



デベロッパーガイド

Amazon Translate



Amazon Translate: デベロッパーガイド

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

Amazon の商標およびトレードドレスは、Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は、Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

Amazon Translate とは	1
ユースケース	1
初めてお使いになる方	2
Amazon Translate の料金	2
Amazon Translate API リファレンス	2
サポートされている言語	3
サポートされている言語	3
Amazon Translate 機能でサポートされる言語	7
仕組み	8
入力コンテンツでサポートされるフォーマット	8
翻訳のカスタマイズ	8
言語の自動検出	9
例外処理	9
設定	10
にサインアップする AWS アカウント	10
管理アクセスを持つユーザーを作成する	11
のセットアップ AWS CLI	12
プログラマチックアクセス権を付与する	12
の使用 AWS SDKs	14
開始	16
開始方法 (コンソール)	16
開始方法 (AWS CLI)	18
コマンドラインを使用したテキストの翻訳	18
次のステップ	19
開始方法 (SDK)	19
SDK for Java の使用	19
AWS SDK for Python の使用	20
その他の SDK サンプル	20
翻訳の処理モード	21
リアルタイム翻訳	21
リアルタイム翻訳 (コンソール)	21
リアルタイム翻訳 (API)	26
非同期バッチ処理	29
利用可能なリージョン	29

前提条件	30
ジョブの実行	35
モニタリングと分析	42
結果の取得	44
翻訳のカスタマイズ	47
do-not-translate タグを使用する	47
do-not-translate コンソールでの使用	48
API do-not-translate との併用	48
カスタム用語によるカスタマイズ	49
カスタム用語の作成	50
カスタム用語の使用	52
Python 用 SDK を使用した例	53
用語の暗号化	55
ベストプラクティス	55
簡潔さを使う	57
簡潔設定を使用する	57
サポートされている言語	57
マスキング、冒涇	58
不敬表現設定の使用	59
サポートされていない言語	59
形式を設定する	60
形式設定の使用	60
サポートされている言語	61
並列データによるカスタマイズ	62
利用可能なリージョン	63
Amazon Translate の並列データ入力ファイル	64
並列データの追加	70
並列データの表示と管理	73
コードの例	78
基本	78
アクション	79
シナリオ	98
Amazon Transcribe ストリーミングアプリケーションを構築する	98
Amazon Lex chatbot を構築する	99
Amazon SNSアプリケーションの構築	100
顧客からのフィードバックを分析するアプリケーションの作成	101

翻訳ジョブを開始する	108
タグ付け	111
新しいリソースへのタグ付け	112
タグの表示、編集、削除	113
セキュリティ	115
データ保護	116
保管中の暗号化	117
転送中の暗号化	118
Identity and Access Management	118
対象者	119
アイデンティティを使用した認証	119
ポリシーを使用したアクセスの管理	123
Amazon Translate と の連携方法 IAM	125
アイデンティティベースのポリシーの例	133
AWS マネージドポリシー	139
トラブルシューティング	142
モニタリング	144
によるモニタリング CloudWatch	147
を使用した Amazon Translate API呼び出しのログ記録 AWS CloudTrail	149
CloudWatch Amazon Translate の メトリクスとディメンション	151
によるモニタリング EventBridge	153
コンプライアンス検証	156
耐障害性	157
インフラストラクチャセキュリティ	157
VPC エンドポイント (AWS PrivateLink)	158
Amazon Translate VPCエンドポイントに関する考慮事項	158
Amazon Translate のインターフェイスVPCエンドポイントの作成	158
Amazon Translate のVPCエンドポイントポリシーの作成	159
ガイドラインとクォータ	161
サポートされている AWS リージョン	161
コンプライアンス	161
スロットリング	161
ガイドライン	161
Service Quotas	162
ドキュメント履歴	165
API リファレンス	180

AWS 用語集	181
.....	clxxxii

Amazon Translate とは

Amazon Translate は高度な機械学習テクノロジーを使用して、高品質の翻訳をオンデマンドで提供するテキスト翻訳サービスです。Amazon Translate を使用すれば、非構造化テキストドキュメントの翻訳や、複数の言語で動作するアプリケーションの構築が可能です。Amazon Translate でサポートされているすべての言語については、「[サポートされている言語と言語コード](#)」を参照してください。

トピック

- [ユースケース](#)
- [Amazon Translate を初めてお使いになる方へ](#)
- [Amazon Translate の料金](#)
- [Amazon Translate API リファレンス](#)

ユースケース

Amazon Translate を使用して以下を行います。

Amazon Translate を組み込めば、アプリケーションで複数言語のユーザーエクスペリエンスを実現できます。

- ミーティングメモ、技術者によるレポート、ナレッジベースの記事、投稿など、社内で作成したコンテンツを翻訳できます。
- メール、ゲーム内チャット、カスタマーサービスなど、対人コミュニケーションを翻訳できるので、顧客と社員のニーズに合わせた言語で連絡を交わすことができます。

会社の受信データを処理および管理できます。

- ソーシャルメディア、ニュースフィードなどのテキストをさまざまな言語で分析できます。
- eDiscovery のケースなど、さまざまな言語で情報を検索できます。

Amazon Translate を他の AWS のサービスに組み込めば、言語に依存しない処理が可能になります。

- [Amazon Comprehend](#) を使用したソーシャルメディアストリームなどといった非構造化テキストから、ネームドエンティティ (固有表現)、センチメント、キーフレーズを抽出できます。
- [Amazon Transcribe](#) を使って字幕やライブキャプションをさまざまな言語で作成できます。
- [Amazon Polly](#) を使用して、翻訳されたコンテンツを読み上げることができます。

- [Amazon S3](#) に保存されているドキュメントレポジトリを翻訳できます。
- データベース [Amazon DynamoDB](#)、[Amazon Aurora](#)、[Amazon Redshift](#) に保存されているテキストを翻訳できます。
- [AWS Lambda](#) または [AWS Glue](#) を使用してワークフローをシームレスに統合できます。

Amazon Translate を初めてお使いになる方へ

Amazon Translate を初めてお使いになる場合は、以下のトピックから開始することをお勧めします。

1. [Amazon Translate の仕組み](#) — Amazon Translate の概要です。
2. [Amazon Translate の開始方法](#) – AWS アカウントを設定し、Amazon Translate の使用を開始する方法が説明されています。
3. [を使用した Amazon Translate のコード例 AWS SDKs](#)— コード例を使用して Amazon Translate API を調べてください。

以下のリソースを使用して Amazon Translate サービスについて学ぶこともできます。

- [AWS 機械学習 ブログ](#)には、Amazon Translate に関する役立つ記事が掲載されています。
- [Amazon Translate ディープダイブビデオシリーズ](#)では、Amazon Translate に関する紹介動画を提供しています。

Amazon Translate の料金

他の AWS 製品と同様、Amazon Translate を使用するための契約や最低契約金は必要ありません。Amazon Translate の使用料金の詳細については、「[Amazon Translate 料金表](#)」を参照してください。

Amazon Translate API リファレンス

「Amazon Translate API リファレンス」は独立したドキュメントになりました。詳細については、「[Amazon Translate API リファレンス](#)」を参照してください。

サポートされている言語と言語コード

Amazon Translate ではソース言語 (入力言語) をターゲット言語 (出力言語) に翻訳されます。ソース言語とターゲット言語の組み合わせは、言語ペアと呼ばれます。

Note

Amazon Translate では、ソース言語とターゲット言語に同じ言語を指定した翻訳に対して課金されることはありません。ソース言語を auto に設定した場合、自動検出の使用に対して課金される場合があります。詳細については、「[言語の自動検出](#)」を参照してください。

トピック

- [サポートされている言語](#)
- [Amazon Translate 機能でサポートされる言語](#)

サポートされている言語

Amazon Translate は、以下の表に示されている言語間のテキスト翻訳をサポートしています。言語コード列には [ISO 639-1](#) の 2 桁の言語コードが使用されています。言語の国別バージョンについては、この表は [RFC 5646](#) 形式に従い、ダッシュの後に [ISO 3166](#) によって定められた 2 桁の国コードを付加しています。例えば、メキシコのスペイン語の言語コードは es-MX です。

言語	言語コード
アフリカーンス語	af
アルバニア語	sq
アムハラ語	am
アラビア語	ar
アルメニア語	hy
アゼルバイジャン語	az

言語	言語コード
ベンガル語	bn
ボスニア語	bs
ブルガリア語	bg
カタロニア語	ca
簡体字中国語	zh
繁体字中国語	zh-TW
クロアチア語	hr
チェコ語	cs
デンマーク語	da
ダリ語	fa-AF
オランダ語	nl
英語	en
エストニア語	et
ファルシ語 (ペルシア語)	fa
フィリピン語、タガログ語	tl
フィンランド語	fi
フランス語	fr
フランス語 (カナダ)	fr-CA
グルジア語	ka
ドイツ語	de

言語	言語コード
ギリシャ語	el
グジャラート語	gu
ハイチクレオール語	ht
ハウサ語	ha
ヘブライ語	he
ヒンディー語	hi
ハンガリー語	hu
アイスランド語	is
インドネシア語	id
アイルランド語	ga
イタリア語	it
日本語	ja
カンナダ語	kn
カザフ語	kk
韓国語	ko
ラトビア語	lv
リトアニア語	lt
マケドニア語	mk
マレー語	ms
マラヤーラム語	ml

言語	言語コード
マルタ語	mt
マラーティー語	mr
モンゴル語	mn
ノルウェー語 (ブークモール)	no
パシュト語	ps
ポーランド語	pl
ポルトガル語 (ブラジル)	pt
ポルトガル語 (ポルトガル)	pt-PT
パンジャブ語	pa
ルーマニア語	ro
ロシア語	ru
セルビア語	sr
シンハラ語	si
スロバキア語	sk
スロベニア語	sl
ソマリ語	so
スペイン語	es
スペイン語 (メキシコ)	es-MX
スワヒリ語	sw
スウェーデン語	sv

言語	言語コード
タミル語	ta
テルグ語	te
タイ語	th
トルコ語	tr
ウクライナ語	uk
ウルドゥー語	ur
ウズベク語	uz
ベトナム語	vi
ウェールズ語	cy

Amazon Translate 機能でサポートされる言語

以下のセクションでは、Amazon Translate 機能でサポートされている言語について説明します。

- リアルタイムのドキュメント翻訳 — 英語からサポートされている任意の言語への翻訳と、サポートされている任意の言語から英語への翻訳をサポートします。リアルタイム翻訳の詳細については、「[リアルタイム翻訳](#)」を参照してください。
- 簡潔さ — この機能でサポートされる言語については、[を参照してください](#) [Amazon Translate での簡潔さの使用](#)。
- 不敬マスキング — この機能でサポートされている言語については「[Amazon Translate における適切な言葉やフレーズのマスキング](#)」を参照してください。
- 形式 — この機能でサポートされている言語については「[Amazon Translate の形式設定](#)」を参照してください。

Amazon Translate の仕組み

Amazon Translate サービスを使用して、コンテンツをソース言語 (入力コンテンツの言語) からターゲット言語 (翻訳出力用に選択した言語) に翻訳します。バッチジョブでは、1 つ以上のソース言語から 1 つ以上のターゲット言語にファイルを翻訳できます。サポートされている言語の詳細については、[を参照してください](#) [サポートされている言語と言語コード](#)。

トピック

- [入力コンテンツでサポートされるフォーマット](#)
- [翻訳のカスタマイズ](#)
- [言語の自動検出](#)
- [例外処理](#)

入力コンテンツでサポートされるフォーマット

Amazon Translate は、入力コンテンツの次の形式をサポートしています。

- リアルタイム翻訳の場合:
 - 入力テキスト — UTF-8 形式のプレーンテキスト。Amazon Translate は、出力コンテンツを UTF-8 テキストとして提供します。
 - 1 つの入力ファイル — プレーンテキスト (.txt)、HTML (.html)、または Word (.docx) のコンテンツを含むファイル。Amazon Translate は、出力コンテンツを入力ファイルと同じ形式のファイルとして提供します。
- バッチ翻訳ジョブの場合:
 - 入力ファイルのコレクション — Amazon S3 の場所にアップロードする 1 つ以上のファイル。サポートされているファイル形式には、プレーンテキスト (.txt)、HTML (.html)、ワード (.docx)、Excel (.xlsx)、(.pptx)、XLIFF 1.2 PowerPoint (.xlf) があります。Amazon Translate は、出力コンテンツをファイルとして提供します。各出力ファイルのファイル形式は、入力ファイル形式と一致します。

翻訳のカスタマイズ

以下の機能を使用して、Amazon Translate で作成した翻訳をカスタマイズできます。

- Do-not-translate タグ — 開始タグと終了タグを使用して、(HTML コンテンツ内の) 翻訳したくないコンテンツを指定します。
- Custom terminology (カスタム用語) — Amazon Translate でブランド名などの特定の用語を翻訳する方法を定義します。
- 簡潔さ — ほとんどの翻訳において、翻訳出力の長さを (簡潔でない翻訳出力と比較して) 短くします。簡潔さはリアルタイムのテキスト翻訳でサポートされています。
- 冒涇 — 翻訳出力に含まれる不敬な単語やフレーズを隠します。
- 形式性 — 翻訳出力における言語の形式性のレベルを設定します。
- パラレルデータ — 提供する翻訳サンプルに含まれるスタイル、トーン、単語の選択を反映するように翻訳出力を調整します。

詳しくは、「[翻訳のカスタマイズ](#)」を参照してください。

言語の自動検出

Amazon Translate ではソーステキストで使用されている言語を自動で検出できます。言語の自動検出を使用するには、`auto` をソース言語として指定します。Amazon Translate がユーザーに代わって Amazon Comprehend を呼び出し、ソーステキストで使用されている言語を判断します。言語の自動検出を選択することで、お客様は Amazon Comprehend のサービス条件および契約に同意するものとみなされます。Amazon Comprehend の料金の詳細については「[Amazon Comprehend の料金](#)」を参照してください。

例外処理

サポートされていないソース言語またはターゲット言語を指定した場合、Amazon Translate から次の例外が返されます。

- `UnsupportedLanguagePairException`— Amazon Translate は、サポートされているすべての言語間の翻訳をサポートしています。この例外は、ソース言語またはターゲット言語のいずれかがサポートされていない場合に返されます。さらなる詳細については、[Supported languages](#) を参照してください。
- `DetectedLanguageLowConfidenceException`— 自動言語検出を使用していて、Amazon Translate が正しいソース言語を検出したという確信が低い場合、この例外が返されます。信頼性のレベルが低い場合でも、それが許容される範囲である場合は例外で返されたソース言語を使用できます。

セットアップ

Amazon Translate を初めて使用する場合は、事前に以下のタスクをすべて実行してください。

タスクのセットアップ

- [にサインアップする AWS アカウント](#)
- [管理アクセスを持つユーザーを作成する](#)
- [のインストールと設定 AWS Command Line Interface \(AWS CLI\)](#)
- [プログラマチックアクセス権を付与する](#)
- [でのこのサービスの使用 AWS SDK](#)

にサインアップする AWS アカウント

をお持ちでない場合 AWS アカウントで、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. <https://portal.aws.amazon.com/billing/サインアップ> を開きます。
2. オンラインの手順に従います。

サインアップ手順の一環として、通話呼び出しを受け取り、電話キーパッドで検証コードを入力するように求められます。

にサインアップするとき AWS アカウント、AWS アカウントのルートユーザー が作成されます。ルートユーザーはすべてののにアクセスできます AWS のサービス アカウントの および リソース。セキュリティのベストプラクティスとして、ユーザーに管理アクセスを割り当て、ルートユーザーのみを使用して[ルートユーザーアクセスが必要なタスク](#)を実行してください。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。<https://aws.amazon.com/> に移動し、マイアカウント を選択すると、いつでも現在のアカウントアクティビティを表示し、アカウントを管理できます。

管理アクセスを持つユーザーを作成する

にサインアップした後 AWS アカウント、を保護する AWS アカウントのルートユーザー、有効化 AWS IAM Identity Center、および 管理ユーザーを作成して、日常的なタスクにルートユーザーを使用しないようにします。

のセキュリティ保護 AWS アカウントのルートユーザー

1. [にサインインします。AWS Management Console](#) ルートユーザーを選択し、AWS アカウント E メールアドレス。次のページでパスワードを入力します。

ルートユーザーを使用してサインインする方法については、[「」の「ルートユーザーとしてサインインする」](#)を参照してください。AWS サインイン ユーザーガイド。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

手順については、[「の仮想MFAデバイスの有効化」](#)を参照してください。[AWS アカウントIAM ユーザーガイドのルートユーザー \(コンソール\)](#)。

管理アクセスを持つユーザーを作成する

1. IAM Identity Center を有効にします。

手順については、[「の有効化」](#)を参照してください。[AWS IAM Identity Center \(\)AWS IAM Identity Center ユーザーガイド](#)。

2. IAM Identity Center で、ユーザーに管理アクセス権を付与します。

の使用に関するチュートリアル IAM アイデンティティセンターディレクトリ ID ソースとして、[「デフォルトを使用してユーザーアクセスを設定する」](#)を参照してください。[IAM アイデンティティセンターディレクトリ \(\)AWS IAM Identity Center ユーザーガイド](#)。

管理アクセス権を持つユーザーとしてサインインする

- IAM Identity Center ユーザーでサインインするには、IAM Identity Center ユーザーの作成時に E メールアドレスに URL 送信されたサインインを使用します。

IAM Identity Center ユーザーを使用してサインインする方法については、[「へのサインイン」](#)を参照してください。[AWS の アクセスポータル](#) AWS サインイン ユーザーガイド。

追加のユーザーにアクセス権を割り当てる

1. IAM Identity Center で、最小特権のアクセス許可を適用するベストプラクティスに従うアクセス許可セットを作成します。

手順については、「」の「[アクセス許可セットの作成](#)」を参照してください。AWS IAM Identity Center ユーザーガイド。

2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

手順については、「」の「[グループの追加](#)」を参照してください。AWS IAM Identity Center ユーザーガイド。

のインストールと設定 AWS Command Line Interface (AWS CLI)

を使用する AWS CLI Amazon Translate をインタラクティブに呼び出すには、を使用します。

をインストールして設定するには AWS CLI

1. のインストール AWS CLI。手順については、「」の次のトピックを参照してください。AWS Command Line Interface ユーザーガイド：

[の最新バージョンのインストールまたは更新 AWS Command Line Interface](#)

2. の設定 AWS CLI。手順については、「」の次のトピックを参照してください。AWS Command Line Interface ユーザーガイド：

[の設定 AWS Command Line Interface](#)

プログラマチックアクセス権を付与する

ユーザーが を操作する場合は、プログラムによるアクセスが必要です AWS の外部 AWS Management Console。プログラムによるアクセスを許可する方法は、アクセスするユーザーのタイプによって異なります。AWS.

ユーザーにプログラマチックアクセス権を付与するには、以下のいずれかのオプションを選択します。

プログラマチックアクセス権を必要とするユーザー	目的	方法
<p>ワークフォースアイデンティティ</p> <p>(IAMIdentity Center で管理されるユーザー)</p>	<p>へのプログラムによるリクエストに署名するために一時的な認証情報を使用する AWS CLI, AWS SDKs、または AWS APIs.</p>	<p>使用するインターフェイス用の手引きに従ってください。</p> <ul style="list-style-type: none"> • 向けの AWS CLI、「の設定」を参照してください。AWS CLI を使用する AWS IAM Identity Center ()AWS Command Line Interface ユーザーガイド。 • [AWS SDKs、ツール、および AWS APIs、「のIAM Identity Center 認証」を参照してください。AWS SDKs および ツールリファレンスガイド。
IAM	<p>へのプログラムによるリクエストに署名するために一時的な認証情報を使用する AWS CLI, AWS SDKs、または AWS APIs.</p>	<p>「での一時的な認証情報の使用」の手順に従います。AWSIAM ユーザーガイドのリソース。</p>
IAM	<p>(非推奨)</p> <p>長期認証情報を使用してへのプログラムによるリクエストに署名する AWS CLI, AWS SDKs、または AWS APIs.</p>	<p>使用するインターフェイス用の手引きに従ってください。</p> <ul style="list-style-type: none"> • 向けの AWS CLI、「のIAM ユーザー認証情報を使用した認証」を参照してください。AWS Command Line Interface ユーザーガイド。 • [AWS SDKs および ツール、「の長期認証情報を使用した認証」を参照し

プログラマチックアクセス権を必要とするユーザー	目的	方法
		<p>てください。AWS SDKs および ツールリファレンスガイド。</p> <ul style="list-style-type: none"> • [AWS APIs、「IAMユーザーガイド」のIAM「ユーザーのアクセスキーの管理」を参照してください。

でのこのサービスの使用 AWS SDK

AWS Software Development Kit (SDKs) は、多くの一般的なプログラミング言語で使用できます。各 SDKにはAPI、開発者が好みの言語でアプリケーションを簡単に構築できるようにする、コード例、およびドキュメントが用意されています。

SDK ドキュメント	コードの例
AWS SDK for C++	AWS SDK for C++ コード例
AWS CLI	AWS CLI コード例
AWS SDK for Go	AWS SDK for Go コード例
AWS SDK for Java	AWS SDK for Java コード例
AWS SDK for JavaScript	AWS SDK for JavaScript コード例
AWS SDK for Kotlin	AWS SDK for Kotlin コード例
AWS SDK for .NET	AWS SDK for .NET コード例
AWS SDK for PHP	AWS SDK for PHP コード例
AWS Tools for PowerShell	PowerShell コード例のツール
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) コード例

SDK ドキュメント	コードの例
AWS SDK for Ruby	AWS SDK for Ruby コード例
AWS SDK for Rust	AWS SDK for Rust コード例
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP コード例
AWS SDK for Swift	AWS SDK for Swift コード例

可用性の例

必要なものが見つからなかった場合。このページの下側にある [Provide feedback (フィードバックを送信)] リンクから、コードの例をリクエストしてください。

Amazon Translate の開始方法

Amazon Translate の使用を開始する最も簡単な方法は、コンソールを使って何らかのテキストを翻訳することです。コマンドラインから API オペレーションを試すこともできます。また、いずれかの AWS SDK をインストールして Amazon Translate API オペレーションを使用できます。

トピック

- [開始方法 \(コンソール\)](#)
- [開始方法 \(AWS CLI\)](#)
- [開始方法 \(SDK\)](#)

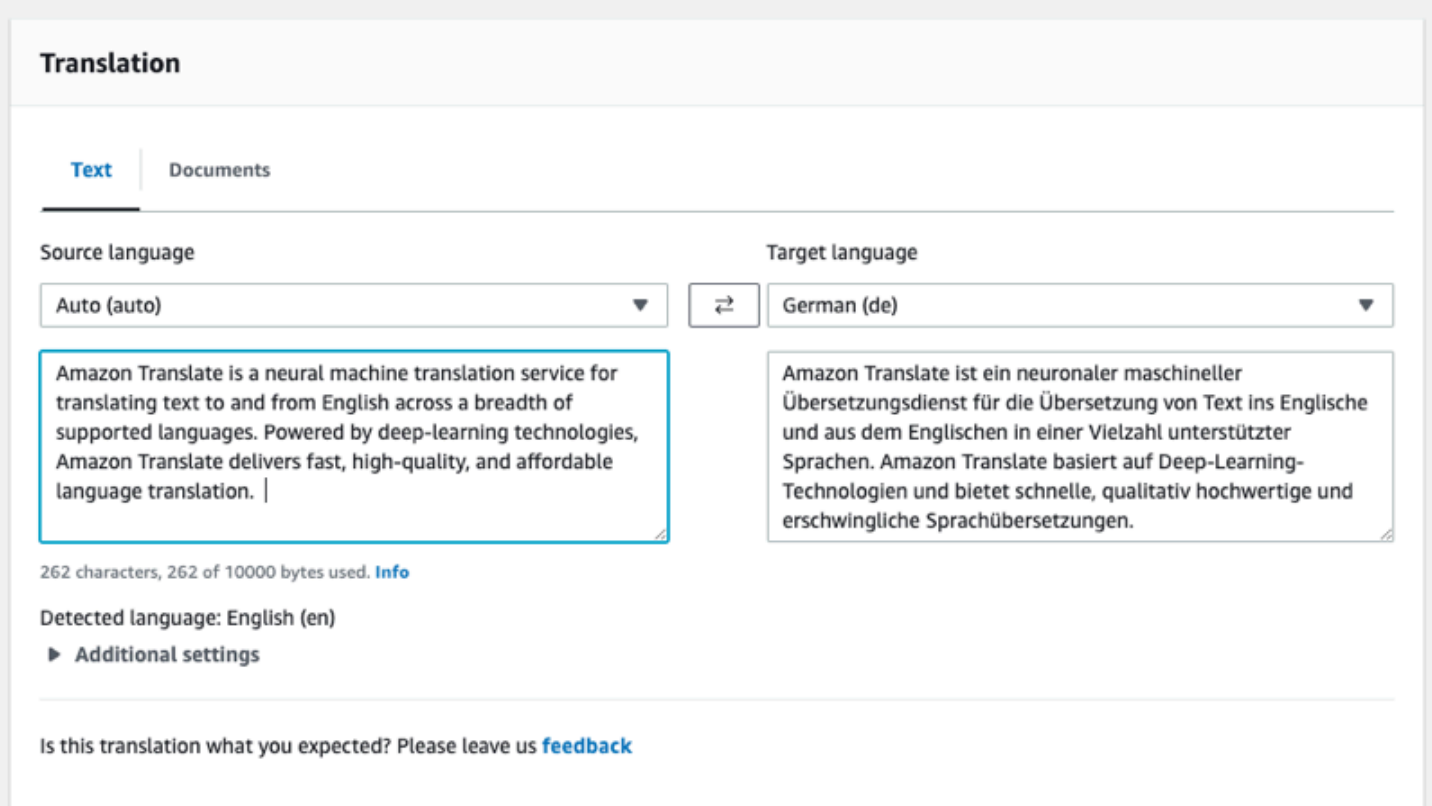
開始方法 (コンソール)

Amazon Translate の使用を開始する最も簡単な方法は、コンソールを使って何らかのテキストを翻訳することです。コンソールを使用して、最大 10,000 バイトのテキストを翻訳できます。「[Amazon Translate の仕組み](#)」の概念と用語を確認してから、先に進むことをお勧めします。

[Amazon Translate コンソール](#) を開きます。

Amazon Translate を初めて使用する場合は、[Launch real-time translation (リアルタイム翻訳を起動)] を選択します。

[リアルタイムでの翻訳] で、ターゲット言語を選択します。ソース言語は Amazon Translate によって自動検出されますが、ユーザーが選択することもできます。左側のテキストボックスに翻訳するテキストを入力します。翻訳されたテキストが右側のテキストボックスに表示されます。



The screenshot shows the Amazon Translate web interface. At the top, there is a "Translation" header. Below it, there are two tabs: "Text" (selected) and "Documents". The interface is divided into two main sections: "Source language" and "Target language".

