
Amazon Translate

Developer Guide



Amazon Translate: Developer Guide

Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

¿Qué es Amazon Translate?	1
Idiomas y códigos de idioma admitidos	1
Casos de uso	3
Usuario principiante	3
Precios de Amazon Translate	4
Cómo funciona	5
Detección automática del idioma	5
Tratamiento de excepciones	5
Pasos siguientes	6
Introducción	7
Paso 1: Configurar una cuenta	7
Inscripción en AWS	7
Creación de un usuario de IAM	7
Paso siguiente	8
Paso 2: Configuración de la AWS CLI	8
Paso siguiente	9
Paso 3: Introducción (consola)	9
Paso siguiente	10
Paso 4: introducción (AWS CLI)	10
Traducción de texto con la línea de comandos	11
Traducción de texto con un archivo JSON	11
Paso siguiente	12
Paso 5: Introducción (SDK)	12
Uso de SDK para Java	12
Uso de AWS SDK para Python	13
Uso de SDK de Mobile para Android	15
Uso de Mobile SDK for iOS	16
Modos de procesamiento de traducción	19
Traducción en tiempo real	19
Procesamiento por lotes asincrónico	19
Disponibilidad por región	19
Requisitos previos	20
Inicio de un trabajo de traducción por lotes	20
Monitorización y análisis de trabajos de traducción por lotes	21
Obtención de resultados de traducción por lotes	22
Terminología personalizada	24
¿Cómo funciona?	24
Creación de terminología personalizada	24
Idiomas compatibles	25
Uso de terminología personalizada	26
Cifrado de la terminología	27
Prácticas recomendadas	27
Ejemplos	29
Uso de Amazon Polly con Amazon Translate	29
Code	29
Uso de Amazon Translate para traducir un canal de chat	33
Uso de Amazon Translate con DynamoDB	41
Código de ejemplo	42
Uso de Amazon Translate para traducir una página web	44
Uso de Amazon Translate para traducir documentos grandes	47
Uso de Signature Version 4 con Amazon Translate	49
Configuración	49
Code	50
Seguridad	53

Protección de los datos	53
Cifrado en reposo	54
Cifrado en tránsito	54
Administración de claves	54
Identity and Access Management	54
Público	55
Autenticación con identidades	55
Administración de acceso mediante políticas	57
Funcionamiento de Amazon Translate con IAM	59
Ejemplos de políticas basadas en identidad	61
Permitir a los usuarios ver sus propios permisos	64
Solución de problemas	64
Referencia de permisos de la API de Amazon Translate	66
Monitoreo	67
Monitorear con CloudWatch	69
Registro de llamadas al API de Amazon Translate con AWS CloudTrail	70
Dimensiones y métricas de CloudWatch para Amazon Translate	72
Validación de la conformidad	73
Resiliencia	74
Seguridad de la infraestructura	74
Directrices y límites	75
Regiones de AWS admitidas	75
Conformidad	75
Limitación controlada	75
Directrices	75
Límites de los servicios	75
Historial de revisión	77
Referencia de la API	81
Encabezados HTTP	81
Actions	81
DeleteTerminology	83
DescribeTextTranslationJob	85
GetTerminology	87
ImportTerminology	90
ListTerminologies	93
ListTextTranslationJobs	96
StartTextTranslationJob	99
StopTextTranslationJob	103
TranslateText	105
Data Types	108
AppliedTerminology	109
EncryptionKey	110
InputDataConfig	111
JobDetails	112
OutputDataConfig	113
Term	114
TerminologyData	115
TerminologyDataLocation	116
TerminologyProperties	117
TextTranslationJobFilter	119
TextTranslationJobProperties	120
Common Errors	122
Common Parameters	123
AWS glossary	126

¿Qué es Amazon Translate?

Amazon Translate es un servicio de traducción de textos que utiliza tecnologías de aprendizaje automático avanzadas para proporcionar traducción de calidad bajo demanda. Puede usar Amazon Translate para traducir documentos de texto no estructurado o crear aplicaciones que funcionen en varios idiomas.

Idiomas y códigos de idioma admitidos

Amazon Translate proporciona traducción entre un idioma de origen (el idioma de entrada) y un idioma de destino (el idioma de salida). La combinación de un idioma de origen y un idioma de destino se denomina par de idiomas.

Amazon Translate puede traducir texto entre los idiomas que se indican en la tabla siguiente.

Idioma	Código de idioma
Afrikáans	af
Albanés	sq
Amárico	am
Árabe	ar
Azerbaiyano	az
Bengalí	bn
Bosnio	bs
Búlgaro	bg
Chino simplificado	zh
Chino tradicional	zh-TW
Croata	hr
Checo	cs
Danés	da
Dari	fa-AF
Neerlandés	nl
Inglés	en
Estonio	et
Finés	fi
Francés	fr
Francés (Canadá)	fr-CA
Georgiano	ka

Idioma	Código de idioma
Alemán	de
Griego	el
Hausa	ha
Hebreo	he
Hindi	hi
Húngaro	hu
Indonesio	id
Italiano	it
Japonés	ja
Coreano	ko
Letón	lv
Malayo	ms
Noruego	no
Persa	fa
Pastún	ps
Polaco	pl
Portugués	pt
Rumano	ro
Ruso	ru
Serbio	sr
Eslovaco	sk
Esloveno	sl
Somalí	so
Español	es
Español (México)	es-MX
Suajili	sw
Sueco	sv
Tagalo	tl
Tamil	ta
Tailandés	th
Turco	tr

Idioma	Código de idioma
Ucraniano	uk
Urdu	ur
Vietnamita	vi

Casos de uso

Utilice Amazon Translate para hacer lo siguiente:

Habilitar experiencias de usuario multilingües en sus aplicaciones mediante la integración de Amazon Translate:

- Traduzca contenido creado en la empresa, como actas de reuniones, informes técnicos, artículos de base de conocimientos, publicaciones, etc.
- Traduzca comunicaciones interpersonales, como correo electrónico, chat en juegos, chat del servicio de atención al cliente, etc., para permitir a los clientes y los empleados conectarse en su idioma preferido.

Procese y administre los datos de entrada de la compañía:

- Analice texto, por ejemplo, de redes sociales y canales de noticias, en numerosos idiomas.
- Busque información, como casos eDiscovery, en muchos idiomas.

Habilite el procesamiento independiente del idioma mediante la integración de Amazon Translate con otros servicios de AWS:

- Extraiga entidades con nombre, sentimientos y expresiones clave del texto no estructurado, como las transmisiones de los medios sociales con [Amazon Comprehend](#).
- Cree subtítulos y subtítulos en directo en muchos idiomas con [Amazon Transcribe](#).
- Dikte el contenido traducido con [Amazon Polly](#).
- Traduzca repositorios de documentos almacenados en [Amazon S3](#).
- Traduzca el texto almacenado en las siguientes bases de datos: [Amazon DynamoDB](#), [Amazon Aurora](#) y [Amazon Redshift](#).
- Integre perfectamente los flujos de trabajo con [AWS Lambda](#) o [AWS Glue](#).

¿Es la primera vez que usa Amazon Translate?

Si es un usuario nuevo, le recomendamos que lea las siguientes secciones en orden:

1. [Funcionamiento de Amazon Translate \(p. 5\)](#) – presenta una introducción a Amazon Translate.
2. [Introducción a Amazon Translate \(p. 7\)](#) – explica cómo configurar su cuenta de AWS y empezar a usar Amazon Translate.
3. [Ejemplos \(p. 29\)](#) – proporciona ejemplos de código en Java y Python. Utilice los ejemplos para saber cómo funciona Amazon Translate.
4. [Referencia de la API \(p. 81\)](#) – contiene documentación de referencia para las operaciones de Amazon Translate.

Precios de Amazon Translate

Al igual que sucede con otros productos de AWS, no hay contratos ni compromisos mínimos para utilizar Amazon Translate. Para obtener más información sobre el costo de usar Amazon Translate, consulte [Precios de Amazon Translate](#).

Funcionamiento de Amazon Translate

El servicio Amazon Translate utiliza redes neurales entrenadas para la traducción de idiomas. De este modo, puede traducir un texto de un idioma de origen (el idioma original del texto) a un idioma de destino (el idioma al que se va a traducir el texto). Para obtener más información, consulte [Idiomas y códigos de idioma admitidos \(p. 1\)](#).

Cuando trabaje con Amazon Translate, deberá proporcionar el texto de origen y obtendrá el texto de salida:

- Texto de origen: texto que desea traducir. El texto de origen se debe proporcionar en formato UTF-8.
- Texto de salida: texto que Amazon Translate ha traducido al idioma de destino. El texto de salida también está en formato UTF-8. En función de los idiomas de origen y de destino, puede haber más caracteres en el texto de salida que en el de entrada.

El modelo de traducción consta de dos componentes, el codificador y el decodificador. El codificador lee una frase de origen palabra a palabra y construye una representación semántica que capta su significado. El decodificador usa la representación semántica para generar una traducción palabra a palabra en el idioma de destino.

Amazon Translate usa mecanismos de atención para comprender el contexto. Esto le ayuda a decidir qué palabras del texto de origen son más relevantes para generar la siguiente palabra de destino. Los mecanismos de atención permiten al decodificador centrarse en las partes más relevantes de una frase de origen. De este modo, se garantiza que el decodificador traduce correctamente palabras o expresiones ambiguas.

La palabra de destino que genera el modelo se convierte en la entrada del decodificador. La red continúa generando palabras hasta que llega al final de la frase.

Detección automática del idioma

Amazon Translate puede detectar automáticamente el idioma que se usa en el texto de origen. Para detectar automáticamente el idioma, especifique `auto` como idioma de origen cuando proporcione el texto de origen. Amazon Translate llama a Amazon Comprehend para que determine qué idioma se utiliza en el texto de origen. Al elegir la detección automática del idioma, acepta los términos de servicio y los acuerdos de Amazon Comprehend. Para obtener más información sobre los precios de Amazon Comprehend, consulte [Precios de Amazon Comprehend](#).

Tratamiento de excepciones

Si especifica un idioma de origen o de destino que no se admite, Amazon Translate devuelve las siguientes excepciones:

- `UnsupportedLanguagePairException`: Amazon Translate permite realizar traducciones entre todos los idiomas admitidos. Esta excepción se devuelve si no se admite el idioma de origen o el idioma de destino. Para obtener más información, consulte [Idiomas admitidos \(p. 1\)](#).

- `DetectedLanguageLowConfidenceException`: si utiliza la detección automática del idioma y Amazon Translate tiene poca confianza en que haya detectado el idioma de origen correcto, devuelve esta excepción. Si un nivel de confianza bajo es aceptable, puede utilizar el idioma de origen que se devuelve en la excepción.

Pasos siguientes

Ahora que ya conoce cómo funciona Amazon Translate, puede explorar las secciones siguientes para obtener información sobre la creación de una solución.

- [Introducción a Amazon Translate \(p. 7\)](#)
- [Ejemplos \(p. 29\)](#)

Introducción a Amazon Translate

Para comenzar a utilizar Amazon Translate, configure una cuenta de AWS y cree un usuario de AWS Identity and Access Management (IAM). Para utilizar la AWS Command Line Interface (AWS CLI), descárguela y configúrela.

Temas

- [Paso 1: Configurar una cuenta de AWS y crear un usuario administrador \(p. 7\)](#)
- [Paso 2: Configuración de la AWS Command Line Interface \(AWS CLI\) \(p. 8\)](#)
- [Paso 3: Introducción \(consola\) \(p. 9\)](#)
- [Paso 4: introducción \(AWS CLI\) \(p. 10\)](#)
- [Paso 5: Introducción \(SDK\) \(p. 12\)](#)

Paso 1: Configurar una cuenta de AWS y crear un usuario administrador

Antes de usar Amazon Translate por primera vez, realice las siguientes tareas:

1. [Inscripción en AWS \(p. 7\)](#)
2. [Creación de un usuario de IAM \(p. 7\)](#)

Inscripción en AWS

Al inscribirse en Amazon Web Services (AWS), su cuenta de AWS se registra automáticamente en todos los servicios de AWS, incluido Amazon Translate. Solo se le cobrará por los servicios que utilice.

Con Amazon Translate, paga solo por los recursos que usa. Si es un cliente nuevo de AWS, puede comenzar con Amazon Translate de forma gratuita. Para obtener más información, consulte [Capa gratuita de AWS](#).

Si ya dispone de una cuenta de AWS, pase a la siguiente sección.

Para crear una cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones en línea.

Parte del procedimiento de inscripción consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Registre su ID de cuenta de AWS ya que lo necesitará en la siguiente tarea.

Creación de un usuario de IAM

Los servicios de AWS, como Amazon Translate, requieren que proporcione credenciales cuando obtiene acceso a ellos. Esto permite al servicio determinar si tiene permisos para obtener acceso a sus recursos.

Se recomienda encarecidamente que obtenga acceso a AWS con AWS Identity and Access Management (IAM), no con las credenciales de su cuenta de AWS. Si desea usar IAM para obtener acceso a AWS,

Cree un usuario de IAM, añádalo a un grupo de IAM con permisos administrativos y, a continuación, conceda permisos administrativos al usuario de IAM que ha creado. Después, podrá obtener acceso a AWS utilizando una URL especial y las credenciales de usuario de IAM.

En los ejercicios de esta guía se presupone que tiene un usuario de IAM con privilegios de administrador llamado `adminuser`.

Para crear un usuario administrador

- En su cuenta de AWS, cree un usuario administrador llamado `adminuser`. Para obtener instrucciones, consulte [Creación del primer grupo de usuarios y administradores de IAM](#) en la Guía del usuario de IAM.

Para obtener más información sobre IAM, consulte lo siguiente:

- [AWS Identity and Access Management \(IAM\)](#)
- [Introducción](#)
- [Guía del usuario de IAM](#)

Paso siguiente

[Paso 2: Configuración de la AWS Command Line Interface \(AWS CLI\) \(p. 8\)](#)

Paso 2: Configuración de la AWS Command Line Interface (AWS CLI)

Puede utilizar la AWS CLI para realizar llamadas interactivas a Amazon Translate.

Para configurar la AWS CLI

1. Descargue y configure la AWS CLI. Para obtener instrucciones, consulte los siguientes temas en la AWS Command Line Interface Guía del usuario:
 - [Configuración inicial de la AWS Command Line Interface](#)
 - [Configuración de AWS Command Line Interface](#)
2. En el archivo `config` de la AWS CLI, añada un perfil con nombre para el usuario administrador:

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

Puede utilizar este perfil cuando ejecute los comandos de la AWS CLI. Para obtener más información sobre los perfiles con nombre, consulte [Perfiles con nombre](#) en la AWS Command Line Interface Guía del usuario. Para ver una lista de las regiones de AWS, consulte [Regiones y puntos de enlace](#) en la Referencia general de Amazon Web Services.

3. Compruebe la configuración; para ello, escriba el siguiente comando de ayuda en el símbolo del sistema:

```
aws translate help
```

Debería ver una breve descripción de Amazon Translate y una lista de los comandos disponibles.

Paso siguiente

[Paso 3: Introducción \(consola\) \(p. 9\)](#)

Paso 3: Introducción (consola)

La forma más sencilla de comenzar a utilizar Amazon Translate consiste en utilizar la consola para traducir un texto. Puede traducir hasta 5000 caracteres con la consola. Si no ha revisado los conceptos y la terminología en [Funcionamiento de Amazon Translate \(p. 5\)](#), le recomendamos que lo haga antes de continuar.

Para comenzar a traducir texto, vaya a la [Consola de administración de AWS](#) y abra la consola de Amazon Translate.

Si es la primera vez que ha utilizado Amazon Translate, elija Launch real-time translation (Lanzar la traducción en tiempo real).

En Real-time translation (Traducción en tiempo real), elija los idiomas de origen y destino. Escriba el texto que desee traducir en el cuadro de texto de la izquierda. El texto traducido aparece en el cuadro de texto de la derecha.

The screenshot displays the 'Try Amazon Translate' interface. It features a 'Translate text' section with two columns: 'Source language' and 'Target language'. The source language is set to 'English (en)' and the target language is 'German (de)'. A text box on the left contains the English text: 'Amazon Translate uses advanced machine learning technologies to provide high-quality translation on demand. Use it to translate unstructured text documents or to build applications that work in multiple languages.' This text is highlighted with a blue border. To the right, a corresponding German translation is partially visible. Below the text boxes, it shows '213 characters, 213 of 5000 bytes used' and a link to 'Please leave us feedback'.

En la sección JSON samples (Muestras JSON) puede ver la entrada y salida JSON de la operación [TranslateText \(p. 105\)](#).

JSON samples

JSON input and output for your AWS CLI or an AWS SDK

Translation

JSON request

```
{
  "Text": "",
  "SourceLanguageCode": "en",
  "TargetLanguageCode": "de"
}
```

JSON response

```
{
  "TranslatedText": "",
  "SourceLanguageCode": "",
  "TargetLanguageCode": ""
}
```

Paso siguiente

[Paso 4: introducción \(AWS CLI\) \(p. 10\)](#)

Paso 4: introducción (AWS CLI)

En los siguientes ejercicios, usará la interfaz de línea de comandos de AWS (AWS CLI) para traducir texto. Para completar estos ejercicios, es necesario conocer el uso de la CLI y disponer de un editor de texto. Para obtener más información, consulte [Paso 2: Configuración de la AWS Command Line Interface \(AWS CLI\) \(p. 8\)](#).

Existen dos maneras de utilizar la CLI para traducir texto con Amazon Translate. Para un texto breve, puede proporcionar el texto que desee traducir como un parámetro del comando `translate-text`. Para un texto largo, puede proporcionar el idioma de origen, el idioma de destino y el texto en un archivo JSON.

Para utilizar Amazon Translate desde la línea de comandos, debe conocer el punto de enlace y la región del servicio. Para ver una lista de los puntos de enlace y las regiones disponibles, consulte [Regiones y puntos de enlace de Amazon Translate](#) en Referencia general de AWS.

Traducción de texto con la línea de comandos

El siguiente ejemplo muestra cómo utilizar la operación `translate-text` desde la línea de comandos para traducir texto. El ejemplo está preparado para Unix, Linux y macOS. Para Windows, sustituya la barra oblicua inversa (`\`) del carácter de continuación de Unix al final de cada línea por el signo de intercalación (`^`). En la línea de comando, escriba lo siguiente.

```
aws translate translate-text \
    --region region \
    --source-language-code "en" \
    --target-language-code "es" \
    --text "hello, world"
```

La respuesta es el siguiente archivo JSON:

```
{
  "TargetLanguageCode": "es",
  "Text": "Hola, mundo",
  "SourceLanguageCode": "en"
}
```

Traducción de texto con un archivo JSON

Este ejemplo muestra cómo utilizar la operación `translate-text` para traducir un bloque de texto largo de un archivo JSON. Puede especificar el idioma de origen y de destino en la línea de comandos, pero en este ejemplo debe especificarlos en el archivo JSON.

Note

El archivo JSON está formateado para facilitar su lectura. Vuelva a formatear el campo `"Text"` para eliminar los saltos de línea.

El ejemplo está preparado para Unix, Linux y macOS. Para Windows, sustituya la barra diagonal invertida (`\`) utilizada como carácter de continuación de Unix al final de cada línea por el signo de intercalación (`^`).

Para traducir texto con un archivo JSON

1. Copie el siguiente texto en un archivo JSON denominado `translate.json`:

```
{
  "Text": "Amazon Translate translates documents between languages in
real time. It uses advanced machine learning technologies
to provide high-quality real-time translation. Use it to
translate documents or to build applications that work in
multiple languages.",
  "SourceLanguageCode": "en",
  "TargetLanguageCode": "fr"
}
```

2. En la AWS CLI, ejecute el siguiente comando:

```
aws translate translate-text \
    --region region \
    --cli-input-json file://translate.json > translated.json
```

El comando devuelve un archivo JSON que contiene el siguiente texto JSON:

```
{
  "TargetLanguageCode": "fr",
  "Text": "Amazon Translate traduit les documents entre
les langue en temps réel. Il utilise des technologies
avancées d'apprentissage de la machine pour fournir
une traduction en temps réel de haute qualité. Utilisez-le
pour traduire des documents ou pour créer des applications
qui fonctionnent en plusieurs langues.",
  "SourceLanguageCode": "en"
}
```

Paso siguiente

Para ver otras maneras de utilizar Amazon Translate, consulte [Ejemplos \(p. 29\)](#).

Paso 5: Introducción (SDK)

En los siguientes ejemplos se muestra cómo se utiliza la operación [TranslateText \(p. 105\)](#) de Amazon Translate en Java y Python. Puede utilizarlos para obtener más información sobre la operación `TranslateText` y como componentes básicos de sus propias aplicaciones.

Para ejecutar los ejemplos de Java, debe instalar AWS SDK for Java. Para obtener instrucciones sobre la instalación de SDK para Java, consulte [Configuración de AWS SDK para Java](#).

