WorkDocsServices .

 Important

Cet exemple montre comment utiliser une requête curl GET pour obtenir les métadonnées d'un document.

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

Exemple JavaScript pour décrire les dossiers racine d'un utilisateur :

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

Un exemple d'appel d'API basé sur Java est décrit ci-après :

```
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 Gestionnaire de contenu

Amazon WorkDocs Le gestionnaire de contenu est un outil utilitaire de haut niveau qui télécharge du contenu ou le télécharge depuis un Amazon WorkDocs site.

Rubriques

- [Construction d'Amazon WorkDocs Gestionnaire de contenu](#)
- [Téléchargement d'un document](#)
- [Chargement d'un document](#)

Construction d'Amazon WorkDocs Gestionnaire de contenu

Vous pouvez utiliser Amazon WorkDocs Gestionnaire de contenu pour les applications administratives et utilisateurs.

Pour les applications utilisateur, un développeur doit créer Amazon WorkDocs Gestionnaire de contenu anonymeAWSdes informations d'identification et un jeton d'authentification.

Pour les applications administratives, Amazon WorkDocs le client doit être initialisé avecAWS Identity and Access Managementinformations d'identification (IAM). En outre, le jeton d'authentification doit être omis dans les appels d'API suivants.

Le code suivant montre comment initialiser Amazon. WorkDocs Gestionnaire de contenu pour les applications utilisateur utilisant Java ou C#.

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);
```

Téléchargement d'un document

Les développeurs peuvent utiliser Amazon WorkDocs Gestionnaire de contenu pour télécharger une version spécifique ou la dernière version d'un document depuis Amazon WorkDocs. Les exemples suivants montrent comment télécharger la version spécifique d'un document en Java et C#.

Note

Pour télécharger la dernière version d'un document, ne spécifiez pas le `VersionId` lors de la construction de la demande `GetDocumentStream`.

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();
```

Chargement d'un document

Amazon WorkDocs Le gestionnaire de contenu fournit une API pour télécharger du contenu sur Amazon WorkDocs site. Les exemples suivants montrent comment charger un document à l'aide de Java et 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();
```