Source language: A dropdown menu is set to "Auto (auto)". Below it, a text box contains the English text: "Amazon Translate is a neural machine translation service for translating text to and from English across a breadth of supported languages. Powered by deep-learning technologies, Amazon Translate delivers fast, high-quality, and affordable language translation." Below the text box, it says "262 characters, 262 of 10000 bytes used. [Info](#)".

Target language: A dropdown menu is set to "German (de)". Below it, a text box contains the German translation: "Amazon Translate ist ein neuronaler maschineller Übersetzungsdienst für die Übersetzung von Text ins Englische und aus dem Englischen in einer Vielzahl unterstützter Sprachen. Amazon Translate basiert auf Deep-Learning-Technologien und bietet schnelle, qualitativ hochwertige und erschwingliche Sprachübersetzungen."

Below the target language section, it says "Detected language: English (en)" and "► Additional settings". At the bottom, there is a feedback prompt: "Is this translation what you expected? Please leave us [feedback](#)".

[アプリケーション統合] セクションでは、[TranslateText](#) オペレーションの JSON 入力と出力を確認できます。

▼ Application integration

Learn more about working with the Translate service using APIs for automation and larger volumes of text. [Info](#)

JSON request

```
1 {
2   "Text": "Amazon Translate is a neural machine
3     translation service for translating text to
4     and from English across a breadth of supported
5     languages. Powered by deep-learning
6     technologies, Amazon Translate delivers fast,
7     high-quality, and affordable language
8     translation. ",
9   "SourceLanguageCode": "auto",
10  "TargetLanguageCode": "de"
11 }
```

Copy

JSON response

```
1 {
2   "TranslatedText": "Amazon Translate ist ein
3     neuronaler maschineller Übersetzungsdienst für
4     die Übersetzung von Text ins Englische und aus
5     dem Englischen in einer Vielzahl unterstützter
6     Sprachen. Amazon Translate basiert auf Deep
7     -Learning-Technologien und bietet schnelle,
8     qualitativ hochwertige und erschwingliche
9     Sprachübersetzungen. ",
10  "SourceLanguageCode": "en",
11  "TargetLanguageCode": "de"
12 }
```

Copy

開始方法 (AWS CLI)

以下の演習では、AWS コマンドラインインターフェイス (AWS CLI) を使用してテキストを翻訳します。以下の演習を行うには、CLI に精通している必要があります。また、テキストエディタが必要です。詳細については、「[のインストールと設定 AWS Command Line Interface \(AWS CLI\)](#)」を参照してください。

コマンドラインから Amazon Translate を使用するには、Amazon Translate サービスをサポートするリージョンからコマンドを実行する必要があります。使用可能なエンドポイントとリージョンのリストについては、[AWS 全般リファレンス](#)の「Amazon Translate のリージョンとエンドポイント」を参照してください。

コマンドラインを使用したテキストの翻訳

次の例は、コマンドラインから [TranslateText](#) オペレーションを使用してテキストを翻訳する方法を示しています。例は、Unix、Linux、および macOS 用にフォーマットされています。Windows の場合は、各行末のバックスラッシュ (\) Unix 連結文字をキャレット (^) に置き換えてください。コマンドラインで、以下のように入力します。

```
aws translate translate-text \
```



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

レスポンスは以下の JSON です。

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

次のステップ

Amazon Translate のその他の使用方法については、「[を使用した Amazon Translate のコード例 AWS SDKs](#)」を参照してください。

開始方法 (SDK)

AWS は、さまざまなコンピュータ言語用の SDK を提供しています。SDK は、署名の計算、リクエストの再試行処理、エラー処理など、クライアントの API 接続の詳細の多くを管理します。詳細については、[AWSSDK](#) をご参照ください。

次の例では、Java および Python で Amazon Translate の [TranslateText](#) オペレーションを使用する方法を示します。SDK を使用して、Amazon Translate API オペレーションについて学び、独自のアプリケーション構成要素として使用します。

トピック

- [AWS SDK for Java を使用したテキストの翻訳](#)
- [AWS SDK for Python \(Boto\) を使用したテキストの翻訳](#)
- [その他の SDK サンプル](#)

AWS SDK for Java を使用したテキストの翻訳

AWS は、Java [GitHub TranslateText](#)での操作の使用法の例を示しています。この例を実行するには、AWS SDK for Java が必要です。SDK for Java をインストールする手順については、「[Java 2.x 用 AWS SDK のセットアップ](#)」を参照してください。

AWS SDK for Python (Boto) を使用したテキストの翻訳

次の例は、Python [TranslateText](#)でオペレーションを使用する方法を示しています。この例を実行するには、を使用して Python SDK をインストールしますAWS CLI。手順については、「[the section called “のセットアップ AWS CLI”](#)」を参照してください。

```
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'))
```

サポートされている言語コードのリストについては、「[サポートされている言語と言語コード](#)」を参照してください

その他の SDK サンプル

.NET と SAP ABAP [を使用した Amazon Translate のコード例 AWS SDKs](#) を使用する例については、を参照してください。

翻訳の処理モード

ドキュメントを翻訳するときは、2つの異なる翻訳処理モードを使用できます。リアルタイム翻訳または非同期バッチ処理です。使用するモードは、ターゲットドキュメントのサイズと種類に基づいており、翻訳ジョブの送信方法と結果の表示方法に影響します。

- [リアルタイム翻訳](#) — 少量のテキスト (またはテキストファイル) を翻訳する同期リクエストを行うと、Amazon Translate は翻訳されたテキストを直ちに応答します。
- [非同期バッチ処理](#) — Amazon Simple Storage Service (Amazon S3) の場所にドキュメントのコレクションを置き、非同期処理ジョブを開始してそれらを翻訳します。Amazon Translate は、翻訳された出力ドキュメントを指定された Amazon S3 の場所に送信します。

リアルタイム翻訳

Amazon Translate では、ドキュメントとテキストの翻訳オペレーションがリアルタイムで行われ、翻訳がすぐに返されます。コンソールまたは API を使用してリアルタイムの翻訳を実行できます。

トピック

- [コンソールを使用したリアルタイム翻訳](#)
- [API を使用したリアルタイム翻訳](#)

コンソールを使用したリアルタイム翻訳

リアルタイム翻訳にコンソールを使用する場合、入力テキストを [ソース言語] テキストボックスに貼り付けるか、入力テキストをファイルとして入力します。オプションで、希望する形式レベル、冒濺的な表現のマスキング、簡潔さ、カスタム用語などの機能を設定できます。

リアルタイム翻訳では言語の自動検出を使用できますが、料金が発生する場合があります。詳細については、「[言語の自動検出](#)」を参照してください。

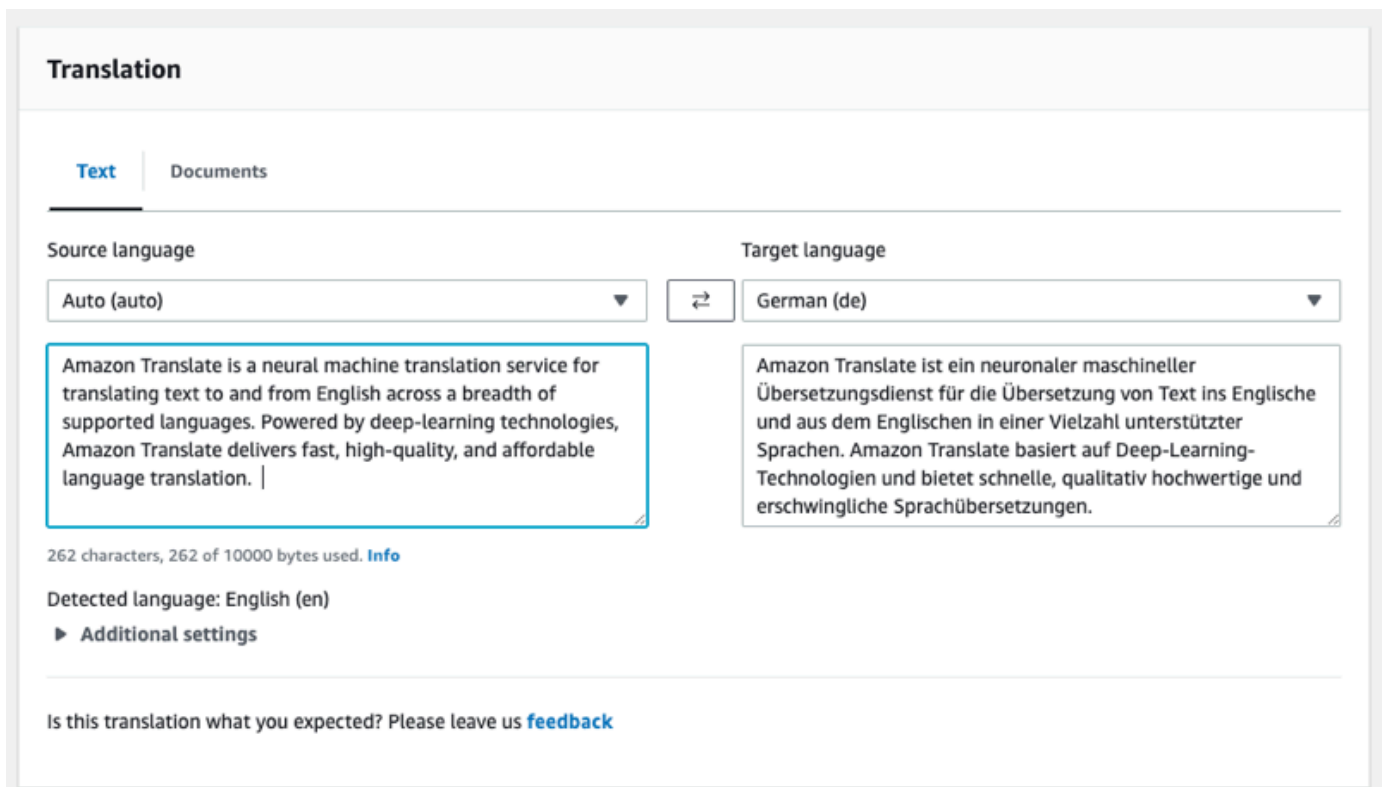
トピック

- [テキストの翻訳](#)
- [ドキュメントの翻訳](#)
- [同等の API リクエストとレスポンスデータを表示する](#)
- [翻訳機能を使用する](#)

テキストの翻訳

Amazon Translate コンソールを使用して、最大 10,000 バイトのテキストを翻訳できます。

1. [Amazon Translate コンソール](#) を開きます。
2. 左側のナビゲーションメニューで、[リアルタイムでの翻訳] を選択します。
3. [ソース言語] では、ソーステキストの言語を選択するか、自動検出の場合は値を [Auto (自動)] のままにします。
4. [ターゲット言語] で言語を選択します。
5. [ソース言語] テキストボックスにテキストを入力するか貼り付けます。コンソールの [ターゲット言語] テキストボックスに、翻訳されたテキストが表示されます。



The screenshot shows the Amazon Translate console interface. At the top, there is a 'Translation' header with two tabs: 'Text' (selected) and 'Documents'. Below the tabs, there are two dropdown menus for 'Source language' and 'Target language'. The source language is set to 'Auto (auto)' and the target language is set to 'German (de)'. A double-headed arrow icon is between the two dropdowns. Below the source language dropdown, there is a text box containing the English text: 'Amazon Translate is a neural machine translation service for translating text to and from English across a breadth of supported languages. Powered by deep-learning technologies, Amazon Translate delivers fast, high-quality, and affordable language translation.' Below this text box, it says '262 characters, 262 of 10000 bytes used. Info'. Below the target language dropdown, there is a text box containing the German translation: 'Amazon Translate ist ein neuronaler maschineller Übersetzungsdienst für die Übersetzung von Text ins Englische und aus dem Englischen in einer Vielzahl unterstützter Sprachen. Amazon Translate basiert auf Deep-Learning-Technologien und bietet schnelle, qualitativ hochwertige und erschwingliche Sprachübersetzungen.' Below this text box, it says 'Detected language: English (en)' and 'Additional settings'. At the bottom, there is a feedback prompt: 'Is this translation what you expected? Please leave us feedback'.

ドキュメントの翻訳

リアルタイムのドキュメント翻訳は、英語からサポートされている任意の言語への翻訳と、サポートされている任意の言語から英語への翻訳をサポートします。

Amazon Translate コンソールを使用してドキュメントを翻訳するには、以下の手順を実行します。

1. [Amazon Translate コンソール](#) を開きます。

2. 左側のナビゲーションメニューで、[リアルタイムでの翻訳] を選択します。
3. 翻訳パネルで、[ドキュメント] タブを選択します。

Translation

Text | **Document**

Source language Target language

Choose English for either the source language or the target language.

Upload file

Supported file extensions include .html and .txt. The maximum file size is 100 KB.

Document type

▶ **Additional settings**

Is this translation what you expected? Please leave us [feedback](#)

4. [ソース言語] では、ソーステキストの言語を選択するか、自動検出の場合は [Auto (自動)] を選択します。
5. [ターゲット言語] で言語を選択します。ソース言語が英語でない場合、ターゲット言語に英語を選択する必要があります。
6. [ファイルをアップロード] で [ファイルの選択] を選択し、ソースファイルへのパスを入力します。ファイルの最大サイズは 100 KB です。
7. [ドキュメントタイプ] では、翻訳ソースファイルの形式を選択します。ドキュメント翻訳では、プレーンテキスト、HTML、または Word (.docx) の入力ファイルがサポートされています。
8. [翻訳] を選択します。

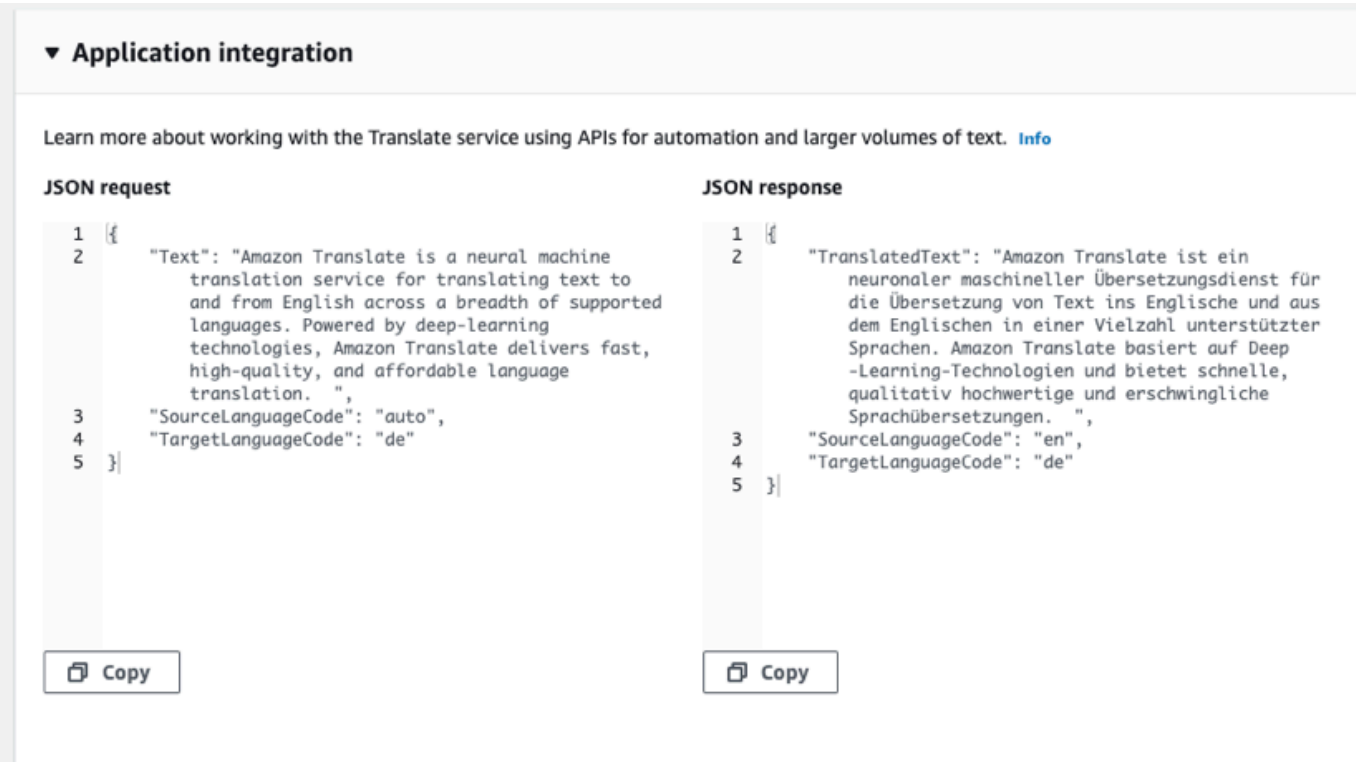
翻訳タスクが完了したら、[翻訳をダウンロード] を選択して、翻訳したドキュメントをローカルのハードドライブにダウンロードします。翻訳したドキュメントの形式 (テキスト、HTML、Word) は入力ドキュメントと一致します。

同等の API リクエストとレスポンスデータを表示する

コンソールを使用して入力テキストまたはドキュメントを翻訳すると、同等の API リクエストデータとレスポンスデータを JSON 形式で表示できます。

1. [翻訳] パネルの下にある [アプリケーション統合] パネルを展開します。

コンソールには同等の翻訳リクエストデータが JSON 形式で表示されます。



The screenshot shows the 'Application integration' section of the Amazon Translate console. It displays two JSON blocks: a request and a response. The request JSON is on the left, and the response JSON is on the right. Both have a 'Copy' button below them.

```
▼ Application integration

Learn more about working with the Translate service using APIs for automation and larger volumes of text. Info

JSON request
1 {
2   "Text": "Amazon Translate is a neural machine
3     translation service for translating text to
4     and from English across a breadth of supported
5     languages. Powered by deep-learning
6     technologies, Amazon Translate delivers fast,
7     high-quality, and affordable language
8     translation. ",
9   "SourceLanguageCode": "auto",
10  "TargetLanguageCode": "de"
11 }

JSON response
1 {
2   "TranslatedText": "Amazon Translate ist ein
3     neuronaler maschineller Übersetzungsdienst für
4     die Übersetzung von Text ins Englische und aus
5     dem Englischen in einer Vielzahl unterstützter
6     Sprachen. Amazon Translate basiert auf Deep
7     -Learning-Technologien und bietet schnelle,
8     qualitativ hochwertige und erschwingliche
9     Sprachübersetzungen. ",
10  "SourceLanguageCode": "en",
11  "TargetLanguageCode": "de"
12 }
```

2. JSON リクエストをコピーして、または API オペレーションで使用できます。[TranslateTextTranslateDocument](#)
3. [JSON レスポンスパネル] の JSON 出力は API が生成する出力と一致します。

翻訳機能を使用する

Amazon Translate コンソールで翻訳機能を使用するには、以下の手順を実行します。

1. [Amazon Translate コンソール](#) を開きます。
2. 左側のナビゲーションメニューで、[リアルタイムでの翻訳] を選択します。
3. 前の手順で説明したように、ソース言語、ターゲット言語、および入力データ (テキストまたはドキュメント) を指定します。
4. [追加設定] では、次の設定を使用して翻訳ジョブの出力をカスタマイズすることができます。

Custom terminology (カスタム用語)

カスタム用語ファイルを選択します。ファイルの入力テキストにソース用語のエントリがある場合、Amazon Translate は用語ファイルの翻訳を使用します。

詳細については、「[カスタム用語による翻訳のカスタマイズ](#)」を参照してください。

簡潔さ

(簡潔でない翻訳出力に比べて) ほとんどの翻訳の翻訳出力の長さが短くなります。Amazon Translate はテキストを簡潔に翻訳できますが、ドキュメントの翻訳はサポートしていません。Amazon Translate は、簡潔にするためにソース言語とターゲット言語がサポートされていない言語ペアを形成している場合、簡潔さの設定を無視します。

サポートされている言語については、[を参照してください](#)。 [Amazon Translate での簡潔さの使用](#)

Profanity (不敬)

翻訳出力内の不適切な単語やフレーズをマスクします。Amazon Translate は、サポートされているすべての言語で不敬表現のマスクングをサポートしているわけではありません。

詳細については、「[Amazon Translate における不適切な言葉やフレーズのマスクング](#)」を参照してください。

形式

一部のターゲット言語では、[形式] をフォーマルまたはインフォーマルに設定できます。形式がターゲット言語をサポートしていない場合、Amazon Translate は形式設定を無視します。

詳細については、「[Amazon Translate の形式設定](#)」を参照してください。

5. ドキュメント翻訳の場合、[翻訳] をクリックすると、選択した翻訳機能を使用してドキュメントが翻訳されます。

テキスト翻訳の場合、各翻訳機能を選択すると、コンソールは翻訳されたテキストにその機能を適用します。

API を使用したリアルタイム翻訳

Amazon Translate では、インタラクティブなアプリケーションをサポートするために、以下のリアルタイム翻訳オペレーションを提供しています。

- [TranslateText](#)— テキストブロックを翻訳します。
- [TranslateDocument](#)— ファイルの内容 (プレーンテキスト、HTML、または.docx) を翻訳します。

これらの同期オペレーションは、翻訳結果をアプリケーションに直接返します。これらのオペレーションで言語の自動検出を使用すると、料金が発生する場合があります。詳細については、「[言語の自動検出](#)」を参照してください。

テキストの翻訳

[TranslateText](#) オペレーションを使用して 1 つのテキストブロックを翻訳します。

コマンドラインを使用したテキストの翻訳

次の例は、[TranslateText](#) コマンドラインから操作を使用する方法を示しています。例は、Unix、Linux、および macOS 用にフォーマットされています。Windows の場合は、各行末のバックスラッシュ (\) Unix 連結文字をキャレット (^) に置き換えてください。

コマンドラインで、以下のコマンドを入力します。

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

コマンドは次のJSONで応答します。

```
{  
    "TargetLanguageCode": "es",  
    "TranslatedText": "Hola, mundo",
```



```
"SourceLanguageCode": "en"
}
```

JSON ファイルを使用したテキストの翻訳

この例では、JSON ファイルを使用して長いテキストブロックを翻訳する方法を示します。ソース言語とターゲットの言語をコマンドラインで指定するか、JSON ファイルで指定します。

Note

JSON ファイルの例は、読みやすいようにフォーマットされています。"Text" フィールドの形式を変更して改行を削除します。

例は、Unix、Linux、および macOS 用にフォーマットされています。Windows の場合は、各行末のバックスラッシュ (\) Unix 連結文字をキャレット (^) に置き換えてください。

JSON ファイルを使用してテキストを翻訳するには

1. 次のテキストを、translate.json という JSON ファイルにコピーします。

```
{
  "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. AWS CLI で、次のコマンドを実行します。

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

このコマンドは、次の JSON テキストを含む JSON ファイルを出力します。

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

ドキュメントの翻訳

[TranslateDocument](#) オペレーションを使用してテキスト、HTML、または Word (.docx) ドキュメントを翻訳し、その翻訳結果をアプリケーションに直接返します。

リアルタイムのドキュメント翻訳は、英語からサポートされている任意の言語への翻訳と、サポートされている任意の言語から英語への翻訳をサポートします。ソース言語コードを指定するか、自動検出を使用できます。

コマンドラインを使用したドキュメントの翻訳

次の例は、[TranslateDocument](#) コマンドラインから操作を使用する方法を示しています。例は、Unix、Linux、および macOS 用にフォーマットされています。Windows の場合は、各行末のバックスラッシュ (\) Unix 連結文字をキャレット (^) に置き換えてください。

コマンドラインで、以下のコマンドを入力します。

```
aws translate translate-document \
    --region region \
    --source-language-code "en" \
    --target-language-code "es" \
    --document-content fileb://source-lang.txt
    --document ContentType=text/plain
    --query "TranslatedDocument.Content"
    --output text | base64
    --decode > target-lang.txt
```

コマンドは次のJSONで応答します。

```
{
  "SourceLanguageCode": "en",
  "TargetLanguageCode": "es",
  "TranslatedDocument": {
```