Temas

- [Traducción de texto con AWS SDK for Java \(p. 12\)](#)
- [Traducción de texto con AWS SDK for Python \(Boto\) \(p. 13\)](#)
- [Traducción de texto con SDK de AWS Mobile para Android \(p. 15\)](#)
- [Traducción de texto con AWS Mobile SDK for iOS \(p. 16\)](#)

Traducción de texto con AWS SDK for Java

El siguiente ejemplo muestra el uso de la operación [TranslateText \(p. 105\)](#) en Java. Para ejecutar este ejemplo, necesita AWS SDK for Java. Para obtener instrucciones sobre la instalación de SDK para Java, consulte [Configuración de AWS SDK para Java](#).

```
import com.amazonaws.auth.AWSSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.client.builder.AwsClientBuilder;
import com.amazonaws.services.translate.AmazonTranslate;
import com.amazonaws.services.translate.AmazonTranslateClient;
import com.amazonaws.services.translate.model.TranslateTextRequest;
import com.amazonaws.services.translate.model.TranslateTextResult;

public class App {
    private static final String REGION = "region";

    public static void main( String[] args ) {

        // Create credentials using a provider chain. For more information, see
        // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/credentials.html
        AWSCredentialsProvider awsCreds = DefaultAWSCredentialsProviderChain.getInstance();
```



```
AmazonTranslate translate = AmazonTranslateClient.builder()
    .withCredentials(new
AWSStaticCredentialsProvider(awsCreds.getCredentials()))
    .withRegion(REGION)
    .build();

TranslateTextRequest request = new TranslateTextRequest()
    .withText("Hello, world")
    .withSourceLanguageCode("en")
    .withTargetLanguageCode("es");
TranslateTextResult result = translate.translateText(request);
System.out.println(result.getTranslatedText());
}
}
```

Para obtener una lista de idiomas y códigos de idioma admitidos, consulte [Idiomas y códigos de idioma admitidos \(p. 1\)](#)

Traducción de texto con AWS SDK for Python (Boto)

El siguiente ejemplo muestra el uso de la operación [TranslateText \(p. 105\)](#) en Python. Para ejecutarlo, primero debe instalar Amazon Translate a través de la AWS CLI. Para obtener instrucciones, consulte [the section called "Paso 2: Configuración de la AWS CLI" \(p. 8\)](#).

```
import boto3

translate = boto3.client(service_name='translate', region_name='region', use_ssl=True)

result = translate.translate_text(Text="Hello, World",
    SourceLanguageCode="en", TargetLanguageCode="de")
print('TranslatedText: ' + result.get('TranslatedText'))
print('SourceLanguageCode: ' + result.get('SourceLanguageCode'))
print('TargetLanguageCode: ' + result.get('TargetLanguageCode'))
```

Para obtener una lista de códigos de idioma admitidos, consulte [Idiomas y códigos de idioma admitidos \(p. 1\)](#)

Terminología personalizada

Veamos otro ejemplo: aquí se ilustra cómo se utilizan las operaciones de terminología personalizada en Python:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import boto3

translate = boto3.client(service_name='translate')

# The terminology file 'my-first-terminology.csv' has the following contents:
'''
en,fr
Amazon Family,Amazon Famille
'''

# Read the terminology from a local file
with open('/tmp/my-first-terminology.csv', 'rb') as f:
    data = f.read()

file_data = bytearray(data)

print("Importing the terminology into Amazon Translate...")
```

```

    response = translate.import_terminology(Name='my-first-terminology',
MergeStrategy='OVERWRITE', TerminologyData={"File": file_data, "Format": 'CSV'})
    print("Terminology imported: "),
    print(response.get('TerminologyProperties'))
    print("\n")

    print("Getting the imported terminology...")
    response = translate.get_terminology(Name='my-first-terminology',
TerminologyDataFormat='CSV')
    print("Received terminology: "),
    print(response.get('TerminologyProperties'))
    print("The terminology data file can be downloaded here: " +
response.get('TerminologyDataLocation').get('Location'))
    print("\n")

    print("Listing the first 10 terminologies for the account...")
    response = translate.list_terminologies(MaxResults=10)
    print("Received terminologies: "),
    print(response.get('TerminologyPropertiesList'))
    print("\n")

    print("Translating 'Amazon Family' from English to French with no terminology...")
    response = translate.translate_text(Text="Amazon Family", SourceLanguageCode="en",
TargetLanguageCode="fr")
    print("Translated text: " + response.get('TranslatedText'))
    print("\n")

    print("Translating 'Amazon Family' from English to French with the 'my-first-
terminology' terminology...")
    response = translate.translate_text(Text="Amazon Family", TerminologyNames=["my-first-
terminology"], SourceLanguageCode="en", TargetLanguageCode="fr")
    print("Translated text: " + response.get('TranslatedText'))
    print("\n")

    # The terminology file 'my-updated-terminology.csv' has the following contents:
    '''
    en,fr
    Amazon Family,Amazon Famille
    Prime Video, Prime Video
    '''

    # Read the terminology from a local file
    with open('/tmp/my-updated-terminology.csv', 'rb') as f:
        data = f.read()

    file_data = bytearray(data)

    print("Updating the imported terminology in Amazon Translate...")
    response = translate.import_terminology(Name='my-first-terminology',
MergeStrategy='OVERWRITE', TerminologyData={"File": file_data, "Format": 'CSV'})
    print("Terminology updated: "),
    print(response.get('TerminologyProperties'))
    print("\n")

    print("Translating 'Prime Video' from English to French with no terminology...")
    response = translate.translate_text(Text="Prime Video", SourceLanguageCode="en",
TargetLanguageCode="fr")
    print("Translated text: " + response.get('TranslatedText'))
    print("\n")

    print("Translating 'Prime Video' from English to French with the 'my-first-terminology'
terminology...")
    response = translate.translate_text(Text="Prime Video", TerminologyNames=["my-first-
terminology"], SourceLanguageCode="en", TargetLanguageCode="fr")
    print("Translated text: " + response.get('TranslatedText'))
    print("\n")

```

```
print("Cleaning up by deleting 'my-first-terminology'...")
translate.delete_terminology(Name="my-first-terminology")
print("Terminology deleted.")
```

Traducción de texto con SDK de AWS Mobile para Android

Puede utilizar Amazon Translate en una aplicación Android para traducir texto.

Para configurar el ejemplo

1. Configure SDK de AWS Mobile para Android. Para obtener instrucciones, consulte [Android: opciones de configuración del SDK](#) en la Guía del desarrollador para móviles de AWS.
2. Cree un usuario de IAM con los permisos mínimos necesarios para ejecutar este ejemplo. Para obtener información acerca de cómo crear un usuario de IAM, consulte [Creación de un usuario de IAM en su cuenta de AWS](#) en la Guía del usuario de AWS Identity and Access Management. Para conocer las políticas de permisos necesarias, consulte [Políticas basadas en identidad de Amazon Translate \(p. 59\)](#). Después de crear el usuario, descargue las credenciales o registre la clave de acceso y la clave de acceso secreta.
3. Cree un nuevo proyecto con Android Studio.
4. Agregue lo siguiente a la sección de dependencias del archivo build.gradle.

```
dependencies {
    implementation 'com.amazonaws:aws-android-sdk-translate:2.6.20'
}
```

5. Añada los siguientes permisos al archivo AndroidManifest.xml.

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

6. Copie el código fuente en su proyecto.
7. Cambie la clave de acceso y la clave de acceso secreta en claves que registró en el paso 1.

Code

Utilice el siguiente código para crear el ejemplo.

```
package com.amazonaws.amazontranslatetester;

import android.app.Activity;
import android.util.Log;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.handlers.AsyncHandler;
import com.amazonaws.services.translate.AmazonTranslateAsyncClient;
import com.amazonaws.services.translate.model.TranslateTextRequest;
import com.amazonaws.services.translate.model.TranslateTextResult;

public class MainActivity extends Activity {

    private static final String LOG_TAG = MainActivity.class.getSimpleName();
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    AWSCredentials awsCredentials = new AWSCredentials() {
        @Override
        public String getAWSSecretKey() {
            return "secret key";
        }

        @Override
        public String getAWSAccessKeyId() {
            return "access key";
        }
    };

    AmazonTranslateAsyncClient translateAsyncClient = new
    AmazonTranslateAsyncClient(awsCredentials);
    TranslateTextRequest translateTextRequest = new TranslateTextRequest()
        .withText("Hello, world")
        .withSourceLanguageCode("en")
        .withTargetLanguageCode("es");
    translateAsyncClient.translateTextAsync(translateTextRequest, new
    AsyncHandler<TranslateTextRequest, TranslateTextResult>() {
        @Override
        public void onError(Exception e) {
            Log.e(LOG_TAG, "Error occurred in translating the text: " +
            e.getLocalizedMessage());
        }

        @Override
        public void onSuccess(TranslateTextRequest request, TranslateTextResult
        translateTextResult) {
            Log.d(LOG_TAG, "Original Text: " + request.getText());
            Log.d(LOG_TAG, "Translated Text: " +
            translateTextResult.getTranslatedText());
        }
    });
}
```

Traducción de texto con AWS Mobile SDK for iOS

Puede utilizar Amazon Translate en una aplicación iOS para traducir texto.

Para configurar el ejemplo

1. Cree un usuario de IAM con los permisos mínimos necesarios para ejecutar este ejemplo. Para obtener información acerca de cómo crear un usuario de IAM, consulte [Creación de un usuario de IAM en su cuenta de AWS](#) en la Guía del usuario de AWS Identity and Access Management. Para conocer las políticas de permisos necesarias, consulte [Políticas basadas en identidad de Amazon Translate \(p. 59\)](#). Después de crear el usuario, descargue las credenciales o registre la clave de acceso y la clave de acceso secreta.
2. Instale Xcode versión 8.0 o posterior. Puede descargar la versión más reciente de Xcode en el sitio web de Apple, <https://developer.apple.com/xcode/>.
3. Instale CocoaPods. En una ventana de terminal, ejecute el siguiente comando:

```
sudo gem install cocoapods
```

4. Cree un proyecto con Xcode. A continuación, en una ventana de terminal, vaya al directorio que contiene el archivo `.xcodproj` del proyecto y ejecute el siguiente comando:

```
pod init
```

5. Agregue los componentes de Mobile SDK for iOS básicos a su archivo pod:

```
platform :ios, '9.0'
target :'app name' do
  use_frameworks!
  pod 'AWSTranslate', '~> 2.6.19'
  # other pods
end
```

6. Instale las dependencias ejecutando el siguiente comando en una ventana del terminal:

```
pod install --repo-update
```

7. Al ejecutar "pod install" se crea un nuevo archivo de espacio de trabajo. Cierre el proyecto Xcode y ábralo utilizando el archivo `./nombre_del_proyecto.xcworkspace`. A partir de ahora solo debe utilizar este archivo para abrir su proyecto Xcode.

Vuelva a compilar la aplicación después de abrirla para resolver las API desde las nuevas bibliotecas a las que se llama en el código.

8. Agregue la siguiente instrucción import al controlador de vista.

```
import AWSTranslate
```

9. Copie el siguiente código en su proyecto XCode. Actualice la clave de acceso y la clave secreta en los valores que registró en el paso 1.

Code

Utilice el siguiente código para crear el ejemplo.

```
var credentialsProvider = AWSStaticCredentialsProvider(accessKey: "access key", secretKey: "secret key")

var configuration = AWSServiceConfiguration(region: AWSRegionUSEast1, credentialsProvider: credentialsProvider)

AWSServiceManager.default().defaultServiceConfiguration = configuration

let translateClient = AWSTranslate.default()
let translateRequest = AWSTranslateTranslateTextRequest()
translateRequest?.sourceLanguageCode = "en"
translateRequest?.targetLanguageCode = "es"
translateRequest?.text = "Hello World"

let callback: (AWSTranslateTranslateTextResponse?, Error?) -> Void = { (response, error) in
    guard let response = response else {
        print("Got error \(error)")
        return
    }

    if let translatedText = response.translatedText {
        print(translatedText)
    }
}
```

```
}  
translateClient.translateText(translateRequest!, completionHandler: callback)
```

Modos de procesamiento de traducción

Al traducir documentos, puede utilizar dos modos de procesamiento de traducción diferentes: traducción en tiempo real o procesamiento por lotes asincrónico. El modo que utilice depende del tamaño y del tipo de los documentos de destino, y afecta a la forma en que se envía el trabajo de traducción y se ven sus resultados.

- [Traducción en tiempo real \(p. 19\)](#): llama a Amazon Translate en un pequeño campo de texto y se envía inmediatamente una respuesta sincrónica a su aplicación.
- [Procesamiento por lotes asíncrono \(p. 19\)](#): coloca una colección de documentos en un bucket de Amazon Simple Storage Service (Amazon S3) e inicia un trabajo de procesamiento asíncrono para traducirlos. Amazon Translate envía el documento de salida traducido a un bucket de Amazon S3 especificado.

Traducción en tiempo real

Para traducir un único documento de texto y devolver los resultados de esa traducción directamente a su aplicación, utilice la traducción en tiempo real de Amazon Translate, [TranslateText \(p. 105\)](#). Utilice este modo de traducción para aplicaciones interactivas que funcionan con un documento cada vez.

Para ver un ejemplo de un flujo de trabajo de traducción en tiempo real, consulte [Traducción de texto con la línea de comandos \(p. 11\)](#).

Procesamiento por lotes asincrónico

Para traducir grandes colecciones de documentos (hasta 5 GB de tamaño), utilice la operación de procesamiento por lotes asincrónico de Amazon Translate, [StartTextTranslationJob \(p. 99\)](#). Esta es la mejor opción para colecciones de documentos cortos, como publicaciones en redes sociales o reseñas de usuarios, o para cualquier situación en la que no se requiera traducción instantánea.

Para realizar una traducción por lotes asincrónica, normalmente se realizan los siguientes pasos:

1. Guarde un conjunto de documentos en una carpeta dentro de un bucket de Amazon S3.
2. Proporcione a Amazon Translate un rol de IAM que tenga acceso de lectura a la carpeta de entrada de Amazon S3 y acceso de lectura y escritura a una carpeta de salida de Amazon S3.
3. Inicie un trabajo de traducción por lotes.
4. Monitoree el progreso del trabajo de traducción por lotes.
5. Recupere los resultados del trabajo de traducción por lotes de la carpeta de salida de S3 especificada.

Disponibilidad por región

La traducción por lotes se admite en las siguientes regiones de AWS:

- US East (N. Virginia)

- EE.UU. Este (Ohio)
- EE.UU. Oeste (Oregón)
- Asia Pacífico (Seúl)
- Europa (Fráncfort)
- Europa (Irlanda)
- Europa (Londres)

Requisitos previos

Se deben cumplir los siguientes requisitos previos para que Amazon Translate pueda realizar un trabajo de traducción por lotes correctamente:

- Un rol de IAM que tenga acceso de lectura a la carpeta de entrada de Amazon S3 y acceso de lectura y escritura a la carpeta de salida debe solicitar un trabajo de traducción por lotes. Para obtener más información, consulte [Uso de políticas de bucket y usuario](#) en la Guía para desarrolladores de Amazon S3.
- Las carpetas de Amazon S3 que contienen los documentos de entrada y salida deben estar en la misma región de AWS que el punto de enlace de la API a la que llama.
- Los documentos deben ser archivos `.txt` o `.html` con formato UTF-8.
- La colección de documentos de entrada por lotes debe tener un tamaño igual o inferior a 5 GB.
- Puede haber un máximo de un millón de documentos enviados en un trabajo de traducción por lotes.
- Cada documento de entrada debe tener un tamaño de 1 MB o menos.

Inicio de un trabajo de traducción por lotes

Para enviar un trabajo de traducción por lotes, utilice la consola de Amazon Translate o la operación [StartTextTranslationJob](#) (p. 99).

En el ejemplo siguiente se muestra cómo utilizar la operación `start-text-translation-job` de la AWS CLI para iniciar un trabajo de traducción por lotes. Si se realiza correctamente, la operación devuelve un mensaje de éxito y el `JobId` del nuevo trabajo de traducción.

Debe consignar el tipo de archivo de los documentos de entrada en el parámetro [InputDataConfig](#) (p. 111) `ContentType`. Si su entrada es un conjunto de documentos `txt`, use `text/plain`. Si su entrada es un conjunto de documentos `html`, use `text/html`.

Important

Amazon Translate no detecta automáticamente un idioma de origen durante los trabajos de traducción por lotes.

```
aws translate start-text-translation-job --job-name batch-test \
  --source-language-code en \
  --target-language-codes fr --input-data-config S3Uri=s3://batchtranslation/
input/,ContentType=text/plain \
  --output-data-config S3Uri=s3://batchtranslation/output/ \
  --data-access-role-arn arn:aws:iam::012345678901:role/service-role/
AmazonTranslateInputOutputAccess

{
  "JobStatus": "SUBMITTED",
  "JobId": "1c1838f470806ab9c3e0057f14717bed"
}
```


Note

Los trabajos de traducción por lotes son operaciones de larga duración y pueden tardar mucho tiempo en completarse. Por ejemplo, la traducción por lotes de un conjunto de datos pequeño puede tardar unos minutos, mientras que los conjuntos de datos muy grandes pueden tardar hasta dos días. El tiempo de finalización depende también de la disponibilidad de los recursos.

Monitorización y análisis de trabajos de traducción por lotes

Puede utilizar el ID de un trabajo para monitorizar su progreso y obtener la ubicación en Amazon S3 de sus documentos de salida. Para monitorizar un trabajo específico, utilice la operación [DescribeTextTranslationJob](#) (p. 85). También puede utilizar la operación [ListTextTranslationJobs](#) (p. 96) para recuperar información sobre todos los trabajos de traducción de su cuenta. Para restringir los resultados a trabajos que satisfagan determinados criterios, utilice el parámetro `filter` de la operación [ListTextTranslationJobs](#) (p. 96). Puede filtrar los resultados por nombre de trabajo, estado de trabajo o fecha y hora en que se envió el trabajo.

```
aws translate describe-text-translation-job --job-id 1c1838f470806ab9c3e0057f14717bed

{
  "TextTranslationJobProperties": {
    "InputDataConfig": {
      "ContentType": "text/plain",
      "S3Uri": "s3://batchtranslation/input/"
    },
    "EndTime": 1576551359.483,
    "SourceLanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::012345678901:role/service-role/AmazonTranslateInputOutputAccess",
    "JobId": "1c1838f470806ab9c3e0057f14717bed",
    "TargetLanguageCodes": [
      "fr"
    ],
    "JobName": "batch-test",
    "SubmittedTime": 1576544017.357,
    "JobStatus": "COMPLETED",
    "Message": "Your job has completed successfully.",
    "JobDetails": {
      "InputDocumentsCount": 77,
      "DocumentsWithErrorsCount": 0,
      "TranslatedDocumentsCount": 77
    },
    "OutputDataConfig": {
      "S3Uri": "s3://batchtranslation/output/012345678901-TranslateText-1c1838f470806ab9c3e0057f14717bed/"
    }
  }
}
```

Puede detener un trabajo de traducción por lotes mientras su estado es `IN_PROGRESS` con la operación [StopTextTranslationJob](#) (p. 103).

```
aws translate stop-text-translation-job --job-id 5236d36ce5192abdb3e2519f3ab8b065

{
  "TextTranslationJobProperties": {
    "InputDataConfig": {
      "ContentType": "text/plain",
      "S3Uri": "s3://batchtranslation/input/"
    },
    ...
  }
}
```

```
    "SourceLanguageCode": "en",
    "DataAccessRoleArn": "arn:aws:iam::012345678901:role/service-role/
AmazonTranslateInputOutputAccess",
    "TargetLanguageCodes": [
        "fr"
    ],
    "JobName": "canceled-test",
    "SubmittedTime": 1576558958.167,
    "JobStatus": "STOP_REQUESTED",
    "JobId": "5236d36ce5192abdb3e2519f3ab8b065",
    "OutputDataConfig": {
        "S3Uri": "s3://batchtranslation/output/012345678901-
TranslateText-5236d36ce5192abdb3e2519f3ab8b065/"
    }
}
}
```

Obtención de resultados de traducción por lotes

Una vez que el estado del trabajo sea `COMPLETED` o `COMPLETED_WITH_ERROR`, los documentos de salida estarán disponibles en la carpeta de Amazon S3 especificada. Los nombres de los documentos de salida coinciden con los nombres de los documentos de entrada, con el código de idioma de destino añadido como prefijo. Por ejemplo, si ha traducido un documento llamado `mySourceText.txt` al francés, el documento de salida se llamará `fr.mySourceText.txt`.

Si el estado de un trabajo de traducción por lotes es `FAILED`, la respuesta de la operación [DescribeTextTranslationJob](#) (p. 85) incluye un campo `Message` que describe el motivo por el que el trabajo no se completó correctamente.

Cada trabajo de traducción por lotes también genera un archivo auxiliar que contiene información sobre las traducciones realizadas, como el número total de caracteres traducidos y el número de errores encontrados. Este archivo, llamado `target-language-code.auxiliary-translation-details.json`, se genera en la subcarpeta `details` de la carpeta de salida.

A continuación, se muestra un ejemplo de un archivo auxiliar de traducción por lotes.

```
{
  "sourceLanguageCode": "en",
  "targetLanguageCode": "fr",
  "charactersTranslated": "105",
  "documentCountWithCustomerError": "0",
  "documentCountWithServerError": "0",
  "inputDataPrefix": "s3://batchtranslation/input/",
  "outputDataPrefix": "s3://batchtranslation/output/012345678901-
TranslateText-1c1838f470806ab9c3e0057f14717bed/",
  "details": [
    {
      "sourceFile": "mySourceText.txt",
      "targetFile": "fr.mySourceText.txt",
      "auxiliaryData": {
        "appliedTerminologies": [
          {
            "name": "TestTerminology",
            "terms": [
              {
                "sourceText": "Amazon",
                "targetText": "Amazon"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```
    },  
    {  
      "sourceFile": "batchText.txt",  
      "targetFile": "fr.batchText.txt",  
      "auxiliaryData": {  
        "appliedTerminologies": [  
          {  
            "name": "TestTerminology",  
            "terms": [  
              {  
                "sourceText": "Amazon",  
                "targetText": "Amazon"  
              }  
            ]  
          }  
        ]  
      }  
    }  
  ]  
}
```

Terminología personalizada

Si utiliza terminología personalizada con sus solicitudes de traducción, tendrá la seguridad de que los nombres de las marcas, los personajes, los modelos y otro contenido único se traducen de forma precisa, con independencia del contexto y la decisión del algoritmo de Amazon Translate.

Resulta muy fácil configurar un archivo de terminología y asociarlo a la cuenta de Amazon Translate. Cuando traduzca un texto, solo tendrá que decidir si desea utilizar la terminología personalizada y los ejemplos de las palabras de origen se traducirán del modo que desee.

Por ejemplo, supongamos que Amazon Family es un conjunto de beneficios que ofrece ofertas exclusivas a los miembros de Amazon Prime, como un descuento de hasta el 20 % en pañales, comida para bebé, etc. Imaginemos que en Francia se llama Amazon Famille. Si traduce Amazon Family al francés con Amazon Translate sin ningún tipo de información adicional, el resultado será Famille Amazon. Aunque esta traducción es correcta, no es lo que busca el equipo francés de Amazon. Sin embargo, si añade información contextual, como en "¿Alguna vez ha utilizado Amazon Family en sus compras?", Amazon Translate determinará que el nombre del programa no debe traducirse y el resultado será "Avez-vous déjà fait des achats avec Amazon Family?". Esta traducción está bien, pero sigue sin ser lo que el equipo de Amazon busca. La terminología personalizada puede resolver problemas como este. Si en la terminología personalizada añade una entrada en la que se especifique que el término Amazon Family debe traducirse como Amazon Famille, el equipo tendrá la seguridad de que, con independencia del contexto, siempre se traducirá de ese modo. Ahora, Amazon Family se traducirá como Amazon Famille y "¿Alguna vez ha utilizado Amazon Family en sus compras?", como "Avez-vous déjà fait des achats avec Amazon Famille?".

Temas

- [¿Cómo funciona? \(p. 24\)](#)
- [Creación de terminología personalizada \(p. 24\)](#)
- [Uso de terminología personalizada \(p. 26\)](#)
- [Cifrado de la terminología \(p. 27\)](#)
- [Prácticas recomendadas \(p. 27\)](#)

¿Cómo funciona?

A grandes rasgos, cuando llega una solicitud de traducción, Amazon Translate lee la frase de origen, crea una representación semántica del contenido (en otras palabras, comprende la frase) y genera una traducción en el idioma de destino.

Cuando se utiliza terminología personalizada en una solicitud de traducción, el motor examina el archivo de terminología antes de devolver el resultado final. Si el motor identifica una coincidencia exacta entre una entrada terminológica y una cadena del texto de origen, busca la cadena correspondiente en la traducción propuesta y la sustituye por la entrada terminológica. En el ejemplo de Amazon Family, primero genera la traducción "Avez-vous déjà fait des achats avec Amazon Family?", pero después se detiene y sustituye Amazon Family por Amazon Famille antes de proporcionar la respuesta.

Creación de terminología personalizada

Puede utilizar un archivo CSV o un archivo TMX con el texto de origen y el término de destino (traducido) como archivo de terminología. En cada término, debe utilizarse una única cadena de origen. Sin embargo,

puede haber varios términos de destino (uno por cada idioma) siempre que los idiomas de origen y destino sean viables.

CSV (valores separados por comas)

La primera columna contiene el texto de origen, mientras que el resto de columnas contienen las traducciones de destino. La primera fila contiene los códigos de idioma. La primera columna alberga el texto de origen y las otras, las traducciones de destino.

en	fr	de	es	
Amazon	Amazon	Amazon	Amazon	

TMX (Translation Memory eXchange)

Un archivo TMX es un archivo de tipo XML que se utiliza habitualmente en el software de traducción. Aunque el formato es diferente al de CSV, el contenido es similar:

```
<?xml version="1.0" encoding="UTF-8"?>
<tmx version="1.4">
  <header
    creationtool="XYZTool" creationtoolversion="0"
    datatype="PlainText" segtype="sentence"
    adminlang="en-us" srclang="en"
    o-tmf="test"/>
  <body>
    <tu>
      <tuv xml:lang="en">
        <seg>Amazon</seg>
      </tuv>
      <tuv xml:lang="fr">
        <seg>Amazon</seg>
      </tuv>
      <tuv xml:lang="de">
        <seg>Amazon</seg>
      </tuv>
      <tuv xml:lang="es">
        <seg>Amazon</seg>
      </tuv>
    </tu>
  </body>
</tmx>
```

Estos archivos se asocian a la cuenta de Amazon Translate. Cuando un trabajo de traducción se ejecuta y se decide emplear terminología personalizada, Amazon Translate utiliza la palabra designada siempre que encuentra el término de origen.

Important

En una terminología personalizada, el término de origen distingue entre mayúsculas y minúsculas y no funcionará con palabras que no coincidan exactamente.

Idiomas compatibles

Algunos idiomas no cambian la forma de una palabra en función del contexto oracional. En estos idiomas, es muy probable que la calidad general de la traducción mejore aplicando terminología personalizada. Sin embargo, algunos idiomas realizan muchos cambios en la forma de las palabras. No es conveniente aplicar esta característica en dichos idiomas, aunque puede hacerse.

En la tabla siguiente, se muestran algunos idiomas y las recomendaciones sobre el uso de esta característica:

Idioma	Recomendado/no recomendado
Lenguas de Asia oriental (chino, indonesio, japonés, coreano, malayo)	Recomendado
Lenguas germánicas (danés, neerlandés, alemán, inglés, noruego, sueco)	Recomendado
Lenguas romances (francés, italiano, rumano, español, portugués)	Recomendado
Lenguas semíticas (árabe, hebreo)	Recomendado
Lenguas indoiránias (hindi, persa, urdu)	Recomendado
Griego	Recomendado
Tailandés	Recomendado
Vietnamita	Recomendado
Lenguas baltoeslavas (checo, polaco, ruso, ucraniano)	No recomendado
Lenguas urálicas (finlandés, húngaro)	No recomendado
Árabe	No recomendado
Turco	No recomendado

Uso de terminología personalizada

Cuando se utiliza terminología personalizada para traducir texto con la operación [TranslateText \(p. 105\)](#), el proceso es muy parecido a cuando no se utiliza esta característica. La diferencia es que, al llamar a la operación con la terminología personalizada, se puede incluir el parámetro opcional `TerminologyNames`.

Por ejemplo, si tiene un archivo de terminología llamado `Amazon_Family.csv` asociado a su cuenta:

```
en,fr
Amazon Family,Amazon Famille
```

Puede utilizar lo siguiente para llamar a la terminología personalizada y traducir el texto a través de la CLI:

Note

Este ejemplo está preparado para Unix, Linux y macOS. En Windows, tiene que sustituir la barra oblicua inversa (`\`) que se utiliza como carácter de continuación de Unix al final de cada línea por el signo de intercalación (`^`).