```
    "Content": blob
  }
}
```

Amazon Translate による非同期バッチ処理

大きいドキュメントの集合 (サイズは 5 GB 以下) を翻訳するには、Amazon Translate 非同期バッチ処理オペレーション [StartTextTranslationJob](#) を使用します。これは、ソーシャルメディアへの投稿やユーザーレビューなどの短いドキュメントの集合に最適です。また、瞬時の翻訳が不要な状況にも最適です。

非同期バッチ翻訳を実行するには通常、以下の手順を実行します。

1. Amazon S3 バケット内の入力フォルダに一連のドキュメントを保存します。
2. バッチ翻訳ジョブを開始します。
3. リクエストの一環として、Amazon S3 の入力フォルダとそのすべてのサブフォルダに対する読み取りアクセス権がある IAM ロールを Amazon Translate に付与します。このロールには、出力 Amazon S3 バケットに対する読み取り/書き込みアクセス権も必要です。
4. バッチ翻訳ジョブの進行状況をモニタリングします。
5. 指定した出力バケットからバッチ翻訳ジョブの結果を取得します。

利用可能なリージョン

バッチ翻訳は以下の AWS リージョンでサポートされています。

- 米国東部 (オハイオ)
- 米国東部 (バージニア北部)
- 米国西部 (北カリフォルニア)
- 米国西部 (オレゴン)
- アジアパシフィック (ムンバイ)
- アジアパシフィック (ソウル)
- アジアパシフィック (シンガポール)
- アジアパシフィック (シドニー)
- アジアパシフィック (東京)
- カナダ (中部)

- 欧州 (フランクフルト)
- 欧州 (アイルランド)
- 欧州 (ロンドン)
- 欧州 (パリ)
- 欧州 (ストックホルム)

トピック

- [バッチ翻訳ジョブの前提条件](#)
- [バッチ翻訳ジョブの実行](#)
- [バッチ翻訳ジョブのモニタリングと分析](#)
- [バッチ翻訳結果の取得](#)

バッチ翻訳ジョブの前提条件

Amazon Translate がバッチ翻訳ジョブを正常に実行するには、以下の前提条件を満たす必要があります。

- 入力および出力ドキュメントを含む Amazon S3 バケツは、呼び出す API エンドポイントと同じ AWS リージョン内に存在している必要があります。
- バッチ入力ドキュメントの集合は、サイズが 5 GB 以下とします。
- バッチ翻訳ジョブでは、最大 100 万個のドキュメントを送信できます。
- 各入力ドキュメントは 20 MB 以下で、100 万字未満とします。
- 入力ファイルは、Amazon S3 バケツのフォルダ内に存在している必要があります。バケツの最上位レベルに入力ファイルを追加すると、バッチ翻訳ジョブを実行するときに Amazon Translate によってエラーがスローされます。この要件は入力ファイルに適用されます。出力ファイルにフォルダは不要で、Amazon Translate を Amazon S3 バケツの最上位レベルに配置できます。
- 入力ファイルフォルダには、ネストされたフォルダを含めることができます。ネストされたどのフォルダにも details という名前が付けられていないことを確認してください。付いていると、バッチ翻訳ジョブを実行するときに Amazon Translate によってエラーがスローされます。

サポートされるファイル形式

Amazon Translate では、バッチ翻訳ジョブで次の種類のファイルがサポートされています。

- プレーンテキスト
- HTML
- Word ドキュメント (.docx)
- PowerPoint プレゼンテーションファイル (.pptx)
- Excel ワークブックファイル (.xlsx)
- XML Localization Interchange File Format (XLIFF) ファイル (.xlf) Amazon Translate では XLIFF バージョン 1.2 のみがサポートされています。

Amazon Translate ではファイルを UTF-8 エンコーディングする必要があります。

前提となるアクセス許可

バッチ翻訳ジョブを実行する前に、AWS アカウントに IAM のサービスロールが必要です。このロールには、Amazon Translate に以下を付与するアクセス許可ポリシーがある必要があります。

- Amazon S3 の入力フォルダとそのすべてのサブフォルダへの読み取りアクセス権
- 出力バケットへの読み取り/書き込みアクセス権

また、ロールを引き受け、アクセス許可を取得することを Amazon Translate に許可する信頼ポリシーが含まれている必要があります。この信頼ポリシーは、translate.amazonaws.com サービスプリンシパルによる sts:AssumeRole アクションの実行を許可する必要があります。

Amazon Translate コンソールを使用してバッチ翻訳ジョブを作成する場合、Amazon Translate でこのロールを自動的に作成できるようにするオプションがあります。AWS CLI または Amazon Translate API を使用してバッチ翻訳ジョブを実行する場合、リクエストでロールの Amazon リソースネーム (ARN) を提供します。

詳細については、「IAM ユーザーガイド」の「[AWS のサービスにアクセス許可を委任するロールの作成](#)」を参照してください。

Example 許可ポリシー

次のアクセス許可ポリシーの例では、Amazon S3 のサンプル入力フォルダへの読み取りアクセス権が付与されます。サンプル出力バケットへの読み取り/書き込みアクセス権が付与されます。

```
{  
  "Version": "2012-10-17",
```

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": "s3:GetObject",  
    "Resource": [  
      "arn:aws:s3:::input-bucket-name/*",  
      "arn:aws:s3:::output-bucket-name/*"  
    ]  
  },  
  {  
    "Effect": "Allow",  
    "Action": "s3:ListBucket",  
    "Resource": [  
      "arn:aws:s3:::input-bucket-name",  
      "arn:aws:s3:::output-bucket-name"  
    ]  
  },  
  {  
    "Effect": "Allow",  
    "Action": [  
      "s3:PutObject"  
    ],  
    "Resource": "arn:aws:s3:::output-bucket-name/*"  
  }  
]
```

Example 信頼ポリシー

次の信頼ポリシーでは、ポリシーが属する IAM ロールが Amazon Translate に引き受けられるようにしています。

[混乱した代理](#)問題を軽減するために、信頼ポリシーを使用している AWS アカウントを確認することをお勧めします。次の例では、aws:SourceArn および aws:SourceAccount 条件キーを使用してソースアカウントを検証しています。バッチ翻訳ジョブを送信する AWS アカウントを入力します。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "Service": "translate.amazonaws.com"  
      }  
    }  
  ]  
}
```

```
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:translate:*:111122223333:"
      },
      "StringEquals": {
        "aws:SourceAccount": "111122223333"
      }
    }
  }
]
```

暗号化をカスタマイズするための前提となるアクセス許可

Amazon Translate で暗号化設定をカスタマイズできますが、最初に IAM のサービスロールにアクセス許可を追加する必要があります。

Amazon Translate では、バッチ翻訳ジョブを実行したときに生成される翻訳出力が暗号化されます。デフォルトでは、この暗号化は AWS マネージドキー を使用して実行されます。このタイプのキーは、AWS により作成されてアカウントの AWS Key Management Service (AWS KMS) に保存されます。ただし、この KMS キーをユーザー自身が管理することはできません。これは、ユーザーに代わって AWS によってのみ管理と使用が行われます。

オプションで、カスタマーマネージドキーによる出力の暗号化を選択することもできます。これは KMS キーで、AWS アカウントにおいてユーザーが作成して所有し、管理します。

キーには、Amazon Translate によるキーの使用を可能にするキーポリシーが必要です。キーポリシーではこれを、Amazon Translate による Amazon S3 バケットへのアクセスを許可するサービスロールにアクセス許可を付与することで実行します。

次のポリシーステートメント例に示されているように、キーポリシーにより、出力の暗号化に必要な AWS KMS オペレーションをサービスロールで実行できるようになります。

Example KMS キーポリシーステートメント

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS":
```

```
[
  "arn:aws:iam::111122223333:role/AmazonTranslateServiceRoleS3FullAccess"
],
"Action":
[
  "kms:Decrypt",
  "kms:GenerateDataKey",
  "kms:CreateGrant",
  "kms:RetireGrant",
  "kms:DescribeKey"
],
"Resource": "*"
}
```

詳細については、AWS Key Management Service デベロッパーガイドの「[AWS KMS でのキーポリシー](#)」を参照してください。

別の AWS アカウントの AWS KMS キーを使用するためのアクセス許可

別の AWS アカウントにある KMS キーを Amazon Translate を使用しているアカウントから使用する場合、以下を行う必要があります。

1. IAM で Amazon Translate のサービスロールを更新する。
2. AWS KMS でキーポリシーを更新する。

サービスロールを更新するには、以下の例に示されているように、別の AWS アカウントにある KMS キーを使用して必要な AWS KMS オペレーションを実行できるようにするためのポリシーをアタッチします。

Example 別のアカウントの KMS キーへのアクセス権を付与する IAM ポリシー

```
{
  "Effect": "Allow",
  "Action":
  [
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:CreateGrant",
    "kms:RetireGrant",
    "kms:DescribeKey"
  ],
```



```
"Resource": "arn:aws:kms:us-west-2:111122223333:key/key-id"
}
```

KMS キーポリシーを更新するには、次のポリシーステートメントの例に示されているように、キーの使用が許可されているプリンシパルとしてサービスロールと管理者ユーザーを追加します。

Example IAM ロールによるキーの使用を許可するための KMS キーポリシーステートメント

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::444455556666:role/AmazonTranslateServiceRoleS3FullAccess",
      "arn:aws:iam::444455556666:admin"
    ]
  },
  "Action": [
    "kms:Decrypt",
    "kms:CreateGrant",
    "kms:GenerateDataKey",
    "kms:RetireGrant",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

詳細については、「AWS Key Management Service デベロッパーガイド」の「[他のアカウントのユーザーに AWS KMS キーの使用を許可する](#)」をご参照ください。

バッチ翻訳ジョブの実行

バッチ翻訳ジョブは、Amazon Translate コンソール、AWS CLI、または Amazon Translate を使用して実行できますAPI。

Note

バッチ翻訳ジョブは長時間実行されるオペレーションであり、完了するまでにかなりの時間がかかる場合があります。たとえば、小さいデータセットのバッチ翻訳に数分かかり、巨大

なデータセットのバッチ翻訳には最大 2 日かかることがあります。完了時間はリソースの可用性によっても異なります。

Amazon Translate コンソール

Amazon Translate コンソールを使用して翻訳ジョブを実行するには、バッチ翻訳ページでジョブを作成します。

1. [Amazon Translate コンソール](#)を開きます。
2. 左側のナビゲーションメニューで、[Batch translation (バッチ翻訳)] を選択します。
3. [Translation jobs (翻訳ジョブ)] ページで、[Create job (ジョブの作成)] を選択します。コンソールに [Create translation job (翻訳ジョブの作成)] ページが表示されます。
4. [Job settings (ジョブ設定)] で、次の作業を実行します。
 - a. [Name (名前)] で、バッチ翻訳ジョブのカスタム名を入力します。
 - b. [ソース言語] で、ソースファイルの言語を選択します。ソースファイルの言語がわからない場合や、入カドキュメントに異なるソース言語が含まれている場合は、auto を選択します。Amazon Translate が各ファイルのソース言語を自動検出します。
 - c. [ターゲット言語] では、最大 10 個の言語を選択できます。Amazon Translate が各ソースファイルを各ターゲット言語に翻訳します。
5. [Input data (入力データ)] で、次の作業を行います。
 - a. [Input S3 location (S3 での入力先)] で、Amazon S3 内にあって、翻訳ソースファイルが含まれている入力フォルダを指定します。Amazon S3 でフォルダに移動することでフォルダを提供するには、[Select folder (フォルダを選択)] を選択します。
 - b. [File format (ファイル形式)] で、翻訳ソースファイルの形式を選択します。
6. [Output data (出力データ)] で、次の作業を行います。
 - a. [Output S3 location (S3 での出力先)] で、Amazon S3 内にあって、Amazon Translate からの翻訳の出力先となる出力フォルダを指定します。Amazon S3 でフォルダに移動することでフォルダを提供するには、[Select folder (フォルダを選択)] を選択します。
 - b. オプションで、() で管理するカスタマーマネージドキーを使用して出力を暗号化する場合は、暗号化設定のカスタマイズ (アドバンスト) を選択しますAWS KMS。AWS Key Management Service

デフォルトでは、Amazon Translate は、 がユーザーに代わって作成、管理、使用するKMS キーを使用して翻訳出力を暗号化します AWS。代わりに独自のKMSキーを使用して出力を暗号化する場合は、このオプションを選択します。

現在の AWS アカウントのKMSキーを使用する場合は、キーの選択 で AWS Key Management Service キーを選択します。または、別のAWSアカウントのKMSキーを使用する場合は、そのキーの Amazon リソースネーム (ARN) を入力します。

Note

独自のKMSキーを使用する前に、 で Amazon Translate のサービスロールにアクセス許可を追加する必要がありますIAM。別のアカウントのKMSキーを使用する場合は、 のキーポリシーも更新する必要があります AWS KMS。詳細については、「[暗号化をカスタマイズするための前提となるアクセス許可](#)」を参照してください。

7. [Customizations - optional (カスタマイズ - オプション)]では、次の設定を使用して翻訳ジョブの出力をカスタマイズすることができます。

Profanity (不敬)

翻訳出力内の不適切な単語やフレーズをマスクします。ジョブに複数のターゲット言語を指定する場合、すべてのターゲット言語が不敬表現のマスクキングをサポートしている必要があります。いずれかのターゲット言語が不敬表現のマスクキングをサポートしていない場合、翻訳ジョブはどのターゲット言語でも不敬表現をマスクしません。

詳細については、「[Amazon Translate における不適切な言葉やフレーズのマスクキング](#)」を参照してください。

明るさ

Amazon Translate は、バッチ翻訳ジョブの簡潔さをサポートしていません。

詳細については、「[Amazon Translate での簡潔さの使用](#)」を参照してください。

形式

一部のターゲット言語では、[形式] をフォーマルまたはインフォーマルに設定できます。ジョブに複数のターゲット言語を指定した場合、Translate はサポートされていないターゲット言語の形式設定を無視します。

詳細については、「[Amazon Translate の形式設定](#)」を参照してください。

Custom terminology (カスタム用語)

各用語のソース用語例と翻訳例で構成されています。ジョブに複数のターゲット言語を指定した場合、Translate は、用語ファイルにソース用語のエントリがある要求されたターゲット言語ごとに、指定された用語を使用します。

詳細については、「[カスタム用語による翻訳のカスタマイズ](#)」を参照してください。

Parallel data (並列データ)

テキストのセグメントの翻訳方法を示した例で構成されています。ジョブに複数のターゲット言語を指定する場合、並列データファイルにはすべてのターゲット言語の翻訳が含まれている必要があります。

並列データをバッチ翻訳ジョブに追加するときは、Active Custom Translation ジョブを作成します。

Note

Active Custom Translation ジョブは、並列データを使用しない他のジョブよりも高いレートで料金が設定されます。詳細については、「[Amazon Translate 料金表](#)」を参照してください。

詳細については、「[並列データによる翻訳のカスタマイズ \(Active Custom Translation\)](#)」を参照してください。

8. 「アクセス許可」で、Amazon S3 の入出力ファイルに必要なアクセス許可を付与する ロールを Amazon Translate Amazon S3に提供します。IAM
 - アカウントにこのIAMロールがすでにある場合は、既存のIAMロールを使用する を選択し、IAMロール で選択します。
 - アカウントにこのIAMロールがまだない場合は、IAMロールの作成 を選択します。IAM ロールで、入力 と出力 S3 バケット を選択します。[Role name (ロール名)] にカスタムの名前を入力します。翻訳ジョブを作成すると、Amazon Translate によってロールが自動的に作成されます。のロール名には、プレフィックス AmazonTranslateServiceRole- IAMが付きません。

Note

翻訳出力を独自のKMSキーで暗号化することを選択した場合、IAMロールの作成を選択することはできません。この場合、既存のIAMロールを使用する必要があります。また、KMSキーには、ロールにキーの使用を許可するキーポリシーが必要です。

詳細については、「[暗号化をカスタマイズするための前提となるアクセス許可](#)」を参照してください

9. [Create job (ジョブの作成)] を選択します。

コンソールが [Translation jobs (翻訳ジョブ)] ページに戻ります。このページの上部にあるバナーにジョブ作成ステータスが表示されます。数分後、ジョブがテーブルに表示されます。

10. [Name (名前)] 列でジョブ名を選択すると、ジョブの詳細ページが開きます。

翻訳ジョブの実行中は、[Status (ステータス)] フィールドに [In progress (進行中)] と表示されます。

11. ステータスが [Completed (完了)] になったら、[Output file location (出力ファイルの場所)] でリンクを選択して翻訳出力に進みます。コンソールが Amazon S3 の出力バケットに移動します。**12. 出力ファイルをダウンロードするには、それぞれのチェックボックスを選択して、[ダウンロード] を選択します。****AWS CLI**

を使用して翻訳ジョブを実行するには AWS CLI、[start-text-translation-job](#) コマンドを使用して、`parallel-data-names` パラメータの並列データリソースの名前を指定します。

Example S tart-text-translation-job コマンド

次の例では、Amazon S3 の入力バケットに保存されている Excel ファイルを送信して、翻訳ジョブを実行します。このジョブは、リクエストに含まれる並列データによってカスタマイズされます。

```
$ aws translate start-text-translation-job \  
> --input-data-config ContentType=application/vnd.openxmlformats-officedocument.spreadsheetml.sheet,S3Uri=s3://amzn-s3-demo-bucket/input/ \  
> --output-data-config S3Uri=s3://amzn-s3-demo-bucket/output/ \  
> --data-access-role-arn arn:aws:iam::111122223333:role/my-iam-role \  
> --source-language-code en \  
>
```

```
> --target-language-codes es it \  
> --job-name my-translation-job
```

コマンドが成功すると、Amazon Translate からジョブ ID と次のステータスでの応答があります。

```
{  
  "JobId": "4446f95f20c88a4b347449d3671fbe3d",  
  "JobStatus": "SUBMITTED"  
}
```

翻訳ジョブの出力をカスタマイズする場合は、次のパラメータを使用できます。

--settings

翻訳出力を構成するための設定で、以下のオプションを含みます。

翻訳出力で簡潔性を有効にします。Amazon Translate は、バッチ翻訳ジョブの簡潔さをサポートしていません。詳細については、「[Amazon Translate での簡潔さの使用](#)」を参照してください。

不敬表現のマスキングを有効にして、不適切な単語やフレーズをマスクします。有効にするには、不敬表現のパラメータを Profanity=MASK に設定します。詳細については、「[Amazon Translate における不適切な言葉やフレーズのマスキング](#)」を参照してください。いずれかのターゲット言語が不敬表現のマスキングをサポートしていない場合、翻訳ジョブはどのターゲット言語でも不敬表現をマスクしません。

翻訳出力の形式レベルを設定します。Formality パラメータを FORMAL または INFORMAL に設定します。ジョブに複数のターゲット言語を指定した場合、Translate はサポートされていないターゲット言語の形式設定を無視します。詳細については、「[Amazon Translate の形式設定](#)」を参照してください。

--terminology-names

翻訳ジョブに追加するカスタム用語リソースの名前。このリソースでは、各用語のソース用語と翻訳例を挙げています。ジョブに複数のターゲット言語を指定した場合、Translate は、用語ファイルにソース用語のエントリがある要求されたターゲット言語ごとに、指定された用語を使用します。

このパラメータでは、カスタム用語リソースが 1 つだけ許容されます。

使用可能なカスタム用語リソースのリストについては、[list-terminologies](#) コマンドを使用してください。

詳細については、「[カスタム用語による翻訳のカスタマイズ](#)」を参照してください。

--parallel-data-names

翻訳ジョブに追加する並列データリソースの名前。このリソースには、テキストのセグメントの翻訳方法を示した例が含まれています。ジョブに複数のターゲット言語を指定する場合、並列データファイルにはすべてのターゲット言語の翻訳が含まれている必要があります。

並列データを翻訳ジョブに追加するときは、Active Custom Translation ジョブを作成します。

このパラメータでは、並列データリソースが 1 つだけ許容されます。

Note

Active Custom Translation ジョブは、並列データを使用しない他のジョブよりも高いレートで料金が設定されます。詳細については、「[Amazon Translate 料金表](#)」を参照してください。

使用可能な並列データリソースのリストについては、[list-parallel-data](#) コマンドを使用してください。

詳細については、「[並列データによる翻訳のカスタマイズ \(Active Custom Translation\)](#)」を参照してください。

翻訳ジョブのステータスを確認するには、[describe-text-translation-job](#) コマンドを使用します。

Example D describe-text-translation-job コマンド

次の例では、ジョブ ID を指定してジョブのステータスを確認します。この ID は、ジョブが `start-text-translation-job` コマンドで開始されたときに Amazon Translate により提供されたものです。

```
$ aws translate describe-text-translation-job \  
> --job-id 4446f95f20c88a4b347449d3671fbe3d
```

Amazon Translate からジョブプロパティでの応答があります。これにはそのジョブのステータスが含まれます。

```
{
  "TextTranslationJobProperties": {
    "JobId": "4446f95f20c88a4b347449d3671fbe3d",
    "JobName": "my-translation-job",
    "JobStatus": "COMPLETED",
    "JobDetails": {
      "TranslatedDocumentsCount": 0,
      "DocumentsWithErrorsCount": 0,
      "InputDocumentsCount": 1
    },
    "SourceLanguageCode": "en",
    "TargetLanguageCodes": [
      "es",
      "it"
    ],
    "SubmittedTime": 1598661012.468,
    "InputDataConfig": {
      "S3Uri": "s3://amzn-s3-demo-bucket/input/",
      "ContentType": "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"
    },
    "OutputDataConfig": {
      "S3Uri": "s3://amzn-s3-demo-bucket/output/111122223333-TranslateText-4446f95f20c88a4b347449d3671fbe3d/"
    },
    "DataAccessRoleArn": "arn:aws:iam::111122223333:role/my-iam-role"
  }
}
```

JobStatus 値が IN_PROGRESS である場合は、数分が経過してから [describe-text-translation-job](#) を再実行し、ステータスが COMPLETED になるまで続けます。ジョブが完了すれば、OutputDataConfig の S3Uri フィールドにある場所に翻訳結果をダウンロードできます。

Amazon Translate API

Amazon Translate を使用してバッチ翻訳ジョブを送信するにはAPI、[StartTextTranslationJob](#) オペレーションを使用します。

バッチ翻訳ジョブのモニタリングと分析

ジョブの ID を使用して、進行状況をモニタリングし、Amazon S3 での出力ドキュメントの場所を取得できます。特定のジョブを監視するには、[DescribeTextTranslationJob](#) オペレーションを

使用します。[ListTextTranslationJobs](#) オペレーションを使用して、アカウント内のすべての翻訳ジョブに関する情報を取得することもできます。結果を特定の条件に一致するジョブに制限するには、[ListTextTranslationJobs](#) オペレーションの `filter` パラメータを使用します。ジョブ名、ジョブステータス、またはジョブが送信された日時で結果をフィルタ処理できます。

Example `describe-text-translation-job` コマンド

次の例では、AWS CLI を使用して [DescribeTextTranslationJob](#) コマンドを実行することで、ジョブのステータスをチェックします。

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

このコマンドから次の出力が返されます。

```
{
  "TextTranslationJobProperties": {
    "InputDataConfig": {
      "ContentType": "text/plain",
      "S3Uri": "s3://input-bucket-name/folder"
    },
    "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://bucket-name/output/012345678901-TranslateText-1c1838f470806ab9c3e0057f14717bed/"
    }
  }
}
```

```
}
```

[StopTextTranslationJob](#) オペレーションを使用して、ステータスが IN_PROGRESS の間にバッチ翻訳ジョブを停止できます。

Example stop-text-translation-job コマンド

次の例では、AWS CLI を使用して [StopTextTranslationJob](#) コマンドを実行することで、バッチ翻訳を停止します。

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

このコマンドから次の出力が返されます。

```
{
  "TextTranslationJobProperties": {
    "InputDataConfig": {
      "ContentType": "text/plain",
      "S3Uri": "s3://input-bucket-name/folder"
    },
    "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://output-bucket-name/012345678901-TranslateText-5236d36ce5192abdb3e2519f3ab8b065/"
    }
  }
}
```

バッチ翻訳結果の取得

ジョブのステータスが COMPLETED または COMPLETED_WITH_ERROR になったら、指定した Amazon S3 フォルダで出力ドキュメントを使用できます。出力ドキュメント名は入力ド

キュメント名と一致し、プレフィックスとしてターゲット言語コードが追加されます。たとえば、mySourceText.txt という名前のドキュメントをフランス語に翻訳した場合、出カドキュメントは fr.mySourceText.txt という名前になります。

バッチ翻訳ジョブのステータスが FAILED の場合、[DescribeTextTranslationJob](#) オペレーションのレスポンスには、ジョブが正常に完了しなかった理由を説明する Message フィールドが含まれます。

各バッチ翻訳ジョブでは、翻訳された文字の合計数や発生したエラーの数など、実行された翻訳に関する情報を含む補助ファイルも生成されます。このファイルは、*target-language-code*.auxiliary-translation-details.json という名前で、出力フォルダの details サブフォルダに生成されます。

以下に示しているのは、バッチ翻訳の補助ファイルの例です。

```
{
  "sourceLanguageCode": "en",
  "targetLanguageCode": "fr",
  "charactersTranslated": "105",
  "documentCountWithCustomerError": "0",
  "documentCountWithServerError": "0",
  "inputDataPrefix": "s3://input-bucket-name/folder",
  "outputDataPrefix": "s3://output-bucket-name/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"
        }
      ]
    }
  ]
}
```

Amazon Translate での翻訳のカスタマイズ

次の設定を使用すれば、Amazon Translate で生成した翻訳をカスタマイズできます。

- **Do-not-translate タグ** — 開始タグと終了タグを使用して、(HTML コンテンツ内の) 翻訳したくないコンテンツを指定します。
- **カスタム用語** — Amazon Translate でブランド名などの特定の用語をどのように翻訳するかを定義します。
- **簡潔さ** — ほとんどの翻訳において、翻訳出力の長さを (簡潔でない翻訳出力と比較して) 短くします。リアルタイムのテキスト翻訳では簡潔にしましょう。
- **冒瀆的な表現** — 翻訳出力に含まれる不適切な単語やフレーズを隠します。
- **形式性** — 翻訳出力における言語の形式性のレベルを設定します。
- **パラレルデータ** — 提供する翻訳サンプルに含まれるスタイル、トーン、単語の選択を反映するように翻訳出力を調整します。

トピック

- [Amazon Translate do-not-translate での使用](#)
- [カスタム用語による翻訳のカスタマイズ](#)
- [Amazon Translate での簡潔さの使用](#)
- [Amazon Translate における不適切な言葉やフレーズのマスキング](#)
- [Amazon Translate の形式設定](#)
- [並列データによる翻訳のカスタマイズ \(Active Custom Translation\)](#)

Amazon Translate do-not-translate での使用

HTML コンテンツの場合、do-not-translate 翻訳したくないテキストにタグを追加できます。この機能はコンソールと API の操作で使用できます。

トピック

- [do-not-translate コンソールでの使用](#)
- [API do-not-translate との併用](#)

do-not-translate コンソールでの使用

ソース HTML `translate="no"` コンテンツで、翻訳したくないコンテンツを囲む HTML タグを指定します。例えば、次のテキストを英語からドイツ語に翻訳するには:

```
In French, the Louvre Museum is Musée du Louvre.
```

「Musée du Louvre」というテキストはフランス語のままではいけないので、このコンテンツの翻訳をスキップするには `span` タグを使います。

```
<p>In French, the Louvre Museum is <span translate="no">Musée du Louvre</span>.</p>
```

この文は、その結果ドイツ語に翻訳されています。

```
<p>Auf Französisch ist <span translate="no">Musée du Louvre</span> das Louvre-Museum.</p>
```

API do-not-translate との併用

リアルタイム API オペレーション (`TranslateText` と `TranslateDocument`) と非同期 `StartTextTranslationJob` API オペレーションで使用できます `do-not-translate`。API リクエストに指定するソーステキスト内で、翻訳する必要のないコンテンツを任意のタイプの HTML 要素で指定することができます。

次の例では、一部のテキストを英語からスペイン語に翻訳し、一部のテキストは英語のままにしたいと考えています。

```
aws translate translate-text \  
  --source-language-code "en" \  
  --target-language-code "es" \  
  --region us-west-2 \  
  --text "<p>You can translate this paragraph to any language.</p> <p translate=no>But do not translate this.</p>"
```

この API リクエストは次のスペイン語の翻訳を返します。

```
{  
  "TranslatedText": "<p>Puede traducir este párrafo a cualquier idioma.</p>"
```

```
        <p translate=no>But do not translate this.</p>",
    "SourceLanguageCode": "en",
    "TargetLanguageCode": "es"
}
```

カスタム用語による翻訳のカスタマイズ

翻訳リクエストでカスタム用語を使用すると、ブランド名、キャラクター名、モデル名、およびその他の独自コンテンツを、目的どおりに翻訳することができます。

用語ファイルを作成して Amazon Translate アカウントにアップロードすることができます。ファイルサイズと用語ファイルの数については、「[Service Quotas](#)」を参照してください。テキストを翻訳する場合、使用するカスタム用語ファイルを任意で選択できます。Amazon Translate は、ソーステキストと用語ファイルの間で一致を検出すると、用語ファイルにある翻訳を使用します。

次の例を考えてみましょう。Amazon Photos では、Amazon プライム会員に写真と動画の無料ストレージを提供しています。フランス語では、Amazon Photos は翻訳されず原語のまま残されます。

Amazon Translate でコンテキストを追加せずに Amazon Photos をフランス語に翻訳した場合、結果は Photos d'Amazon となりますが、これは望ましい翻訳ではありません。

Amazon Photos という用語にカスタム用語のエントリを追加し、フランス語訳が Amazon Photos であることを指定すると、Amazon Translate はカスタム用語を使用してフレーズを意図したとおりに翻訳します。

Amazon Translate は、すべての翻訳にターゲット用語を使用することを保証するものではありません。カスタム用語では、翻訳コンテキストにおけるソース用語とターゲット用語の意味を使用して、ターゲット用語を使用するかどうかを決定します。詳細については、「[ベストプラクティス](#)」を参照してください。

トピック

- [カスタム用語の作成](#)
- [カスタム用語の使用](#)
- [を使用したカスタム用語の例 AWS SDK for Python \(Boto\)](#)
- [用語の暗号化](#)
- [ベストプラクティス](#)

カスタム用語の作成

用語ファイルを作成してカスタム用語を定義します。Amazon Translate は、CSV、TSV、または TMX ファイル形式をサポートしています。ファイル内の各エントリには、ソース用語とそれに対応する (翻訳された) 各ターゲット言語での用語が含まれています。

用語ファイルを作成したら、Amazon Translate アカウントにアップロードします。

Important

カスタム用語のソーステキストでは大文字と小文字が区別されます。Amazon Translate は、翻訳中に入力ドキュメントに完全に一致するものを見つけると、カスタム用語を使用します。

用語ファイルの形式

次の例では、CSV 形式の用語ファイルを示しています。

CSV (カンマ区切り値)

```
en,fr,es
Amazon Photos,Amazon Photos,Amazon Photos
```

次の例では、TMX 形式の用語ファイルを示しています。TMX ファイルは、翻訳ソフトウェアでよく使用される XML 形式を使用しています。

TMX (Translation Memory eXchange)

```
<?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 Photos</seg>
      </tuv>
    </tu>
  </body>
</tmx>
```



```
<tuv xml:lang="fr">
  <seg>Amazon Photos</seg>
</tuv>
<tuv xml:lang="es">
  <seg>Amazon Photos</seg>
</tuv>
</tu>
</body>
</tmx>
```

方向性

カスタム用語ファイルをアップロードするときは、カスタム用語の方向性の値を設定します。方向性とは、用語ファイルが1つのソース言語を指定しているのか、複数のソース言語を指定しているのかを示します。

方向性を指定するには、次のいずれかの値を設定します。

Uni-directional (単一方向)

用語ファイルには1つのソース言語 (リストの最初の言語) が含まれています。その他の言語はすべてターゲット言語です。

例えば、CSV ファイルの場合、最初の列にソース言語のテキストが記載され、他のすべての列にはターゲット言語のテキストが記載されます。

Multi-directional (多方向)

ファイル内の任意の言語が、ソース言語またはターゲット言語になります。例えば、用語ファイルに英語、フランス語、スペイン語のテキストが含まれている場合、次の言語ペアを翻訳するジョブでファイルを使用できます。

- 英語 → フランス語
- 英語 → スペイン語
- フランス語 → 英語
- フランス語 → スペイン語
- スペイン語 → 英語
- スペイン語 → フランス語

一方で、これら6つの翻訳ジョブに対して単一方向の用語ファイルを3つ (ソース言語ごとに1つずつ) 作成する必要があります。

カスタム用語の使用

[TranslateText](#) オペレーションでテキストを翻訳するときにカスタム用語を使用するには、オプションの `TerminologyNames` パラメータを含めます。

例えば、`Amazon_Family.csv` という次の用語ファイルをアカウントにアップロードするとします。

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

以下の CLI コマンドによって、カスタム用語を使用してテキストを翻訳することができます。

Note

この例は、Unix、Linux、および macOS 用にフォーマットされています。Windows の場合は、各行末のバックスラッシュ (\) Unix 連結文字をキャレット (^) に置き換えてください。

```
aws translate translate-text \
  --region region \
  --source-language-code "en" \
  --target-language-code "fr" \
  --terminology-names "Amazon_Family" \
  --text "Have you ever stored videos in Amazon Family?"
```

ここでは、選択したカスタム用語を使用してこのテキストが「Avez-vous déjà fait des achats avec Amazon Famille?」と訳されます。望ましくない直訳「Avez-vous déjà fait des achats avec Famille Amazon?」にはなりません。

次の例では、同じ用語ファイルを Python で使用方法を示しています。

```
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")
```

カスタム用語を使った Amazon Translate オペレーションの使用の詳細については、「[API オペレーション](#)」を参照してください。

を使用したカスタム用語の例 AWS SDK for Python (Boto)

次の例は、Python でカスタム用語操作を使用する方法を示しています。この例を実行するには、を使用して Python SDK をインストールします AWS CLI。手順については、「[the section called “のセットアップ AWS CLI”](#)」を参照してください。

```
#!/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.")
```

用語の暗号化

Amazon Translate はお客様のすべてのデータを保護するべく取り組んでおり、カスタム用語についても同様です。各カスタム用語は、作成時に暗号化され、お客様のみがアクセス可能になります。

使用できる暗号化オプションは 3 つです。

- AWS 暗号化を使用する。AWS 暗号化は情報を保護するためのデフォルトオプションです。
- お客様のアカウントに関連付けられた暗号化キーの使用。コンソールのメニューで、使用する関連付けられた暗号化キーを選択できます。
- お客様のアカウントに関連付けられていない暗号化キーの使用。暗号化キーの Amazon リソースネーム (ARN) を入力する入力フィールドがコンソールに表示されます。

ベストプラクティス

カスタム用語を使用する際は、以下の一般的なベストプラクティスを参考にしてください。

- カスタム用語は整理しておきましょう。翻訳された値を制御する必要がある用語のみを含めてください。
- カスタム用語は大文字と小文字が区別されます。大文字を使ったバージョンと小文字を使ったバージョンに単語を翻訳する必要がある場合は、各バージョンのエントリを含めます。
- カスタム用語は、スペース、句読点、大文字と小文字の使い分けを制御するためのツールではありません。たとえば、次のようなエントリは避けてください。

- スペースの追加 — EN: 米国 FR: U S A
- 句読点の追加 — EN: 米国 FR: 米国
- 大文字の変更 — EN: 米国 FR: 米国
- 同じソースフレーズに対して複数の異なる翻訳を含めないでください。例:
 - エントリ 1 – EN: Amazon、FR: Amazon
 - エントリ 2 – EN: Amazon、FR: Amazone
- Amazon Translate でサポートされているすべての言語のカスタム用語ファイルを作成できます。

Amazon Translate は、カスタム用語がすべての翻訳でターゲット用語を使用することを保証しません。カスタム用語集を高い精度で作成するには、用語ファイル用のコンテンツを作成する際に以下のベストプラクティスに従ってください。

- カスタム用語は、動詞や同形異義語を含むあらゆる単語に適しています。ブランド名や製品名などの固有名詞は理想的なエントリです。
- 訳語はターゲット言語で流暢に読める必要があります。特殊文字や書式が多数含まれている対象用語には、カスタム用語集はお勧めしません。
- 用語ファイルには、複数の単語からなるフレーズや節を含めることができます。ただし、複数の単語を含む用語は、ターゲット言語で流暢に読める可能性が低くなります。
- カスタム用語では、翻訳コンテキストにおける原文用語と訳文用語の意味に基づいて、訳文用語を使用するかどうかが決まります。ターゲット用語が特定の翻訳コンテキストに合わない場合、Amazon Translate はターゲット用語を使用しない場合があります。

たとえば、用語ファイルに次のような英語からフランス語へのエントリが含まれているとします。

EN: order, FR: commande(英語の「to order」をフランス語の「commander」に翻訳したものに基づく)。

Amazon Translate は、翻訳コンテキストが一致しないため、次の文を翻訳するときこのエントリを使用しません。

「私たちがお手伝いできるよう、お名前を教えてください。」

このような状況を避けるための提案:

- 各言語のターゲット用語がソース用語と意味的に同等であることを確認してください。
- 複数の意味を持つ原文用語や訳文用語は避けてください。

Amazon Translate での簡潔さの使用

言語間の翻訳では、翻訳出力が (文字数で) 希望よりも長くなることがあります。一部のシナリオ (キャプション、字幕、見出し、フォームフィールドなど) では、余分な文字を入れるスペースがないと、出力が長いと問題が発生する可能性があります。

Amazon Translate でリアルタイムのテキスト翻訳を実行するときに、簡潔設定を有効にできます。簡潔にすると、ほとんどの翻訳で翻訳出力の長さが (簡潔でない翻訳出力に比べて) 短くなります。

トピック

- [簡潔設定を使用する](#)
- [サポートされている言語](#)

簡潔設定を使用する

簡潔設定はリアルタイムのテキスト翻訳に使用できます。Amazon Translate は、リアルタイムのドキュメント翻訳や非同期翻訳ジョブの簡潔化をサポートしていません。

リアルタイムのテキスト翻訳リクエストを簡潔にするには、次のいずれかを実行してください。

- コンソール — リアルタイム翻訳ページの「テキスト」タブの「その他の設定」で、「簡潔さ」設定を選択します。
- AWS CLI — `translate-text` コマンドで、`--settings`パラメータの簡潔さをに設定します。Brevity=ON詳細については、AWS CLI コマンドリファレンスの「[translate-text](#)」を参照してください。
- AWSAPI — [TranslateText](#)API オペレーションでは、設定パラメータで簡潔さを設定します。

サポートされている言語

Amazon Translate では、以下の言語ペアの簡潔化がサポートされています。

- [任意のソース言語から次の表のいずれかの言語へ。](#)
- 次の表のいずれかの言語から英語へ。

言語	言語コード
フランス語	fr
ドイツ語	de
イタリア語	it
ポルトガル語 (ブラジル)	pt
スペイン語	es

サポートされていない言語ペアで簡潔な翻訳を依頼すると、簡潔さの設定をオフにしたまま翻訳が実行されます。

Amazon Translate における不適切な言葉やフレーズのマスキング

Amazon Translate で翻訳を実行する際に不敬設定を有効にすると、翻訳出力内の不適切な単語とフレーズが隠されます (マスク)。

不適切な単語やフレーズを隠す場合、Amazon Translate ではそれらがグローバル文字列「?#\$@ \$」に置換されます。この 5 文字のシーケンスは、不適切な各単語または各フレーズに対して、その長さや数に関係なく使用されます。

Amazon Translate では、ソース言語とターゲット言語が同一である翻訳リクエストには不敬マスキングは実行されません。

場合によっては、ソース入力内の不適切な単語が翻訳出力内で自然に無害な表現になることがあります。このような場合、マスキングは適用されません。

Amazon Translate では、文脈上ではなく、文字通り不敬な単語やフレーズを 1 つ 1 つ検出します。つまり、不適切な言葉は、文脈上は無害である場合でも、マスクされる可能性があるということです。たとえば、Amazon Translate により「jerk」(ばか) が不適切な単語として検出される場合、「jerk chicken」(鶏肉料理の一種) というフレーズは、無害な言葉であっても「?#\$@\$ chicken」と書き込まれます。(ここでは「jerk」をあくまで一例として使用しています。Amazon Translate ではこの単語は不敬語として検出されません。)

トピック

- [不敬表現設定の使用](#)

• [サポートされていない言語](#)

不敬表現設定の使用

不敬設定は、Amazon Translate のリアルタイム翻訳と非同期バッチ処理という翻訳オペレーションの両方で使用できます。

リアルタイム翻訳リクエストで不敬語をマスクするには、次のいずれかの作業を実行します。

- Amazon Translate コンソールの [Real-time translation (リアルタイム翻訳)] ページの [Additional settings (追加設定)] で、[Profanity (不敬)] 設定を有効にします。
- `translate-text` のコマンドで AWS CLI、パラメータをに設定します。 `--settings Profanity=MASK` 詳細については、AWS CLI コマンドリファレンスの「[translate-text](#)」を参照してください。
- Amazon Translate API `TranslateText` のアクションでは、Settings パラメータを使用して不適切な表現のマスクを設定します。詳細については、API [TranslateText](#) リファレンスのを参照してください。

非同期バッチ操作で不敬語をマスクするには、[バッチ翻訳ジョブの実行](#) を参照してください。

サポートされていない言語

Amazon Translate でサポートされているターゲット言語を翻訳する場合は不敬表現をマスクできません。ただし、次の例外があります。

言語	言語コード
ベンガル語	bn
ヒンディー語	hi
マラーラム語	ml
パンジャブ語	pa
シンハラ語	si
ベトナム語	vi

Amazon Translate でサポートされているすべての言語については、「[サポートされている言語と言語コード](#)」を参照してください。

Amazon Translate の形式設定

必要に応じて、サポートされているターゲット言語への翻訳に必要な形式のレベルを指定できます。形式設定は、翻訳出力における形式的な言語の使用レベル ([敬語](#)または[言語使用域](#)とも呼ばれます) を制御します。形式設定は、リアルタイム翻訳および非同期バッチ処理で利用できます。

形式では、以下の値がサポートされています。

- **インフォーマル** — 翻訳されたテキスト内のすべての文で、インフォーマルなコミュニケーションに関連する言語構造が使用されます。例えば、翻訳されたテキストでは、親しみのある二人称代名詞とその動詞の一致が使用されます (日本語の場合はくだけた表現)。
- **フォーマル** — 翻訳されたテキスト内のすべての文で、フォーマルで丁寧なコミュニケーションに関連する言語構造が使用されます。例えば、翻訳されたテキストでは、格式ばった二人称代名詞とその動詞の一致が使用されます (日本語の場合は丁寧語)。

「Are you sure? (よろしいですか)」という文を例に挙げます。ドイツ語では 2 つの正しい翻訳が可能です。フォーマルな言語使用域の場合は「Sind Sie sicher?」、インフォーマルな場合は「Bist du sicher?」となります。

Amazon Translate がターゲット言語の形式レベルをサポートしていない場合、または形式パラメータが指定されなかった場合、翻訳ジョブは形式設定を無視します。

トピック

- [形式設定の使用](#)
- [サポートされている言語](#)

形式設定の使用

リアルタイムの翻訳リクエストで形式を設定するには、次のいずれかの手順を実行します。

- Amazon Translate コンソールの [リアルタイムでの翻訳] ページで、[追加設定] から [形式] 設定を有効にして、いずれかの値を選択します。
- Amazon Translate API [TranslateText](#) のオペレーションで設定パラメータを使用してください。

- AWS CLI の `translate-text` コマンドに対して、`--settings` パラメータを `Formality=FORMAL` または `Formality=INFORMAL` に設定します。詳細については、AWS CLI コマンドリファレンスの「[translate-text](#)」を参照してください。

バッチ翻訳リクエストの形式を設定するには、翻訳ジョブを開始するときに [形式] パラメータを設定します。詳細と例については、「[バッチ翻訳ジョブの実行](#)」を参照してください。

CLI または API リクエストの場合、レスポンスの `AppliedSettings` フィールドにはリクエストの形式設定 (ある場合) が含まれます。ターゲット言語が形式をサポートしていない場合、レスポンスの `AppliedSettings` の値は `NULL` です。

サポートされている言語

Amazon Translate は、任意のソース言語から以下のターゲット言語への翻訳の形式設定をサポートしています。

言語	言語コード
オランダ語	nl
フランス語	fr
フランス語 (カナダ)	fr-CA
ドイツ語	de
ヒンディー語	hi
イタリア語	it
日本語	ja
韓国語	ko
ポルトガル語 (ポルトガル)	pt-PT
スペイン語	es
スペイン語 (メキシコ)	es-MX

Amazon Translate でサポートされているすべての言語については、「[サポートされている言語と言語コード](#)」を参照してください。

並列データによる翻訳のカスタマイズ (Active Custom Translation)

並列データを Amazon Translate に追加してバッチ翻訳ジョブの出力をカスタマイズします。並列データは、テキストのセグメントの翻訳方法を示した例で構成されています。これには、ソース言語でのテキスト例の一群が含まれており、各例については、1 つ以上のターゲット言語での対象の翻訳出力が含まれています。

並列データをバッチ翻訳ジョブに追加するときは、Active Custom Translation ジョブを作成します。これらのジョブを実行すると、Amazon Translate ではランタイムに並列データが使用されて、カスタマイズされた機械翻訳出力が生成されます。並列データで見つかったスタイル、トーン、単語の選択肢を反映させるために翻訳を適応させます。並列データを使用すると、ライフサイエンス、法律、財務など、特定のドメインに固有の用語やフレーズに合わせて翻訳を調整できます。

Note

Active Custom Translation ジョブは、並列データを使用しない他のジョブよりも高いレートで料金が設定されます。詳細については、「[Amazon Translate 料金表](#)」を参照してください。

たとえば、次の並列データは CSV ファイルで定義されています。

```
"en","fr"  
"How are you?","Comment ça va ?"
```

この例では、英語 (en) がソース言語で、フランス語 (fr) がターゲット言語です。この例は、ソースフレーズ "How are you?" がどのようにしてフランス語に翻訳されるかを示しています。この例の入力ファイルは、Amazon Translate にインポートされた後で翻訳ジョブに適用されて、出力に影響を及ぼす可能性があります。このジョブの中で、Amazon Translate では "How are you?" が日常的な "Comment ça va ?" に翻訳されます。これは格式ばった "Comment allez-vous ?" とは対照的です。たとえば、このジョブで次のソーステキストを受け取ったとします。

```
Hello, how are you?  
How are you?  
Hi, how are you?
```

```
How are you doing?
```

このテキストから、ジョブでは次の翻訳が生成されます。

```
Bonjour, comment ça va ?  
Comment ça va ?  
Salut, comment ça va ?  
Comment ça va ?
```

対照的に、並列データを使用せずにジョブが実行された場合、出力には硬めの表現である "comment allez-vous" が含まれる可能性があります。

```
Bonjour, comment allez-vous ?  
Comment allez-vous ?  
Salut, comment allez-vous ?  
Comment allez-vous ?
```

並列データを使用してバッチ翻訳ジョブをカスタマイズすることで、翻訳例を使って学習させたカスタム翻訳モデルを使用するのと同じような感覚で、出力に影響を与えることができます。Active Custom Translation にはカスタムモデルのトレーニングは不要であるため、このようなトレーニングにかかる時間と費用を節約できます。翻訳要件の経時的な変化に伴い、並列データを更新して出力を絞り込むことができます。これはカスタムモデルの再トレーニングよりも簡単です。

利用可能なリージョン

Active Custom Translation は以下のリージョンで使用できます。

- 米国東部 (バージニア北部)
- 米国西部 (オレゴン)
- 欧州 (アイルランド)

トピック

- [Amazon Translate の並列データ入力ファイル](#)
- [Amazon Translate への並列データの追加](#)
- [Amazon Translate における並列データの表示と管理](#)

Amazon Translate の並列データ入力ファイル

Amazon Translate で並列データリソースを作成する前に、翻訳例を含む入力ファイルを作成する必要があります。並列データ入力ファイルでは、Amazon Translate でサポートされている言語を使用する必要があります。これらの言語のリストについては、「[サポートされている言語と言語コード](#)」を参照してください。

並列データの例

次の表に示すテキストは、並列データ入力ファイルにフォーマットできる翻訳セグメントの例を示しています。

en	es
Amazon Translate is a neural machine translation service.	Amazon Translate es un servicio de traducción automática basado en redes neuronales.
Neural machine translation is a form of language translation automation that uses deep learning models.	La traducción automática neuronal es una forma de automatizar la traducción de lenguajes utilizando modelos de aprendizaje profundo.
Amazon Translate allows you to localize content for international users.	Amazon Translate le permite localizar contenido para usuarios internacionales.

表の最初の行に表示されているのは言語コードです。1 つ目の言語である英語 (en) がソース言語です。スペイン語 (es) がターゲット言語です。1 列目にはソーステキストの例が示されています。もう一つの列には翻訳の例が記載されています。この並列データでバッチジョブをカスタマイズすると、Amazon Translate ではその例が反映されて翻訳が適応されます。

入力ファイル形式

Amazon Translate では、並列データ入力ファイルに対して次の形式がサポートされています。

- Translation Memory eXchange (TMX)
- カンマ区切り値 (CSV)
- タブ区切り値 (TSV)

TMX

Example TMX 入力ファイル

次のサンプル TMX ファイルでは、Amazon Translate で使用できる形式で並列データが定義されています。このファイルでは、英語 (en) がソース言語です。スペイン語 (es) がターゲット言語です。バッチジョブの出力を調整するために Amazon Translate で並列データの入力ファイルとして使用できるいくつかの例が提示されています。

```
<?xml version="1.0" encoding="UTF-8"?>
<tmx version="1.4">
  <header srclang="en"/>
  <body>
    <tu>
      <tuv xml:lang="en">
        <seg>Amazon Translate is a neural machine translation service.</seg>
      </tuv>
      <tuv xml:lang="es">
        <seg>Amazon Translate es un servicio de traducción automática basado
en redes neuronales.</seg>
      </tuv>
    </tu>
    <tu>
      <tuv xml:lang="en">
        <seg>Neural machine translation is a form of language translation
automation that uses deep learning models.</seg>
      </tuv>
      <tuv xml:lang="es">
        <seg>La traducción automática neuronal es una forma de automatizar
la traducción de lenguajes utilizando modelos de aprendizaje profundo.</seg>
      </tuv>
    </tu>
    <tu>
      <tuv xml:lang="en">
        <seg>Amazon Translate allows you to localize content for
international users.</seg>
      </tuv>
      <tuv xml:lang="es">
        <seg>Amazon Translate le permite localizar contenido para usuarios
internacionales.</seg>
      </tuv>
    </tu>
  </body>
</tmx>
```

```
</body>  
</tmx>
```

TMX の要件

TMX ファイルで並列データを定義する場合は、Amazon Translate の次の要件に注意してください。

- Amazon Translate では TMX 1.4b がサポートされています。詳細については、Globalization and Localization Association ウェブサイトの [TMX 1.4b の仕様](#) を参照してください。
- header 要素には必ず srclang 属性を含めてください。この属性の値によって並列データのソース言語が決まります。
- body 要素には必ず、少なくとも 1 つの翻訳単位 (tu) 要素を含めてください。
- 各 tu 要素には必ず、少なくとも 2 つの翻訳単位バリエーション (tuv) 要素を含めてください。これらの tuv 要素のうち 1 つに、必ず、header 要素の srclang 属性に割り当てられている値と値が同じである xml:lang 属性を含めてください。
- すべての tuv 要素に必ず xml:lang 属性を含めてください。
- すべての tuv 要素に必ずセグメント (seg) 要素を含めてください。
- 入力ファイルの処理中、空である、またはホワイトスペースしかない seg 要素が発生した場合は、Amazon Translate では特定の tu 要素または tuv 要素がスキップされます。
 - seg 要素がソース言語に対応している場合は、Amazon Translate では seg 要素が占めている tu 要素がスキップされます。
 - seg 要素がターゲット言語に対応している場合は、Amazon Translate では seg 要素が占めている tuv 要素のみがスキップされます。
- 入力ファイルの処理中、空である、または 1000 バイトを超えている seg 要素が発生した場合は、Amazon Translate では特定の tu 要素または tuv 要素がスキップされます。
 - seg 要素がソース言語に対応している場合は、Amazon Translate では seg 要素が占めている tu 要素がスキップされます。
 - seg 要素がターゲット言語に対応している場合は、Amazon Translate では seg 要素が占めている tuv 要素のみがスキップされます。
- ソーステキストが同一である複数の tu 要素が入力ファイルに含まれている場合、Amazon Translate では次のいずれかの処理が実行されます。
 - tu 要素に changedate 属性が含まれている場合は、直近の日付の要素が使用されます。
 - それ以外の場合は、そのファイルの最後に最も近い時期に生じる要素が使用されます。

CSV

次のサンプル CSV ファイルでは、Amazon Translate で使用できる形式で並列データが定義されています。このファイルでは、英語 (en) がソース言語です。スペイン語 (es) がターゲット言語です。バッチジョブの出力を調整するために Amazon Translate で並列データの入力ファイルとして使用できるいくつかの例が提示されています。