```
aws translate translate-text \
  --region region \
  --source-language-code "en" \
  --target-language-code "fr" \
  --terminology-names "Amazon_Family" \
```

```
--text "Have you ever shopped with Amazon Family?"
```

De este modo, se utilizará la terminología personalizada seleccionada para traducir el texto a "Avez-vous déjà fait des achats avec Amazon Famille?" en lugar de utilizar la traducción directa (y no deseada) "Avez-vous déjà fait des achats avec Famille Amazon?".

En Python, la misma tarea puede verse del modo siguiente:

```
import boto3

translate = boto3.client(service_name='translate')

print("Translating 'Have you ever shopped with Amazon Family?' from English to French with
the 'Amazon_Family' custom terminology...")
response = translate.translate_text(Text="Have you ever shopped with Amazon Family?",
TerminologyNames=["Amazon_Family"], SourceLanguageCode="en", TargetLanguageCode="fr")
print("Translated text: " + response.get('TranslatedText'))
print("\n")
```

Para obtener más información acerca del uso de las operaciones de Amazon Translate con terminología personalizada, consulte [Actions \(p. 81\)](#).

Cifrado de la terminología

Amazon Translate utiliza todos los medios a su alcance para proteger los datos y las terminologías personalizadas. Cuando se crea una terminología personalizada, se cifra para que solo usted pueda obtener acceso a ella.

Hay tres opciones de cifrado disponibles:

- Uso del cifrado de AWS. Esta es la opción predeterminada y se utiliza para proteger la información si no se selecciona ningún otro método.
- Uso de una clave de cifrado asociada a la cuenta. Si se selecciona esta opción, aparecerá un menú en la consola con una selección de las claves de cifrado asociadas que se van a utilizar.
- Uso de una clave de cifrado no asociada a la cuenta. Si elige esta opción, aparecerá un cuadro en la consola para que escriba el nombre de recurso de Amazon (ARN) de la clave de cifrado.

Prácticas recomendadas

A continuación, se indican algunas prácticas recomendadas que conviene aplicar cuando se utiliza terminología personalizada.

- Reduzca la terminología personalizada al mínimo posible. Incluya únicamente las palabras que quiere controlar y que no presentan ninguna ambigüedad. Utilice únicamente palabras que sepa que nunca se van a utilizar con otra acepción y que quiere que se traduzcan siempre de la misma manera. Si es posible, utilice exclusivamente nombres propios en la lista, como nombres de marcas y de productos.
- Las terminologías personalizadas distinguen entre mayúsculas y minúsculas. Si necesita tener una versión en mayúscula y otra en minúscula de una misma palabra, deberá incluir una entrada para cada versión.
- No incluya diferentes traducciones para el mismo término de origen (por ejemplo, entrada 1-EN: Amazon, FR: Amazon; entrada 2-EN: Amazon, FR: Amazone).
- Algunos idiomas no cambian la forma de una palabra en función del contexto oracional. En estos idiomas, es muy probable que la calidad general de la traducción mejore aplicando terminología

personalizada. Sin embargo, algunos idiomas realizan muchos cambios en la forma de las palabras. No es conveniente aplicar esta característica en dichos idiomas, aunque puede hacerse. Para obtener más información, consulte [Idiomas compatibles \(p. 25\)](#).

Ejemplos

En los siguientes ejemplos se muestran distintas formas de usar Amazon Translate.

Temas

- [Uso de Amazon Polly con Amazon Translate \(p. 29\)](#)
- [Uso de Amazon Translate para traducir un canal de chat \(p. 33\)](#)
- [Uso de Amazon Translate con Amazon DynamoDB \(p. 41\)](#)
- [Uso de Amazon Translate para traducir una página web \(p. 44\)](#)
- [Uso de Amazon Translate para traducir documentos grandes \(p. 47\)](#)
- [Uso de Signature Version 4 con Amazon Translate \(p. 49\)](#)

Uso de Amazon Polly con Amazon Translate

Para dictar texto traducido, puede utilizar Amazon Polly con Amazon Translate. En este ejemplo, creará una página web en la que puede traducir texto mediante Amazon Translate y después dictarlo con Amazon Polly. El código se puede resumir de la forma siguiente:

- CSS y HTML para crear la página web.
- Código de inicialización que crea los controladores de Amazon Translate y Amazon Polly.
- Función que lee los datos de la página web y llama a Amazon Translate.
- Función que lee los datos de la página web y llama a Amazon Polly.
- Funciones de utilidad para administrar la página web.

Para configurar el ejemplo

1. Instale y configure AWS SDK for JavaScript. Para obtener instrucciones acerca de cómo instalar SDK for JavaScript, consulte [Instalación del SDK para JavaScript](#).
2. Copie el código del ejemplo en un archivo HTML en el servidor web.
3. Actualice la etiqueta `<script>` con la ubicación en la que ha instalado el SDK for JavaScript.
4. Cambie la región y el punto de enlace a la región en la que desea ejecutar las operaciones de Amazon Translate y Amazon Polly. Para obtener una lista de las regiones admitidas para Amazon Translate, consulte [Regiones y puntos de enlace de AWS](#). Para ver una lista de las regiones admitidas para Amazon Polly, consulte [Regiones y puntos de enlace de AWS](#) en Referencia general de AWS.
5. Cree un usuario de IAM con los permisos mínimos necesarios para ejecutar este ejemplo. Para obtener información acerca de cómo crear un usuario de IAM, consulte [Creación de un usuario de IAM en su cuenta de AWS](#) en la Guía del usuario de AWS Identity and Access Management. Para obtener información sobre las políticas de permisos necesarias, consulte [Políticas basadas en identidad de Amazon Translate \(p. 59\)](#) y [Uso de políticas basadas en identidades \(políticas de IAM\) para Amazon Polly](#) en la Guía para desarrolladores de Amazon Polly.
6. Proporcione el ID de acceso y la clave secreta del usuario de IAM que ha creado en el paso anterior.

Code

A continuación se incluye el código completo de la página web de ejemplo. Puede copiar este código en un archivo HTML para ejecutar el ejemplo en su propio servidor web.

```
<!DOCTYPE html>
```

```
<html>

<head>
  <title>Amazon Translate</title>
  <script src="aws-sdk/dist/aws-sdk.js"></script>
</head>

<body>
  <h1 style="text-align: left">Amazon Translate Demo</h1>
  <br/>
  <table class="tg">
    <tr>
      <th align="left">
Source Language Code:
        <select id="sourceLanguageCodeDropdown">
          <option value="en">en</option>
          <option value="ar">ar</option>
          <option value="cs">cs</option>
          <option value="de">de</option>
          <option value="es">es</option>
          <option value="fr">fr</option>
          <option value="it">it</option>
          <option value="ja">ja</option>
          <option value="pt">pt</option>
          <option value="ru">ru</option>
          <option value="tr">tr</option>
          <option value="zh">zh</option>
          <option value="zh-TW">zh-TW</option>
        </select>
      </th>
      <th align="left">
Target Language Code:
        <select id="targetLanguageCodeDropdown">
          <option value="en">en</option>
          <option value="ar">ar</option>
          <option value="cs">cs</option>
          <option value="de">de</option>
          <option value="es">es</option>
          <option value="fr">fr</option>
          <option value="it">it</option>
          <option value="ja">ja</option>
          <option value="pt">pt</option>
          <option value="ru">ru</option>
          <option value="tr">tr</option>
          <option value="zh">zh</option>
          <option value="zh-TW">zh-TW</option>
        </select>
      </th>
    </tr>
    <tr>
      <th>
        <textarea id="inputText" name="inputText" rows="10" cols="50"
placeholder="Text to translate..."></textarea>
      </th>
      <th>
        <textarea id="outputText" name="outputText" rows="10" cols="50"
placeholder="Translated text..."></textarea>
      </th>
    </tr>
    <tr>
      <th align="left">
        <button type="button" name="translateButton"
onclick="doTranslate()">Translate</button>
        <button type="button" name="synthesizeButton"
onclick="doSynthesizeInput()">Synthesize Input Speech</button>
      </th>
    </tr>
  </table>

```

```

        <button type="button" name="clearButton" onclick="clearInputs()">Clear</
button>
        </th>
        <th align="left">
            <button type="button" name="synthesizeButton"
onclick="doSynthesizeOutput()">Synthesize Output Speech</button>
        </th>
    </tr>
</table>
<script type="text/javascript">
    // set the focus to the input box
    document.getElementById("inputText").focus();

    /**
     * Change the region and endpoint.
     */
    AWS.config.region = 'region'; // Region

    /**
     * In a production application you should use a secure method of authenticating
uses, such as the ones
     * described here:
     *   https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/setting-
credentials-browser.html
     *
     * Note that Amazon Translate does not work with Identidad de Amazon Cognito.
     *
     * For this example you place the credentials of an IAM user in the HTML page. The
IAM user associated
     * with these credentials must have permissions to call Amazon Translate. We
recommend
using the following
     * permissions policy and nothing more, as anyone that has access to this HTML page
will also have access to
     * these hard-coded credentials.
     * {
     *   "Version": "2012-10-17",
     *   "Statement": [
     *     {
     *       "Action": [
     *         "translate:TranslateText",
     *         "polly:SynthesizeSpeech"
     *       ],
     *       "Resource": "*",
     *       "Effect": "Allow"
     *     }
     *   ]
     * }
     *
     * For more information about the AWS Credentials object, see:
     *   http://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/Credentials.html
     */
    AWS.config.credentials = new AWS.Credentials("access key", "secret key");

    var translate = new AWS.Translate({region: AWS.config.region});
    var polly = new AWS.Polly();

    function doTranslate() {
        var inputText = document.getElementById('inputText').value;
        if (!inputText) {
            alert("Input text cannot be empty.");
            exit();
        }

        // get the language codes
        var sourceDropdown = document.getElementById("sourceLanguageCodeDropdown");

```

```
        var sourceLanguageCode =
sourceDropdown.options[sourceDropdown.selectedIndex].text;

        var targetDropdown = document.getElementById("targetLanguageCodeDropdown");
        var targetLanguageCode =
targetDropdown.options[targetDropdown.selectedIndex].text;

        var params = {
            Text: inputText,
            SourceLanguageCode: sourceLanguageCode,
            TargetLanguageCode: targetLanguageCode
        };

        translate.translateText(params, function(err, data) {
            if (err) {
                console.log(err, err.stack);
                alert("Error calling Amazon Translate. " + err.message);
                return;
            }
            if (data) {
                var outputTextArea = document.getElementById('outputText');
                outputTextArea.value = data.TranslatedText;
            }
        });
    }

    function doSynthesizeInput() {
        var text = document.getElementById('inputText').value.trim();
        if (!text) {
            return;
        }
        var sourceLanguageCode =
document.getElementById("sourceLanguageCodeDropdown").value;
        doSynthesize(text, sourceLanguageCode);
    }

    function doSynthesizeOutput() {
        var text = document.getElementById('outputText').value.trim();
        if (!text) {
            return;
        }
        var targetLanguageCode =
document.getElementById("targetLanguageCodeDropdown").value;
        doSynthesize(text, targetLanguageCode);
    }

    function doSynthesize(text, languageCode) {
        var voiceId;
        switch (languageCode) {
            case "de":
                voiceId = "Marlene";
                break;
            case "en":
                voiceId = "Joanna";
                break;
            case "es":
                voiceId = "Penelope";
                break;
            case "fr":
                voiceId = "Celine";
                break;
            case "pt":
                voiceId = "Vitoria";
                break;
            default:
                voiceId = null;
        }
    }
}
```

```
        break;
    }
    if (!voiceId) {
        alert("Speech synthesis unsupported for language code: \"" + languageCode +
            "\"");
        return;
    }
    var params = {
        OutputFormat: "mp3",
        SampleRate: "8000",
        Text: text,
        TextType: "text",
        VoiceId: voiceId
    };
    polly.synthesizeSpeech(params, function(err, data) {
        if (err) {
            console.log(err, err.stack); // an error occurred
            alert("Error calling Amazon Polly. " + err.message);
        }
        else {
            var uInt8Array = new Uint8Array(data.AudioStream);
            var arrayBuffer = uInt8Array.buffer;
            var blob = new Blob([arrayBuffer]);
            var url = URL.createObjectURL(blob);

            audioElement = new Audio([url]);
            audioElement.play();
        }
    });
}

function clearInputs() {
    document.getElementById('inputText').value = "";
    document.getElementById('outputText').value = "";
    document.getElementById("sourceLanguageCodeDropdown").value = "en";
    document.getElementById("targetLanguageCodeDropdown").value = "en";
}
}
</script>
</body>
</html>
```

Uso de Amazon Translate para traducir un canal de chat

Puede utilizar Amazon Translate para traducir mensajes de chat en tiempo real. Este ejemplo utiliza un canal de Twitch, pero puede utilizarlo como punto de partida para otro texto de streaming en tiempo real, como otras plataformas de chat, interacciones de servicio de atención al cliente, foros y mucho más.

Este ejemplo utiliza una página web que muestra mensajes en tiempo real en inglés y sus traducciones en tiempo real al lado. Puede enviar los mensajes a Amazon Polly para dictar el texto. Para seguir a una persona en el chat, escriba su nombre de usuario. La aplicación solo dictará los mensajes de ese usuario.

El código se puede resumir de la forma siguiente:

- CSS y HTML para crear la página web.
- Código de inicialización que crea controladores para Amazon Translate y Amazon Polly.
- Una función de llamada que se ejecuta cuando se recibe un mensaje de chat.
- Una función que envía un mensaje de chat.

- Una función que llama a Amazon Translate para traducir mensajes.
- Una función que llama a Amazon Polly para sintetizar la voz.
- Funciones de utilidad para administrar la página web.

Para configurar el ejemplo

1. Instale y configure AWS SDK for JavaScript. Para obtener instrucciones acerca de cómo instalar SDK for JavaScript, consulte [Instalación del SDK para JavaScript](#).
2. Copie el código del ejemplo en un archivo HTML en el servidor web.
3. Actualice la etiqueta `<script>` con la ubicación en la que ha instalado el SDK for JavaScript.
4. Cambie la región y el punto de enlace a la región en la que desea ejecutar las operaciones de Amazon Translate y Amazon Polly. Para ver una lista de las regiones admitidas para Amazon Translate, consulte [Regiones y puntos de enlace de AWS](#) en Referencia general de AWS.
5. Cree un usuario de IAM con los permisos mínimos necesarios para ejecutar este ejemplo. Para obtener información acerca de cómo crear un usuario de IAM, consulte [Creación de un usuario de IAM en su cuenta de AWS](#) en la Guía del usuario de AWS Identity and Access Management. Para obtener información sobre las políticas de permisos necesarias, consulte [Políticas basadas en identidad de Amazon Translate \(p. 59\)](#) y [Uso de políticas basadas en identidades \(políticas de IAM\) para Amazon Polly](#) en la Guía para desarrolladores de Amazon Polly.
6. Proporcione el ID de acceso y la clave secreta del usuario de IAM que ha creado en el paso anterior.
7. Proporcione un nombre de usuario de Twitch y token de OAuth para su cuenta. Puede crear una cuenta de Twitch en <https://www.twitch.tv>. Puede crear un token de Twitch OAuth en <https://twitchapps.com/tmi>.

```
<!doctype html>
<html lang="en">
<head>
  <title>Amazon Translate</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Latest compiled and minified CSS for Bootstrap -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/
bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/
K68vbdEjh4u" crossorigin="anonymous">

  <!-- Custom CSS -->
  <style>
    .topHeader
    {
      background-color: #6441a4;
      padding: 10px;
      border-bottom: solid 1px #cacaca;
      color: white
    }

    .panelHeading
    {
      background-color: #6441a4 !important;
    }

    .panelBody
    {
      min-height: 450px; max-height: 450px;overflow-y: scroll;
    }

    body{
```

```
        margin-left: 0px;
        margin-right: 0px;
        height: 100%;
    }
</style>
</head>
<body>
    <div class="container-fluid">
        <!--Top Header-->
        <div class="row topHeader">
            <div class="col-md-12">
                <h4>Amazon Translate - Artificial Intelligence on AWS - Powerful machine learning
for all Developers and Data Scientists</h4>
            </div>
        </div>

        <!--Status Label-->
        <div class="row">
            <div class="col-md-12">
                <p class="bg-info">
                    <div id="connecting-div"></div>
                </p>
            </div>
        </div>

        <div class="row" style="padding: 10px;">
            <div class="col-md-6">
                <div class="form-inline">
                    <div class="form-group">
                        <input type="text" id="channel" class="form-control" value=""
placeholder="Channel"/>
                    </div>
                    <div class="form-group">
                        <select id="sourceLanguage" class="form-control">
                            <option value="en">en</option>
                            <option value="ar">ar</option>
                            <option value="de" selected="selected">de</option>
                            <option value="es">es</option>
                            <option value="fr">fr</option>
                            <option value="pt">pt</option>
                            <option value="zh">zh</option>
                        </select>
                    </div>
                    <div class="form-group">
                        <select id="targetLanguage" class="form-control">
                            <option value="en" selected="selected">en</option>
                            <option value="ar">ar</option>
                            <option value="de">de</option>
                            <option value="es">es</option>
                            <option value="fr">fr</option>
                            <option value="pt">pt</option>
                            <option value="zh">zh</option>
                        </select>
                    </div>
                    <div class="form-group">
                        <button type="button" class="form-control" id="btn-go"
onclick="connect()">Go</button>
                        <button type="button" class="form-control" id="btn-stop"
onclick="location.href='index.html';">Stop</button>
                        <span id="status"></span>
                    </div>
                </div>
            </div>
            <div class="col-md-6">
                <div class="form-inline">
                    <div class="form-group">
```

```

        <input type="checkbox" id="cbSpeak" value="Speak"> Speak Live Translation
        <input type="text" id="follow" class="form-control" value=""
placeholder="follow"/>
      </div>
    </div>
  </div>
</div>

<!--Chat Boxes-->
<div class="row">
  <!--Live Chat-->
  <div class="col-md-6">
    <div class="panel panel-primary">
      <div class="panel-heading panelHeading">Live Chat</div>
      <div id="livechatc" class="panel-body panelBody">
        <div class="subscribe" id="livechat"></div>
      </div>
    </div>
  </div>
  <!--Live Chat-->
  <!--Translated Chat-->
  <div class="col-md-6">
    <div class="panel panel-primary">
      <div class="panel-heading panelHeading">Live Translation</div>
      <div id="livetranslationc" class="panel-body panelBody">
        <div class="imageDetected" id="livetranslation"></div>
      </div>
    </div>
  </div>
  <!--Translated Chat-->
</div>

<!--Send Message-->
<div class="row">
  <div class="col-md-11">
    <input type="text" id="message" class="form-control"/>
  </div>
  <div class="col-md-1">
    <button type="button" class="form-control btn btn-default" id="btn-send"
onclick="sendMessage()">Send</button>
  </div>
</div>
</div>

<!-- Latest compiled and minified JavaScript -->
<!-- jQuery first, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnPnCJA712mCWNIPG9mGCD8wGNICPD7Txa"
crossorigin="anonymous"></script>

<script src="aws-js-sdk/dist/aws-sdk-all.js"></script>
<script src="http://cdn.tmijs.org/js/1.2.1/tmi.min.js" integrity="sha384-
eEOn7sm1W7DOUI2Xh5I4qSpZTe6hupAO0ovLfqEy0yVJtGRBNfssdmjbJhEYm6Bw"
crossorigin="anonymous"></script>
<script>
  cred = {
    twitchUsername: "Twitch user name",
    twitchOAuthToken: "Twitch OAuth token",
    awsAccessKeyId: "access key",
    awsSecretAccessKey: "secret key"
  };

  AWS.config.region = 'region';

```



```
ep = new AWS.Endpoint('endpoint');

AWS.config.credentials = new AWS.Credentials(cred.awsAccessKeyId,
cred.awsSecretAccessKey);
window.translator = new AWS.Translate({endpoint: ep, region: AWS.config.region});

/*****Init and Connect to Chat*****/
function connect(){
    init();

    //Twitch Client
    var options = {
        options: {
            debug: false
        },
        connection: {
            cluster: "aws",
            reconnect: true
        },
        identity: {
            username: cred.twitchUsername,
            password: cred.twitchOAuthToken
        },
        channels: [con.channel]
    };

    window.client = tmi.client(options);

    window.client.connect();

    //Attached Handlers
    window.client.on("chat", onChat);
    window.client.on("connecting", onConnecting);
    window.client.on("connected", onConnected);

    //Disable UI Elements
    document.getElementById("sourceLanguage").disabled = true;
    document.getElementById("targetLanguage").disabled = true;
    document.getElementById("channel").disabled = true;
    document.getElementById("btn-go").disabled = true;
}

function init(){
    //Get UI Controls
    var lc = document.getElementById("livechat");
    var lt = document.getElementById("livetranslation")
    var lcc = document.getElementById("livechatc");
    var ltc = document.getElementById("livetranslationc")
    var cbspeak = document.getElementById("cbSpeak")
    var follow = document.getElementById("follow");
    var sendMessage = document.getElementById("message");

    //Cache values
    con = {
        channel: document.getElementById("channel").value,
        sourceLanguage: document.getElementById("sourceLanguage").value,
        targetLanguage: document.getElementById("targetLanguage").value,
        liveChatUI: lc,
        liveTranslationUI: lt,
        liveChatUIContainer: lcc,
        liveTranslationUIContainer: ltc,
        cbSpeak: cbspeak,
        follow: follow,
        sendMessage: sendMessage
    }
}
```

```
lc.innerHTML = '';
lt.innerHTML = '';

//Speaker
var voiceId = "Joanna";
if(con.targetLanguage == "en")
    voiceId = "Joanna";
else if(con.targetLanguage == "de")
    voiceId = "Marlene";
else if(con.targetLanguage == "es")
    voiceId = "Conchita";
else if(con.targetLanguage == "fr")
    voiceId = "Celine";
else if(con.targetLanguage == "pt")
    voiceId = "Ines";
else
    voiceId = "Joanna";
window.audioPlayer = AudioPlayer(voiceId);
}
/*****Init and Connect to Chat*****/

/*****Receive and Translate Chat*****/
function onChat (channel, userstate, message, self) {
    // Don't listen to my own messages..
    if (self) return;

    //Translate
    if (message) {
        var username = userstate['username'];

        var params = {
            Text: message,
            SourceLanguageCode: con.sourceLanguage,
            TargetLanguageCode: con.targetLanguage
        };

        window.translator.translateText(params, function onIncomingMessageTranslate(err,
data) {
            if (err) {
                console.log("Error calling Translate. " + err.message + err.stack);
            }
            if (data) {
                console.log("M: " + message);
                console.log("T: " + data.TranslatedText);

                //Print original message in chat UI
                con.liveChatUI.innerHTML += '<strong>' + username + '</strong>: ' +
message + '<br>';

                //Print translation in translation UI
                con.liveTranslationUI.innerHTML += '<strong>' + username + '</strong>: '
+ data.TranslatedText + '<br>';

                //If speak translation is enabled, speak translated message
                if(con.cbSpeak.checked){
                    if(con.follow.value == "" || username == con.follow.value)
                        audioPlayer.Speak(username + " says " + data.TranslatedText);
                }

                //Scroll chat and translated UI to bottom to keep focus on latest
messages
                con.liveChatUIContainer.scrollTop = con.liveChatUIContainer.scrollHeight;
                con.liveTranslationUIContainer.scrollTop =
con.liveTranslationUIContainer.scrollHeight;
            }
        });
    }
};
```

```

    }
  }
  /*****Receive and Translate Chat*****/

  /*****Client Connecting*****/
  function onConnecting (address, port) {
    document.getElementById("status").innerHTML = " [ Connecting...]"
  }

  function onConnected (address, port) {
    document.getElementById("status").innerHTML = " [ Connected ]"
    window.audioPlayer.Speak("Connected to channel " + con.channel + ". You should now
be getting live chat messages.");
  }
  /*****Client Connecting*****/

  /*****Send Message*****/
  function sendMessage(){
    if(con.sendMessage.value){
      message = con.sendMessage.value;
      var params = {
        Text: con.sendMessage.value,
        SourceLanguageCode: con.targetLanguage,
        TargetLanguageCode: con.sourceLanguage
      };

      window.translator.translateText(params, function onSendMessageTranslate(err,
data) {
        if (err) {
          console.log("Error calling Translate. " + err.message + err.stack);
        }
        if (data) {
          console.log("M: " + message);
          console.log("T: " + data.TranslatedText);

          //Send message to chat
          window.client.action(con.channel, data.TranslatedText);

          //Clear send message UI
          con.sendMessage.value = "";

          //Print original message in Translated UI
          con.liveTranslationUI.innerHTML += '<strong> ME: </strong>: ' +
message + '<br>';

          //Print translated message in Chat UI
          con.liveChatUI.innerHTML += '<strong> ME: </strong>: ' +
data.TranslatedText + '<br>';

          //Scroll chat and translated UI to bottom to keep focus on latest
messages
          con.liveChatUIContainer.scrollTop =
con.liveChatUIContainer.scrollHeight;
          con.liveTranslationUIContainer.scrollTop =
con.liveTranslationUIContainer.scrollHeight;
        }
      });
    }
  }
  /*****Send Message*****/

  /*****Audio player*****/
  function AudioPlayer(voiceId) {
    var audioPlayer = document.createElement('audio');
    audioPlayer.setAttribute("id", "audioPlayer");
    document.body.appendChild(audioPlayer);
  }

```

```
var isSpeaking = false;

var speaker = {
  self: this,
  playlist:[],

  Speak: function (text) {
    //If currently speaking a message, add new message to the playlist
    if (isSpeaking) {
      this.playlist.push(text);
    } else {
      speakTextMessage(text).then(speakNextTextMessage)
    }
  }
}

// Speak text message
function speakTextMessage(text) {
  return new Promise(function (resolve, reject) {
    isSpeaking = true;
    getAudioStream(text).then(playAudioStream).then(resolve);
  });
}

// Speak next message in the list
function speakNextTextMessage() {
  var pl = speaker.playlist;
  if (pl.length > 0) {
    var txt = pl[0];
    pl.splice(0, 1);
    speakTextMessage(txt).then(speakNextTextMessage);
  }
}

// Get synthesized speech from Amazon polly
function getAudioStream(textMessage) {
  return new Promise(function (resolve, reject) {
    var polly = new AWS.Polly();
    var params = {
      OutputFormat: 'mp3',
      Text: textMessage,
      VoiceId: voiceId
    }
    polly.synthesizeSpeech(params, function (err, data) {
      if (err)
        reject(err);
      else
        resolve(data.AudioStream);
    });
  });
}

// Play audio stream
function playAudioStream(audioStream) {
  return new Promise(function (resolve, reject) {
    var uint8Array = new Uint8Array(audioStream);
    var arrayBuffer = uint8Array.buffer;
    var blob = new Blob([arrayBuffer]);

    var url = URL.createObjectURL(blob);
    audioPlayer.src = url;
    audioPlayer.addEventListener("ended", function () {
      isSpeaking = false;
      resolve();
    });
  });
}
```

```
        audioPlayer.play();
    });
}

return speaker;
}
/*****Audio player*****/
</script>
</body>
</html>
```

Uso de Amazon Translate con Amazon DynamoDB

En este ejemplo, se explica cómo se traduce una opinión sobre un producto y se almacena en Amazon DynamoDB. Si solicita la misma opinión más tarde, DynamoDB la devolverá sin necesidad de que Amazon Translate la traduzca de nuevo.

En este ejemplo:

- Utilice AWS CloudFormation si desea crear tablas de DynamoDB para almacenar la traducción y una función Lambda que llama a la operación [TranslateText](#) (p. 105).
- Pruebe la función utilizando la consola de AWS Lambda.

Para ejecutar el ejemplo

1. Copie el contenido de `example.py`, que puede encontrar en [Función Lambda de Python](#) (p. 42), en un archivo denominado `example.py`. `example.py` es una función Lambda que llama a la operación [TranslateText](#) (p. 105). Comprima el archivo en un archivo zip denominado `example.zip`. Almacénelo en un bucket de S3 en la misma región de AWS donde desea ejecutar la función.
2. Cree un nuevo archivo llamado `template.yaml`. Copie el código de plantilla de AWS CloudFormation, que encontrará en [Plantilla de AWS CloudFormation](#) (p. 44), en el archivo. AWS CloudFormation utiliza la plantilla para crear recursos para la aplicación de ejemplo. Cambie `BUCKET_NAME` el nombre del bucket de S3 que contiene `example.zip`. Guarde el archivo en un directorio local.
3. Inicie sesión en la Consola de administración de AWS y abra la consola de AWS CloudFormation en <https://console.aws.amazon.com/cloudformation>.
4. Elija Create new stack.
5. Elija Upload a template to Amazon S3 y, a continuación, elija Choose file. Elija `template.yaml`, que ha creado en el Paso 2, a continuación Next.
6. Escriba un nombre para la pila y, a continuación, elija Next.
7. En la página Options, seleccione Next.
8. Elija I acknowledge that AWS CloudFormation might create IAM resources (Confirmando que AWS CloudFormation puede crear recursos de IAM) y I acknowledge that AWS CloudFormation might create IAM resources with custom names (Confirmando que AWS CloudFormation puede crear recursos de IAM con nombres personalizados). Para obtener más información, consulte [Control del acceso con AWS Identity and Access Management](#) en la Guía del usuario de AWS CloudFormation.
9. Elija Create Change Set.
10. Después de que AWS CloudFormation cree el conjunto de cambios, elija Execute (Ejecutar). Espere a que AWS CloudFormation cree la pila.
11. Inicie sesión en la Consola de administración de AWS y abra la consola de AWS Lambda en <https://console.aws.amazon.com/lambda/>.

12. Elija la nueva función. Su nombre comienza por `TestTranslate-ReviewTranslate`.
13. En la página de detalle de la función, elija `Test`.
14. En `Event Name` (Nombre del evento), escriba **`TestTranslate`**. Para `Configure test event`, reemplace el JSON por lo siguiente:

```
{
  "review": "hello world",
  "target_language": "es",
  "source_language": "en",
  "review_id": "1"
}
```

Seleccione `Create`.

15. Asegúrese de que `TestTranslate` esté seleccionado, a continuación elija `Test`. Cuando la prueba finaliza, recibe el siguiente mensaje:

Execution result: succeeded (logs)

▼ Details

The area below shows the result returned by your function execution. [Learn more](#) about returning r

```
"¡Hola mundo!"
```

Código de ejemplo

Utilice el siguiente código para crear el ejemplo.

Función Lambda de Python

El contenido de la función Lambda de Python es el siguiente. La función Lambda llama a la operación `TranslateText` y pasa la opinión, el idioma de origen y el idioma de destino para obtener la traducción de la opinión. Guarde este archivo como `example.py` y comprímalo en un archivo `.zip` denominado `example.zip`. Guarde el archivo en un bucket de S3 en la misma región en la que se ejecuta el ejemplo.

```
import logging
import json
import boto3
import os

translate = boto3.client('translate')
dynamodb = boto3.client('dynamodb')
firehose = boto3.client('firehose')

TABLE_NAME = os.getenv('TABLE_NAME')

logger = logging.getLogger()
logger.setLevel(logging.INFO)

def lambda_handler(event, context):

    logger.info(event)
```

```
    if 'source_language' in event and 'target_language' in event and 'review' in event and
    'review_id' in event:
        review_id = event['review_id']
        source_language = event['source_language']
        target_language = event['target_language']
        review = event['review']

    try:
        # The Lambda function queries the Amazon DynamoDB table to check whether
        # the review has already been translated. If the translated review
        # is already stored in Amazon DynamoDB, the function returns it.
        response = dynamodb.get_item(
            TableName=TABLE_NAME,
            Key={
                'review_id': {
                    'N': review_id,
                },
                'language': {
                    'S': target_language,
                },
            }
        )
        logger.info(response)
        if 'Item' in response:
            return response['Item']['review']['S']
    except Exception as e:
        logger.error(response)
        raise Exception("[ErrorMessage]: " + str(e))

    try:
        # The Lambda function calls the TranslateText operation and passes the
        # review, the source language, and the target language to get the
        # translated review.
        result = translate.translate_text(Text=review,
SourceLanguageCode=source_language, TargetLanguageCode=target_language)
        logging.info("Translation output: " + str(result))
    except Exception as e:
        logger.error(response)
        raise Exception("[ErrorMessage]: " + str(e))

    try:
        # After the review is translated, the function stores it using
        # the Amazon DynamoDB putItem operation. Subsequent requests
        # for this translated review are returned from Amazon DynamoDB.
        response = dynamodb.put_item(
            TableName=TABLE_NAME,
            Item={
                'review_id': {
                    'N': review_id,
                },
                'language': {
                    'S': target_language,
                },
                'review': {
                    'S': result.get('TranslatedText')
                }
            }
        )
        logger.info(response)
    except Exception as e:
        logger.error(e)
        raise Exception("[ErrorMessage]: " + str(e))
    return result.get('TranslatedText')
else:
    logger.error(e)
```

```
raise Exception("[ErrorMessage]: Invalid input ")
```

Plantilla de AWS CloudFormation

A continuación, se presenta el archivo de plantilla que utiliza con AWS CloudFormation para crear y configurar la función Lambda y las tablas de DynamoDB. Utilice este archivo al crear la pila de AWS CloudFormation del ejemplo. Actualice `BUCKET_NAME` para el nombre del bucket de S3 que contiene el archivo `example.zip` y guárdelo en un directorio local como `template.yaml`.

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: 'AWS::Serverless-2016-10-31'
Resources: ReviewTranslate:
  Type: 'AWS::Serverless::Function'
  Properties:
    Handler: example.lambda_handler
    Runtime: python2.7
    CodeUri:
      Bucket: BUCKET_NAME
      Key: example.zip
    Policies:
      - AWSLambdaFullAccess
      - TranslateReadOnly
    Environment:
      Variables:
        TABLE_NAME: !Ref ReviewTable
    Tracing: "Active"
  ReviewTable:
    Type: 'AWS::DynamoDB::Table'
    Properties:
      AttributeDefinitions:
        - AttributeName: "review_id"
          AttributeType: "N"
        - AttributeName: "language"
          AttributeType: "S"
      KeySchema:
        - AttributeName: "review_id"
          KeyType: "HASH"
        - AttributeName: "language"
          KeyType: "RANGE"
      ProvisionedThroughput:
        ReadCapacityUnits: 5
        WriteCapacityUnits: 5
```

Uso de Amazon Translate para traducir una página web

Puede utilizar Amazon Translate para traducir el contenido de una página web. El siguiente programa de Java traduce una página web especificada del inglés al español y crea un archivo HTML que contiene el resultado de la traducción. Hay dos funciones en el programa:

- Una función que lee los datos de la página web de origen, los separa en elementos HTML y, a continuación, llama a la segunda función para traducirlos. Al final del documento, escribe los resultados en un archivo HTML.
- Una función que llama al servicio de Amazon Translate para traducir el contenido de un elemento HTML.

Este ejemplo funciona en páginas HTML sencillas sin elementos anidados.

Para configurar el ejemplo

1. Instale y configure AWS SDK for Java. Para obtener instrucciones acerca de la instalación de SDK para Java, consulte [Configuración de AWS SDK for Java](#).
2. Instale el analizador HTML de Java jsoup. Para obtener instrucciones, consulte [jsoup](#).
3. Cree un usuario de IAM con los permisos mínimos necesarios para ejecutar este ejemplo. Para obtener información acerca de cómo crear un usuario de IAM, consulte [Creación de un usuario de IAM en su cuenta de AWS](#) en la Guía del usuario de AWS Identity and Access Management. Para conocer las políticas de permisos necesarias, consulte [Políticas basadas en identidad de Amazon Translate \(p. 59\)](#).
4. Configure las credenciales necesarias para ejecutar la muestra. Para obtener instrucciones, consulte [Configuración de credenciales y regiones de AWS](#) en la Guía para desarrolladores de AWS SDK for Java.
5. Cree un nuevo proyecto en el IDE de Java y copie el código fuente.
6. Cambie la región y el punto de enlace a la región en la que desea ejecutar la operación de Amazon Translate. Para ver una lista de las regiones admitidas para Amazon Translate, consulte [Regiones y puntos de enlace de AWS](#) en Referencia general de AWS.

```
package com.amazonaws.translateweb;

import com.amazonaws.auth.AWSCredentialsProviderChain;
import com.amazonaws.auth.EnvironmentVariableCredentialsProvider;
import com.amazonaws.auth.SystemPropertiesCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.client.builder.AwsClientBuilder;
import com.amazonaws.services.translate.AmazonTranslate;
import com.amazonaws.services.translate.AmazonTranslateClient;
import com.amazonaws.services.translate.model.TranslateTextRequest;
import com.amazonaws.services.translate.model.TranslateTextResult;
import com.amazonaws.AmazonServiceException;

import java.io.IOException;
import java.io.PrintWriter;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;

import org.jsoup.select.Elements;

public class TranslateWebPage {

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

        // Define the URL of the HTML content to translate
        String url = "http://example.com/source.html";

        // Create credentials using a provider chain that will evaluate in order;
        // a) Any Java system properties
        // b) Any environment variables
        // c) Any profile file
        AWSCredentialsProviderChain DefaultAWSCredentialsProviderChain = new
        AWSCredentialsProviderChain(
            new SystemPropertiesCredentialsProvider(),
            new EnvironmentVariableCredentialsProvider(),
            new ProfileCredentialsProvider()
        );

        // Create an endpoint configuration for the Translate service
```

```
AwsClientBuilder.EndpointConfiguration endpointConfiguration = new
AwsClientBuilder.EndpointConfiguration(
    "endpoint", "region");

// Create a client for the Translate service
AmazonTranslate translate = AmazonTranslateClient.builder()
    .withCredentials(DefaultAWSCredentialsProviderChain)
    .withEndpointConfiguration(endpointConfiguration).build();

// Record the beginning of translating the HTML content at the url
System.out.println("Translating URL: " + url);

// Create an empty HTML document to store the parsed data
Document doc;

try {
    // Retrieve the HTML located at the URL
    doc = Jsoup.connect(url).get();

    // Select all of the elements in the HTML
    Elements eles = doc.select("*");

    // For each element
    for (Element ele : eles) {

        // Translate the element
        translateElement(ele, translate);

        // If you encounter service throttling when translating large web
        // pages, you can request a service limit increase. For details,
        // see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-
limits/,
        // or you can throttle your requests by inserting a sleep statement.
        // Thread.sleep(1000);
    }

    // Configure an output file for the translated HTML
    String fname = "output HTML file name";
    PrintWriter pw = new PrintWriter(fname, "UTF-8");

    // Write our translated HTML to the output file
    pw.println(doc);
    pw.close();

    // Record that the file has been saved
    System.out.println("Saved file "+fname);

    // Catch any exceptions in retrieving the HTML
} catch (IOException e1) {
    e1.printStackTrace();
}

// This function is used to translate each individual element
public static void translateElement(Element ele, AmazonTranslate translate) {

    // Check if the element has any text
    if (!ele.ownText().isEmpty()) {

        // Retrieve the text of the HTML element
        String text = ele.ownText();

        // Now translate the element's text
        try {
```

```
// Translate from English to Spanish
TranslateTextRequest request = new TranslateTextRequest()
    .withText(text)
    .withSourceLanguageCode("en")
    .withTargetLanguageCode("es");

// Retrieve the result
TranslateTextResult result = translate.translateText(request);

// Record the original and translated text
System.out.println("Original text: " + text + " - Translated text: " +
result.getTranslatedText());

// Update the HTML element with the translated text
ele.text(result.getTranslatedText());

// Catch any translation errors
} catch (AmazonServiceException e) {
    System.err.println(e.getErrorMessage());
    System.exit(1);
}
} else {
    // We have found a non-text HTML element. No action required.
}
}
}
```

Uso de Amazon Translate para traducir documentos grandes

Puede dividir los documentos de gran tamaño en partes más pequeñas que no sobrepasen el límite de tamaño de los documentos. Para obtener más información sobre el límite de tamaño de los documentos, consulte [Límites de los servicios \(p. 75\)](#). El siguiente programa de Java divide documentos de texto grandes en frases individuales y, a continuación, traduce cada frase del idioma de origen al idioma de destino. El programa contiene dos secciones:

- La clase `SentenceSegmenter`, que se encarga de dividir la cadena de origen en frases individuales. La muestra usa la clase `BreakIterator` de Java.
- La función `main`, que llama a la operación `Translate` para cada frase de la cadena de origen. La función `main` también se encarga de la autenticación con Amazon Translate.

Para configurar el ejemplo

1. Instale y configure AWS SDK for Java. Para obtener instrucciones acerca de la instalación de SDK para Java, consulte [Configuración de AWS SDK for Java](#).
2. Cree un usuario de IAM con los permisos mínimos necesarios para ejecutar este ejemplo. Para obtener información acerca de cómo crear un usuario de IAM, consulte [Creación de un usuario de IAM en su cuenta de AWS](#) en la Guía del usuario de AWS Identity and Access Management. Para conocer las políticas de permisos necesarias, consulte [Políticas basadas en identidad de Amazon Translate \(p. 59\)](#).
3. Configure las credenciales necesarias para ejecutar la muestra. Para obtener instrucciones, consulte [Configuración de credenciales y regiones de AWS](#) en la Guía para desarrolladores de AWS SDK for Java.

4. Cree un nuevo proyecto en el IDE de Java y copie el código fuente.
5. Cambie a la región en la que desea ejecutar la operación de Amazon Translate. Para ver una lista de las regiones admitidas para Amazon Translate, consulte [Regiones y puntos de enlace de AWS](#) en Referencia general de AWS.
6. Cambie los idiomas de origen y de destino a los idiomas entre los que va a traducir.
7. Ejecute la muestra para ver el texto traducido en salida estándar.

```
import com.amazonaws.auth.AWSCredentialsProviderChain;
import com.amazonaws.auth.EnvironmentVariableCredentialsProvider;
import com.amazonaws.auth.SystemPropertiesCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.translate.AmazonTranslate;
import com.amazonaws.services.translate.AmazonTranslateClient;
import com.amazonaws.services.translate.model.TranslateTextRequest;
import com.amazonaws.services.translate.model.TranslateTextResult;
import java.text.BreakIterator;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

public class MultiSentenceTranslator {

    public static void main(String[] args) {
        // Define the text to be translated here
        String region = "region";
        String text = "Text to be translated";

        String sourceLang = "source language";
        String targetLang = "target language";

        // Break text into sentences
        SentenceSegmenter sentenceSegmenter = new SentenceSegmenter();
        List<String> sentences = new ArrayList<>();
        try {
            sentences = sentenceSegmenter.segment(text, sourceLang);
        } catch (Exception e) {
            System.out.println(e);
            System.exit(1);
        }

        // Create credentials using a provider chain that will evaluate in order;
        // a) Any Java system properties
        // b) Any environment variables
        // c) Any profile file
        AWSCredentialsProviderChain DefaultAWSCredentialsProviderChain = new
        AWSCredentialsProviderChain(
            new SystemPropertiesCredentialsProvider(),
            new EnvironmentVariableCredentialsProvider(),
            new ProfileCredentialsProvider()
        );

        // Create an Amazon Translate client
        AmazonTranslate translate = AmazonTranslateClient.builder()
            .withCredentials(DefaultAWSCredentialsProviderChain)
            .withRegion(region)
            .build();

        // Translate sentences and print the results to stdout
        for (String sentence : sentences) {
            TranslateTextRequest request = new TranslateTextRequest()
                .withText(sentence)
                .withSourceLanguageCode(sourceLang)
```

```
        .withTargetLanguageCode(targetLang);
        TranslateTextResult result = translate.translateText(request);
        System.out.println("Original text: " + sentence);
        System.out.println("Translated text: " + result.getTranslatedText());
    }
}

}

class SentenceSegmenter {
    public List<String> segment(final String text, final String lang) throws Exception {
        List<String> res = new ArrayList<>();
        BreakIterator sentenceIterator = BreakIterator.getSentenceInstance(new
        Locale(lang));
        sentenceIterator.setText(text);
        int prevBoundary = sentenceIterator.first();
        int curBoundary = sentenceIterator.next();
        while (curBoundary != BreakIterator.DONE) {
            String sentence = text.substring(prevBoundary, curBoundary);
            res.add(sentence);
            prevBoundary = curBoundary;
            curBoundary = sentenceIterator.next();
        }
        return res;
    }
}
```

Uso de Signature Version 4 con Amazon Translate

En este programa en Python de ejemplo, se muestra cómo se utiliza Signature Version 4 para agregar información de autenticación a las solicitudes de Amazon Translate. El ejemplo realiza una solicitud POST, crea una estructura JSON que contiene el texto que se traducirá en el cuerpo (carga) de la solicitud y pasa la información de autenticación en un encabezado Authorization. Para obtener más información sobre el uso de Signature Version 4, consulte [Proceso de firma de Signature Version 4](#) en la Referencia general de Amazon Web Services.

Configuración

Para ejecutar el ejemplo, realice los siguientes pasos:

1. Instale la AWS Command Line Interface (AWS CLI). La AWS SDK for Python (Boto) está incluida en la instalación de AWS CLI. Para obtener instrucciones, consulte [Paso 2: Configuración de la AWS Command Line Interface \(AWS CLI\)](#) (p. 8).
2. Cree un usuario de AWS Identity and Access Management (IAM) con la política de permisos mínimos necesarios para ejecutar este ejemplo. Para obtener información acerca de cómo crear un usuario de IAM, consulte [Creación de un usuario de IAM en su cuenta de AWS](#) en la Guía del usuario de AWS Identity and Access Management. Para conocer las políticas de permisos necesarias, consulte [Políticas basadas en identidad de Amazon Translate](#) (p. 59). Registre el ID de clave de acceso de usuario y la clave de acceso secreta.
3. Guarde el ID de clave de acceso y la clave de acceso secreta en las variables de entorno denominadas `AWS_ACCESS_KEY` and `AWS_SECRET_ACCESS_KEY`, respectivamente. Como práctica recomendada, es conveniente que no incluya las credenciales en el código.
4. Cree un nuevo archivo en el equipo, copie el código del ejemplo (que puede encontrar en la siguiente sección), péguelo en el archivo y guarde el archivo con la extensión `.py`.

5. En el código, reemplace `region` por el nombre de la región de AWS en la que desea ejecutar la operación `TranslateText` de Amazon Translate. Para ver una lista de las regiones admitidas, consulte [Regiones y puntos de enlace de AWS](#) en Referencia general de AWS.