Example CSV 入力ファイル

```
en,es
Amazon Translate is a neural machine translation service.,Amazon Translate es un
servicio de traducción automática basado en redes neuronales.
Neural machine translation is a form of language translation automation that uses
deep learning models.,La traducción automática neuronal es una forma de automatizar
la traducción de lenguajes utilizando modelos de aprendizaje profundo.
Amazon Translate allows you to localize content for international users.,Amazon
Translate le permite localizar contenido para usuarios internacionales.
```

CSV 要件

CSV ファイルで並列データを定義する場合は、Amazon Translate の次の要件に注意してください。

- 1 行目は言語コードで構成されます。最初のコードはソース言語で、後続の各コードはターゲット言語です。
- 1 列目の各フィールドにはソーステキストが記載されています。後続の列の各フィールドにはターゲット翻訳が記載されています。
- いずれかのフィールドのテキストにカンマが含まれている場合、そのテキストを二重引用符 (") で囲む必要があります。
- テキストフィールドを複数の行に渡って表示することはできません。
- フィールドの最初の文字として +、-、=、@ は使用できません。この要件は、フィールドが二重引用符 (") で囲まれているかどうかにかかわらず適用されます。
- フィールド内のテキストに二重引用符 (") が含まれている場合は、二重引用符でエスケープする必要があります。たとえば、次のようなテキストがあるとします。

```
34" monitor
```

次のように記述する必要があります。

```
34"" monitor
```

- 入力ファイルの処理中、空である、またはホワイトスペースしかないフィールドが発生した場合は、Amazon Translate では特定の行またはフィールドがスキップされます。
- ソーステキストフィールドが空である場合、Amazon Translate ではそのフィールドが占有している行がスキップされます。
- ターゲット翻訳フィールドが空白である場合、Amazon Translate ではそのフィールドだけがスキップされます。
- 入力ファイルの処理中、空である、または 1000 バイトを超えているフィールドが発生した場合は、Amazon Translate では特定の行またはフィールドがスキップされます。
 - あるソーステキストフィールドが上記のバイト上限を超えている場合、Amazon Translate ではそのフィールドが占有している行がスキップされます。
 - あるターゲット翻訳フィールドが上記のバイト上限を超えている場合、Amazon Translate ではそのフィールドだけがスキップされます。
- ソーステキストが同じである複数のレコードが入力ファイルに含まれている場合、Amazon Translate ではそのファイルの最後に最も近いレコードが使用されます。

TSV

次のサンプル TSV ファイルでは、Amazon Translate で使用できる形式で並列データが定義されています。このファイルでは、英語 (en) がソース言語です。スペイン語 (es) がターゲット言語です。バッチジョブの出力を調整するために Amazon Translate で並列データの入力ファイルとして使用できるいくつかの例が提示されています。

Example TSV 入力ファイル

```
en es
Amazon Translate is a neural machine translation service. Amazon Translate es un
servicio de traducción automática basado en redes neuronales.
Neural machine translation is a form of language translation automation that uses
deep learning models. La traducción automática neuronal es una forma de automatizar
la traducción de lenguajes utilizando modelos de aprendizaje profundo.
Amazon Translate allows you to localize content for international users. Amazon
Translate le permite localizar contenido para usuarios internacionales.
```

TSV 要件

TSV ファイルで並列データを定義する場合は、Amazon Translate の次の要件に注意してください。

- 1 行目は言語コードで構成されます。最初のコードはソース言語で、後続の各コードはターゲット言語です。
- 1 列目の各フィールドにはソーステキストが記載されています。後続の列の各フィールドにはターゲット翻訳が記載されています。
- いずれかのフィールドのテキストにタブ文字が含まれている場合、そのテキストを二重引用符 (") で囲む必要があります。
- テキストフィールドを複数の行に渡って表示することはできません。
- フィールドの最初の文字として +、-、=、@ は使用できません。この要件は、フィールドが二重引用符 (") で囲まれているかどうかにかかわらず適用されます。
- フィールド内のテキストに二重引用符 (") が含まれている場合は、二重引用符でエスケープする必要があります。たとえば、次のようなテキストがあるとします。

```
34" monitor
```

次のように記述する必要があります。

```
34"" monitor
```

- 入力ファイルの処理中、空である、またはホワイトスペースしかないフィールドが発生した場合は、Amazon Translate では特定の行またはフィールドがスキップされます。
 - ソーステキストフィールドが空である場合、Amazon Translate ではそのフィールドが占有している行がスキップされます。
 - ターゲット翻訳フィールドが空白である場合、Amazon Translate ではそのフィールドだけがスキップされます。
- 入力ファイルの処理中、空である、または 1000 バイトを超えているフィールドが発生した場合は、Amazon Translate では特定の行またはフィールドがスキップされます。
 - あるソーステキストフィールドが上記のバイト上限を超えている場合、Amazon Translate ではそのフィールドが占有している行がスキップされます。
 - あるターゲット翻訳フィールドが上記のバイト上限を超えている場合、Amazon Translate ではそのフィールドだけがスキップされます。

- ソーステキストが同じである複数のレコードが入力ファイルに含まれている場合、Amazon Translate ではそのファイルの最後に最も近いレコードが使用されます。

Amazon Translate への並列データの追加

Amazon Translate に並列データを追加するには、Amazon S3 から並列データ入力ファイルをインポートします。その後、並列データを使用して、バッチ翻訳ジョブによって生成される出力をカスタマイズできます。

前提条件

Amazon Translate に並列データを追加する前に、次の作業を行う必要があります。

- 並列データ入力ファイルを用意してください。作成する場合は「[Amazon Translate の並列データ入力ファイル](#)」を参照してください。
- AWS アカウント内に Amazon S3 バケットを用意してください。作成する場合は、Amazon Simple Storage Service ユーザーガイドの「[S3 バケットを作成する方法](#)」を参照してください。
- 入力ファイルを Amazon S3 バケットにアップロードしてください。詳細については、[Amazon Simple Storage Service ユーザーガイド](#)の「S3 バケットにファイルとフォルダをアップロードする方法」を参照してください。

並列データの追加 (Amazon Translate コンソール)

Amazon Translate コンソールを使用して並列データを追加するには、並列データページを使用します。

1. [Amazon Translate コンソール](#) を開きます。
2. 左側のナビゲーションメニューで、[Customization (カスタマイズ)] を選択し、[Parallel data (並列データ)] を選択します。
3. [Parallel data (並列データ)] ページで [Create parallel data (並列データの作成)] を選択します。コンソールに [Create parallel data (並列データの作成)] ページが表示されます。
4. 以下を指定します。

Name (名前)

並列データリソースのカスタム名。アカウントとリージョンで一意的な名前を割り当てる必要があります。

Description (説明) - オプション

カスタム説明。

Parallel data location on S3 (S3 での並列データの場所)

Amazon S3 での並列データ入力ファイルの場所。Amazon S3 のファイルに移動して場所を設けるには、[Select file (ファイルを選択)] を選択します。

File format (ファイル形式)

並列データ入力ファイルの形式。サポートされている形式は、Translation Memory eXchange (TMX)、カンマ区切り値 (CSV)、タブ区切り値 (TSV) です。

5. [Encryption key (暗号化キー)] で、AWS KMS key を選択して並列データを保護します。これらの KMS キーは AWS Key Management Service (AWS KMS) によって管理されます。AWS KMS の詳細については、「[AWS Key Management Service デベロッパーガイド](#)」を参照してください。

Use AWS owned key (AWS 所有キーを使用する)

Amazon Translate によって所有され管理されている KMS キーを使用します。これはデフォルトのオプションであり、他の方法を選択しない場合に情報を暗号化するために使用されます。詳細については、AWS Key Management Service デベロッパーガイドの「[AWS 所有キー](#)」を参照してください。

Use key from current account (現在のアカウントのキーを使用する)

AWS アカウントの AWS KMS で管理している KMS キーのうちの 1 つを使用します。このオプションを選択すると、選択可能な KMS キーのリストがメニューに表示されます。詳細については、[AWS Key Management Service デベロッパーガイド](#)の「カスタマーマネージドキー」を参照してください。

Use key from different account (別のアカウントのキーを使用する)

別の AWS アカウントの AWS KMS で管理している KMS キーを使用します。このオプションを選択した場合、KMS キーの Amazon リソースネーム (ARN) を入力するためのフィールドがコンソールに表示されます。

暗号化キーの詳細については、[AWS Key Management Service デベロッパーガイド](#)を参照してください。

6. [Create parallel data (並列データの作成)] を選択します。

コンソールが [Parallel data (並列データ)] ページに戻ります。このページの上部にあるバナーにインポートステータスが表示されます。数分後、並列データリソースが表に表示されます。[Status (ステータス)] 列の値が [Active (アクティブ)] である場合、並列データがバッチ翻訳ジョブで使用できる状態です。

トラブルシューティング用エラーファイル

Amazon Translate で入力ファイルの処理中にエラーや警告が発生した場合は、コンソールからエラーファイルが提供され、これをダウンロードするとエラーメッセージを確認できます。次に示すのは、エラーファイルのコンテンツの例です。

```
{
  "summary": {
    "record_error_count": 1,
    "record_skipped_count": 0
  },
  "messages": [
    {
      "content": "Number 1 TU element",
      "message": "Invalid TMX format. One tu element should contain exactly one tuv element with the source language code: en"
    }
  ]
}
```

並列データの追加 (AWS CLI)

AWS CLI を使用して並列データを追加するには、`create-parallel-data` コマンドを使用します。

Example create-parallel-data コマンド

次の例では、Amazon S3 から TSV ファイルをインポートして並列データオブジェクトを作成します。

```
$ aws translate create-parallel-data \  
> --name my-parallel-data \  
> --parallel-data-config S3Uri=s3://input-bucket/parallel-data-file.tsv,Format=TSV
```

コマンドが成功すると、Amazon Translate から、新しい並列データオブジェクトのステータスでの応答があります。

```
{  
  "Name": "my-parallel-data",  
  "Status": "CREATING"  
}
```

get-parallel-data コマンドを使用して並列データの進行状況を監視することができます。ステータスが ACTIVE である場合、並列データがバッチ翻訳ジョブで使用できる状態です。get-parallel-data コマンドの例については「[並列データオブジェクトの詳細を表示する方法](#)」を参照してください。

並列データの使用

並列データリソースを作成したので、バッチ翻訳ジョブに適用して出力をカスタマイズすることができます。バッチジョブを実行するには、[バッチ翻訳ジョブの実行](#) を参照してください。

Amazon Translate における並列データの表示と管理

Amazon Translate に追加したすべての並列データリソースを表示でき、各リソースの詳細なサマリーにアクセスできます。翻訳要件の変化に伴い、並列データを更新して翻訳出力を絞り込むことができます。

並列データの表示と管理 (Amazon Translate コンソール)

Amazon Translate コンソールで並列データの表示と管理を行うには、並列データページを使用します。

並列データリソースの一覧を表示する方法

1. [Amazon Translate コンソール](#) を開きます。
2. 左側のナビゲーションメニューで、[Customization (カスタマイズ)] を選択し、[Parallel data (並列データ)] を選択します。このページの表には、Amazon Translate に追加した並列データリソースが表示されます。

並列データリソースの詳細を表示する方法

- [Parallel data (並列データ)] ページの [Name (名前)] 列で、並列データリソースの名前を選択します。コンソールで詳細ページを開きます。この詳細ページには、ステータス、最終更新日、ソース言語、ターゲット言語などの情報が含まれています。

並列データリソースを更新する方法

1. 並列データの更新バージョンを、新しい入力ファイルとして Amazon S3 バケットにアップロードします。
2. Amazon Translate コンソールで、[Parallel data (並列データ)] ページに進みます。
3. 更新する並列データを選択し、[Update (更新)] を選択します。コンソールに [Update parallel data (並列データの更新)] ページが表示されます。
4. 以下を指定します。

Description (説明) - オプション

更新された説明。

Parallel data location on S3 (S3 での並列データの場所)

Amazon S3 での更新済み並列データ入力ファイルの場所。Amazon S3 のファイルに移動して場所を設けるには、[Select file (ファイルを選択)] を選択します。

Select parallel data file format (並列データファイル形式の選択)

並列データ入力ファイルの形式。サポートされている形式は、Translation Memory eXchange (TMX)、カンマ区切り値 (CSV)、タブ区切り値 (TSV) です。

5. [Save (保存)] を選択します。Amazon Translate により古い並列データが新しい入力ファイルに置き換えられます。

並列データの表示と管理 (AWS CLI)

AWS CLI を使用して並列データリソースを表示して更新することができます。

並列データリソースの一覧を表示する方法

Amazon Translate に追加した並列データリソースのリストを表示するには、`list-parallel-data` コマンドを使用します。

Example list-parallel-data コマンド

次の例では並列データリソースとそのプロパティの一覧が返されます。

```
$ aws translate list-parallel-data
```

コマンドが成功すると、Amazon Translate から以下のような配列が返されます。

```
{
  "ParallelDataPropertiesList": [
    {
      "Name": "my-parallel-data",
      "Arn": "arn:aws:translate:us-west-2:111122223333:parallel-data/my-parallel-
data",
      "Status": "ACTIVE",
      "SourceLanguageCode": "en",
      "TargetLanguageCodes": [
        "es",
        "ja",
        "zh"
      ],
      "ParallelDataConfig": {
        "S3Uri": "s3://input-bucket/parallel-data-file.tsv",
        "Format": "TSV"
      },
      "ImportedDataSize": 2283,
      "ImportedRecordCount": 3,
      "FailedRecordCount": 0,
      "CreatedAt": 1598597751.406,
      "LastUpdatedAt": 1598597911.675
    }
  ]
}
```

並列データオブジェクトの詳細を表示する方法

単一の並列データリソースの詳細を調べるには、`get-parallel-data` コマンドを使用します。このコマンドにより、並列データのプロパティと、作成に使用された入力ファイルをダウンロードできる署名付き S3 URL が返されます。

Example get-parallel-data コマンド

次の例では、`my-parallel-data` オブジェクトのプロパティとダウンロード先を取得します。

```
$ aws translate get-parallel-data \  
> --name my-parallel-data
```

コマンドが成功すると、Amazon Translate によりプロパティとダウンロード先が返されます。

```
{  
  "ParallelDataProperties": {  
    "Name": "my-parallel-data",  
    "Arn": "arn:aws:translate:us-west-2:111122223333:parallel-data/my-parallel-  
data",  
    "Status": "ACTIVE",  
    "SourceLanguageCode": "en",  
    "TargetLanguageCodes": [  
      "es",  
      "ja",  
      "zh"  
    ],  
    "ParallelDataConfig": {  
      "S3Uri": "s3://input-bucket/parallel-data-file.tsv",  
      "Format": "TSV"  
    },  
    "ImportedDataSize": 2283,  
    "ImportedRecordCount": 3,  
    "FailedRecordCount": 0,  
    "CreatedAt": 1598597751.406,  
    "LastUpdatedAt": 1598597911.675  
  },  
  "DataLocation": {  
    "RepositoryType": "S3",  
    "Location": "pre-signed S3 URL"  
  }  
}
```

並列データリソースを更新する方法

並列データリソースを更新するには、まず、新しい入力ファイルを Amazon S3 入力バケットにアップロードします。次に、`update-parallel-data` コマンドを使用し、更新する並列データリソースを指定します。Amazon Translate により古い並列データが新しい入力ファイルの情報に置き換えられます。

Example update-parallel-data コマンド

次のコマンドにより、Amazon S3 の新しい入力ファイルで my-parallel-data が更新されます。

```
$ aws translate update-parallel-data \  
> --name my-parallel-data \  
> --parallel-data-config S3Uri=s3://input-bucket/parallel-data-file.tsv,Format=TSV
```

コマンドが成功すると、Amazon Translate から以下のようなレスポンスが出ます。

```
{  
  "Name": "my-parallel-data",  
  "Status": "ACTIVE",  
  "LatestUpdateAttemptStatus": "UPDATING",  
  "LatestUpdateAttemptAt": 1598601455.844  
}
```

このレスポンスでは、Status フィールドに既存の並列データオブジェクトのステータスが表示され、LatestUpdateAttemptStatus フィールドに現在の更新試行のステータスが表示されます。

を使用した Amazon Translate のコード例 AWS SDKs

次のコード例は、AWS Software Development Kit () で Amazon Translate を使用する方法を示していますSDK。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

「シナリオ」は、1つのサービス内から、または他のAWSのサービスと組み合わせて複数の関数を呼び出し、特定のタスクを実行する方法を示すコード例です。

開発者ガイドとコード例の完全なリストAWS SDKについては、「」を参照してください[でのこのサービスの使用 AWS SDK](#)。このトピックには、開始方法に関する情報と以前のSDKバージョンに関する詳細も含まれています。

コードの例

- [を使用した Amazon Translate の基本的な例 AWS SDKs](#)
 - [を使用した Amazon Translate のアクション AWS SDKs](#)
 - [DescribeTextTranslationJob で を使用する AWS SDK](#)
 - [ListTextTranslationJobs で を使用する AWS SDK](#)
 - [StartTextTranslationJob で を使用する AWS SDK](#)
 - [StopTextTranslationJob で を使用する AWS SDK](#)
 - [または TranslateTextAWS SDKで を使用する CLI](#)
 - [を使用した Amazon Translate のシナリオ AWS SDKs](#)
 - [Amazon Transcribe ストリーミングアプリケーションを構築する](#)
 - [Amazon Lex chatbot を作成して、ウェブサイトの訪問者に対応する](#)
 - [メッセージを翻訳する公開およびサブスクリプションアプリケーションを構築する](#)
 - [顧客からのフィードバックを分析し、音声を作成するアプリケーションの作成](#)
 - [を使用して Amazon Translate ジョブを開始する AWS SDK](#)

を使用した Amazon Translate の基本的な例 AWS SDKs

次のコード例は、 で AWS Amazon Translate の基本を使用する方法を示していますSDKs。

例

- [を使用した Amazon Translate のアクション AWS SDKs](#)
- [DescribeTextTranslationJob で を使用する AWS SDK](#)
- [ListTextTranslationJobs で を使用する AWS SDK](#)
- [StartTextTranslationJob で を使用する AWS SDK](#)
- [StopTextTranslationJob で を使用する AWS SDK](#)
- [または TranslateTextAWS SDKで を使用する CLI](#)

を使用した Amazon Translate のアクション AWS SDKs

次のコード例は、を使用して個々の Amazon Translate アクションを実行する方法を示しています AWS SDKs。各例にはへのリンクが含まれており GitHub、コードの設定と実行の手順を確認できます。

これらの抜粋は Amazon Translate を呼び出しAPI、コンテキスト内で実行する必要がある大規模なプログラムからのコード抜粋です。アクションは [を使用した Amazon Translate のシナリオ AWS SDKs](#) のコンテキスト内で確認できます。

以下の例には、最も一般的に使用されるアクションのみ含まれています。詳細なリストについては、[Amazon Translate APIリファレンス](#)」を参照してください。

例

- [DescribeTextTranslationJob で を使用する AWS SDK](#)
- [ListTextTranslationJobs で を使用する AWS SDK](#)
- [StartTextTranslationJob で を使用する AWS SDK](#)
- [StopTextTranslationJob で を使用する AWS SDK](#)
- [または TranslateTextAWS SDKで を使用する CLI](#)

DescribeTextTranslationJob で を使用する AWS SDK

以下のコード例は、DescribeTextTranslationJob の使用方法を示しています。

アクション例は、より大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。次のコード例で、このアクションのコンテキストを確認できます。

- [翻訳ジョブを開始する](#)

.NET

AWS SDK for .NET

Note

詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Translate;
using Amazon.Translate.Model;

/// <summary>
/// The following example shows how to retrieve the details of
/// a text translation job using Amazon Translate.
/// </summary>
public class DescribeTextTranslation
{
    public static async Task Main()
    {
        var client = new AmazonTranslateClient();

        // The Job Id is generated when the text translation job is started
        // with a call to the StartTextTranslationJob method.
        var jobId = "1234567890abcdef01234567890abcde";

        var request = new DescribeTextTranslationJobRequest
        {
            JobId = jobId,
        };

        var jobProperties = await DescribeTranslationJobAsync(client,
request);

        DisplayTranslationJobDetails(jobProperties);
    }

    /// <summary>
    /// Retrieve information about an Amazon Translate text translation job.
```

```
    /// </summary>
    /// <param name="client">The initialized Amazon Translate client
    object.</param>
    /// <param name="request">The DescribeTextTranslationJobRequest object.</
    param>
    /// <returns>The TextTranslationJobProperties object containing
    /// information about the text translation job.</returns>
    public static async Task<TextTranslationJobProperties>
    DescribeTranslationJobAsync(
        AmazonTranslateClient client,
        DescribeTextTranslationJobRequest request)
    {
        var response = await client.DescribeTextTranslationJobAsync(request);
        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            return response.TextTranslationJobProperties;
        }
        else
        {
            return null;
        }
    }

    /// <summary>
    /// Displays the properties of the text translation job.
    /// </summary>
    /// <param name="jobProperties">The properties of the text translation
    /// job returned by the call to DescribeTextTranslationJobAsync.</param>
    public static void
    DisplayTranslationJobDetails(TextTranslationJobProperties jobProperties)
    {
        if (jobProperties is null)
        {
            Console.WriteLine("No text translation job properties found.");
            return;
        }

        // Display the details of the text translation job.
        Console.WriteLine($"{jobProperties.JobId}: {jobProperties.JobName}");
    }
}
```

- API 詳細については、「AWS SDK for .NET APIリファレンス [DescribeTextTranslationJob](#)」の「」を参照してください。

SAP ABAP

SDK の SAP ABAP

Note

詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Gets the properties associated with an asynchronous batch translation job."
"Includes properties such as name, ID, status, source and target languages,
and input/output Amazon Simple Storage Service (Amazon S3) buckets."
TRY.
    oo_result = lo_xl8->describetexttranslationjob(      "oo_result is
returned for testing purposes."
    EXPORTING
        iv_jobid      = iv_jobid
    ).
    MESSAGE 'Job description retrieved.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex .
    MESSAGE 'An internal server error occurred. Retry your request.' TYPE
'E'.
CATCH /aws1/cx_xl8resourcenotfoundex .
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex.
    MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
ENDTRY.
```

- API 詳細については、[DescribeTextTranslationJob](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

開発者ガイドとコード例の完全なリスト AWS SDKについては、「」を参照してください[でのこのサービスの使用 AWS SDK](#)。このトピックには、開始方法に関する情報と以前のSDKバージョンに関する詳細も含まれています。

ListTextTranslationJobs で使用する AWS SDK

以下のコード例は、ListTextTranslationJobs の使用方法を示しています。

.NET

AWS SDK for .NET

Note

詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.Translate;
using Amazon.Translate.Model;

/// <summary>
/// List Amazon Translate translation jobs, along with details about each
job.
/// </summary>
public class ListTranslationJobs
{
    public static async Task Main()
    {
        var client = new AmazonTranslateClient();
        var filter = new TextTranslationJobFilter
        {
            JobStatus = "COMPLETED",
        };

        var request = new ListTextTranslationJobsRequest
        {
            MaxResults = 10,
            Filter = filter,
        };
    }
}
```

```
};

    await ListJobsAsync(client, request);
}

/// <summary>
/// List Amazon Translate text translation jobs.
/// </summary>
/// <param name="client">The initialized Amazon Translate client
object.</param>
/// <param name="request">An Amazon Translate
/// ListTextTranslationJobsRequest object detailing which text
/// translation jobs are of interest.</param>
public static async Task ListJobsAsync(
    AmazonTranslateClient client,
    ListTextTranslationJobsRequest request)
{
    ListTextTranslationJobsResponse response;

    do
    {
        response = await client.ListTextTranslationJobsAsync(request);

        ShowTranslationJobDetails(response.TextTranslationJobPropertiesList);

        request.NextToken = response.NextToken;
    }
    while (response.NextToken is not null);
}

/// <summary>
/// List existing translation job details.
/// </summary>
/// <param name="properties">A list of Amazon Translate text
/// translation jobs.</param>
public static void
ShowTranslationJobDetails(List<TextTranslationJobProperties> properties)
{
    properties.ForEach(prop =>
    {
        Console.WriteLine($"{prop.JobId}: {prop.JobName}");
        Console.WriteLine($"Status: {prop.JobStatus}");
        Console.WriteLine($"Submitted time: {prop.SubmittedTime}");
    });
}
```

```
}  
}
```

- API 詳細については、「AWS SDK for .NET APIリファレンス[ListTextTranslationJobs](#)」の「」を参照してください。

SAP ABAP

SDK の SAP ABAP

Note

詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Gets a list of the batch translation jobs that you have submitted."  
  
DATA lo_filter TYPE REF TO /aws1/cl_xl8textxlationjobfilt.  
  
"Create an ABAP object for filtering using jobname."  
CREATE OBJECT lo_filter  
  EXPORTING  
    iv_jobname = iv_jobname.  
  
TRY.  
  oo_result = lo_xl8->listtexttranslationjobs(      "oo_result is returned  
for testing purposes."  
    EXPORTING  
      io_filter      = lo_filter  
    ).  
  MESSAGE 'Jobs retrieved.' TYPE 'I'.  
CATCH /aws1/cx_xl8internalserverex .  
  MESSAGE 'An internal server error occurred. Retry your request.' TYPE  
'E'.  
CATCH /aws1/cx_xl8invalidfilterex .  
  MESSAGE 'The filter specified for the operation is not valid. Specify a  
different filter.' TYPE 'E'.  
CATCH /aws1/cx_xl8invalidrequestex .
```

```
MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex.
MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
ENDTRY.
```

- API 詳細については、[ListTextTranslationJobs](#)「」のAWS SDKSAPABAPI「」を参照してください。

開発者ガイドとコード例の完全なリスト AWS SDKについては、「」を参照してください[でのこのサービスの使用 AWS SDK](#)。このトピックには、開始方法に関する情報と以前のSDKバージョンに関する詳細も含まれています。

StartTextTranslationJob で使用する AWS SDK

以下のコード例は、StartTextTranslationJob の使用方法を示しています。

アクション例は、より大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。次のコード例で、このアクションのコンテキストを確認できます。

- [翻訳ジョブを開始する](#)

.NET

AWS SDK for .NET

Note

詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.Translate;
using Amazon.Translate.Model;