Code

A continuación se incluye el código completo del programa en Python de ejemplo.

Después de crear los valores de solicitud, como la URL del punto de enlace y el cuerpo de la solicitud, el código hace lo siguiente:

1. Cree una solicitud canónica para la operación `TranslateText` de Amazon Translate.
2. Crea la cadena a la que aplicará la función hash para crear la firma.
3. Calcule la firma.
4. Agrega la firma al encabezado de solicitud.
5. Envía la solicitud a la operación `TranslateText`.

Para ejecutar el ejemplo en su equipo, copie el código en un archivo Python.

```
# AWS Version 4 signing example

# Translate API (TranslateText)

# For more information about using Signature Version 4, see http://docs.aws.amazon.com/
# general/latest/gr/sigv4_signing.html.
# This example makes a POST request to Amazon Translate and
# passes the text to translate JSON in the body (payload)
# of the request. Authentication information is passed in an
# Authorization header.
import sys, os, base64, datetime, hashlib, hmac
import requests # pip install requests

# ***** REQUEST VALUES *****
method = 'POST'
service = 'translate'
region = 'region'
host = service + '.' + region + '.amazonaws.com'
endpoint = 'https://' + host + '/'

# POST requests use a content type header. For Amazon Translate,
# the content is JSON.
content_type = 'application/x-amz-json-1.1'
# Amazon Translate requires an x-amz-target header that has this format:
# AWSShineFrontendService_20170701.<operationName>.
amz_target = 'AWSShineFrontendService_20170701.TranslateText'

# Pass request parameters for the TranslateText operation in a JSON block.
request_parameters = '{'
request_parameters += '"Text": "Hello world.",'
request_parameters += '"SourceLanguageCode": "en",'
request_parameters += '"TargetLanguageCode": "de"'
request_parameters += '}'

# The following functions derive keys for the request. For more information, see
# http://docs.aws.amazon.com/general/latest/gr/signature-v4-examples.html#signature-v4-
# examples-python.
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
```

```
def getSignatureKey(key, date_stamp, regionName, serviceName):
    kDate = sign(('AWS4' + key).encode('utf-8'), date_stamp)
    kRegion = sign(kDate, regionName)
    kService = sign(kRegion, serviceName)
    kSigning = sign(kService, 'aws4_request')
    return kSigning

# Python can read the AWS access key from environment variables or the configuration file.
# In this example, keys are stored in environment variables. As a best practice, do not
# embed credentials in code.
access_key = os.environ.get('AWS_ACCESS_KEY_ID')
secret_key = os.environ.get('AWS_SECRET_ACCESS_KEY')
if access_key is None or secret_key is None:
    print 'No access key is available.'
    sys.exit()

# Create a timestamp for headers and the credential string.
t = datetime.datetime.utcnow()
amz_date = t.strftime('%Y%m%dT%H%M%SZ')
date_stamp = t.strftime('%Y%m%d') # The date without time is used in the credential scope.

# ***** TASK 1: CREATE A CANONICAL REQUEST *****
# For information about creating a canonical request, see http://docs.aws.amazon.com/general/latest/gr/sigv4-create-canonical-request.html.

# Step 1: Define the verb (GET, POST, etc.), which you have already done.

# Step 2: Create a canonical URI. A canonical URI is the part of the URI from domain to
# query.
# string (use '/' if no path)
canonical_uri = '/'

## Step 3: Create the canonical query string. In this example, request
# parameters are passed in the body of the request and the query string
# is blank.
canonical_querystring = ''

# Step 4: Create the canonical headers. Header names must be trimmed,
# lowercase, and sorted in code point order from low to high.
# Note the trailing \n.
canonical_headers = 'content-type:' + content_type + '\n' + 'host:' + host + '\n' + 'x-amz-
date:' + amz_date + '\n' + 'x-amz-target:' + amz_target + '\n'

# Step 5: Create the list of signed headers by listing the headers
# in the canonical_headers list, delimited with ";" and in alphabetical order.
# Note: The request can include any headers. Canonical_headers and
# signed_headers should contain headers to include in the hash of the
# request. "Host" and "x-amz-date" headers are always required.
# For Amazon Translate, content-type and x-amz-target are also required.
signed_headers = 'content-type;host;x-amz-date;x-amz-target'

# Step 6: Create the payload hash. In this example, the request_parameters
# variable contains the JSON request parameters.
payload_hash = hashlib.sha256(request_parameters).hexdigest()

# Step 7: Combine the elements to create a canonical request.
canonical_request = method + '\n' + canonical_uri + '\n' + canonical_querystring + '\n' +
    canonical_headers + '\n' + signed_headers + '\n' + payload_hash

# ***** TASK 2: CREATE THE STRING TO SIGN*****
# Set the algorithm variable to match the hashing algorithm that you use, either SHA-256
# (recommended) or SHA-1.
#
algorithm = 'AWS4-HMAC-SHA256'
```

```
credential_scope = date_stamp + '/' + region + '/' + service + '/' + 'aws4_request'
string_to_sign = algorithm + '\n' + amz_date + '\n' + credential_scope + '\n' +
    hashlib.sha256(canonical_request).hexdigest()

# ***** TASK 3: CALCULATE THE SIGNATURE *****
# Create the signing key using the getSignatureKey function defined above.
signing_key = getSignatureKey(secret_key, date_stamp, region, service)

# Sign the string_to_sign using the signing_key.
signature = hmac.new(signing_key, (string_to_sign).encode('utf-8'),
    hashlib.sha256).hexdigest()

# ***** TASK 4: ADD SIGNING INFORMATION TO THE REQUEST *****
# Put the signature information in a header named Authorization.
authorization_header = algorithm + ' ' + 'Credential=' + access_key + '/' +
    credential_scope + ', ' + 'SignedHeaders=' + signed_headers + ', ' + 'Signature=' +
    signature

# For Amazon Translate, the request can include any headers, but it must include "host,"
# "x-amz-date,"
# "x-amz-target," "content-type," and "Authorization" headers. Except for the authorization
# header, the headers must be included in the canonical_headers and signed_headers values,
# as
# noted earlier. Header order is not significant.
# Note: The Python 'requests' library automatically adds the 'host' header.
headers = {'Content-Type':content_type,
    'X-Amz-Date':amz_date,
    'X-Amz-Target':amz_target,
    'Authorization':authorization_header}

# ***** TASK 5: SEND THE REQUEST *****
print 'Request:\n\t' + request_parameters

response = requests.post(endpoint, data=request_parameters, headers=headers)
print 'Response:\n\t' + response.text
```


Seguridad en Amazon Translate

La seguridad en la nube de AWS es la mayor prioridad. Como cliente de AWS, se beneficiará de una arquitectura de red y un centro de datos diseñados para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

La seguridad es una responsabilidad compartida entre AWS y usted. El [modelo de responsabilidad compartida](#) la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube – AWS es responsable de proteger la infraestructura que ejecuta servicios de AWS en la nube de AWS. AWS también proporciona servicios que puede utilizar de forma segura. Auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los [programas de conformidad de AWS](#). Para obtener más información acerca de los programas de conformidad que se aplican a Amazon Translate, consulte [Servicios de AWS en el ámbito del programa de conformidad](#).
- Seguridad en la nube: su responsabilidad viene determinada por el servicio de AWS que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables.

Este tema le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS. En los siguientes temas, se le mostrará cómo configurar AWS para satisfacer sus objetivos de seguridad y conformidad. También puede aprender a utilizar otros servicios de AWS que le ayudan a monitorizar y proteger sus recursos de AWS.

Temas

- [Protección de los datos en Amazon Translate \(p. 53\)](#)
- [Administración de identidades y accesos en Amazon Translate \(p. 54\)](#)
- [Monitoreo de Amazon Translate \(p. 67\)](#)
- [Validación de la conformidad en Amazon Translate \(p. 73\)](#)
- [Resiliencia de Amazon Translate \(p. 74\)](#)
- [Seguridad de la infraestructura en Amazon Translate \(p. 74\)](#)

Protección de los datos en Amazon Translate

Amazon Translate cumple los requisitos del [modelo de responsabilidad compartida](#) de AWS, que incluye normas y directrices para la protección de los datos. AWS es responsable de proteger la infraestructura global que ejecuta todos los servicios de AWS. AWS mantiene el control de los datos alojados en esta infraestructura, incluidos los controles de configuración de la seguridad para el tratamiento del contenido y los datos personales de los clientes. Los clientes de AWS y los socios de APN, que actúan como controladores o procesadores de datos, son responsables de todos los datos personales que guardan en la nube de AWS.

En aras de la protección de datos, le recomendamos que proteja las credenciales de la cuenta de AWS y configure cuentas de usuario individuales con AWS Identity and Access Management (IAM), de modo que a cada usuario se le concedan únicamente los permisos necesarios para llevar a cabo su trabajo. También le recomendamos proteger sus datos de las siguientes formas:

- Utilice la autenticación multifactor (MFA) con cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos de AWS.
- Configure la API y el registro de actividad del usuario con AWS CloudTrail.
- Utilice las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los servicios de AWS.
- Utilice servicios de seguridad administrados avanzados como Amazon Macie, que le ayudará a detectar y proteger los datos personales almacenados en Amazon Simple Storage Service (Amazon S3).

Le recomendamos encarecidamente que nunca introduzca información de identificación confidencial, como los números de cuenta de los clientes, en los campos de formato libre; por ejemplo, en el campo Name (Nombre). No debe introducir esta información cuando trabaje con Amazon Translate u otros servicios de AWS a través de la consola, la API, la AWS CLI o los SDK de AWS. Cualquier dato que escriba en Amazon Translate o en otros servicios se puede incluir en los registros de diagnóstico. Cuando proporcione una URL a un servidor externo, no incluya información de credenciales en la URL para validar la solicitud para ese servidor.

Para obtener más información sobre la protección de datos, consulte la entrada relativa al [modelo de responsabilidad compartida de AWS y GDPR](#) del blog de seguridad de AWS.

Temas

- [Cifrado en reposo](#) (p. 54)
- [Cifrado en tránsito](#) (p. 54)
- [Administración de claves](#) (p. 54)

Cifrado en reposo

Amazon Translate utiliza la clave predeterminada de Amazon S3 (SSE-S3) para el cifrado del lado del servidor de las traducciones almacenadas en el bucket de S3.

Amazon Translate utiliza un volumen de Amazon Elastic Block Store (Amazon EBS) cifrado con la clave predeterminada.

Cifrado en tránsito

Para cifrar datos en tránsito, Amazon Translate utiliza TLS 1.2 con certificados de AWS.

Administración de claves

AWS administra la clave de Amazon S3 predeterminada. Es responsabilidad del cliente administrar todas las claves de [AWS Key Management Service \(AWS KMS\)](#) proporcionadas por el cliente.

Administración de identidades y accesos en Amazon Translate

AWS Identity and Access Management (IAM) es un servicio de AWS que ayuda a un administrador a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién puede ser autenticado (iniciar sesión) y estar autorizado (tener permisos) para utilizar los recursos de Amazon Translate. IAM es un servicio de AWS que se puede utilizar sin costo adicional.

Temas

- [Público \(p. 55\)](#)
- [Autenticación con identidades \(p. 55\)](#)
- [Administración de acceso mediante políticas \(p. 57\)](#)
- [Funcionamiento de Amazon Translate con IAM \(p. 59\)](#)
- [Ejemplos de políticas basadas en identidad de Amazon Translate \(p. 61\)](#)
- [Permitir a los usuarios ver sus propios permisos \(p. 64\)](#)
- [Solución de problemas de identidad y acceso en Amazon Translate \(p. 64\)](#)
- [Permisos de la API de Amazon Translate: información acerca de acciones, recursos y condiciones \(p. 66\)](#)

Público

La forma en que utilice AWS Identity and Access Management (IAM) difiere, en función del trabajo que realice en Amazon Translate.

Usuario de servicio: si utiliza el servicio Amazon Translate para realizar su trabajo, su administrador le proporciona las credenciales y los permisos que necesita. A medida que utilice más características de Amazon Translate para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos a su administrador. Si no puede acceder a una característica en Amazon Translate, consulte [Solución de problemas de identidad y acceso en Amazon Translate \(p. 64\)](#).

Administrador de servicio: si está a cargo de los recursos de Amazon Translate en su empresa, probablemente tenga acceso completo a Amazon Translate. Su trabajo consiste en determinar qué a características y recursos de Amazon Translate deben acceder sus empleados. A continuación, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con Amazon Translate, consulte [Funcionamiento de Amazon Translate con IAM \(p. 59\)](#).

Administrador de IAM: si es un administrador de IAM, es posible que quiera conocer información sobre cómo escribir políticas para administrar el acceso a Amazon Translate. Para ver ejemplos de políticas basadas en la identidad de Amazon Translate que puede utilizar en IAM, consulte [Ejemplos de políticas basadas en identidad de Amazon Translate \(p. 61\)](#).

Autenticación con identidades

La autenticación es la manera de iniciar sesión en AWS mediante credenciales de identidad. Para obtener más información acerca del inicio de sesión con la Consola de administración de AWS, consulte [La consola de IAM y la página de inicio de sesión](#) en la Guía del usuario de IAM.

Debe estar autenticado (haber iniciado sesión en AWS) como Usuario de la cuenta raíz de AWS, usuario de IAM o asumiendo un rol de IAM. También puede utilizar la autenticación de inicio de sesión único de su empresa o incluso iniciar sesión con Google o Facebook. En estos casos, su administrador habrá configurado previamente la federación de identidad mediante roles de IAM. Cuando obtiene acceso a AWS mediante credenciales de otra empresa, asume un rol indirectamente.

Para iniciar sesión directamente en la [Consola de administración de AWS](#), use su contraseña con su correo electrónico usuario raíz o su nombre de usuario de IAM. Puede obtener acceso a AWS mediante programación utilizando sus claves de acceso usuario raíz o de usuario de IAM. AWS proporciona SDK y herramientas de línea de comandos para firmar criptográficamente su solicitud con sus credenciales. Si no utiliza las herramientas de AWS, debe firmar usted mismo la solicitud. Para ello, utilice Signature Version

4, un protocolo para autenticar solicitudes de API de entrada. Para obtener más información acerca de la autenticación de solicitudes, consulte [Proceso de firma Signature Version 4](#) en la AWS General Reference.

Independientemente del método de autenticación que utilice, es posible que también deba proporcionar información de seguridad adicional. Por ejemplo, AWS le recomienda el uso de la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte [Uso de Multi-Factor Authentication \(MFA\) en AWS](#) en la Guía del usuario de IAM.

Usuario raíz de la cuenta de AWS

Cuando se crea por primera vez una cuenta de AWS, se comienza con una única identidad de inicio de sesión que tiene acceso completo a todos los servicios y recursos de AWS de la cuenta. Esta identidad recibe el nombre de AWS de la cuenta de usuario raíz y se obtiene acceso a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizó para crear la cuenta. Le recomendamos que no utilice usuario raíz en sus tareas cotidianas, ni siquiera en las tareas administrativas. En lugar de ello, es mejor ceñirse a la [práctica recomendada de utilizar exclusivamente usuario raíz para crear el primer usuario de IAM](#). A continuación, guarde las credenciales de usuario raíz en un lugar seguro y utilícelas únicamente para algunas tareas de administración de cuentas y servicios.

Usuarios y grupos de IAM

Un [usuario de IAM](#) es una entidad de la cuenta de AWS que dispone de permisos específicos para una sola persona o aplicación. Un usuario de IAM puede tener credenciales a largo plazo, como un nombre de usuario y una contraseña o un conjunto de claves de acceso. Para obtener más información acerca de cómo generar claves de acceso, consulte [Administración de las claves de acceso de los usuarios de IAM](#) en la Guía del usuario de IAM. Al generar claves de acceso para un usuario de IAM, asegúrese de ver y guardar de forma segura el par de claves. No puede recuperar la clave de acceso secreta en el futuro. En su lugar, debe generar un nuevo par de claves de acceso.

Un [grupo de IAM](#) es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese Administradores de IAM y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para obtener más información, consulte [Cuándo crear un usuario de IAM \(en lugar de un rol\)](#) en la Guía del usuario de IAM.

Roles de IAM

Un [rol de IAM](#) es una entidad de la cuenta de AWS que dispone de permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente un rol de IAM en la Consola de administración de AWS [cambiando de roles](#). Puede asumir un rol llamando a una operación de la AWS CLI o de la API de AWS, o utilizando una URL personalizada. Para obtener más información acerca de los métodos para el uso de roles, consulte [Uso de roles de IAM](#) en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- **Permisos de usuario temporales de IAM:** un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- **Acceso de usuario federado:** En lugar de crear un usuario de IAM, puede utilizar identidades existentes de AWS Directory Service, del directorio de usuarios de la empresa o de un proveedor de identidades web. A estas identidades se les llama usuarios federados. AWS asigna una función a un usuario

federado cuando se solicita acceso a través de un [proveedor de identidad](#). Para obtener más información acerca de los usuarios federados, consulte [Usuarios federados y roles](#) en la Guía del usuario de IAM.

- **Acceso entre cuentas:** puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta obtenga acceso a los recursos de su cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunos servicios de AWS, puede asociar una política directamente a un recurso (en lugar de utilizar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.
- **Acceso a servicios de AWS:** Un rol de servicio es un rol de IAM que un servicio asume para realizar acciones en su cuenta en su nombre. Al configurar algunos de los entornos de los servicios de AWS, debe definir un rol que el servicio asumirá. Este rol de servicio debe incluir todos los permisos que son necesarios para que el servicio pueda acceder a los recursos de AWS que necesita. Los roles de servicio varían de servicio a servicio, pero muchos le permiten elegir sus permisos, siempre y cuando se cumplan los requisitos documentados para dicho servicio. Los roles de servicio ofrecen acceso solo dentro de su cuenta y no se pueden utilizar para otorgar acceso a servicios en otras cuentas. Puede crear, modificar y eliminar un rol de servicio desde IAM. Por ejemplo, puede crear un rol que permita a Amazon Redshift tener acceso a un bucket de Amazon S3 en su nombre y, a continuación, cargar los datos de ese bucket en un clúster de Amazon Redshift. Para obtener más información, consulte [Creación de un rol para delegar permisos a un servicio de AWS](#) en la Guía del usuario de IAM.
- **Aplicaciones que se ejecutan en Amazon EC2:** Puede utilizar un rol de IAM para administrar credenciales temporales para las aplicaciones que se ejecutan en una instancia EC2 y realizan solicitudes de la AWS CLI o la API de AWS. Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia EC2. Para asignar un rol de AWS a una instancia EC2 y ponerla a disposición de todas las aplicaciones, cree un perfil de instancia asociado a la misma. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia EC2 obtener credenciales temporales. Para obtener más información, consulte [Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2](#) en la Guía del usuario de IAM.

Para obtener información acerca del uso de los roles de IAM, consulte [Cuándo crear un rol de IAM \(en vez de un usuario\)](#) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

Para controlar el acceso en AWS, se crean políticas y se asocian a identidades de IAM o recursos de AWS. Una política es un objeto de AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando una entidad principal (usuario raíz, usuario de IAM o rol de IAM) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. Las mayoría de las políticas se almacenan en AWS como documentos JSON. Para obtener más información acerca de la estructura y el contenido de los documentos de política JSON, consulte [Información general de las políticas de JSON](#) en la Guía del usuario de IAM.

Un administrador de IAM puede utilizar las políticas para especificar quién tiene acceso a los recursos de AWS y qué acciones se pueden realizar en dichos recursos. Cada entidad de IAM (usuario o rol) comienza sin permisos. En otras palabras, de forma predeterminada, los usuarios no pueden hacer nada, ni siquiera cambiar sus propias contraseñas. Para conceder permiso a un usuario para hacer algo, el administrador debe asociarle una política de permisos. O bien el administrador puede añadir al usuario a un grupo que tenga los permisos necesarios. Cuando el administrador concede permisos a un grupo, todos los usuarios de ese grupo obtienen los permisos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con dicha política puede obtener información del usuario de la Consola de administración de AWS, la AWS CLI o la API de AWS.

Políticas basadas en la identidad

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como por ejemplo un usuario, un rol o un grupo de IAM. Estas políticas controlan qué acciones puede realizar dicha identidad, en qué recursos y en qué condiciones. Para obtener más información acerca de cómo crear una política basada en identidad, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Las políticas basadas en identidad pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede asociar a varios usuarios, grupos y roles de su cuenta de AWS. Las políticas administradas incluyen las políticas administradas por AWS y las políticas administradas por el cliente. Para obtener más información acerca de cómo elegir una política administrada o una política insertada, consulte [Elegir entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que puede asociar a un recurso como, por ejemplo, un bucket de Amazon S3. Los administradores de servicios pueden utilizar estas políticas para definir qué acciones puede realizar un principal especificado (miembro de cuenta, usuario o rol) en dicho recurso y bajo qué condiciones. Las políticas basadas en recursos son políticas insertadas. No existen políticas basadas en recursos que sean administradas.

Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) son un tipo de política que controlan qué entidades principales (cuentas, miembros, usuarios o roles) tienen permisos para obtener acceso a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de política JSON. Amazon S3, AWS WAF y Amazon VPC son ejemplos de servicios que admiten ACL. Para obtener más información sobre las ACL, consulte [Información general de las Access Control Lists \(ACL, Listas de control de acceso\)](#) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite otros tipos de políticas menos frecuentes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le otorgan.

- **Límites de permisos:** un límite de permisos es una característica avanzada que le permite definir los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una identidad. Los permisos resultantes son la intersección de las políticas basadas en identidades de la entidad y los límites de sus permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información acerca de los límites de permisos, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- **Políticas de control de servicios (SCP):** las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU) en AWS Organizations. AWS Organizations es un servicio que le permite agrupar y administrar de forma centralizada varias cuentas de AWS que posee su negocio. Si habilita todas las funciones en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o todas sus cuentas. Una SCP limita los permisos para las entidades de las cuentas de miembros, incluido cada Usuario de la cuenta raíz de AWS. Para obtener más información acerca de Organizaciones y las SCP, consulte [Funcionamiento de las SCP](#) en la Guía del usuario de AWS Organizations.
- **Políticas de sesión:** las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los

permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política basada en recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para obtener información acerca de cómo AWS determina si permitir una solicitud cuando hay varios tipos de políticas implicados, consulte [Lógica de evaluación de políticas](#) en la Guía del usuario de IAM.

Funcionamiento de Amazon Translate con IAM

Puede administrar el acceso a las operaciones y los recursos de Amazon Translate con AWS Identity and Access Management. Antes de utilizar AWS Identity and Access Management (IAM) para administrar el acceso a Amazon Translate, debe saber qué características de IAM están disponibles para su uso con Amazon Translate. Para obtener una perspectiva general acerca de cómo funciona Amazon Translate y otros servicios de AWS con IAM, consulte [Servicios de AWS que funcionan con IAM](#) en la Guía del usuario de IAM.

Temas

- [Políticas basadas en identidad de Amazon Translate](#) (p. 59)
- [Políticas basadas en recursos de Amazon Translate](#) (p. 60)
- [Autorización basada en etiquetas de Amazon Translate](#) (p. 60)
- [Roles de IAM de Amazon Translate](#) (p. 60)

Políticas basadas en identidad de Amazon Translate

Con las políticas basadas en identidad de IAM, puede especificar las acciones permitidas o denegadas y los recursos además de las condiciones en las que se permiten o deniegan las acciones. Amazon Translate admite acciones, recursos y claves de condiciones específicos. Para obtener más información acerca de los elementos que utiliza en una política de JSON, consulte [Referencia de los elementos de las políticas de JSON de IAM](#) en la Guía del usuario de IAM.

Acciones

El elemento `Action` de una política basada en la identidad de IAM describe la acción o las acciones específicas que la política permitirá o denegará. Las acciones de la política generalmente tienen el mismo nombre que la operación de API de AWS asociada. La acción se utiliza en una política para otorgar permisos para realizar la operación asociada.