///  
/// <summary>
```

```
/// This example shows how to use Amazon Translate to process the files in
/// an Amazon Simple Storage Service (Amazon S3) bucket. The translated
results
/// will also be stored in an Amazon S3 bucket.
/// </summary>
public class BatchTranslate
{
    public static async Task Main()
    {
        var contentType = "text/plain";

        // Set this variable to an S3 bucket location with a folder."
        // Input files must be in a folder and not at the bucket root."
        var s3InputUri = "s3://amzn-s3-demo-bucket1/FOLDER/";
        var s3OutputUri = "s3://amzn-s3-demo-bucket2/";

        // This role must have permissions to read the source bucket and to
read and
        // write to the destination bucket where the translated text will be
stored.
        var dataAccessRoleArn = "arn:aws:iam::0123456789ab:role/
S3TranslateRole";

        var client = new AmazonTranslateClient();

        var inputConfig = new InputDataConfig
        {
            ContentType = contentType,
            S3Uri = s3InputUri,
        };

        var outputConfig = new OutputDataConfig
        {
            S3Uri = s3OutputUri,
        };

        var request = new StartTextTranslationJobRequest
        {
            JobName = "ExampleTranslationJob",
            DataAccessRoleArn = dataAccessRoleArn,
            InputDataConfig = inputConfig,
            OutputDataConfig = outputConfig,
            SourceLanguageCode = "en",
            TargetLanguageCodes = new List<string> { "fr" },
        };
    }
}
```

```
};

var response = await StartTextTranslationAsync(client, request);

if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
{
    Console.WriteLine($"{response.JobId}: {response.JobStatus}");
}


}

/// <summary>
/// Start the Amazon Translate text translation job.
/// </summary>
/// <param name="client">The initialized AmazonTranslateClient object.</
param>
/// <param name="request">The request object that includes details such
/// as source and destination bucket names and the IAM Role that will
/// be used to access the buckets.</param>
/// <returns>The StartTextTranslationResponse object that includes the
/// details of the request response.</returns>
public static async Task<StartTextTranslationJobResponse>
StartTextTranslationAsync(AmazonTranslateClient client,
StartTextTranslationJobRequest request)
{
    var response = await client.StartTextTranslationJobAsync(request);
    return response;
}
}
```

- API 詳細については、「AWS SDK for .NET APIリファレンス[StartTextTranslationJob](#)」の「」を参照してください。

SAP ABAP

SDK の SAP ABAP

 Note

詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Starts an asynchronous batch translation job."
"Use batch translation jobs to translate large volumes of text across
multiple documents at once."

DATA lo_inputdataconfig TYPE REF TO /aws1/cl_xl8inputdataconfig.
DATA lo_outputdataconfig TYPE REF TO /aws1/cl_xl8outputdataconfig.
DATA lt_targetlanguagecodes TYPE /aws1/
cl_xl8tgtlanguagecodes00=>tt_targetlanguagecodestrlist.
DATA lo_targetlanguagecodes TYPE REF TO /aws1/cl_xl8tgtlanguagecodes00.

"Create an ABAP object for the input data config."
CREATE OBJECT lo_inputdataconfig
  EXPORTING
    iv_s3uri          = iv_input_data_s3uri
    iv_contenttype    = iv_input_data_contenttype.

"Create an ABAP object for the output data config."
CREATE OBJECT lo_outputdataconfig
  EXPORTING
    iv_s3uri = iv_output_data_s3uri.

"Create an internal table for target languages."
CREATE OBJECT lo_targetlanguagecodes
  EXPORTING
    iv_value = iv_targetlanguagecode.
INSERT lo_targetlanguagecodes INTO TABLE lt_targetlanguagecodes.

TRY.
  oo_result = lo_xl8->starttexttranslationjob(      "oo_result is returned
for testing purposes."
  EXPORTING
    io_inputdataconfig = lo_inputdataconfig
```

```
        io_outputdataconfig = lo_outputdataconfig
        it_targetlanguagecodes = lt_targetlanguagecodes
        iv_dataaccessrolelearn = iv_dataaccessrolelearn
        iv_jobname = iv_jobname
        iv_sourcelanguagecode = iv_sourcelanguagecode
    ).
    MESSAGE 'Translation job started.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex .
    MESSAGE 'An internal server error occurred. Retry your request.' TYPE
'E'.
CATCH /aws1/cx_xl8invparamvalueex .
    MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8invalidrequestex.
    MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8resourcenotfoundex .
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex.
    MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
CATCH /aws1/cx_xl8unsuppdedlanguage00 .
    MESSAGE 'Amazon Translate does not support translation from the language
of the source text into the requested target language.' TYPE 'E'.
ENDTRY.
```

- API 詳細については、[StartTextTranslationJob](#) 「」のAWS SDKSAPABAPI 「」を参照してください。

開発者ガイドとコード例の完全なリスト AWS SDKについては、「」を参照してください[でのこのサービスの使用 AWS SDK](#)。このトピックには、開始方法に関する情報と以前のSDKバージョンに関する詳細も含まれています。

StopTextTranslationJob で を使用する AWS SDK

以下のコード例は、StopTextTranslationJob の使用方法を示しています。

.NET

AWS SDK for .NET

Note

詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Translate;
using Amazon.Translate.Model;

/// <summary>
/// Shows how to stop a running Amazon Translation Service text translation
/// job.
/// </summary>
public class StopTextTranslationJob
{
    public static async Task Main()
    {
        var client = new AmazonTranslateClient();
        var jobId = "1234567890abcdef01234567890abcde";

        var request = new StopTextTranslationJobRequest
        {
            JobId = jobId,
        };

        await StopTranslationJobAsync(client, request);
    }

    /// <summary>
    /// Sends a request to stop a text translation job.
    /// </summary>
    /// <param name="client">Initialized AmazonTrnslateClient object.</param>
    /// <param name="request">The request object to be passed to the
    /// StopTextJobAsync method.</param>
    public static async Task StopTranslationJobAsync(
        AmazonTranslateClient client,
```

```

        StopTextTranslationJobRequest request)
    {
        var response = await client.StopTextTranslationJobAsync(request);
        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"{response.JobId} as status:
{response.JobStatus}");
        }
    }
}

```

- API 詳細については、「AWS SDK for .NET APIリファレンス[StopTextTranslationJob](#)」の「」を参照してください。

SAP ABAP

SDK の SAP ABAP

Note

詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

"Stops an asynchronous batch translation job that is in progress."

```

TRY.
    oo_result = lo_xl8->stoptexttranslationjob(      "oo_result is returned
for testing purposes."
    EXPORTING
        iv_jobid      = iv_jobid
    ).
    MESSAGE 'Translation job stopped.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex .
    MESSAGE 'An internal server error occurred.' TYPE 'E'.
CATCH /aws1/cx_xl8resourcenotfoundex .
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex.

```

```
MESSAGE 'You have made too many requests within a short period of time.'  
TYPE 'E'.  
ENDTRY.
```

- API 詳細については、[StopTextTranslationJob](#)「」のAWS SDKSAPABAPI「」を参照してください。

開発者ガイドとコード例の完全なリスト AWS SDKについては、「」を参照してください[でのこのサービスの使用 AWS SDK](#)。このトピックには、開始方法に関する情報と以前のSDKバージョンに関する詳細も含まれています。

または TranslateTextAWS SDKで を使用する CLI

以下のコード例は、TranslateText の使用方法を示しています。

.NET

AWS SDK for .NET

Note

詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
using System;  
using System.IO;  
using System.Threading.Tasks;  
using Amazon.S3;  
using Amazon.S3.Transfer;  
using Amazon.Translate;  
using Amazon.Translate.Model;  
  
/// <summary>  
/// Take text from a file stored a Amazon Simple Storage Service (Amazon S3)  
/// object and translate it using the Amazon Transfer Service.  
/// </summary>  
public class TranslateText  
{  
    public static async Task Main()  
    {  
        // ...  
    }  
}
```

```
{
    // If the region you want to use is different from the region
    // defined for the default user, supply it as a parameter to the
    // Amazon Translate client object constructor.
    var client = new AmazonTranslateClient();

    // Set the source language to "auto" to request Amazon Translate to
    // automatically detect the language of the source text.

    // You can get a list of the languages supposed by Amazon Translate
    // in the Amazon Translate Developer's Guide here:
    //     https://docs.aws.amazon.com/translate/latest/dg/what-is.html
    string srcLang = "en"; // English.
    string destLang = "fr"; // French.

    // The Amazon Simple Storage Service (Amazon S3) bucket where the
    // source text file is stored.
    string srcBucket = "amzn-s3-demo-bucket";
    string srcTextFile = "source.txt";

    var srcText = await GetSourceTextAsync(srcBucket, srcTextFile);
    var destText = await TranslatingTextAsync(client, srcLang, destLang,
srcText);

    ShowText(srcText, destText);
}

/// <summary>
/// Use the Amazon S3 TransferUtility to retrieve the text to translate
/// from an object in an S3 bucket.
/// </summary>
/// <param name="srcBucket">The name of the S3 bucket where the
/// text is stored.
/// </param>
/// <param name="srcTextFile">The key of the S3 object that
/// contains the text to translate.</param>
/// <returns>A string representing the source text.</returns>
public static async Task<string> GetSourceTextAsync(string srcBucket,
string srcTextFile)
{
    string srcText = string.Empty;

    var s3Client = new AmazonS3Client();
    TransferUtility utility = new TransferUtility(s3Client);
```

```
        using var stream = await utility.OpenStreamAsync(srcBucket,
srcTextFile);

        StreamReader file = new System.IO.StreamReader(stream);

        srcText = file.ReadToEnd();
        return srcText;
    }

    /// <summary>
    /// Use the Amazon Translate Service to translate the document from the
    /// source language to the specified destination language.
    /// </summary>
    /// <param name="client">The Amazon Translate Service client used to
    /// perform the translation.</param>
    /// <param name="srcLang">The language of the source text.</param>
    /// <param name="destLang">The destination language for the translated
    /// text.</param>
    /// <param name="text">A string representing the text to ranslate.</
param>
    /// <returns>The text that has been translated to the destination
    /// language.</returns>
    public static async Task<string>
TranslatingTextAsync(AmazonTranslateClient client, string srcLang, string
destLang, string text)
    {
        var request = new TranslateTextRequest
        {
            SourceLanguageCode = srcLang,
            TargetLanguageCode = destLang,
            Text = text,
        };

        var response = await client.TranslateTextAsync(request);

        return response.TranslatedText;
    }

    /// <summary>
    /// Show the original text followed by the translated text.
    /// </summary>
    /// <param name="srcText">The original text to be translated.</param>
    /// <param name="destText">The translated text.</param>
```

```
public static void ShowText(string srcText, string destText)
{
    Console.WriteLine("Source text:");
    Console.WriteLine(srcText);
    Console.WriteLine();
    Console.WriteLine("Translated text:");
    Console.WriteLine(destText);
}
}
```

- API 詳細については、「AWS SDK for .NET APIリファレンス[TranslateText](#)」の「」を参照してください。

PowerShell

のツール PowerShell

例 1: 指定された英語テキストをフランス語に変換します。変換するテキストは、-Text パラメータとして渡すこともできます。

```
"Hello World" | ConvertTo-TRNTargetLanguage -SourceLanguageCode en -
TargetLanguageCode fr
```

- API 詳細については、「コマンドレットリファレンス[TranslateText](#)」の「」を参照してください。AWS Tools for PowerShell

SAP ABAP

SDK の SAP ABAP

Note

詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
"Translates input text from the source language to the target language."
```

```
TRY.
    oo_result = lo_xl8->translatetext(      "oo_result is returned for
testing purposes."
    EXPORTING
        iv_text          = iv_text
        iv_sourcelanguagecode = iv_sourcelanguagecode
        iv_targetlanguagecode = iv_targetlanguagecode
    ).
    MESSAGE 'Translation completed.' TYPE 'I'.
CATCH /aws1/cx_xl8detectedlanguage00 .
    MESSAGE 'The confidence that Amazon Comprehend accurately detected the
source language is low.' TYPE 'E'.
CATCH /aws1/cx_xl8internalserverex .
    MESSAGE 'An internal server error occurred.' TYPE 'E'.
CATCH /aws1/cx_xl8invalidrequestex .
    MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8resourcenotfoundex .
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8serviceunavailex .
    MESSAGE 'The Amazon Translate service is temporarily unavailable.' TYPE
'E'.
CATCH /aws1/cx_xl8textsizefmtexcdex .
    MESSAGE 'The size of the text you submitted exceeds the size limit. '
TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex .
    MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
CATCH /aws1/cx_xl8unsuppdedlanguage00 .
    MESSAGE 'Amazon Translate does not support translation from the language
of the source text into the requested target language. ' TYPE 'E'.
ENDTRY.
```

- API 詳細については、[TranslateText](#)「」のAWS SDKSAPABAPAPI「」を参照してください。

開発者ガイドとコード例の完全なリスト AWS SDKについては、「」を参照してください[でのこのサービスの使用 AWS SDK](#)。このトピックには、開始方法に関する情報と以前のSDKバージョンに関する詳細も含まれています。

を使用した Amazon Translate のシナリオ AWS SDKs

次のコード例は、を使用して Amazon Translate で一般的なシナリオを実装する方法を示しています AWS SDKs。これらのシナリオでは、Amazon Translate 内で複数の関数を呼び出すか、他の関数と組み合わせて特定のタスクを実行する方法を示します AWS のサービス。各シナリオには、完全なソースコードへのリンクが含まれており、そこからコードの設定方法と実行方法に関する手順を確認できます。

シナリオは、サービスアクションをコンテキストで理解するのに役立つ中級レベルの経験を対象としています。

例

- [Amazon Transcribe ストリーミングアプリケーションを構築する](#)
- [Amazon Lex chatbot を作成して、ウェブサイトの訪問者に対応する](#)
- [メッセージを翻訳する公開およびサブスクリプションアプリケーションを構築する](#)
- [顧客からのフィードバックを分析し、音声を作成するアプリケーションの作成](#)
- [を使用して Amazon Translate ジョブを開始する AWS SDK](#)

Amazon Transcribe ストリーミングアプリケーションを構築する

次のコード例は、ライブ音声をリアルタイムで記録、転写、翻訳し、結果を E メールで送信するアプリケーションを構築する方法を示しています。

JavaScript

SDK for JavaScript (v3)

Amazon Transcribe を使用して、ライブ音声をリアルタイムで記録、文字起こし、翻訳し、Amazon Simple Email Service (Amazon) を使用して結果を E メールで送信するアプリを構築する方法を示します SES。

完全なソースコードと、セットアップと実行の手順については、「」の詳細な例を参照してください [GitHub](#)。

この例で使用されているサービス

- Amazon Comprehend
- Amazon SES

- Amazon Transcribe
- Amazon Translate

開発者ガイドとコード例の完全なリスト AWS SDKについては、「」を参照してください[でのこのサービスの使用 AWS SDK](#)。このトピックには、開始方法に関する情報と以前のSDKバージョンに関する詳細も含まれています。

Amazon Lex chatbot を作成して、ウェブサイトの訪問者に対応する

次のコード例は、Web サイトの訪問者を引き付けるチャットボットの作成方法を示しています。

Java

SDK for Java 2.x

Amazon Lex を使用してウェブアプリケーション内に Chatbot API を作成し、ウェブサイトの訪問者を引き付ける方法を示します。

完全なソースコードと、セットアップと実行の手順については、「」の詳細な例を参照してください[GitHub](#)。

この例で使用されているサービス

- Amazon Comprehend
- Amazon Lex
- Amazon Translate

JavaScript

SDK for JavaScript (v3)

Amazon Lex を使用してウェブアプリケーション内に Chatbot API を作成し、ウェブサイトの訪問者を引き付ける方法を示します。

完全なソースコードとセットアップと実行の手順については、デ AWS SDK for JavaScript ベロッパーガイドの[Amazon Lex チャットボットの構築](#)」の完全な例を参照してください。

この例で使用されているサービス

- Amazon Comprehend

- Amazon Lex
- Amazon Translate

開発者ガイドとコード例の完全なリスト AWS SDKについては、「」を参照してください[でのこのサービスの使用 AWS SDK](#)。このトピックには、開始方法に関する情報と以前のSDKバージョンに関する詳細も含まれています。

メッセージを翻訳する公開およびサブスクリプションアプリケーションを構築する

次のコード例は、サブスクリプションと発行機能を持ち、メッセージを翻訳するアプリケーションを作成する方法を示しています。

.NET

AWS SDK for .NET

Amazon Simple Notification Service NETを使用して、サブスクリプションと公開機能を持つウェブアプリケーションAPIを作成する方法を示します。さらに、このサンプルアプリケーションではメッセージを翻訳します。

完全なソースコードと、セットアップと実行の手順については、「」の詳細な例を参照してください[GitHub](#)。

この例で使用されているサービス

- Amazon SNS
- Amazon Translate

Java

SDK for Java 2.x

Amazon Simple Notification Service Java を使用して、サブスクリプションと公開機能を持つウェブアプリケーションAPIを作成する方法を示します。さらに、このサンプルアプリケーションではメッセージを翻訳します。

完全なソースコードと、セットアップと実行の手順については、「」の詳細な例を参照してください[GitHub](#)。

完全なソースコードと、Java 非同期 を使用する例をセットアップして実行する方法についてはAPI、「」の詳細な例を参照してください[GitHub](#)。

この例で使用されているサービス

- Amazon SNS
- Amazon Translate

Kotlin

SDK Kotlin 用の

Amazon SNS Kotlin を使用して、サブスクリプションと公開機能を持つアプリケーションAPIを作成する方法を示します。さらに、このサンプルアプリケーションではメッセージを翻訳します。

完全なソースコードとウェブアプリケーションの作成方法については、「」の詳細な例を参照してください[GitHub](#)。

完全なソースコードとネイティブ Android アプリの作成方法については、 の完全な例を参照してください[GitHub](#)。

この例で使用されているサービス

- Amazon SNS
- Amazon Translate

開発者ガイドとコード例の完全なリスト AWS SDKについては、「」を参照してください[でのこのサービスの使用 AWS SDK](#)。このトピックには、開始方法に関する情報と以前のSDKバージョンに関する詳細も含まれています。

顧客からのフィードバックを分析し、音声を合成するアプリケーションの作成

次のコード例は、顧客のコメントカードを分析し、それを元の言語から翻訳し、顧客の感情を判断し、翻訳されたテキストから音声ファイルを生成するアプリケーションの作成方法を示しています。

.NET

AWS SDK for .NET

このサンプルアプリケーションは、顧客フィードバックカードを分析し、保存します。具体的には、ニューヨーク市の架空のホテルのニーズを満たします。このホテルでは、お客様からのフィードバックをさまざまな言語で書かれた実際のコメントカードの形で受け取ります。そのフィードバックは、ウェブクライアントを通じてアプリにアップロードされます。コメントカードの画像をアップロードされると、次の手順が発生します。

- テキストは Amazon Textract を使用して、画像から抽出されます。
- Amazon Comprehend は、抽出されたテキストの感情とその言語を決定します。
- 抽出されたテキストは、Amazon Translate を使用して英語に翻訳されます。
- Amazon Polly は抽出されたテキストからオーディオファイルを合成します。

完全なアプリは AWS CDK を使用してデプロイすることができます。ソースコードとデプロイ手順については、「」のプロジェクトを参照してください [GitHub](#)。

この例で使用されているサービス

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Java

SDK for Java 2.x

このサンプルアプリケーションは、顧客フィードバックカードを分析し、保存します。具体的には、ニューヨーク市の架空のホテルのニーズを満たします。このホテルでは、お客様からのフィードバックをさまざまな言語で書かれた実際のコメントカードの形で受け取ります。そのフィードバックは、ウェブクライアントを通じてアプリにアップロードされます。コメントカードの画像をアップロードされると、次の手順が発生します。

- テキストは Amazon Textract を使用して、画像から抽出されます。
- Amazon Comprehend は、抽出されたテキストの感情とその言語を決定します。
- 抽出されたテキストは、Amazon Translate を使用して英語に翻訳されます。

- Amazon Polly は抽出されたテキストからオーディオファイルを合成します。

完全なアプリは AWS CDK を使用してデプロイすることができます。ソースコードとデプロイ手順については、「」のプロジェクトを参照してください [GitHub](#)。

この例で使用されているサービス

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

JavaScript

SDK for JavaScript (v3)

このサンプルアプリケーションは、顧客フィードバックカードを分析し、保存します。具体的には、ニューヨーク市の架空のホテルのニーズを満たします。このホテルでは、お客様からのフィードバックをさまざまな言語で書かれた実際のコメントカードの形で受け取ります。そのフィードバックは、ウェブクライアントを通じてアプリにアップロードされます。コメントカードの画像をアップロードされると、次の手順が発生します。

- テキストは Amazon Textract を使用して、画像から抽出されます。
- Amazon Comprehend は、抽出されたテキストの感情とその言語を決定します。
- 抽出されたテキストは、Amazon Translate を使用して英語に翻訳されます。
- Amazon Polly は抽出されたテキストからオーディオファイルを合成します。

完全なアプリは AWS CDK を使用してデプロイすることができます。ソースコードとデプロイ手順については、「」のプロジェクトを参照してください [GitHub](#)。次の抜粋は、AWS SDK for JavaScript が Lambda 関数内でどのように使用されるかを示しています。

```
import {
  ComprehendClient,
  DetectDominantLanguageCommand,
  DetectSentimentCommand,
} from "@aws-sdk/client-comprehend";

/**
```

```
* Determine the language and sentiment of the extracted text.
*
* @param {{ source_text: string}} extractTextOutput
*/
export const handler = async (extractTextOutput) => {
  const comprehendClient = new ComprehendClient({});

  const detectDominantLanguageCommand = new DetectDominantLanguageCommand({
    Text: extractTextOutput.source_text,
  });

  // The source language is required for sentiment analysis and
  // translation in the next step.
  const { Languages } = await comprehendClient.send(
    detectDominantLanguageCommand,
  );

  const languageCode = Languages[0].LanguageCode;

  const detectSentimentCommand = new DetectSentimentCommand({
    Text: extractTextOutput.source_text,
    LanguageCode: languageCode,
  });

  const { Sentiment } = await comprehendClient.send(detectSentimentCommand);

  return {
    sentiment: Sentiment,
    language_code: languageCode,
  };
};
```

```
import {
  DetectDocumentTextCommand,
  TextractClient,
} from "@aws-sdk/client-textract";

/**
 * Fetch the S3 object from the event and analyze it using Amazon Textract.
 *
 * @param {import("@types/aws-lambda").EventBridgeEvent<"Object Created">}
  eventBridgeS3Event
 */
```

```
export const handler = async (eventBridgeS3Event) => {
  const textractClient = new TextractClient();

  const detectDocumentTextCommand = new DetectDocumentTextCommand({
    Document: {
      S3Object: {
        Bucket: eventBridgeS3Event.bucket,
        Name: eventBridgeS3Event.object,
      },
    },
  });

  // Textract returns a list of blocks. A block can be a line, a page, word, etc.
  // Each block also contains geometry of the detected text.
  // For more information on the Block type, see https://docs.aws.amazon.com/textract/latest/dg/API\_Block.html.
  const { Blocks } = await textractClient.send(detectDocumentTextCommand);

  // For the purpose of this example, we are only interested in words.
  const extractedWords = Blocks.filter((b) => b.BlockType === "WORD").map(
    (b) => b.Text,
  );

  return extractedWords.join(" ");
};
```

```
import { PollyClient, SynthesizeSpeechCommand } from "@aws-sdk/client-polly";
import { S3Client } from "@aws-sdk/client-s3";
import { Upload } from "@aws-sdk/lib-storage";

/**
 * Synthesize an audio file from text.
 *
 * @param {{ bucket: string, translated_text: string, object: string }}
  sourceDestinationConfig
 */
export const handler = async (sourceDestinationConfig) => {
  const pollyClient = new PollyClient({});

  const synthesizeSpeechCommand = new SynthesizeSpeechCommand({
    Engine: "neural",
    Text: sourceDestinationConfig.translated_text,
    VoiceId: "Ruth",
```

```
    OutputFormat: "mp3",
  });

  const { AudioStream } = await pollyClient.send(synthesizeSpeechCommand);

  const audioKey = `${sourceDestinationConfig.object}.mp3`;

  // Store the audio file in S3.
  const s3Client = new S3Client();
  const upload = new Upload({
    client: s3Client,
    params: {
      Bucket: sourceDestinationConfig.bucket,
      Key: audioKey,
      Body: AudioStream,
      ContentType: "audio/mp3",
    },
  });

  await upload.done();
  return audioKey;
};
```

```
import {
  TranslateClient,
  TranslateTextCommand,
} from "@aws-sdk/client-translate";

/**
 * Translate the extracted text to English.
 *
 * @param {{ extracted_text: string, source_language_code: string }}
  textAndSourceLanguage
 */
export const handler = async (textAndSourceLanguage) => {
  const translateClient = new TranslateClient({});

  const translateCommand = new TranslateTextCommand({
    SourceLanguageCode: textAndSourceLanguage.source_language_code,
    TargetLanguageCode: "en",
    Text: textAndSourceLanguage.extracted_text,
  });
```



```
const { TranslatedText } = await translateClient.send(translateCommand);

return { translated_text: TranslatedText };
};
```

この例で使用されているサービス

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

Ruby

SDK Ruby 用の

このサンプルアプリケーションは、顧客フィードバックカードを分析し、保存します。具体的には、ニューヨーク市の架空のホテルのニーズを満たします。このホテルでは、お客様からのフィードバックをさまざまな言語で書かれた実際のコメントカードの形で受け取ります。そのフィードバックは、ウェブクライアントを通じてアプリにアップロードされます。コメントカードの画像をアップロードされると、次の手順が発生します。

- テキストは Amazon Textract を使用して、画像から抽出されます。
- Amazon Comprehend は、抽出されたテキストの感情とその言語を決定します。
- 抽出されたテキストは、Amazon Translate を使用して英語に翻訳されます。
- Amazon Polly は抽出されたテキストからオーディオファイルを合成します。

完全なアプリは AWS CDK を使用してデプロイすることができます。ソースコードとデプロイ手順については、「」のプロジェクトを参照してください [GitHub](#)。

この例で使用されているサービス

- Amazon Comprehend
- Lambda
- Amazon Polly
- Amazon Textract
- Amazon Translate

開発者ガイドとコード例の完全なリスト AWS SDKについては、「」を参照してください[でのこのサービスの使用 AWS SDK](#)。このトピックには、開始方法に関する情報と以前のSDKバージョンに関する詳細も含まれています。

を使用して Amazon Translate ジョブを開始する AWS SDK

次のコードサンプルは、以下の操作方法を示しています。

- 非同期バッチ翻訳ジョブを開始します。
- 非同期ジョブが完了するまで待ちます。
- 非同期ジョブを記述します。

SAP ABAP

SDK の SAP ABAP

Note

詳細については、「」を参照してください GitHub。用例一覧を検索し、[AWS コード例リポジトリ](#)での設定と実行の方法を確認してください。

```
DATA lo_inputdataconfig TYPE REF TO /aws1/cl_xl8inputdataconfig.
DATA lo_outputdataconfig TYPE REF TO /aws1/cl_xl8outputdataconfig.
DATA lt_targetlanguagecodes TYPE /aws1/
cl_xl8tgtlanguagecodes00=>tt_targetlanguagecodestrlist.
DATA lo_targetlanguagecodes TYPE REF TO /aws1/cl_xl8tgtlanguagecodes00.