Las acciones de políticas de Amazon Translate utilizan el siguiente prefijo antes de la acción: `translate:`. Por ejemplo, para conceder a alguien permiso para realizar una traducción con la operación Amazon Translate `TranslateText` de la API, inicie sesión como administrador e incluya la acción `translate:TranslateText` en su política. Las instrucciones de política deben incluir un elemento `NotAction` o `Action`. Amazon Translate define su propio conjunto de acciones que describen las tareas que se pueden realizar con este servicio. Para ver una lista de acciones de Amazon Translate, consulte [Actions Defined by Amazon Translate](#) en la Guía del usuario de IAM.

Para especificar varias acciones en una única instrucción, sepárelas con comas del siguiente modo:

```
"Action": [
```

```
"translate:action1",  
"translate:action2"
```

Recursos

Amazon Translate no admite la especificación de los nombres de recursos de Amazon (ARN) de los recursos en una política.

Claves de condición

Amazon Translate no proporciona ninguna clave de condición específica del servicio, pero sí admite el uso de algunas claves de condición globales. Para ver todas las claves de condición globales de AWS, consulte [Claves de contexto de condición globales de AWS](#) en la Guía del usuario de IAM.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilizan [operadores de condición](#), tales como igual o menor que, para que coincida la condición de la política con valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación AND lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición con una operación lógica OR. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para obtener más información, consulte [Elementos de la política de IAM: Variables y etiquetas](#) en la Guía del usuario de IAM.

Ejemplos

Para obtener ejemplos de políticas basadas en identidad de Amazon Translate, consulte [Ejemplos de políticas basadas en identidad de Amazon Translate \(p. 61\)](#).

Políticas basadas en recursos de Amazon Translate

Amazon Translate no admite políticas basadas en recursos.

Autorización basada en etiquetas de Amazon Translate

Amazon Translate no admite el etiquetado de recursos o el control de acceso basado en etiquetas.

Roles de IAM de Amazon Translate

Un [rol de IAM](#) es una entidad de la cuenta de AWS que dispone de permisos específicos.

Uso de credenciales temporales con Amazon Translate

Puede utilizar credenciales temporales para iniciar sesión con federación, asumir un rol de IAM o asumir un rol de acceso entre cuentas. Las credenciales de seguridad temporales se obtienen mediante una llamada a operaciones de la API de AWS Security Token Service (AWS STS), como [AssumeRole](#) o [GetFederationToken](#).

Amazon Translate admite el uso de credenciales temporales.

Roles vinculados a servicios

Los [roles vinculados a servicios](#) permiten a los servicios de AWS obtener acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles vinculados a servicios aparecen en la cuenta de IAM y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Amazon Translate no admite roles vinculados a servicios.

Ejemplos de políticas basadas en identidad de Amazon Translate

De forma predeterminada, los usuarios y roles de IAM no tienen permiso para crear, ver ni modificar recursos de Amazon Translate. Tampoco pueden realizar tareas mediante la Consola de administración de AWS, la AWS CLI, o la API de AWS. Un administrador de IAM debe crear políticas de IAM que concedan permisos a los usuarios y a los roles para realizar operaciones de la API concretas en los recursos específicos que necesiten. El administrador debe adjuntar esas políticas a los usuarios o grupos de IAM que necesiten esos permisos.

Para obtener más información acerca de cómo crear una política basada en identidad de IAM con los documentos de políticas de JSON de ejemplo siguientes, consulte [Creación de políticas en la pestaña JSON](#) en la Guía del usuario de IAM.

Temas

- [Prácticas recomendadas de políticas basadas en identidad \(p. 61\)](#)
- [Uso de la consola de Amazon Translate \(p. 62\)](#)
- [Permisos para utilizar claves administradas por el cliente con terminologías personalizadas \(p. 62\)](#)

Prácticas recomendadas de políticas basadas en identidad

Las políticas basadas en identidad son muy eficaces. Determinan si alguien puede crear, acceder o eliminar los recursos de Amazon Translate de su cuenta. Estas acciones pueden generar costes adicionales para su cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidad:

- **Introducción sobre el uso de políticas administradas de AWS:** para comenzar a utilizar Amazon Translate rápidamente, utilice las políticas administradas de AWS para proporcionar a los empleados los permisos necesarios. Estas políticas ya están disponibles en su cuenta y las mantiene y actualiza AWS. Para obtener más información, consulte [Introducción sobre el uso de permisos con políticas administradas de AWS](#) en la Guía del usuario de IAM.
- **Conceder privilegios mínimos:** al crear políticas personalizadas, conceda solo los permisos necesarios para llevar a cabo una tarea. Comience con un conjunto mínimo de permisos y conceda permisos adicionales según sea necesario. Por lo general, es más seguro que comenzar con permisos que son demasiado tolerantes e intentar hacerlos más severos más adelante. Para obtener más información, consulte [Conceder privilegios mínimos](#) en la Guía del usuario de IAM.
- **Habilitar MFA para operaciones confidenciales:** para mayor seguridad, obligue a los usuarios de IAM a que utilicen la autenticación multifactor (MFA) para acceder a recursos u operaciones de API confidenciales. Para obtener más información, consulte [Uso de Multi-Factor Authentication \(MFA\) en AWS](#) en la Guía del usuario de IAM.
- **Utilizar condiciones de política para mayor seguridad:** en la medida en que sea práctico, defina las condiciones en las que sus políticas basadas en identidad permitan el acceso a un recurso. Por ejemplo, puede escribir condiciones para especificar un rango de direcciones IP permitidas desde el que debe proceder una solicitud. También puede escribir condiciones para permitir solicitudes solo en un intervalo

de hora o fecha especificado o para solicitar el uso de SSL o MFA. Para obtener más información, consulte [Elementos de la política de JSON de IAM: condición](#) en la Guía del usuario de IAM.

Uso de la consola de Amazon Translate

Para acceder a la consola de Amazon Translate, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle registrar y consultar los detalles sobre los recursos de Amazon Translate en su cuenta de AWS. Si crea una política basada en identidad que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles de IAM) que tengan esa política.

Para asegurarse de que esas entidades puedan seguir usando la consola de Amazon Translate, asocie también la política administrada de AWS siguiente a las entidades.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "translate:*",
        "comprehend:DetectDominantLanguage",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics"
      ],
      "Resource": "*"
    }
  ]
}
```

La política tiene cuatro declaraciones. La primera concede permisos para usar la operación `TranslateText`. La segunda concede permisos a la operación `DetectDominantLanguage` de Amazon Comprehend para que habilite la detección automática de idioma. Los dos últimos permisos conceden permisos para que Amazon CloudWatch proporcione compatibilidad con las métricas.

No es necesario que conceda permisos mínimos para la consola a los usuarios que solo realizan llamadas a la AWS CLI o a la API de AWS. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar. Para obtener más información, consulte [Adición de permisos a un usuario](#) en la Guía del usuario de IAM:

Permisos para utilizar claves administradas por el cliente con terminologías personalizadas

Si utiliza claves AWS Key Management Service (AWS KMS) claves administradas por el cliente (CMK) con terminologías de Amazon Translate personalizadas, es posible que necesite permisos adicionales en la política de claves de AWS KMS.

Para llamar a la operación `ImportTerminology` con un a CMK de AWS KMS, agregue los siguientes permisos a la política de claves de AWS KMS existente.

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow access for use with Amazon Translate",
      "Effect": "Allow",
      "Principal": {
```

```
        "AWS": "IAM USER OR ROLE ARN"
      },
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:RetireGrant"
      ],
      "Resource": "*"
    }
  ]
}
```

Para llamar a la operación `GetTerminology` para una terminología personalizada que se importó con una CMK de KMS, añada los siguientes permisos a la política de claves de AWS KMS.

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow access for use with Amazon Translate",
      "Effect": "Allow",
      "Principal": {
        "AWS": "IAM USER OR ROLE ARN"
      },
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "*"
    }
  ]
}
```

Para llamar a las operaciones `DeleteTerminology` o `ListTerminologies` para una terminología personalizada importada con una CMK de AWS KMS, no necesita tener permisos de AWS KMS especiales.

Para utilizar CMK con todas las operaciones de terminologías personalizadas, añada los siguientes permisos en la política de claves de AWS KMS.

```
{
  "Id": "key-consolepolicy-3",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow access for use with Amazon Translate",
      "Effect": "Allow",
      "Principal": {
        "AWS": "IAM USER OR ROLE ARN"
      },
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:RetireGrant",
        "kms:Decrypt"
      ],
      "Resource": "*"
    }
  ]
}
```

Para ver las acciones de la API de Amazon Translate y los recursos a los que se aplican, consulte [Permisos de la API de Amazon Translate: información acerca de acciones, recursos y condiciones \(p. 66\)](#).

Permitir a los usuarios ver sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para llevar a cabo esta acción en la consola o mediante programación con la AWS CLI o la API de AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Solución de problemas de identidad y acceso en Amazon Translate

Utilice la información siguiente para diagnosticar y solucionar los problemas comunes que puedan surgir cuando trabaje con Amazon Translate e AWS Identity and Access Management (IAM).

Temas

- [No tengo autorización para realizar una acción en Amazon Translate \(p. 65\)](#)
- [No tengo autorización para realizar la operación iam:PassRole \(p. 65\)](#)
- [Quiero ver mis claves de acceso \(p. 65\)](#)
- [Soy administrador y deseo permitir que otros obtengan acceso a Amazon Translate \(p. 66\)](#)
- [Quiero permitir a personas externas a mi cuenta de AWS el acceso a mis recursos de Amazon Translate \(p. 66\)](#)

No tengo autorización para realizar una acción en Amazon Translate

Si la Consola de administración de AWS le indica que no está autorizado para llevar a cabo una acción, póngase en contacto con su administrador para recibir ayuda. Su administrador es la persona que le facilitó su nombre de usuario y contraseña.

Por ejemplo, el siguiente error se produce cuando el usuario de IAM `mateojackson` intenta utilizar la consola para traducir un texto, pero no cuenta con el permiso `translate:TranslateText`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
translate:TranslateText
```

Mateo le pediría a su administrador que actualizara sus políticas para permitirle usar la acción `translate:TranslateText`.

No tengo autorización para realizar la operación `iam:PassRole`

Si recibe un error que indica que no está autorizado para llevar a cabo la acción `iam:PassRole`, debe ponerse en contacto con su administrador para recibir ayuda. Su administrador es la persona que le facilitó su nombre de usuario y contraseña. Pida a la persona que actualice sus políticas de forma que pueda transferir un rol a Amazon Translate.

Algunos servicios de AWS le permiten transferir un rol existente a dicho servicio en lugar de crear un nuevo rol de servicio o uno vinculado al servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en Amazon Translate. Sin embargo, la acción requiere que el servicio cuente con permisos otorgados por un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

En este caso, Mary pide a su administrador que actualice sus políticas para que pueda realizar la acción `iam:PassRole`.

Quiero ver mis claves de acceso

Después de crear sus claves de acceso de usuario de IAM, puede ver su ID de clave de acceso en cualquier momento. Sin embargo, no puede volver a ver su clave de acceso secreta. Si pierde la clave de acceso secreta, debe crear un nuevo par de claves de acceso.

Las claves de acceso se componen de dos partes: un ID de clave de acceso (por ejemplo, `AKIAIOSFODNN7EXAMPLE`) y una clave de acceso secreta (por ejemplo, `wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY`). El ID de clave de acceso y la clave de acceso secreta se utilizan juntos, como un nombre de usuario y contraseña, para autenticar sus solicitudes. Administre sus claves de acceso con el mismo nivel de seguridad que para el nombre de usuario y la contraseña.

Important

No proporcione las claves de acceso a terceras personas, ni siquiera para que le ayuden a [buscar el ID de usuario canónico](#). Si lo hace, podría conceder a otra persona acceso permanente a su cuenta.

Cuando cree un par de claves de acceso, se le pide que guarde el ID de clave de acceso y la clave de acceso secreta en un lugar seguro. La clave de acceso secreta solo está disponible en el momento de su

creación. Si pierde la clave de acceso secreta, debe añadir nuevas claves de acceso a su usuario de IAM. Puede tener un máximo de dos claves de acceso. Si ya cuenta con dos, debe eliminar un par de claves antes de crear uno nuevo. Para ver las instrucciones, consulte [Administración de las claves de acceso](#) en la Guía del usuario de IAM.

Soy administrador y deseo permitir que otros obtengan acceso a Amazon Translate

Para permitir que otros obtengan acceso a Amazon Translate, debe crear una entidad de IAM (usuario o rol) para la persona o aplicación que necesita acceso. Esta persona utilizará las credenciales de la entidad para obtener acceso a AWS. A continuación, debe asociar una política a la entidad que le conceda los permisos correctos en Amazon Translate.

Para comenzar de inmediato, consulte [Creación del primer grupo y usuario delegado de IAM](#) en la Guía del usuario de IAM.

Quiero permitir a personas externas a mi cuenta de AWS el acceso a mis recursos de Amazon Translate

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para obtener información acerca de si Amazon Translate admite estas características, consulte [Funcionamiento de Amazon Translate con IAM \(p. 59\)](#).
- Para aprender cómo proporcionar acceso a sus recursos en cuentas de AWS de su propiedad, consulte [Proporcionar acceso a un usuario de IAM a otra cuenta de AWS de la que es propietario](#) en la Guía del usuario de IAM.
- Para obtener información acerca de cómo ofrecer acceso a sus recursos a cuentas de AWS de terceros, consulte [Proporcionar acceso a las cuentas de AWS propiedad de terceros](#) en la Guía del usuario de IAM.
- Para obtener información acerca de cómo ofrecer acceso a la identidad federada, consulte [Proporcionar acceso a usuarios autenticados externamente \(identidad federada\)](#) en la Guía del usuario de IAM.
- Para obtener información acerca de la diferencia entre utilizar los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.

Permisos de la API de Amazon Translate: información acerca de acciones, recursos y condiciones

Utilice la siguiente tabla como referencia cuando escriba una política de permisos para poder asociarla a una identidad de IAM (una política basada en identidad). La lista incluye cada operación API de Amazon Translate, la acción correspondiente para la que puede conceder permisos y el recurso de AWS para el que puede conceder los permisos. Las acciones se especifican en el campo `Action` de la política y el valor del recurso se especifica en el campo `Resource` de la política.

Para expresar condiciones, puede utilizar claves de condiciones generales de AWS en sus políticas de Amazon Translate. Para ver una lista completa de claves generales de AWS, consulte [Claves disponibles](#) en la Guía del usuario de IAM.

Note

Para especificar una acción, use el prefijo `translate:` seguido del nombre de la operación API (por ejemplo, `translate:TranslateText`).

Monitoreo de Amazon Translate

El monitoreo es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el desempeño de Amazon Translate y las soluciones. AWS dispone de diversas herramientas que se pueden utilizar para monitorear Amazon Translate. Puede configurar algunas de estas herramientas para monitorizar las soluciones automáticamente. Le recomendamos que automatice las tareas de monitorización en la medida de lo posible.

Amazon Translate cuenta con gráficos preconfigurados en los que se muestran las métricas más importantes para la solución. Cada gráfico ofrece una ventana al desempeño de su solución. Para obtener diferentes vistas del rendimiento de su solución a lo largo del tiempo, puede cambiar el intervalo de tiempo que muestran los gráficos.

También puede utilizar Amazon CloudWatch para monitorear Amazon Translate. Con CloudWatch, puede automatizar el monitoreo de métricas específicas de las soluciones. Recibirá un aviso siempre que una métrica esté fuera de los umbrales que haya establecido. También puede utilizar la API de CloudWatch para crear una aplicación de monitoreo personalizada adecuada a sus necesidades. Para obtener más información, consulte [¿Qué es Amazon CloudWatch?](#) en la Guía del usuario de Amazon CloudWatch.

En la tabla siguiente se describe cada uno de los gráficos preconfigurados que proporciona Amazon Translate.

Gráfico	Descripción
<p>Successful request count</p> <p>Go to CloudWatch</p> 	<p>Recuento de solicitudes correcto</p> <p>Número de solicitudes realizadas correctamente a Amazon Translate durante el periodo de tiempo especificado.</p>
<p>Throttled request count</p> <p>Go to CloudWatch</p> 	<p>Recuento de solicitudes limitadas</p> <p>Número de solicitudes a Amazon Translate que se han limitado durante el periodo de tiempo especificado. Utilice esta información para determinar si la aplicación envía solicitudes a Amazon Translate demasiado rápido.</p>

Gráfico	Descripción																
<p>Average response time (milliseconds)</p> <p>Go to CloudWatch</p> <table border="1"> <caption>Average response time (milliseconds) - All language pairs</caption> <thead> <tr> <th>Time</th> <th>Average response time (milliseconds)</th> </tr> </thead> <tbody> <tr> <td>08:00</td> <td>~250</td> </tr> <tr> <td>08:30</td> <td>~600</td> </tr> <tr> <td>09:00</td> <td>~1250</td> </tr> <tr> <td>09:30</td> <td>~650</td> </tr> <tr> <td>13:00</td> <td>~300</td> </tr> <tr> <td>13:30</td> <td>~350</td> </tr> <tr> <td>14:00</td> <td>~300</td> </tr> </tbody> </table> <p>■ All language pairs</p>	Time	Average response time (milliseconds)	08:00	~250	08:30	~600	09:00	~1250	09:30	~650	13:00	~300	13:30	~350	14:00	~300	<p>Promedio de tiempo de respuesta</p> <p>Promedio de tiempo que tardó Amazon Translate en procesar su solicitud durante el periodo de tiempo especificado.</p>
Time	Average response time (milliseconds)																
08:00	~250																
08:30	~600																
09:00	~1250																
09:30	~650																
13:00	~300																
13:30	~350																
14:00	~300																
<p>Character count</p> <p>Go to CloudWatch</p> <table border="1"> <caption>Character count - All language pairs</caption> <thead> <tr> <th>Time</th> <th>Character count</th> </tr> </thead> <tbody> <tr> <td>08:00</td> <td>~1150</td> </tr> <tr> <td>08:30</td> <td>~1300</td> </tr> <tr> <td>09:00</td> <td>~100</td> </tr> <tr> <td>13:00</td> <td>~1550</td> </tr> <tr> <td>14:00</td> <td>~1250</td> </tr> </tbody> </table> <p>■ All language pairs</p>	Time	Character count	08:00	~1150	08:30	~1300	09:00	~100	13:00	~1550	14:00	~1250	<p>Recuento de caracteres</p> <p>Número total de caracteres que ha enviado a Amazon Translate durante el periodo de tiempo especificado. Es el número de caracteres que se le facturarán.</p>				
Time	Character count																
08:00	~1150																
08:30	~1300																
09:00	~100																
13:00	~1550																
14:00	~1250																
<p>User error count</p> <p>Go to CloudWatch</p> <table border="1"> <caption>User error count - All language pairs</caption> <thead> <tr> <th>Time</th> <th>User error count</th> </tr> </thead> <tbody> <tr> <td>09:00</td> <td>1</td> </tr> </tbody> </table> <p>■ All language pairs</p>	Time	User error count	09:00	1	<p>Recuento de errores de usuario</p> <p>Número de errores de usuario que se produjeron durante el periodo de tiempo especificado. Los errores del usuario se encuentran en el intervalo de códigos de error HTTP de 400 a 499.</p>												
Time	User error count																
09:00	1																
<p>System error count</p> <p>Go to CloudWatch</p> <p>■ All language pairs</p>	<p>Recuento de errores del sistema</p> <p>Número de errores del sistema que se produjeron durante el periodo de tiempo especificado. Los errores del sistema se encuentran en el intervalo de códigos de error HTTP de 500 a 599.</p>																

Monitoreo de Amazon Translate

Con Amazon CloudWatch, obtendrá métricas de las operaciones de Amazon Translate individuales o las métricas de Amazon Translate globales de su cuenta. Utilice las métricas para realizar el seguimiento del estado de las soluciones de Amazon Translate y configurar alarmas que le avisen cuando una o varias métricas superen un umbral definido. Por ejemplo, puede monitorear el número de solicitudes realizadas a Amazon Translate durante un periodo de tiempo concreto, ver la latencia de las solicitudes correctas o crear una alarma para cuando los errores superen un umbral.

Descripción de las métricas de CloudWatch para Amazon Translate

Para obtener métricas de las operaciones de Amazon Translate, especifique la siguiente información:

- La dimensión de la métrica. Una dimensión es un conjunto de pares nombre/valor que se emplea para identificar una métrica. Amazon Translate tiene dos dimensiones:
 - `Operation`
 - `Language pair`
- El nombre de la métrica, como `SuccessfulRequestCount` o `RequestCharacters`. Para obtener una lista completa de métricas, consulte [Métricas de CloudWatch para Amazon Translate \(p. 72\)](#).

Puede obtener métricas de Amazon Translate con la Consola de administración de AWS, la AWS CLI o la API de CloudWatch. También puede utilizar la API de CloudWatch con uno de los kits de desarrollo de software (SDK) de Amazon AWS o las herramientas de API de CloudWatch.

En la siguiente tabla, se indican algunos usos frecuentes de las métricas de CloudWatch. Se trata de sugerencias que puede usar como punto de partida y no de una lista completa.

¿Cómo?	Monitorizar esta métrica
Realizar el seguimiento el número de solicitudes realizadas correctamente	La estadística sum de la métrica <code>SuccessfulRequestCount</code> .
Saber si mi aplicación ha alcanzado su desempeño máximo	La estadística sum de la métrica <code>ThrottledCount</code> .
Buscar el tiempo de respuesta de mi aplicación	La estadística average de la métrica <code>ResponseTime</code> .
Buscar el número de errores de mi aplicación	La estadística sum de las métricas <code>ServerErrorCount</code> y <code>UserErrorCount</code> .
Buscar el número de caracteres facturables	La estadística sum de la métrica <code>CharacterCount</code> .

Debe disponer de los permisos de CloudWatch adecuados para monitorizar Amazon Translate con CloudWatch. Para obtener más información, consulte [Autenticación y control de acceso de Amazon CloudWatch](#) en la Guía de usuario de Amazon CloudWatch.

Visualización de métricas de Amazon Translate

Consulte las métricas de Amazon Translate en la consola de CloudWatch.

Para ver las métricas (consola de CloudWatch)

1. Inicie sesión en la Consola de administración de AWS y abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
2. Elija Metrics (Métricas), All Metrics (Todas las métricas) y AWS/Translate (AWS/Traducir).
3. Elija la dimensión, un nombre de métrica y Add to graph (Añadir al gráfico).
4. Elija un valor para el intervalo de fechas. El recuento de las métricas del intervalo de fechas especificado se muestra en el gráfico.

Registro de llamadas al API de Amazon Translate con AWS CloudTrail

Amazon Translate está integrado con AWS CloudTrail, un servicio que proporciona un registro de las acciones realizadas por un usuario de IAM, un rol de IAM o un servicio de AWS en Amazon Translate. CloudTrail captura todas las llamadas a la API de Amazon Translate como eventos. Esto incluye llamadas desde la consola de Amazon Translate y llamadas de código a las operaciones de la API de Amazon Translate. Si crea un registro de seguimiento de CloudTrail, puede habilitar la entrega continua de eventos de CloudTrail, incluidos los eventos de Amazon Translate, a un bucket de Amazon Simple Storage Service (Amazon S3). Si no configura un registro de seguimiento, puede ver los eventos más recientes en la consola de CloudTrail en el Event history (Historial de eventos). Se puede usar la información que recopila CloudTrail para determinar la solicitud que se envió a Amazon Translate, la dirección IP desde la que se realizó la solicitud, quién realizó la solicitud, cuándo la realizó y detalles adicionales.

Para obtener más información sobre CloudTrail, consulte la [AWS CloudTrail User Guide](#).

Temas

- [Información de Amazon Translate en CloudTrail \(p. 70\)](#)
- [Descripción de las entradas de archivos de registro de Amazon Translate \(p. 71\)](#)

Información de Amazon Translate en CloudTrail

CloudTrail se habilita en una cuenta de AWS al crearla. Cuando se produce una actividad en Amazon Translate, dicha actividad se registra en un evento de CloudTrail junto con los eventos de los demás servicios de AWS en el Event history (Historial de eventos). Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS. Para obtener más información, consulte [Visualización de eventos con el historial de eventos de CloudTrail](#).

Para mantener un registro continuo de los eventos de la cuenta de AWS, incluidos los eventos de Amazon Translate, cree un registro de seguimiento. Un registro de seguimiento permite a CloudTrail enviar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento mediante la consola, este se aplica a todas las regiones de AWS. El registro de seguimiento registra los eventos de todas las regiones de la partición de AWS y envía los archivos de registro al bucket de S3 especificado. También puede configurar otros servicios de AWS para analizar y actuar según los datos de eventos recopilados en los registros de CloudTrail. Para obtener más información, consulte los siguientes temas:

- [Introducción a la creación de registros de seguimiento](#)
- [Servicios e integraciones compatibles con CloudTrail](#)
- [Configuración de notificaciones de Amazon SNS para CloudTrail](#)
- [Recibir archivos de registro de CloudTrail de varias regiones y Recepción de archivos de registro de CloudTrail de varias cuentas](#)

Todas las acciones de Amazon Translate se registran en CloudTrail y se documentan en la sección [Referencia de la API de](#) . Por ejemplo, las llamadas a las acciones `DeleteTerminology`, `ImportTerminology` y `TranslateText` generan entradas en los archivos de registro de CloudTrail.

Cada entrada de registro o evento contiene información acerca de quién generó la solicitud. Esta información le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario raíz o AWS Identity and Access Management (IAM)
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el [elemento `userIdentity` de CloudTrail](#).

Descripción de las entradas de archivos de registro de Amazon Translate

Un registro de seguimiento es una configuración que permite la entrega de eventos como archivos de registro al bucket de Amazon S3 que se especifique. Los archivos de registro de CloudTrail contienen una o varias entradas de registro. Un evento representa una única solicitud de cualquier origen e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etcétera. Los archivos de registro de CloudTrail no están un rastro de la pila ordenada de las llamadas a la API públicas, por lo que no aparecen en ningún orden específico.

En el ejemplo siguiente, se muestra una entrada de registro de CloudTrail que ilustra la acción `TranslateText`.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/Administrator",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Administrator"
  },
  "eventTime": "2019-09-03T20:32:50Z",
  "eventSource": "translate.amazonaws.com",
  "eventName": "TranslateText",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "aws-cli/1.16.207 Python/3.4.7
Linux/4.9.184-0.1.ac.235.83.329.metall1.x86_64 botocore/1.12.197",
  "requestParameters": {
    "text": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "sourceLanguageCode": "en",
    "targetLanguageCode": "fr"
  },
  "responseElements": {
    "translatedText": "HIDDEN_DUE_TO_SECURITY_REASONS",
    "sourceLanguageCode": "en",
    "targetLanguageCode": "fr"
  },
  "requestID": "f56da956-284e-4983-b6fc-59befa20e2bf",
  "eventID": "1dc75278-84d7-4bb2-861a-493d08d67391",
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Dimensiones y métricas de CloudWatch para Amazon Translate

Para monitorizar el rendimiento de la solución, utilice las métricas y las dimensiones de Amazon CloudWatch para Amazon Translate.

Métricas de CloudWatch para Amazon Translate

Métrica	Descripción
CharacterCount	<p>El número de caracteres facturables en las solicitudes.</p> <p>Dimensiones válidas: Language pair, Operation</p> <p>Estadísticas válidas: Average, Maximum, Minimum, Sum</p> <p>Unidad: recuento</p>
ResponseTime	<p>El tiempo que se ha tardado en responder a la solicitud.</p> <p>Dimensiones válidas: Language pair, Operation</p> <p>Estadísticas válidas: Data samples, Average</p> <p>Unidad: en las muestras de datos, recuento. En las estadísticas de promedio, milisegundos.</p>
ServerErrorCount	<p>El número de errores de servidor. El intervalo de códigos de respuesta HTTP de un error de servidor va de 500 a 599.</p> <p>Dimensión válida: Operation</p> <p>Estadísticas válidas: Average, Sum</p> <p>Unidad: recuento</p>
SuccessfulRequestCount	<p>El número de solicitudes de traducción realizadas correctamente. El código de respuesta para una solicitud realizada correctamente va de 200 a 299.</p> <p>Dimensión válida: Operation</p> <p>Estadísticas válidas: Average, Sum</p> <p>Unidad: recuento</p>
ThrottledCount	<p>El número de solicitudes sujetas a limitaciones. Utilice ThrottledCount para determinar si la aplicación está enviando solicitudes a Amazon Translate más rápido de lo que admite la configuración de su cuenta. Para obtener más información, consulte Límites de Amazon Translate en la Referencia general de Amazon Web Services.</p> <p>Dimensión válida: Operation</p> <p>Estadísticas válidas: Average, Sum</p> <p>Unidad: recuento</p>

Métrica	Descripción
UserErrorCount	<p>El número de errores de usuario que se han producido. El intervalo de códigos de respuesta HTTP de un error de usuario va de 400 a 499.</p> <p>Dimensión válida: Operation</p> <p>Estadísticas válidas: Average, Sum</p> <p>Unidad: recuento</p>

Dimensiones de CloudWatch para Amazon Translate

Utilice las siguientes dimensiones para filtrar métricas de Amazon Translate. Las métricas están agrupadas por idioma de origen e idioma de destino.

Dimensión	Descripción
LanguagePair	Restringe las métricas solo a las que contienen los idiomas especificados.
Operation	Restringe las métricas solo a las que tienen la operación especificada.

Validación de la conformidad en Amazon Translate

Hay auditores externos que evalúan la seguridad y la conformidad de Amazon Translate en diferentes programas de conformidad de AWS: SOC, PCI, FedRAMP, HIPAA y otros. Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte la sección [Descarga de informes en AWS Artifact](#).

Su responsabilidad de conformidad al utilizar Amazon Translate se determina en función de la confidencialidad de los datos, los objetivos de cumplimiento de su empresa y la legislación, así como los reglamentos correspondientes. AWS proporciona los siguientes recursos para ayudarle en la conformidad:

- [Guías de inicio rápido de seguridad y conformidad](#) – estas guías de implementación tratan consideraciones sobre arquitectura y ofrecen pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- [Documento técnico sobre arquitectura para seguridad y conformidad de HIPAA](#) – este documento técnico describe cómo las empresas pueden utilizar AWS para crear aplicaciones conformes con HIPAA.
- [Recursos de conformidad de AWS](#) – este conjunto de manuales y guías podría aplicarse a su sector y ubicación.
- [AWS Config](#) – este servicio de AWS evalúa en qué medida las configuraciones de los recursos cumplen las prácticas internas, las directrices del sector y las normativas.
- [AWS Security Hub](#) – este servicio de AWS proporciona una vista integral del estado de seguridad en AWS, lo que le ayudará a comprobar la conformidad con los estándares sectoriales de seguridad y las prácticas recomendadas.

Para obtener una lista de los servicios de AWS incluidos en el ámbito de programas de conformidad específicos, consulte [Servicios de AWS en el ámbito del programa de conformidad](#). Para obtener información general, consulte [Programas de conformidad de AWS](#).

Resiliencia de Amazon Translate

La infraestructura global de AWS se compone de regiones y zonas de disponibilidad de AWS. Las regiones de AWS proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información sobre las zonas de disponibilidad y las regiones de AWS, consulte [Infraestructura global de AWS](#).

Seguridad de la infraestructura en Amazon Translate

Al tratarse de un servicio administrado, Amazon Translate está protegido por los procedimientos de seguridad de red globales de AWS que se describen en el documento técnico [Amazon Web Services: Información general sobre procesos de seguridad](#).

Puede utilizar las llamadas a la API publicadas en Amazon Translate para obtener acceso a AWS a través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.0 o una versión posterior. Le recomendamos TLS 1.2 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS), como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La mayoría de los sistemas modernos, como Java 7 y posteriores, son compatibles con estos modos.

Además, las solicitudes deben estar firmadas con un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de AWS Identity and Access Management (IAM). También puede utilizar [AWS Security Token Service](#) (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Directrices y límites

En las siguientes secciones se ofrece información sobre las directrices y los límites de Amazon Translate.

Temas

- [Regiones de AWS admitidas](#) (p. 75)
- [Conformidad](#) (p. 75)
- [Limitación controlada](#) (p. 75)
- [Directrices](#) (p. 75)
- [Límites de los servicios](#) (p. 75)

Regiones de AWS admitidas

Para ver una lista de las regiones de AWS en las que admite Amazon Translate, consulte [Tabla de regiones de AWS](#) o [Regiones y puntos de enlace de AWS](#) en Referencia general de Amazon Web Services.

Conformidad

Para obtener más información sobre los programas de conformidad de Amazon Translate, consulte [Conformidad de AWS](#), [Programas de conformidad de AWS](#) y [Servicios de AWS en ámbito por programa de conformidad](#).

Limitación controlada

Amazon Translate escala la infraestructura para atender el tráfico operativo del cliente. Si encuentra limitaciones constantes, póngase en contacto con [AWS Support](#).

Directrices

Para mejorar continuamente la calidad de sus modelos de análisis, Amazon Translate puede almacenar sus datos. Para obtener más información, consulte las [preguntas frecuentes de Amazon Translate](#).

Para solicitar que eliminemos sus datos y que los datos futuros asociados a su cuenta no se almacenen, puede ponerse en contacto con [AWS Support](#). Sin embargo, dado que la eliminación de sus datos también puede eliminar datos de entrenamiento únicos que son útiles para mejorar la traducción, hacerlo podría reducir la calidad de sus traducciones.

Límites de los servicios

Amazon Translate tiene las siguientes limitaciones de servicio.

Límites de traducción síncrona en tiempo real

Descripción	Límite
Codificación de caracteres	UTF-8
Tamaño máximo de documento (caracteres UTF-8)	5000 bytes

Límites de conversión por lotes asincrónica

Descripción	Límite
Codificación de caracteres	UTF-8
Tamaño máximo por documento	1 MB
Número máximo de documentos en un lote	1 000 000
Tamaño máximo del total de documentos en un lote	5 GB

Límites de la terminología personalizada

Descripción	Límite
Tamaño máximo del archivo de terminología personalizada	10 MB
Número máximo de terminologías personalizadas por cuenta y por región de AWS	100
Número máximo de idiomas de destino por cada archivo de terminología personalizada	10
Longitud máxima del texto de origen y de destino por término de la terminología personalizada	200 bytes
Número máximo de archivos de terminología por solicitud <code>TranslateText</code> o <code>StartTextTranslationJob</code> .	1

Historial de revisión de Amazon Translate

En la siguiente tabla se describe la documentación de esta versión de Amazon Translate.

- Última actualización de la documentación: 25 de noviembre de 2019

update-history-change	update-history-description	update-history-date
Nuevo idioma	Amazon Translate admite ahora el idioma español (México) para la traducción. Para ver todos los idiomas admitidos, consulte Idiomas y códigos de idioma admitidos .	April 30, 2020
Nueva región	Amazon Translate admite el procesamiento por lotes asincrónico en la región Europa (Londres). Para ver todas las regiones de AWS en las que está disponible el procesamiento por lotes asincrónico, consulte Disponibilidad por región .	April 20, 2020
Nueva característica	Amazon Translate añade funcionalidad de traducción por lotes asincrónica. Para obtener más información, consulte Procesamiento por lotes asincrónico .	December 23, 2019
Nuevas regiones	Amazon Translate añade soporte para las regiones de Asia Pacífico (Hong Kong), Asia Pacífico (Sídney), UE (Londres), UE (París), UE (Estocolmo) y EE.UU. Oeste (Norte de California). Para ver una lista completa de las regiones de AWS que admite Amazon Translate, consulte Tabla de regiones de AWS o Regiones y puntos de enlace de AWS en Referencia general de Amazon Web Services.	November 25, 2019
Idiomas nuevos	Amazon Translate añade nuevos idiomas de traducción: afrikáans, albanés, amárico, azerbaiyano, bengalí, bosnio, búlgaro, croata, dari, eslovaco, esloveno,	November 25, 2019

	estonio, francés (canadiense), georgiano, hausa, letón, pastún, serbio, somalí, suajili, tagalo y tamil. Para ver una lista de las combinaciones de idiomas que puede traducir Amazon Translate directamente, consulte Pares de idiomas compatibles .	
Idiomas nuevos	Amazon Translate añade nuevos idiomas para la traducción: griego, húngaro, rumano, tailandés, ucraniano, urdu y vietnamita. Para ver una lista de las combinaciones de idiomas que puede traducir Amazon Translate directamente, consulte Pares de idiomas compatibles .	October 3, 2019
Nueva característica	Amazon Translate añade Conformidad con FedRAMP . Para obtener más información, consulte Conformidad .	July 31, 2019
Nueva característica	Amazon Translate añade Conformidad con SOC . Para obtener más información, consulte Conformidad .	May 30, 2019
Nuevas regiones	Amazon Translate añade compatibilidad con las regiones de Asia Pacífico (Mumbai), Asia Pacífico (Singapur), Asia Pacífico (Tokio) y Canadá (Central). Para ver una lista completa de las regiones de AWS que admite Amazon Translate, consulte Tabla de regiones de AWS o Regiones y puntos de enlace de AWS en Referencia general de Amazon Web Services.	May 8, 2019
Idiomas nuevos	Amazon Translate agrega idiomas nuevos para la traducción: hindi, malayo, noruego y persa. Para ver una lista de las combinaciones de idiomas que puede traducir Amazon Translate directamente, consulte Pares de idiomas compatibles .	May 6, 2019

Nueva región	Amazon Translate añade compatibilidad con las regiones de UE (Fráncfort) y Asia Pacífico (Seúl). Para ver una lista completa de las regiones de AWS que admite Amazon Translate, consulte Tabla de regiones de AWS o Regiones y puntos de enlace de AWS en Referencia general de Amazon Web Services.	February 28, 2019
Nueva característica	Amazon Translate añade conformidad con PCI . Para obtener más información, consulte Conformidad .	December 12, 2018
Nueva característica	Amazon Translate ha incorporado cuatro nuevas API y la característica de terminología personalizada para ofrecerle más control sobre su traducción. Al utilizar terminología personalizada en sus solicitudes de traducción, tendrá la seguridad de que los nombres de las marcas, los personajes, los modelos y otro contenido único se traducen siempre de la forma deseada, con independencia de la traducción estándar o del contexto. Para obtener más información, consulte Terminología personalizada .	November 27, 2018
Idiomas nuevos	Amazon Translate ahora traduce documentos en los siguientes idiomas: danés, holandés, finés, hebreo, indonesio, coreano, polaco y sueco. Amazon Translate sigue mejorando la traducción directa mediante la reducción significativa del número de pares de idiomas no compatibles. Para ver qué combinaciones de idiomas puede traducir Amazon Translate directamente, consulte Idiomas admitidos .	November 20, 2018

Nueva característica	Amazon Translate ha incorporado la traducción directa entre otros idiomas distintos del inglés. Para ver qué combinaciones de idiomas puede traducir Amazon Translate directamente, consulte Idiomas admitidos .	October 29, 2018
Nueva característica	Amazon Translate ha incorporado la conformidad con HIPAA . Para obtener más información, consulte Conformidad .	October 25, 2018
Nueva característica	Amazon Translate añade nuevos idiomas para la traducción: chino (tradicional), checo, italiano, japonés, ruso y turco. Para obtener una lista de los idiomas que admite Amazon Translate, consulte Idiomas admitidos .	July 17, 2018
Nueva característica	Amazon Translate agrega compatibilidad con la detección automática de idioma de origen. Para obtener más información, consulte Funcionamiento de Amazon Translate .	April 4, 2018
Nueva guía (p. 77)	Esta es la primera versión de la Guía para desarrolladores de Amazon Translate.	November 29, 2017

Referencia de la API

Esta sección contiene la documentación de referencia de la API.

Encabezados HTTP

Además de los encabezados HTTP habituales, las operaciones de Amazon Translate tienen los siguientes encabezados obligatorios:

Encabezado	Valor	Descripción
Content-Type:	application/x-amz-json-1.1	Especifica que el contenido de la solicitud es JSON. También especifica la versión de JSON.
X-Amz-Date:	<Fecha>	La fecha que se utiliza para crear la firma en el encabezado de autorización. El formato debe ser ISO 8601 básico con el formato AAAAMMDD'T'HHMMSS'Z'. Por ejemplo, la siguiente fecha/hora 20180820T184626Z es un valor x-amz-date válido que puede usarse con Amazon Translate. Para obtener más información sobre el uso del encabezado Autorización, consulte Uso de Signature Version 4 con Amazon Translate .
X-Amz-Target:	AWSShineFrontendService_20170701::AmazonTranslate:TranslateText	Operación de Amazon Translate de destino. Por ejemplo, utilice AWSShineFrontendService_20170701::AmazonTranslate:TranslateText para llamar a la operación TranslateText.

Actions

The following actions are supported:

- [DeleteTerminology](#) (p. 83)
- [DescribeTextTranslationJob](#) (p. 85)
- [GetTerminology](#) (p. 87)
- [ImportTerminology](#) (p. 90)
- [ListTerminologies](#) (p. 93)
- [ListTextTranslationJobs](#) (p. 96)
- [StartTextTranslationJob](#) (p. 99)

- [StopTextTranslationJob](#) (p. 103)
- [TranslateText](#) (p. 105)

DeleteTerminology

A synchronous action that deletes a custom terminology.

Request Syntax

```
{  
  "Name": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 123\)](#).

The request accepts the following data in JSON format.

Name (p. 83)

The name of the custom terminology being deleted.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[A-Za-z0-9- _?]+$`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 122\)](#).

InternalServerErrorException

An internal server error occurred. Retry your request.

HTTP Status Code: 500

ResourceNotFoundException

The resource you are looking for has not been found. Review the resource you're looking for and see if a different resource will accomplish your needs before retrying the revised request.

HTTP Status Code: 400

TooManyRequestsException

You have made too many requests within a short period of time. Wait for a short time and then try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DescribeTextTranslationJob

Gets the properties associated with an asynchronous batch translation job including name, ID, status, source and target languages, input/output S3 buckets, and so on.

Request Syntax

```
{  
  "JobId": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 123\)](#).

The request accepts the following data in JSON format.

[JobId \(p. 85\)](#)

The identifier that Amazon Translate generated for the job. The [StartTextTranslationJob \(p. 99\)](#) operation returns this identifier in its response.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32.

Pattern: ^([\p{L}\p{Z}\p{N}_.: /+=\-%@]*)\$

Required: Yes

Response Syntax

```
{  
  "TextTranslationJobProperties": {  
    "DataAccessRoleArn": "string",  
    "EndTime": number,  
    "InputDataConfig": {  
      "ContentType": "string",  
      "S3Uri": "string"  
    },  
    "JobDetails": {  
      "DocumentsWithErrorsCount": number,  
      "InputDocumentsCount": number,  
      "TranslatedDocumentsCount": number  
    },  
    "JobId": "string",  
    "JobName": "string",  
    "JobStatus": "string",  
    "Message": "string",  
    "OutputDataConfig": {  
      "S3Uri": "string"  
    },  
    "SourceLanguageCode": "string",  
    "SubmittedTime": number,  
    "TargetLanguageCodes": [ "string" ],  
    "TerminologyNames": [ "string" ]  
  }  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[TextTranslationJobProperties](#) (p. 85)

An object that contains the properties associated with an asynchronous batch translation job.

Type: [TextTranslationJobProperties](#) (p. 120) object

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 122).

InternalServerError

An internal server error occurred. Retry your request.

HTTP Status Code: 500

ResourceNotFoundException

The resource you are looking for has not been found. Review the resource you're looking for and see if a different resource will accomplish your needs before retrying the revised request.

HTTP Status Code: 400

TooManyRequestsException

You have made too many requests within a short period of time. Wait for a short time and then try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetTerminology

Retrieves a custom terminology.

Request Syntax

```
{  
  "Name": "string",  
  "TerminologyDataFormat": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 123\)](#).

The request accepts the following data in JSON format.

[Name \(p. 87\)](#)

The name of the custom terminology being retrieved.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: ^([A-Za-z0-9-]_?)+\$

Required: Yes

[TerminologyDataFormat \(p. 87\)](#)

The data format of the custom terminology being retrieved, either CSV or TMX.

Type: String

Valid Values: CSV | TMX

Required: Yes

Response Syntax

```
{  
  "TerminologyDataLocation": {  
    "Location": "string",  
    "RepositoryType": "string"  
  },  
  "TerminologyProperties": {  
    "Arn": "string",  
    "CreatedAt": number,  
    "Description": "string",  
    "EncryptionKey": {  
      "Id": "string",  
      "Type": "string"  
    },  
    "LastUpdatedAt": number,  
    "Name": "string",  
    "SizeBytes": number,  
    "SourceLanguageCode": "string",  
    "TargetLanguageCodes": [ "string" ],  
  },  
}
```

```
    "TermCount": number
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[TerminologyDataLocation](#) (p. 87)

The data location of the custom terminology being retrieved. The custom terminology file is returned in a presigned url that has a 30 minute expiration.

Type: [TerminologyDataLocation](#) (p. 116) object

[TerminologyProperties](#) (p. 87)

The properties of the custom terminology being retrieved.

Type: [TerminologyProperties](#) (p. 117) object

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 122).

InternalServerErrorException

An internal server error occurred. Retry your request.

HTTP Status Code: 500

InvalidParameterValueException

The value of the parameter is invalid. Review the value of the parameter you are using to correct it, and then retry your operation.

HTTP Status Code: 400

ResourceNotFoundException

The resource you are looking for has not been found. Review the resource you're looking for and see if a different resource will accomplish your needs before retrying the revised request.

HTTP Status Code: 400

TooManyRequestsException

You have made too many requests within a short period of time. Wait for a short time and then try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ImportTerminology

Creates or updates a custom terminology, depending on whether or not one already exists for the given terminology name. Importing a terminology with the same name as an existing one will merge the terminologies based on the chosen merge strategy. Currently, the only supported merge strategy is OVERWRITE, and so the imported terminology will overwrite an existing terminology of the same name.

If you import a terminology that overwrites an existing one, the new terminology take up to 10 minutes to fully propagate and be available for use in a translation due to cache policies with the DataPlane service that performs the translations.

Request Syntax

```
{
  "Description": "string",
  "EncryptionKey": {
    "Id": "string",
    "Type": "string"
  },
  "MergeStrategy": "string",
  "Name": "string",
  "TerminologyData": {
    "File": blob,
    "Format": "string"
  }
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 123\)](#).

The request accepts the following data in JSON format.

[Description \(p. 90\)](#)

The description of the custom terminology being imported.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `[\P{M}\p{M}] { 0 , 256 }`

Required: No

[EncryptionKey \(p. 90\)](#)

The encryption key for the custom terminology being imported.

Type: [EncryptionKey \(p. 110\)](#) object

Required: No

[MergeStrategy \(p. 90\)](#)

The merge strategy of the custom terminology being imported. Currently, only the OVERWRITE merge strategy is supported. In this case, the imported terminology will overwrite an existing terminology of the same name.

Type: String

Valid Values: `OVERWRITE`

Required: Yes

[Name \(p. 90\)](#)

The name of the custom terminology being imported.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^([A-Za-z0-9-]_?)+$`

Required: Yes

[TerminologyData \(p. 90\)](#)

The terminology data for the custom terminology being imported.

Type: [TerminologyData \(p. 115\)](#) object

Required: Yes

Response Syntax

```
{
  "TerminologyProperties": {
    "Arn": "string",
    "CreatedAt": number,
    "Description": "string",
    "EncryptionKey": {
      "Id": "string",
      "Type": "string"
    },
    "LastUpdatedAt": number,
    "Name": "string",
    "SizeBytes": number,
    "SourceLanguageCode": "string",
    "TargetLanguageCodes": [ "string" ],
    "TermCount": number
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[TerminologyProperties \(p. 91\)](#)

The properties of the custom terminology being imported.

Type: [TerminologyProperties \(p. 117\)](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 122\)](#).

InternalServerErrorException

An internal server error occurred. Retry your request.

HTTP Status Code: 500

InvalidParameterValueException

The value of the parameter is invalid. Review the value of the parameter you are using to correct it, and then retry your operation.

HTTP Status Code: 400

LimitExceededException

The specified limit has been exceeded. Review your request and retry it with a quantity below the stated limit.

HTTP Status Code: 400

TooManyRequestsException

You have made too many requests within a short period of time. Wait for a short time and then try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListTerminologies

Provides a list of custom terminologies associated with your account.

Request Syntax

```
{  
  "MaxResults": number,  
  "NextToken": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 123\)](#).