"Create an ABAP object for the input data config."
CREATE OBJECT lo_inputdataconfig
  EXPORTING
    iv_s3uri      = iv_input_data_s3uri
    iv_contenttype = iv_input_data_contenttype.

"Create an ABAP object for the output data config."
CREATE OBJECT lo_outputdataconfig
  EXPORTING
    iv_s3uri = iv_output_data_s3uri.
```

```

"Create an internal table for target languages."
CREATE OBJECT lo_targetlanguagecodes
  EXPORTING
    iv_value = iv_targetlanguagecode.
INSERT lo_targetlanguagecodes INTO TABLE lt_targetlanguagecodes.

TRY.
  DATA(lo_translationjob_result) = lo_xl8->starttexttranslationjob(
    EXPORTING
      io_inputdataconfig = lo_inputdataconfig
      io_outputdataconfig = lo_outputdataconfig
      it_targetlanguagecodes = lt_targetlanguagecodes
      iv_dataaccessrolelearn = iv_dataaccessrolelearn
      iv_jobname = iv_jobname
      iv_sourcelanguagecode = iv_sourcelanguagecode
    ).
  MESSAGE 'Translation job started.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex .
  MESSAGE 'An internal server error occurred. Retry your request.' TYPE
'E'.
CATCH /aws1/cx_xl8invparamvalueex .
  MESSAGE 'The value of the parameter is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8invalidrequestex.
  MESSAGE 'The request that you made is not valid.' TYPE 'E'.
CATCH /aws1/cx_xl8resourcenotfoundex .
  MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex.
  MESSAGE 'You have made too many requests within a short period of time. '
TYPE 'E'.
CATCH /aws1/cx_xl8unsuppdedlanguage00 .
  MESSAGE 'Amazon Translate does not support translation from the language
of the source text into the requested target language.' TYPE 'E'.
ENDTRY.

"Get the job ID."
DATA(lv_jobid) = lo_translationjob_result->get_jobid( ).

"Wait for translate job to complete."
DATA(lo_des_translation_result) = lo_xl8-
>describetexttranslationjob( iv_jobid = lv_jobid ).
  WHILE lo_des_translation_result->get_textxlationjobproperties( )-
>get_jobstatus( ) <> 'COMPLETED'.
    IF sy-index = 30.
      EXIT.
      "Maximum 900 seconds."

```

```
ENDIF.
WAIT UP TO 30 SECONDS.
lo_des_translation_result = lo_xl8->describetexttranslationjob( iv_jobid =
lv_jobid ).
ENDWHILE.

TRY.
    oo_result = lo_xl8->describetexttranslationjob(      "oo_result is
returned for testing purposes."
    EXPORTING
        iv_jobid      = lv_jobid
    ).
    MESSAGE 'Job description retrieved.' TYPE 'I'.
CATCH /aws1/cx_xl8internalserverex .
    MESSAGE 'An internal server error occurred. Retry your request.' TYPE
'E'.
CATCH /aws1/cx_xl8resourcenotfoundex .
    MESSAGE 'The resource you are looking for has not been found.' TYPE 'E'.
CATCH /aws1/cx_xl8toomanyrequestsex.
    MESSAGE 'You have made too many requests within a short period of time.'
TYPE 'E'.
ENDTRY.
```

- API 詳細については、の以下のトピックを参照してくださいAWS SDKSAPABAPI。
- [DescribeTextTranslationJob](#)
- [StartTextTranslationJob](#)

開発者ガイドとコード例の完全なリスト AWS SDKについては、「」を参照してください[でのこのサービスの使用 AWS SDK](#)。このトピックには、開始方法に関する情報と以前のSDKバージョンに関する詳細も含まれています。

リソースのタグ付け

タグは、Amazon Translate リソースに関連付けることができるメタデータです。タグは、キーと値のペアから構成されます。タグは [並列データ] リソースと [カスタム用語] リソースに追加できません。

タグには主に 2 つの機能があります。1 つ目はリソースを整理することと、2 つ目はタグベースのアクセス制御を行うことです。リソースにタグを追加し、そのタグに基づいてリソースへのアクセスを許可または制限する IAM ポリシーを作成できます。

ポリシーでは、リクエストで指定されたタグ (request-tags) または呼び出すリソースに関連するタグ (resource-tags) に基づいてオペレーションを許可または禁止できます。IAM でタグを使用する方法については、「IAM ユーザーガイド」の「[タグを使用してアクセスを制御する](#)」を参照してください。

Amazon Translate でタグを使用する際の考慮事項:

- リソースごとに最大 50 個のユーザータグを追加できます。
- リソースの作成時または作成後いつでもタグを追加できます。
- タグキーは必須フィールドですが、タグ値はオプションです。
- タグはリソース間で一意である必要はありませんが、特定のリソースのタグには一意のキーが必要です。
- タグのキーと値は大文字と小文字が区別されます。
- タグキーは最大 128 文字、タグ値は最大 256 文字です。
- AWS システムタグはタグキーまたは値のプレフィックス `aws:` で始まります。このプレフィックスが含まれるタグの名前や値は、編集または削除できません。システムタグは、リソースごとのタグクォータには含まれません。

Note

複数の AWS サービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスで許可される文字の制限が異なることがあるのでご注意ください。

トピック

- [新しいリソースへのタグ付け](#)

- [リソースに関連付けられたタグの表示、更新、削除](#)

新しいリソースへのタグ付け

ParallelData またはカスタム用語リソースを作成するときに、タグを追加できます。

新しいリソースにタグを追加する (コンソール)

1. [Amazon Translate コンソール](#) にサインインします。
2. 左のナビゲーションペインから、作成するリソース (Parallel data または Custom terminology) を選択します。
3. [並列データの作成] または [用語の作成] を選択します。コンソールに、リソース作成のメインページが表示されます。このページの最後に [タグ - オプション] パネルがあります。

Tags - optional

A tag is a metadata label you can add to a resource to help you organize, search, or filter your data. Each tag consists of a key and an option value, in the form 'key:value'.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

4. [新しいタグを追加] を選択してリソースのタグを追加します。タグキーおよびタグ値 (オプション) を入力します。

Tags - optional

A tag is a metadata label you can add to a resource to help you organize, search, or filter your data. Each tag consists of a key and an option value, in the form 'key:value'.

Key: X

Value - optional: X

Remove

Add new tag

You can add up to 49 more tags.

5. ステップ 4 を繰り返して、すべてのタグを追加します。各キーは、このリソースに一意である必要があります。

Tags - optional

A tag is a metadata label you can add to a resource to help you organize, search, or filter your data. Each tag consists of a key and an option value, in the form 'key:value'.

Key	Value - optional	
🔍 hello ✕	🔍 world ✕	Remove
🔍 Enter key	🔍 Enter value	Remove

⚠️ You must specify a tag key

Add new tag

You can add up to 48 more tags.

6. [並列データの作成] または [用語の作成] を選択してリソースを作成します。

Amazon Translate [CreateParallelData](#) API オペレーションを使用してタグを追加することもできます。次の例は、create-parallel-data CLI コマンドを使用してタグを追加する方法を示しています。

```
aws translate create-parallel-data \
--name "myTest" \
--parallel-data-config "{\"format\": \"CSV\", \
    \"S3Uri\": \"s3://test-input/TEST.csv\"}" \
--tags "[{\"Key\": \"color\", \"Value\": \"orange\"}]"
```

リソースに関連付けられたタグの表示、更新、削除

Parallel data または Custom terminology リソースに関連付けられているタグを表示、更新、削除できます。

既存のリソースのタグを更新する方法 (コンソール)

1. [Amazon Translate コンソール](#) にサインインします。
2. 左のナビゲーションペインで Parallel data または Custom terminology を選択します。
3. 表示、更新、または削除するタグを含むリソースを選択します。コンソールにリソースの詳細ページが表示されます。
4. [タグ] パネルが表示されるまで下にスクロールします。ここでは、選択したリソースに関連付けられているすべてのタグが表示されます。

Tags (2)		Manage tags
Key	▲	Value
color		orange
type		PDF

[タグを管理] を選択して、リソースからタグを編集または削除します。

5. 変更するテキストを選択し、タグを編集します。[削除] を選択してタグを削除することもできます。
6. 新しいタグを追加するには、[新しいタグを追加] を選択し、空白のフィールドにキーと値を入力します。
7. タグの変更が完了したら、[保存] を選択します。

Amazon Translate のセキュリティ

でのクラウドセキュリティが最優先事項 AWS です。AWS のお客様は、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、AWS とお客様の間での責任共有です。[責任共有モデル](#)では、この責任がクラウドのセキュリティおよびクラウド内のセキュリティとして説明されています。

- **クラウドのセキュリティ** — クラウドで AWS AWS サービスを実行するインフラストラクチャを保護する AWS 責任があります。AWS また、では、安全に使用できるサービスも提供しています。[「AWS」コンプライアンスプログラム](#)の一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。Amazon Translate に適用されるコンプライアンスプログラムについては、[「コンプライアンスプログラムによる対象範囲のAWS のサービス」](#)を参照してください。
- **クラウド内のセキュリティ** — お客様の責任は、使用する AWS サービスによって決まります。また、お客様は、お客様のデータの機密性、企業の要件、および適用可能な法律や規制といった他の要因についても責任を担います。

このトピックは、AWSを使用する際に責任共有モデルを適用する方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成する AWS ように を設定する方法について説明します。また、AWS リソースのモニタリングや保護に役立つ他の AWS のサービスの使用方法についても説明します。

トピック

- [Amazon Translate におけるデータ保護](#)
- [Amazon Translate のための Identity and Access Management](#)
- [Amazon Translate のモニタリング](#)
- [Amazon Translate のコンプライアンス検証](#)
- [Amazon Translate の耐障害性](#)
- [Amazon Translate のインフラストラクチャセキュリティ](#)
- [Amazon Translate とインターフェイスVPCエンドポイント \(AWS PrivateLink \)](#)

Amazon Translate におけるデータ保護

Amazon Translate は、AWS [の責任共有モデル](#)に準拠しています。データ保護の規制とガイドラインを含む。AWS は、すべての AWS サービスを実行するグローバルインフラストラクチャを保護する責任を負います。は、このインフラストラクチャでホストされるデータの制御 AWS を維持します。顧客コンテンツと個人データを処理するためのセキュリティ設定コントロールを含めます。AWS の顧客および APN パートナー、データコントローラーまたはデータ処理者として動作し、は、AWS クラウドに保存した個人データについて責任を負います。

データ保護の目的で、AWS アカウント認証情報を保護し、AWS Identity and Access Management (IAM) でロールを設定することをお勧めします。これにより、各ユーザーに各自の職務を果たすために必要なアクセス許可のみが付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。
- API とユーザーアクティビティのログ記録をセットアップします AWS CloudTrail。
- AWS 暗号化ソリューションと、AWS サービス内のすべてのデフォルトのセキュリティコントロールを使用します。
- Amazon Simple Storage Service (Amazon S3) に保存されている個人情報の発見と保護を支援する Amazon Macie などの高度なマネージド・セキュリティ・サービスを利用します。

顧客のアカウント番号などの機密の識別情報は、[名前] フィールドなどの自由形式のフィールドに配置しないことを強くお勧めします。これは、コンソール、またはを使用して Amazon Translate または他の AWS サービスを使用する場合も同様です API AWS CLI AWS SDKs。Amazon Translate や他のサービスに入力したすべてのデータは、診断ログに取り込まれる可能性があります。URL を外部サーバーに提供するときは、そのサーバーへのリクエストを検証 URL するために認証情報を に含めないでください。

データ保護の詳細については、AWS セキュリティブログの [AWS 責任共有モデルと GDPR](#) ブログ投稿を参照してください。

トピック

- [保管中の暗号化](#)
- [転送中の暗号化](#)

保管中の暗号化

Amazon Translate で実行するバッチ翻訳ジョブについては、翻訳の入力と出力の両方が保存時に暗号化されます。ただし、暗号化方法はそれぞれ異なります。

Amazon Translate は、デフォルトキーで暗号化された Amazon Elastic Block Store (Amazon EBS) ボリュームも使用します。

翻訳入力

Amazon Translate を使用してドキュメントをバッチ単位で翻訳する場合、一連の入力ドキュメントを Amazon S3 バケットに保存します。これらのドキュメントを保管中に暗号化するには、Amazon SSE-S3 が提供する -S3 サーバー側の暗号化オプションを使用できます。Amazon S3 このオプションを使用すると、各オブジェクトは、Amazon S3 が管理する一意のキーで暗号化されます。

詳細については、[「Amazon Simple Storage Service ユーザーガイド」の「Amazon S3-managed暗号化キーによるサーバー側の暗号化 \(SSE-S3\) を使用したデータの保護」](#)を参照してください。

翻訳出力

Amazon Translate がバッチ翻訳ジョブを完了すると、出力は AWS アカウントの Amazon S3 バケットに配置されます。保管中の出力を暗号化するために、Amazon Translate は Amazon S3 が提供する SSE-KMS 暗号化オプションを使用します。このオプションでは、出力は AWS Key Management Service () に保存されているキーで暗号化されますAWS KMS。

SSE- の詳細についてはKMS、[「Amazon Simple Storage Service ユーザーガイド」のAWS「Key Management Service \(SSE-KMS\) によるサーバー側の暗号化を使用したデータの保護」](#)を参照してください。

KMS キーの詳細については、「AWS Key Management Service デベロッパーガイド[AWS KMS keys](#)」の「」を参照してください。

この暗号化のために、Amazon Translate では次のいずれかのタイプのキーを使用できます。

AWS マネージドキー

デフォルトでは Amazon Translate では AWS マネージドキー が使用されます。このタイプの KMSキーは、によって作成 AWS され、アカウントに保存されます。ただし、このKMSキーを自分で管理することはできません。これは、ユーザーに代わって AWSによってのみ管理と使用が行われます。

カスタマーマネージドキー

オプションで、AWS アカウントで作成、所有、管理するキーであるカスタマーマネージドKMS キーを使用して出力を暗号化することもできます。

独自のKMSキーを使用する前に、Amazon Translate が Amazon S3 の出力バケットにアクセスするために使用するIAMサービスロールにアクセス許可を追加する必要があります。別の AWS アカウントにあるKMSキーを使用する場合は、キーポリシーも更新する必要があります AWS KMS。詳細については、「[暗号化をカスタマイズするための前提となるアクセス許可](#)」を参照してください。

バッチ翻訳ジョブの実行時におけるカスタマーマネージドキーの使用を選択できます。詳細については、「[バッチ翻訳ジョブの実行](#)」を参照してください。

転送中の暗号化

転送中のデータを暗号化するために、Amazon Translate は証明書で TLS AWS 1.2 を使用します。

Amazon Translate のための Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つです。IAM管理者は、誰を認証(サインイン)し、誰に Amazon Translate リソースの使用を承認する(アクセス許可を付与 AWS のサービス する)かを制御します。IAMは、追加料金なしで使用できる です。

トピック

- [対象者](#)
- [アイデンティティを使用した認証](#)
- [ポリシーを使用したアクセスの管理](#)
- [Amazon Translate と の連携方法 IAM](#)
- [Amazon Translate のアイデンティティベースのポリシーの例](#)
- [AWS Amazon Translate の マネージドポリシー](#)
- [Amazon Translate のアイデンティティとアクセスに関するトラブルシューティング](#)

対象者

AWS Identity and Access Management (IAM) の使用方法は、Amazon Translate で行う作業によって異なります。

サービスユーザー – 業務を行うために Amazon Translate サービスを使用する場合は、管理者から必要な認証情報と許可が提供されます。さらに多くの Amazon Translate 機能を使用して作業を行う場合は、追加のアクセス許可が必要になることがあります。アクセスの管理方法を理解すると、管理者に適切なアクセス許可をリクエストするのに役に立ちます。Amazon Translate の機能にアクセスできない場合は、「[Amazon Translate のアイデンティティとアクセスに関するトラブルシューティング](#)」を参照してください。

サービス管理者 – 社内の Amazon Translate リソースを担当している場合は、通常、Amazon Translate へのフルアクセスがあります。サービスユーザーがアクセスする Amazon Translate 機能やリソースを決めるのは管理者の仕事です。その後で、サービスユーザーのアクセス許可を変更するために、IAM 管理者にリクエストを送信する必要があります。IAM の基本概念については、このページの情報を確認します。会社 IAM で Amazon Translate を使用方法の詳細については、「」を参照してください [Amazon Translate との連携方法 IAM](#)。

IAM 管理者 – IAM 管理者は、Amazon Translate へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。で使用できる Amazon Translate アイデンティティベースのポリシーの例を表示するには IAM、「」を参照してください [Amazon Translate のアイデンティティベースのポリシーの例](#)。

アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。として、IAM ユーザーとして AWS アカウントのルートユーザー、または IAM ロールを引き受けることによって、認証 (にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーテッド ID AWS として にサインインできます。AWS IAM Identity Center (IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報は、フェデレーテッド ID の例です。フェデレーテッドアイデンティティとしてサインインする場合、管理者は以前に IAM ロールを使用して ID フェデレーションを設定しています。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることとなります。

ユーザーのタイプに応じて、AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、「AWS サインイン ユーザーガイド」の「[にサインインする方法 AWS アカウント](#)」を参照してください。

AWS プログラムで にアクセスする場合、 はソフトウェア開発キット (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、 認証情報を使用してリクエストに暗号で署名します。AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。推奨される方法を使用してリクエストに署名する方法の詳細については、「IAM ユーザーガイド」の [AWS API 「リクエストの署名バージョン 4」](#) を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。たとえば、では、アカウントのセキュリティを強化するために多要素認証 (MFA) を使用する AWS ことをお勧めします。詳細については、「AWS IAM Identity Center ユーザーガイド」の [「多要素認証」](#) および「ユーザーガイド」の [AWS 「での多要素認証IAMIAM」](#) を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての およびリソースへの AWS のサービス 完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、IAM ユーザー ガイドの [「ルートユーザー資格情報が必要なタスク」](#) を参照してください。

フェデレーテッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、ID プロバイダーとのフェデレーションを使用して一時的な認証情報 AWS のサービス を使用して にアクセスすることを要求します。

フェデレーテッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、アイデンティティセンターディレクトリのユーザー、または ID ソースを通じて提供された認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーテッド ID がアクセスすると AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成することも、独自の ID ソースのユーザーとグループのセットに接続して同期し、すべての AWS アカウント とアプリケーションで使用できるようにすることもできます。IAM Identity Center の詳細については、AWS IAM Identity Center 「ユーザーガイド」の [IAM 「Identity Center とは」](#) を参照してください。

IAM ユーザーとグループ

[IAM ユーザー](#)は、単一のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウントを持つ内のアイデンティティです。可能な場合は、パスワードやアクセスキーなどの長期的な認証情報を持つIAMユーザーを作成するのではなく、一時的な認証情報を使用することをお勧めします。ただし、IAMユーザーとの長期的な認証情報を必要とする特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、IAMユーザーガイドの「[長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする](#)」を参照してください。

[IAM グループ](#)は、IAM ユーザーのコレクションを指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、という名前のグループを作成しIAMAdmins、そのグループにIAMリソースを管理するアクセス許可を付与できます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時認証情報が提供されます。詳細については、「IAMユーザーガイド」のIAM「[ユーザーのユースケース](#)」を参照してください。

IAM ロール

[IAM ロール](#)は、特定のアクセス許可 AWS アカウントを持つ内のアイデンティティです。これはIAMユーザーに似ていますが、特定のユーザーに関連付けられていません。IAMロールを一時的に引き受けるにはAWS Management Console、[ユーザーからIAMロールに切り替えることができます \(コンソール\)](#)。ロールを引き受けるには、またはAWS API オペレーションをAWS CLI 呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAMユーザーガイド」の「[ロールを引き受ける方法](#)」を参照してください。

IAM ロールと一時認証情報は、次の状況で役立ちます。

- フェデレーションユーザーアクセス – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロールの詳細については、「[ユーザーガイド](#)」の「[サードパーティー ID プロバイダー \(フェデレーション\) のロールを作成する](#)」を参照してください。IAM IAM Identity Center を使用する場合は、アクセス許可セットを設定します。ID が認証後にアクセスできる内容を制御するために、IAMIdentity

Center はアクセス許可セットを のロールに関連付けますIAM。アクセス許可セットの詳細については、「AWS IAM Identity Center User Guide」の「[Permission sets](#)」を参照してください。

- 一時的なIAMユーザーアクセス許可 – IAM ユーザーまたはロールは、IAMロールを引き受けて、特定のタスクに対して異なるアクセス許可を一時的に引き受けることができます。
- クロスアカウントアクセス IAMロールを使用して、自分のアカウントのリソースにアクセスすることを別のアカウントの信頼済みプリンシパルに許可できます。クロスアカウントアクセスを許可する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、(ロールをプロキシとして使用する代わりに) リソースに直接ポリシーをアタッチできます。クロスアカウントアクセスのロールとリソースベースのポリシーの違いについては、「IAMユーザーガイド」の「[でのクロスアカウントリソースアクセスIAM](#)」を参照してください。
- クロスサービスアクセス — 一部の は他の の機能 AWS のサービス を使用します AWS のサービス。例えば、サービスで呼び出しを行うと、そのサービスが Amazon でアプリケーションを実行 EC2したり、Amazon S3 にオブジェクトを保存したりするのが一般的です。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
- 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用してアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用すると、別のサービスで別のアクションを開始するアクションを実行できます。FASは、 を呼び出すプリンシパルのアクセス許可と AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストのリクエストを使用します。FAS リクエストは、他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストをサービスが受信した場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「[転送アクセスセッション](#)」を参照してください。
- サービスロール – サービスロールは、ユーザーに代わってアクションを実行するためにサービスが引き受ける [IAMロール](#)です。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAMユーザーガイド」の「[にアクセス許可を委任するロールを作成する AWS のサービス](#)」を参照してください。
- サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント、 サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。
- Amazon で実行されているアプリケーション EC2 – IAMロールを使用して、EC2インスタンスで実行され、AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。AWS ロール

ルをEC2インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時認証情報を取得することができます。詳細については、[「ユーザーガイド」の「IAMロールを使用して Amazon EC2 インスタンスで実行されているアプリケーションにアクセス許可を付与する」](#)を参照してください。

IAM

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、ID AWS またはリソースにアタッチします。ポリシーは のオブジェクト AWS であり、アイデンティティまたはリソースに関連付けられると、そのアクセス許可を定義します。は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーはJSON、ドキュメント AWS として に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「IAMユーザーガイド」の[JSON「ポリシーの概要」](#)を参照してください。

管理者はポリシーを使用して AWS JSON、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。IAM管理者は、リソースで必要なアクションを実行するためのアクセス許可をユーザーに付与する IAMポリシーを作成できます。その後、管理者はロールに IAMポリシーを追加し、ユーザーはロールを引き受けることができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションのアクセス許可を定義します。例えば、iam:GetRole アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、AWS Management Console、AWS CLIまたは AWS からロール情報を取得できます API。

アイデンティティベースのポリシー

アイデンティティベースのポリシーは、IAM ユーザー、ユーザーのグループ、ロールなど、アイデンティティにアタッチできるJSONアクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。ID ベースのポリシーを作成する方法については、「IAMユーザーガイド」の[「カスタマー管理ポリシーによるカスタムIAMアクセス許可の定義」](#)を参照してください。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには、AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。管理ポリシーまたはインラインポリシーを選択する方法については、「IAM ユーザーガイド」の[「管理ポリシーとインラインポリシーの選択」](#)を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースにアタッチするJSONポリシードキュメントです。リソースベースポリシーの例としては、IAMロール信頼ポリシーや Amazon S3 バケットポリシーなどがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、[プリンシパルを指定する](#)必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、またはを含めることができます AWS のサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーIAMで から AWS 管理ポリシーを使用することはできません。

アクセスコントロールリスト (ACLs)

アクセスコントロールリスト (ACLs) は、リソースへのアクセス許可を持つプリンシパル (アカウントメンバー、ユーザー、またはロール) を制御します。ACLsは、ポリシードキュメント形式を使用しませんが、リソースベースのJSONポリシーに似ています。

Amazon S3、AWS WAF、および Amazon VPCは、をサポートするサービスの例ですACLs。の詳細についてはACLs、Amazon Simple Storage Service デベロッパーガイドの[「アクセスコントロールリスト \(ACL\) の概要」](#)を参照してください。

その他のポリシータイプ

AWS は、一般的ではない追加のポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- **アクセス許可の境界** – アクセス許可の境界は、アイデンティティベースのポリシーがIAMエンティティ (IAM ユーザーまたはロール) に付与できるアクセス許可の上限を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。Principal

フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、IAM ユーザーガイドの「[IAM エンティティのアクセス許可の境界](#)」を参照してください。

- サービスコントロールポリシー (SCPs) – SCPs は、の組織または組織単位 (OU) の最大アクセス許可を指定する JSON ポリシーです AWS Organizations。AWS Organizations は、ビジネスが所有する複数の AWS アカウント をグループ化して一元管理するためのサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCPs) をアカウントの一部またはすべてに適用できます。は、各 を含むメンバーアカウントのエンティティのアクセス許可 SCP を制限します AWS アカウントのルートユーザー。Organizations との詳細については SCPs、「AWS Organizations ユーザーガイド」の「[サービスコントロールポリシー](#)」を参照してください。
- リソースコントロールポリシー (RCPs) – 所有する各リソースにアタッチされた JSON ポリシーを更新することなく、アカウント内のリソースに使用可能なアクセス許可の上限を設定するために使用できる IAM ポリシー RCPs です。は、メンバーアカウントのリソースのアクセス許可 RCP を制限し、組織に属しているかどうかにかかわらず AWS アカウントのルートユーザー、を含む ID の有効なアクセス許可に影響を与える可能性があります。をサポートする のリストを含む Organizations との詳細については RCPs、AWS Organizations 「ユーザーガイド RCPs」の AWS のサービス「[リソースコントロールポリシー \(RCPs\)](#)」を参照してください。
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、IAM ユーザーガイドの「[セッションポリシー](#)」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。複数のポリシータイプが関係する場合に、ガリクエストを許可するかどうか AWS を決定する方法については、「IAM ユーザーガイド」の「[ポリシーの評価ロジック](#)」を参照してください。

Amazon Translate との連携方法 IAM

IAM を使用して Amazon Translate へのアクセスを管理する前に、Amazon Translate で使用できる IAM 機能を確認してください。

IAM Amazon Translate で使用できる の機能

IAM 機能	Amazon Translate のサポート
アイデンティティベースポリシー	はい
リソースベースのポリシー	いいえ
ポリシーアクション	はい
ポリシーリソース	はい
ポリシー条件キー (サービス固有)	あり
ACLs	いいえ
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	あり
転送アクセスセッション (FAS)	あり
サービスロール	はい
サービスリンクロール	いいえ

Amazon Translate およびその他の AWS のサービスがほとんどの IAM 機能と連携する方法の概要を把握するには、IAM 「ユーザーガイド」の[AWS 「と連携する のサービスIAM」](#)を参照してください。

Amazon Translate のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースのポリシーは、IAM ユーザー、ユーザーのグループ、ロールなど、アイデンティティにアタッチできるJSONアクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。ID ベースのポリシーを作成する方法については、「IAMユーザーガイド」の[「カスタマー管理ポリシーを使用してカスタムIAMアクセス許可を定義する」](#)を参照してください。

IAM のアイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、またアクションが許可または拒否される条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素については、「IAMユーザーガイド」の「[IAMJSON ポリシー要素リファレンス](#)」を参照してください。

Amazon Translate のアイデンティティベースのポリシーの例

Amazon Translate のアイデンティティベースのポリシーの例を確認するには、「[Amazon Translate のアイデンティティベースのポリシーの例](#)」を参照してください。

Amazon Translate 内のリソースベースのポリシー

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースにアタッチするJSONポリシードキュメントです。リソースベースポリシーの例としては、IAMロール信頼ポリシーや Amazon S3 バケットポリシーなどがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、[プリンシパルを指定する](#)必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、またはを含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、全体のアカウント、または別のアカウントの IAM エンティティを、リソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる場合 AWS アカウント、信頼されたアカウントの IAM 管理者は、リソースへのアクセス許可をプリンシパルエンティティ (ユーザーまたはロール) に付与する必要もあります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーをさらに付与する必要はありません。詳細については、「IAMユーザーガイド」の「[でのクロスアカウントリソースアクセスIAM](#)」を参照してください。

Amazon Translate のポリシーアクション

ポリシーアクションのサポート: あり

管理者はポリシーを使用して AWS JSON、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの Action 要素は、ポリシーでアクセスを許可または拒否するために使用できるアクションを記述します。ポリシーアクションの名前は通常、関連する AWS API オペレーションと同じです。一致する API オペレーションがないアクセス許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

Amazon Translate アクションのリストを確認するには、「サービス認可リファレンス」の [Amazon Translate で定義されるアクション](#) を参照してください。

Amazon Translate のポリシーアクションは、アクションの前に次のプレフィックスを使用します。