The request accepts the following data in JSON format.

MaxResults (p. 93)

The maximum number of custom terminologies returned per list request.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500.

Required: No

NextToken (p. 93)

If the result of the request to ListTerminologies was truncated, include the NextToken to fetch the next group of custom terminologies.

Type: String

Length Constraints: Maximum length of 8192.

Pattern: \p{ASCII}{0,8192}

Required: No

Response Syntax

```
{  
  "NextToken": "string",  
  "TerminologyPropertiesList": [  
    {  
      "Arn": "string",  
      "CreatedAt": number,  
      "Description": "string",  
      "EncryptionKey": {  
        "Id": "string",  
        "Type": "string"  
      },  
      "LastUpdatedAt": number,  
      "Name": "string",  
      "SizeBytes": number,  
      "SourceLanguageCode": "string",  
      "TargetLanguageCodes": [ "string" ],  
      "TermCount": number  
    },  
    ...  
  ]  
}
```

```
}  
  ]  
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[NextToken](#) (p. 93)

If the response to the `ListTerminologies` was truncated, the `NextToken` fetches the next group of custom terminologies.

Type: String

Length Constraints: Maximum length of 8192.

Pattern: `\p{ASCII}{0,8192}`

[TerminologyPropertiesList](#) (p. 93)

The properties list of the custom terminologies returned on the list request.

Type: Array of [TerminologyProperties](#) (p. 117) objects

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 122).

`InternalServerError`

An internal server error occurred. Retry your request.

HTTP Status Code: 500

`InvalidParameterValueException`

The value of the parameter is invalid. Review the value of the parameter you are using to correct it, and then retry your operation.

HTTP Status Code: 400

`TooManyRequestsException`

You have made too many requests within a short period of time. Wait for a short time and then try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)

- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListTextTranslationJobs

Gets a list of the batch translation jobs that you have submitted.

Request Syntax

```
{
  "Filter": {
    "JobName": "string",
    "JobStatus": "string",
    "SubmittedAfterTime": number,
    "SubmittedBeforeTime": number
  },
  "MaxResults": number,
  "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 123\)](#).

The request accepts the following data in JSON format.

[Filter \(p. 96\)](#)

The parameters that specify which batch translation jobs to retrieve. Filters include job name, job status, and submission time. You can only set one filter at a time.

Type: [TextTranslationJobFilter \(p. 119\)](#) object

Required: No

[MaxResults \(p. 96\)](#)

The maximum number of results to return in each page. The default value is 100.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500.

Required: No

[NextToken \(p. 96\)](#)

The token to request the next page of results.

Type: String

Length Constraints: Maximum length of 8192.

Pattern: `\p{ASCII}{0,8192}`

Required: No

Response Syntax

```
{
  "NextToken": "string",
  "TextTranslationJobPropertiesList": [
    {

```

```
    "DataAccessRoleArn": "string",
    "EndTime": number,
    "InputDataConfig": {
      "ContentType": "string",
      "S3Uri": "string"
    },
    "JobDetails": {
      "DocumentsWithErrorsCount": number,
      "InputDocumentsCount": number,
      "TranslatedDocumentsCount": number
    },
    "JobId": "string",
    "JobName": "string",
    "JobStatus": "string",
    "Message": "string",
    "OutputDataConfig": {
      "S3Uri": "string"
    },
    "SourceLanguageCode": "string",
    "SubmittedTime": number,
    "TargetLanguageCodes": [ "string" ],
    "TerminologyNames": [ "string" ]
  }
]
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[NextToken \(p. 96\)](#)

The token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

Type: String

Length Constraints: Maximum length of 8192.

Pattern: `\p{ASCII}{0,8192}`

[TextTranslationJobPropertiesList \(p. 96\)](#)

A list containing the properties of each job that is returned.

Type: Array of [TextTranslationJobProperties \(p. 120\)](#) objects

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 122\)](#).

InternalServerError

An internal server error occurred. Retry your request.

HTTP Status Code: 500

InvalidFilterException

The filter specified for the operation is invalid. Specify a different filter.

HTTP Status Code: 400

InvalidRequestException

The request that you made is invalid. Check your request to determine why it's invalid and then retry the request.

HTTP Status Code: 400

TooManyRequestsException

You have made too many requests within a short period of time. Wait for a short time and then try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

StartTextTranslationJob

Starts an asynchronous batch translation job. Batch translation jobs can be used to translate large volumes of text across multiple documents at once. For more information, see [Procesamiento por lotes asincrónico](#) (p. 19).

Batch translation jobs can be described with the [DescribeTextTranslationJob](#) (p. 85) operation, listed with the [ListTextTranslationJobs](#) (p. 96) operation, and stopped with the [StopTextTranslationJob](#) (p. 103) operation.

Note

Amazon Translate does not support batch translation of multiple source languages at once.

Request Syntax

```
{
  "ClientToken": "string",
  "DataAccessRoleArn": "string",
  "InputDataConfig": {
    "ContentType": "string",
    "S3Uri": "string"
  },
  "JobName": "string",
  "OutputDataConfig": {
    "S3Uri": "string"
  },
  "SourceLanguageCode": "string",
  "TargetLanguageCodes": [ "string" ],
  "TerminologyNames": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 123).

The request accepts the following data in JSON format.

[ClientToken](#) (p. 99)

A unique identifier for the request. This token is auto-generated when using the Amazon Translate SDK.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-]+$`

Required: Yes

[DataAccessRoleArn](#) (p. 99)

The Amazon Resource Name (ARN) of an AWS Identity Access and Management (IAM) role that grants Amazon Translate read access to your input data. For more information, see [Administración de identidades y accesos en Amazon Translate](#) (p. 54).

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:aws(-[^\:]+)?:iam::[0-9]{12}:role/.+`

Required: Yes

[InputDataConfig \(p. 99\)](#)

Specifies the format and S3 location of the input documents for the translation job.

Type: [InputDataConfig \(p. 111\)](#) object

Required: Yes

[JobName \(p. 99\)](#)

The name of the batch translation job to be performed.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^([\p{L}\p{Z}\p{N}_.: /+=\-%@]*)$`

Required: No

[OutputDataConfig \(p. 99\)](#)

Specifies the S3 folder to which your job output will be saved.

Type: [OutputDataConfig \(p. 113\)](#) object

Required: Yes

[SourceLanguageCode \(p. 99\)](#)

The language code of the input language. For a list of language codes, see [Idiomas y códigos de idioma admitidos \(p. 1\)](#).

Amazon Translate does not automatically detect a source language during batch translation jobs.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 5.

Required: Yes

[TargetLanguageCodes \(p. 99\)](#)

The language code of the output language.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Minimum length of 2. Maximum length of 5.

Required: Yes

[TerminologyNames \(p. 99\)](#)

The name of the terminology to use in the batch translation job. For a list of available terminologies, use the [ListTerminologies \(p. 93\)](#) operation.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[A-Za-z0-9-]_?)+$`

Required: No

Response Syntax

```
{
  "JobId": "string",
  "JobStatus": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[JobId \(p. 101\)](#)

The identifier generated for the job. To get the status of a job, use this ID with the [DescribeTextTranslationJob \(p. 85\)](#) operation.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32.

Pattern: `^([\p{L}\p{Z}\p{N}_.: /+=\-%@]*)$`

[JobStatus \(p. 101\)](#)

The status of the job. Possible values include:

- **SUBMITTED** - The job has been received and is queued for processing.
- **IN_PROGRESS** - Amazon Translate is processing the job.
- **COMPLETED** - The job was successfully completed and the output is available.
- **COMPLETED_WITH_ERROR** - The job was completed with errors. The errors can be analyzed in the job's output.
- **FAILED** - The job did not complete. To get details, use the [DescribeTextTranslationJob \(p. 85\)](#) operation.
- **STOP_REQUESTED** - The user who started the job has requested that it be stopped.
- **STOPPED** - The job has been stopped.

Type: String

Valid Values: **SUBMITTED** | **IN_PROGRESS** | **COMPLETED** | **COMPLETED_WITH_ERROR** | **FAILED** | **STOP_REQUESTED** | **STOPPED**

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 122\)](#).

InternalServerErrorException

An internal server error occurred. Retry your request.

HTTP Status Code: 500

InvalidRequestException

The request that you made is invalid. Check your request to determine why it's invalid and then retry the request.

HTTP Status Code: 400

ResourceNotFoundException

The resource you are looking for has not been found. Review the resource you're looking for and see if a different resource will accomplish your needs before retrying the revised request.

HTTP Status Code: 400

TooManyRequestsException

You have made too many requests within a short period of time. Wait for a short time and then try your request again.

HTTP Status Code: 400

UnsupportedLanguagePairException

Amazon Translate does not support translation from the language of the source text into the requested target language. For more information, see [Tratamiento de excepciones \(p. 5\)](#).

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

StopTextTranslationJob

Stops an asynchronous batch translation job that is in progress.

If the job's state is `IN_PROGRESS`, the job will be marked for termination and put into the `STOP_REQUESTED` state. If the job completes before it can be stopped, it is put into the `COMPLETED` state. Otherwise, the job is put into the `STOPPED` state.

Asynchronous batch translation jobs are started with the [StartTextTranslationJob](#) (p. 99) operation. You can use the [DescribeTextTranslationJob](#) (p. 85) or [ListTextTranslationJobs](#) (p. 96) operations to get a batch translation job's `JobId`.

Request Syntax

```
{
  "JobId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 123).

The request accepts the following data in JSON format.

[JobId](#) (p. 103)

The job ID of the job to be stopped.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32.

Pattern: `^([\p{L}\p{Z}\p{N}_.: /+=\-%@]*)$`

Required: Yes

Response Syntax

```
{
  "JobId": "string",
  "JobStatus": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[JobId](#) (p. 103)

The job ID of the stopped batch translation job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32.

Pattern: `^([\p{L}\p{Z}\p{N}_.: /+=\-%@]*)$`

[JobStatus](#) (p. 103)

The status of the designated job. Upon successful completion, the job's status will be `STOPPED`.

Type: String

Valid Values: `SUBMITTED` | `IN_PROGRESS` | `COMPLETED` | `COMPLETED_WITH_ERROR` | `FAILED` | `STOP_REQUESTED` | `STOPPED`

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 122).

InternalServerErrorException

An internal server error occurred. Retry your request.

HTTP Status Code: 500

ResourceNotFoundException

The resource you are looking for has not been found. Review the resource you're looking for and see if a different resource will accomplish your needs before retrying the revised request.

HTTP Status Code: 400

TooManyRequestsException

You have made too many requests within a short period of time. Wait for a short time and then try your request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

TranslateText

Translates input text from the source language to the target language. For a list of available languages and language codes, see [Idiomas y códigos de idioma admitidos \(p. 1\)](#).

Request Syntax

```
{
  "SourceLanguageCode": "string",
  "TargetLanguageCode": "string",
  "TerminologyNames": [ "string" ],
  "Text": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 123\)](#).

The request accepts the following data in JSON format.

[SourceLanguageCode \(p. 105\)](#)

The language code for the language of the source text. The language must be a language supported by Amazon Translate. For a list of language codes, see [Idiomas y códigos de idioma admitidos \(p. 1\)](#).

To have Amazon Translate determine the source language of your text, you can specify `auto` in the `SourceLanguageCode` field. If you specify `auto`, Amazon Translate will call [Amazon Comprehend](#) to determine the source language.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 5.

Required: Yes

[TargetLanguageCode \(p. 105\)](#)

The language code requested for the language of the target text. The language must be a language supported by Amazon Translate.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 5.

Required: Yes

[TerminologyNames \(p. 105\)](#)

The name of the terminology list file to be used in the `TranslateText` request. You can use 1 terminology list at most in a `TranslateText` request. Terminology lists can contain a maximum of 256 terms.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[A-Za-z0-9-]_?)+$`

Required: No

[Text \(p. 105\)](#)

The text to translate. The text string can be a maximum of 5,000 bytes long. Depending on your character set, this may be fewer than 5,000 characters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5000.

Pattern: `[\P{M}\p{M}]{1,5000}`

Required: Yes

Response Syntax

```
{
  "AppliedTerminologies": [
    {
      "Name": "string",
      "Terms": [
        {
          "SourceText": "string",
          "TargetText": "string"
        }
      ]
    }
  ],
  "SourceLanguageCode": "string",
  "TargetLanguageCode": "string",
  "TranslatedText": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

[AppliedTerminologies \(p. 106\)](#)

The names of the custom terminologies applied to the input text by Amazon Translate for the translated text response.

Type: Array of [AppliedTerminology \(p. 109\)](#) objects

[SourceLanguageCode \(p. 106\)](#)

The language code for the language of the source text.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 5.

[TargetLanguageCode \(p. 106\)](#)

The language code for the language of the target text.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 5.

TranslatedText (p. 106)

The translated text.

Type: String

Length Constraints: Maximum length of 10000.

Pattern: [\P{M}\p{M}]{0,10000}

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 122\)](#).

DetectedLanguageLowConfidenceException

The confidence that Amazon Comprehend accurately detected the source language is low. If a low confidence level is acceptable for your application, you can use the language in the exception to call Amazon Translate again. For more information, see the [DetectDominantLanguage](#) operation in the Amazon Comprehend Developer Guide.

HTTP Status Code: 400

InternalServerErrorException

An internal server error occurred. Retry your request.

HTTP Status Code: 500

InvalidRequestException

The request that you made is invalid. Check your request to determine why it's invalid and then retry the request.

HTTP Status Code: 400

ResourceNotFoundException

The resource you are looking for has not been found. Review the resource you're looking for and see if a different resource will accomplish your needs before retrying the revised request.

HTTP Status Code: 400

ServiceUnavailableException

The Amazon Translate service is temporarily unavailable. Please wait a bit and then retry your request.

HTTP Status Code: 500

TextSizeLimitExceededException

The size of the text you submitted exceeds the size limit. Reduce the size of the text or use a smaller document and then retry your request.

HTTP Status Code: 400

TooManyRequestsException

You have made too many requests within a short period of time. Wait for a short time and then try your request again.

HTTP Status Code: 400

UnsupportedLanguagePairException

Amazon Translate does not support translation from the language of the source text into the requested target language. For more information, see [Tratamiento de excepciones \(p. 5\)](#).

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

Data Types

The following data types are supported:

- [AppliedTerminology \(p. 109\)](#)
- [EncryptionKey \(p. 110\)](#)
- [InputDataConfig \(p. 111\)](#)
- [JobDetails \(p. 112\)](#)
- [OutputDataConfig \(p. 113\)](#)
- [Term \(p. 114\)](#)
- [TerminologyData \(p. 115\)](#)
- [TerminologyDataLocation \(p. 116\)](#)
- [TerminologyProperties \(p. 117\)](#)
- [TextTranslationJobFilter \(p. 119\)](#)
- [TextTranslationJobProperties \(p. 120\)](#)

AppliedTerminology

The custom terminology applied to the input text by Amazon Translate for the translated text response. This is optional in the response and will only be present if you specified terminology input in the request. Currently, only one terminology can be applied per TranslateText request.

Contents

Name

The name of the custom terminology applied to the input text by Amazon Translate for the translated text response.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^[A-Za-z0-9-]_?)+$`

Required: No

Terms

The specific terms of the custom terminology applied to the input text by Amazon Translate for the translated text response. A maximum of 250 terms will be returned, and the specific terms applied will be the first 250 terms in the source text.

Type: Array of [Term \(p. 114\)](#) objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

EncryptionKey

The encryption key used to encrypt the custom terminologies used by Amazon Translate.

Contents

Id

The Amazon Resource Name (ARN) of the encryption key being used to encrypt the custom terminology.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 400.

Pattern: `(arn:aws((-us-gov)|(-iso)|(-iso-b)|(-cn))?:kms:)?([a-z]{2}-[a-z]+(-[a-z]+)?-\d:)?(\d{12}:)?(((key/)?[a-zA-Z0-9-_.]+)|(alias/[a-zA-Z0-9:_-]+))`

Required: Yes

Type

The type of encryption key used by Amazon Translate to encrypt custom terminologies.

Type: String

Valid Values: `KMS`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

InputDataConfig

The input configuration properties for requesting a batch translation job.

Contents

ContentType

The multipurpose internet mail extension (MIME) type of the input files. Valid values are `text/plain` for plaintext files and `text/html` for HTML files.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `^[-\w.]+\./[-\w.+]+$`

Required: Yes

S3Uri

The URI of the AWS S3 folder that contains the input file. The folder must be in the same Region as the API endpoint you are calling.

Type: String

Length Constraints: Maximum length of 1024.

Pattern: `s3://[a-z0-9][\.-\a-z0-9]{1,61}[a-z0-9](/*.*)?`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

JobDetails

The number of documents successfully and unsuccessfully processed during a translation job.

Contents

DocumentsWithErrorsCount

The number of documents that could not be processed during a translation job.

Type: Integer

Required: No

InputDocumentsCount

The number of documents used as input in a translation job.

Type: Integer

Required: No

TranslatedDocumentsCount

The number of documents successfully processed during a translation job.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

OutputDataConfig

The output configuration properties for a batch translation job.

Contents

S3Uri

The URI of the S3 folder that contains a translation job's output file. The folder must be in the same Region as the API endpoint that you are calling.

Type: String

Length Constraints: Maximum length of 1024.

Pattern: `s3://[a-z0-9][\.-\a-z0-9]{1,61}[a-z0-9](/*.*)?`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

Term

The term being translated by the custom terminology.

Contents

SourceText

The source text of the term being translated by the custom terminology.

Type: String

Length Constraints: Maximum length of 10000.

Pattern: `[\P{M}\p{M}] { 0 , 10000 }`

Required: No

TargetText

The target text of the term being translated by the custom terminology.

Type: String

Length Constraints: Maximum length of 10000.

Pattern: `[\P{M}\p{M}] { 0 , 10000 }`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

TerminologyData

The data associated with the custom terminology.

Contents

File

The file containing the custom terminology data. Your version of the AWS SDK performs a Base64-encoding on this field before sending a request to the AWS service. Users of the SDK should not perform Base64-encoding themselves.

Type: Base64-encoded binary data object

Length Constraints: Maximum length of 10485760.

Required: Yes

Format

The data format of the custom terminology. Either CSV or TMX.

Type: String

Valid Values: `CSV` | `TMX`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

TerminologyDataLocation

The location of the custom terminology data.

Contents

Location

The location of the custom terminology data.

Type: String

Length Constraints: Maximum length of 10000.

Pattern: `[\P{M}\p{M}] { 0 , 10000 }`

Required: Yes

RepositoryType

The repository type for the custom terminology data.

Type: String

Length Constraints: Maximum length of 10000.

Pattern: `[\P{M}\p{M}] { 0 , 10000 }`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

TerminologyProperties

The properties of the custom terminology.

Contents

Arn

The Amazon Resource Name (ARN) of the custom terminology.

Type: String

Pattern: `^arn:aws((-us-gov)|(-iso)|(-iso-b)|(-cn))?:translate:[a-zA-Z0-9-]+:[0-9]{12}:terminology/.+?/.+?$`

Required: No

CreatedAt

The time at which the custom terminology was created, based on the timestamp.

Type: Timestamp

Required: No

Description

The description of the custom terminology properties.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `[\P{M}\p{M}]{0,256}`

Required: No

EncryptionKey

The encryption key for the custom terminology.

Type: [EncryptionKey](#) (p. 110) object

Required: No

LastUpdatedAt

The time at which the custom terminology was last update, based on the timestamp.

Type: Timestamp

Required: No

Name

The name of the custom terminology.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^([A-Za-z0-9-]_?)+$`

Required: No

SizeBytes

The size of the file used when importing a custom terminology.

Type: Integer

Required: No

SourceLanguageCode

The language code for the source text of the translation request for which the custom terminology is being used.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 5.

Required: No

TargetLanguageCodes

The language codes for the target languages available with the custom terminology file. All possible target languages are returned in array.

Type: Array of strings

Length Constraints: Minimum length of 2. Maximum length of 5.

Required: No

TermCount

The number of terms included in the custom terminology.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

TextTranslationJobFilter

Provides information for filtering a list of translation jobs. For more information, see [ListTextTranslationJobs](#) (p. 96).

Contents

JobName

Filters the list of jobs by name.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^([\p{L}\p{Z}\p{N}_.: /+=\-%@]*)$`

Required: No

JobStatus

Filters the list of jobs based by job status.

Type: String

Valid Values: `SUBMITTED` | `IN_PROGRESS` | `COMPLETED` | `COMPLETED_WITH_ERROR` | `FAILED` | `STOP_REQUESTED` | `STOPPED`

Required: No

SubmittedAfterTime

Filters the list of jobs based on the time that the job was submitted for processing and returns only the jobs submitted after the specified time. Jobs are returned in descending order, newest to oldest.

Type: Timestamp

Required: No

SubmittedBeforeTime

Filters the list of jobs based on the time that the job was submitted for processing and returns only the jobs submitted before the specified time. Jobs are returned in ascending order, oldest to newest.

Type: Timestamp

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

TextTranslationJobProperties

Provides information about a translation job.

Contents

DataAccessRoleArn

The Amazon Resource Name (ARN) of an AWS Identity Access and Management (IAM) role that granted Amazon Translate read access to the job's input data.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:aws(-[^:]+)?:iam:[0-9]{12}:role/.+`

Required: No

EndTime

The time at which the translation job ended.

Type: Timestamp

Required: No

InputDataConfig

The input configuration properties that were specified when the job was requested.

Type: [InputDataConfig \(p. 111\)](#) object

Required: No

JobDetails

The number of documents successfully and unsuccessfully processed during the translation job.

Type: [JobDetails \(p. 112\)](#) object

Required: No

JobId

The ID of the translation job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32.

Pattern: `^([\p{L}\p{Z}\p{N}_.:/+=\-%@]*)$`

Required: No

JobName

The user-defined name of the translation job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^([\p{L}\p{Z}\p{N}_.:/+=\-%@]*)$`

Required: No

JobStatus

The status of the translation job.

Type: String

Valid Values: SUBMITTED | IN_PROGRESS | COMPLETED | COMPLETED_WITH_ERROR | FAILED | STOP_REQUESTED | STOPPED

Required: No

Message

An explanation of any errors that may have occurred during the translation job.

Type: String

Required: No

OutputDataConfig

The output configuration properties that were specified when the job was requested.

Type: [OutputDataConfig \(p. 113\)](#) object

Required: No

SourceLanguageCode

The language code of the language of the source text. The language must be a language supported by Amazon Translate.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 5.

Required: No

SubmittedTime

The time at which the translation job was submitted.

Type: Timestamp

Required: No

TargetLanguageCodes

The language code of the language of the target text. The language must be a language supported by Amazon Translate.

Type: Array of strings

Array Members: Fixed number of 1 item.

Length Constraints: Minimum length of 2. Maximum length of 5.

Required: No

TerminologyNames

A list containing the names of the terminologies applied to a translation job. Only one terminology can be applied per [StartTextTranslationJob \(p. 99\)](#) request at this time.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `^([A-Za-z0-9-]_?)+$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java](#)
- [AWS SDK for Ruby V3](#)

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information

about Signature Version 4, see [Signature Version 4 Signing Process](#) in the Amazon Web Services General Reference.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: `access_key/YYYYMMDD/region/service/aws4_request`.

For more information, see [Task 2: Create a String to Sign for Signature Version 4](#) in the Amazon Web Services General Reference.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see [Handling Dates in Signature Version 4](#) in the Amazon Web Services General Reference.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to [AWS Services That Work with IAM](#) in the IAM User Guide.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see [Task 1: Create a Canonical Request For Signature Version 4](#) in the Amazon Web Services General Reference.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

AWS glossary

For the latest AWS terminology, see the [AWS glossary](#) in the AWS General Reference.