```
translate
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [  
  "translate:ListLanguages",  
  "translate:TranslateText"  
]
```

ワイルドカード (*) を使用して複数アクションを指定できます。例えば、List という単語で始まるすべてのアクションを指定するには、次のアクションを含めます。

```
"Action": "translate:List*"
```

1 つのサービスに対してすべてのアクションを指定するワイルドカードを使用しないでください。最小特権を認めるというベストプラクティスに従い、ポリシー内のアクセス許可を絞り込みます。

Amazon Translate のアイデンティティベースのポリシーの例を確認するには、「[Amazon Translate のアイデンティティベースのポリシーの例](#)」を参照してください。

Amazon Translate のポリシーリソース

ポリシーリソースのサポート: あり

管理者はポリシーを使用して AWS JSON、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Resource JSON ポリシー要素は、アクションが適用されるオブジェクトを指定します。ステートメントには、Resource または NotResource 要素を含める必要があります。ベストプラクティスとして、[Amazon リソースネーム \(ARN\) を使用してリソース](#)を指定します。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*" 
```

Amazon Translate リソースタイプとその のリストを確認するには ARNs、「サービス認可リファレンス」の[Amazon Translate で定義されるリソース](#)を参照してください。各リソースARNの を指定できるアクションについては、[Amazon Translate で定義されるアクション](#)を参照してください。

Amazon Translate ポリシーでリソースを使用する方法の例については、[ポリシーでのリソースの指定](#)を参照してください。

Amazon Translate のポリシー条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者はポリシーを使用して AWS JSON、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルが、どのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの [条件演算子](#) を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、AWS では AND 論理演算子を使用してそれらを評価します。1 つの条

件キーに複数の値を指定すると、は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。たとえば、IAM ユーザー名でタグ付けされている場合のみ、リソースにアクセスする IAM ユーザーアクセス許可を付与できます。詳細については、IAMユーザーガイドの「[IAMポリシーエレメント: 変数およびタグ](#)」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートしています。すべての AWS グローバル条件キーを確認するには、「IAMユーザーガイド」の[AWS 「グローバル条件コンテキストキー」](#)を参照してください。

Amazon Translate の条件キーのリストを確認するには、「サービス認可リファレンス」の[Amazon Translate の条件キー](#)」を参照してください。条件キーを使用できるアクションとリソースについては、「[Amazon Translate で定義されるアクション](#)」を参照してください。

Amazon Translate のアイデンティティベースのポリシーの例を確認するには、「[Amazon Translate のアイデンティティベースのポリシーの例](#)」を参照してください。

ACLs Amazon Translate の

をサポートACLs：なし

アクセスコントロールリスト (ACLs) は、リソースへのアクセス許可を持つプリンシパル (アカウントメンバー、ユーザー、またはロール) を制御します。ACLsは、ポリシードキュメント形式を使用しませんが、リソースベースのJSONポリシーに似ています。

ABAC Amazon Translate での

サポート ABAC (ポリシー内のタグ): 一部

属性ベースのアクセスコントロール (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。では AWS、これらの属性はタグと呼ばれます。タグは、IAMエンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、の最初のステップですABAC。次に、プリンシパルのタグがアクセスしようとしているリソースのタグと一致する場合に、オペレーションを許可するABACポリシーを設計します。

ABAC は、急速に成長している環境や、ポリシー管理が煩雑になる状況で役立ちます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの[条件要素](#)でタグ情報を提供します。

サービスがすべてのリソースタイプに対して3つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ3つの条件キーのすべてをサポートする場合、値は「部分的」になります。

の詳細についてはABAC、「IAMユーザーガイド」の[ABAC「認可によるアクセス許可の定義」](#)を参照してください。をセットアップする手順を含むチュートリアルを表示するにはABAC、「IAMユーザーガイド」の[「属性ベースのアクセスコントロール \(ABAC\)」](#)を使用する」を参照してください。

Amazon Translate リソースのタグ付けの詳細については、「[リソースのタグ付け](#)」を参照してください。

Amazon Translate での一時的な認証情報の使用

一時的な認証情報のサポート: あり

一部のAWSのサービスは、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報とAWSのサービス連携するなどの詳細については、「IAMユーザーガイド」の[AWSのサービス「と連携するIAM」](#)を参照してください。

ユーザー名とパスワード以外のAWS Management Console方法でサインインする場合は、一時的な認証情報を使用します。たとえば、会社のシングルサインオン(SSO)リンクAWSを使用してアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えの詳細については、「IAMユーザーガイド」の[「ユーザーからIAMロールへの切り替え\(コンソール\)」](#)を参照してください。

一時的な認証情報は、AWS CLIまたはを使用して手動で作成できますAWS API。その後、これらの一時的な認証情報を使用してアクセスすることができますAWS。AWSでは、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することをお勧めします。詳細については、「IAM」の[「一時的なセキュリティ認証情報IAM」](#)を参照してください。

Amazon Translate の転送アクセスセッション

転送アクセスセッションをサポート (FAS): はい

ユーザーIAMまたはロールを使用してアクションを実行するとAWS、プリンシパルと見なされます。一部のサービスを使用すると、別のサービスで別のアクションを開始するアクションを実行できます。FASは、を呼び出すプリンシパルのアクセス許可とAWSのサービス、ダウンストリームサービスAWSのサービスへのリクエストのリクエストを使用します。FASリクエストは、他のAWSのサービスまたはリソースとのやり取りを完了する必要があるリクエストをサービスが受信

した場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、[「転送アクセスセッション」](#)を参照してください。

Amazon Translate のサービスロール

サービスロールのサポート: あり

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける [IAM ロール](#) です。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAMユーザーガイド」の「[にアクセス許可を委任するロールを作成する AWS のサービス](#)」を参照してください。

Warning

サービスロールの許可を変更すると、Amazon Translate の機能が破損する可能性があります。Amazon Translate が指示する場合以外は、サービスロールを編集しないでください。

Amazon Translate 非同期オペレーションを使用するには、入カドキュメントが含まれている Amazon S3 バケットへのアクセスを Amazon Translate に許可する必要があります。これを行うには、Amazon Translate サービスプリンシパルを信頼するように、アカウントにサービスロールを作成します。

ポリシーの例については、「[バッチ翻訳ジョブの前提条件](#)」を参照してください。

Amazon Translate のサービスにリンクされたロール

サービスにリンクされたロールのサポート: なし

サービスにリンクされたロールは、にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

サービスにリンクされたロールの作成または管理の詳細については、「[IAMと連携するAWS サービス](#)」を参照してください。表の中から、[サービスリンクロール] 列に Yes と記載されたサービスを見つけます。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[はい] リンクを選択します。

Amazon Translate のアイデンティティベースのポリシーの例

デフォルトでは、ユーザーおよびロールには Amazon Translate リソースを作成または変更するアクセス許可がありません。また、AWS Management Console、AWS CLI、またはを使用してタスクを実行することはできません AWS API。IAM 管理者は、必要な特定のリソースに対して特定のAPIオペレーションを実行するアクセス許可を付与するIAMポリシーを作成する必要があります。続いて、管理者はそれらのアクセス許可が必要なユーザーまたはロールに、そのポリシーをアタッチします。

次のポリシードキュメント例を使用してIAMアイデンティティベースのJSONポリシーを作成する方法については、IAMユーザーガイドの「[JSONタブでのポリシーの作成](#)」を参照してください。

トピック

- [アイデンティティベースのポリシーのベストプラクティス](#)
- [Amazon Translate コンソールへのアクセスを許可する](#)
- [自分の権限の表示をユーザーに許可する](#)
- [ポリシーでのリソースの指定](#)
- [カスタム用語でカスタマーマネージドキーを使用するためのアクセス許可](#)

アイデンティティベースのポリシーのベストプラクティス

アイデンティティベースのポリシーは、ユーザーのアカウント内で誰かが Amazon Translate リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションを実行すると、AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください：

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可を付与するには、多くの一般的なユースケースにアクセス許可を付与する AWS 管理ポリシーを使用します。これらはで使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、「IAMユーザーガイド」の「[AWS マネージドポリシー](#)」または「[AWS ジョブ機能の管理ポリシー](#)」を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要なアクセス許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用してアクセス許可を適用する方法の詳細については、IAM ユーザーガイドの「[IAM のポリシーとアクセス許可](#)」を参照してください。

- IAMポリシーで条件を使用してアクセスをさらに制限する – ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを `SSL` を使用して送信するように指定できます。条件を使用して、サービスアクションが `aws:iam:*` などの特定の `aws:iam:*` を通じて使用される場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます `AWS CloudFormation`。詳細については、「IAMユーザーガイド」の [IAMJSON「ポリシー要素: 条件」](#) を参照してください。
- IAM Access Analyzer を使用してIAMポリシーを検証し、安全で機能的なアクセス許可を確保する – IAM Access Analyzer は、ポリシーがポリシー言語 (JSON) とIAMベストプラクティスに準拠するように、新規および既存のIAMポリシーを検証します。IAM Access Analyzer には、安全で機能的なポリシーの作成に役立つ 100 を超えるポリシーチェックと実用的な推奨事項が用意されています。詳細については、「IAMユーザーガイド」の [IAM「Access Analyzer を使用したポリシーの検証」](#) を参照してください。
- 多要素認証を要求する (MFA) – IAMユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化MFAするために `aws:iam:*` をオンにします。API オペレーションが呼び出されるMFAタイミングを要求するには、ポリシーにMFA条件を追加します。詳細については、「IAMユーザーガイド」の [「を使用した安全なAPIアクセスMFA」](#) を参照してください。

IAM でのベストプラクティスの詳細については、「IAMユーザーガイド」の [「IAMでのセキュリティのベストプラクティス」](#) を参照してください。

Amazon Translate コンソールへのアクセスを許可する

Amazon Translate コンソールにアクセスするには、許可の最小限のセットが必要です。これらのアクセス許可により、AWS アカウントの Amazon Translate リソースの詳細を一覧表示および表示できます。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザー、グループ、またはロール) に対してコンソールが意図したとおりに機能しません。

Amazon Translate コンソールのアクセス許可については、エンティティに `TranslateFullAccess` AWS 管理ポリシーをアタッチできます。詳細については、「[AWS Amazon Translate の マネージドポリシー](#)」を参照してください。

また、次のポリシーに示すアクションにアクセス許可を付与する必要があります。これらのアクセス許可は、`TranslateFullAccess` ポリシーに含まれています。

```
{
  "Version": "2012-10-17",
```

```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "iam:ListRoles",
      "iam:GetRole",
      "s3:ListAllMyBuckets",
      "s3:ListBucket",
      "s3:GetBucketLocation"
    ],
    "Resource": "*"
  }
]
```

AWS CLI または のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません AWS API。代わりに、実行しようとしているAPIオペレーションに一致するアクションのみへのアクセスを許可します。詳細については、「IAMユーザーガイド」の「[ユーザーへのアクセス許可の追加](#)」を参照してください。

自分の権限の表示をユーザーに許可する

この例では、ユーザー ID にアタッチされたインラインおよび管理ポリシーの表示を IAM ユーザーに許可するポリシーを作成する方法を示します。このポリシーには、コンソールで、または AWS CLI または を使用してプログラムでこのアクションを実行するアクセス許可が含まれています AWS API。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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": "*"
    }
  ]
}
```

ポリシーでのリソースの指定

多くの Amazon Translate API アクションでは、アクションに許可される (または許可されない) リソースを指定することで、ポリシーの範囲を制限できます。リソースを指定できるアクションのリストについては、「[Amazon Translate で定義されるアクション](#)」を参照してください。ポリシーでは以下のリソースを指定できます。

- Custom terminology – 次のARN形式を使用します。

```
arn:partition:translate:region:account:terminology/terminology-name/LATEST
```

- Parallel data – 次のARN形式を使用します。

```
arn:partition:translate:region:account:parallel-data/parallel-data-name
```

ワイルドカード文字を使用すると、ポリシーで複数のリソースを指定できます。以下のポリシー例では、すべての Amazon Translate アクションに対してすべてのカスタム用語リソースを許可します。

Example

```
{
  "Sid": "Example1",
```

```
    "Effect": "Allow",
    "Action": "translate:*",
    "Resource": [
        "arn:aws:translate:us-west-2:123456789012:terminology/*"
    ]
}
```

次のポリシー例では、GetParallelData アクションの特定の並列データリソースへのアクセスを拒否します。

Example

```
{
    "Sid": "Example2",
    "Effect": "Deny",
    "Action": "translate:GetParallelData",
    "Resource": [
        "arn:aws:translate:us-west-2:123456789012:parallel-data/test-parallel-
data"
    ]
}
```

カスタム用語でカスタマーマネージドキーを使用するためのアクセス許可

Amazon Translate カスタム用語で AWS Key Management Service (AWS KMS) カスタマーマネージドキーを使用する場合は、KMSキーポリシーで追加のアクセス許可が必要になる場合があります。

カスタマーマネージドキーを使用して ImportTerminologyオペレーションを呼び出すには、既存のKMSキーポリシーに次のアクセス許可を追加します。

```
{
    "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:CreateAlias",
        "kms:CreateGrant",
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:GetKeyPolicy",
        "kms:PutKeyPolicy",
        "kms:RetireGrant"
    ],
    "Resource": "*"
}
]
}

```

KMS カスタマーマネージドキーでインポートされたカスタム用語に対して GetTerminology オペレーションを呼び出すには、KMS キーポリシーに次のアクセス許可を追加します。

```

{
  "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",
        "kms:GetKeyPolicy",
        "kms:PutKeyPolicy"
      ],
      "Resource": "*"
    }
  ]
}

```

カスタマーマネージドキーでインポートされたカスタム用語の ListTerminologies または DeleteTerminology オペレーションを呼び出すには、特別な AWS KMS アクセス許可は必要ありません。

すべてのカスタム用語オペレーションでカスタマーマネージドキーを使用するには、KMS キーポリシーに次のアクセス許可を追加します。


```
{
  "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:Decrypt",
        "kms:DescribeKey",
        "kms:GenerateDataKey",
        "kms:GetKeyPolicy",
        "kms:PutKeyPolicy",
        "kms:RetireGrant"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon Translate のオペレーションとリソースの詳細については、「サービス認可リファレンス」の[Amazon Translate のアクション、リソース、および条件キー](#)を参照してください。

AWS Amazon Translate の マネージドポリシー

AWS 管理ポリシーは、によって作成および管理されるスタンドアロンポリシーです AWS。AWS 管理ポリシーは、多くの一般的なユースケースにアクセス許可を付与するように設計されているため、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できます。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケースに固有の[カスタマー管理ポリシー](#)を定義して、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS 管理ポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) が更新されます。AWS AWS のサービスは、新しいが起動されたたと

き、または既存のサービスで新しいAPIオペレーションが利用可能になったときに、AWS 管理ポリシーを更新する可能性が最も高くなります。

詳細については、「IAM ユーザーガイド」の「[AWS 管理ポリシー](#)」を参照してください。

トピック

- [AWS 管理ポリシー: TranslateFullAccess](#)
- [AWS 管理ポリシー: TranslateReadOnly](#)
- [AWS マネージドポリシーに対する Amazon Translate の更新](#)

AWS 管理ポリシー: TranslateFullAccess

このポリシーは、Amazon Translate リソース、Amazon Comprehend DetectDominantLanguage API オペレーション、および必要な CloudWatch API オペレーションへのフルアクセスを許可します。このポリシーは、Amazon S3 バケットと IAM ロールのリストと取得のアクセス許可も付与します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "translate:*",
        "comprehend:DetectDominantLanguage",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "iam:ListRoles",
        "iam:GetRole"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

AWS 管理ポリシー: TranslateReadOnly

このポリシーは、アカウントに関連付けられたリソースを変更しない Amazon Translate API オペレーションにアクセスする許可を付与します。このポリシーは、Amazon Comprehend DetectDominantLanguage API オペレーションおよび必要な CloudWatch API オペレーションにアクセスするためのアクセス許可も付与します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "translate:TranslateText",
        "translate:TranslateDocument",
        "translate:GetTerminology",
        "translate:ListTerminologies",
        "translate:ListTextTranslationJobs",
        "translate:DescribeTextTranslationJob",
        "translate:GetParallelData",
        "translate:ListParallelData",
        "comprehend:DetectDominantLanguage",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

AWS マネージドポリシーに対する Amazon Translate の更新

Amazon Translate の AWS マネージドポリシーの更新に関する詳細を、このサービスがこれらの変更の追跡を開始した以降の分について表示します。このページの変更に関する自動通知については、Amazon Translate [ドキュメント履歴](#) ページのRSSフィードにサブスクライブしてください。

変更	説明	日付
TranslateReadOnly - 既存ポリシーへの更新	Amazon Translate が TranslateReadOnly ポリシーで TranslateDocument アクションを許可するようになりました	2023 年 5 月 23 日
Amazon Translate が変更の追跡を開始	Amazon Translate が AWS マネージドポリシーの変更の追跡を開始しました。	2023 年 5 月 23 日

Amazon Translate のアイデンティティとアクセスに関するトラブルシューティング

次の情報は、Amazon Translate と の使用時に発生する可能性がある一般的な問題の診断と修復に役立ちますIAM。

トピック

- [Amazon Translate でアクションを実行する権限がありません](#)
- [iam を実行する権限がありません。PassRole](#)
- [自分の 以外のユーザーに Amazon Translate リソース AWS アカウント へのアクセスを許可したい](#)

Amazon Translate でアクションを実行する権限がありません

アクションを実行する権限がないというエラーが表示された場合は、そのアクションを実行できるようにポリシーを更新する必要があります。

次の例のエラーは、mateojacksonIAMユーザーがコンソールを使用して架空の`my-example-widget`リソースの詳細を表示しようとしているが、架空の`translate:GetWidget`アクセス許可がない場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
translate:GetWidget on resource: my-example-widget
```

この場合、Mateo のポリシーでは、*my-example-widget* アクションを使用して `translate:GetWidget` リソースへのアクセスを許可するように更新する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

iam を実行する権限がありません。PassRole

`iam:PassRole` アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して Amazon Translate にロールを渡せるようにする必要があります。

一部の AWS のサービスでは、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、既存のロールをそのサービスに渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

次の例のエラーは、という名前の IAM ユーザーがコンソールを使用して Amazon Translate `marymajor` でアクションを実行しようとする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。メアリーには、ロールをサービスに渡す許可がありません。

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

この場合、Mary のポリシーを更新してメアリーに `iam:PassRole` アクションの実行を許可する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

自分の 以外のユーザーに Amazon Translate リソース AWS アカウント へのアクセスを許可したい

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACLs) をサポートするサービスでは、それらのポリシーを使用して、リソースへのアクセスをユーザーに許可できます。

詳細については、以下を参照してください。

- Amazon Translate がこれらの機能をサポートしているかどうかについては、「[Amazon Translate との連携方法 IAM](#)」を参照してください。

- 所有 AWS アカウント する 全体で リソースへのアクセスを提供する方法については、「IAM ユーザーガイド」の [「所有 AWS アカウント する別の の IAM ユーザーへのアクセスを提供する」](#) を参照してください。
- リソースへのアクセスをサードパーティーに提供する方法については AWS アカウント、「IAM ユーザーガイド」の [「サードパーティー AWS アカウント が所有する へのアクセスを提供する」](#) を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、「IAM ユーザーガイド」の [「外部で認証されたユーザー \(ID フェデレーション\) へのアクセスの許可」](#) を参照してください。
- クロスアカウントアクセスでのロールとリソースベースのポリシーの使用の違いについては、「IAM ユーザーガイド」の [「でのクロスアカウントリソースアクセスIAM」](#) を参照してください。

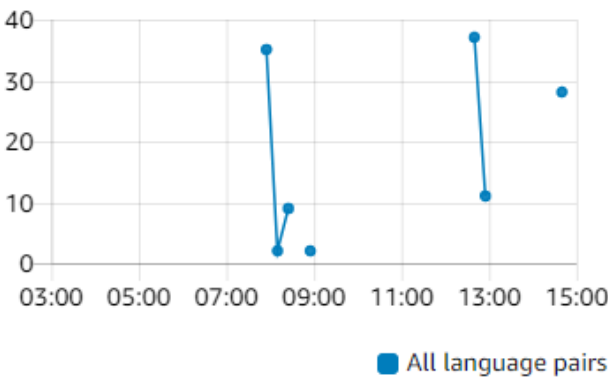
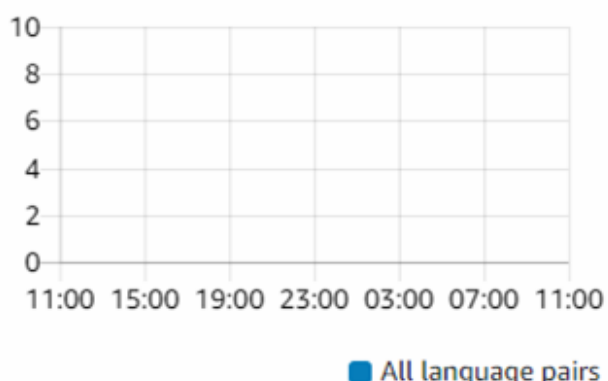
Amazon Translate のモニタリング

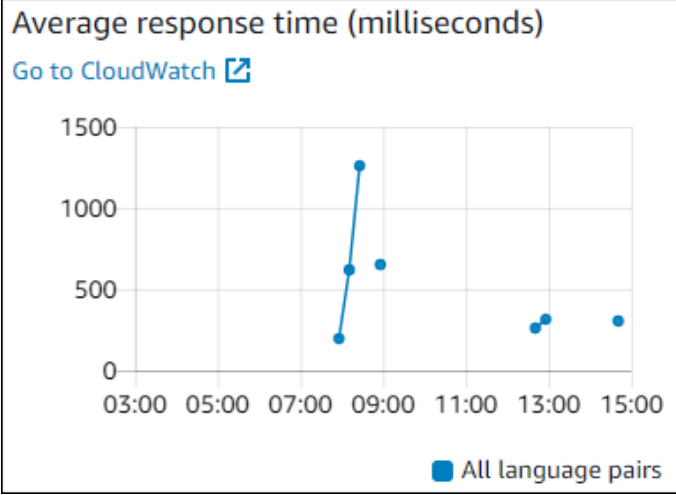
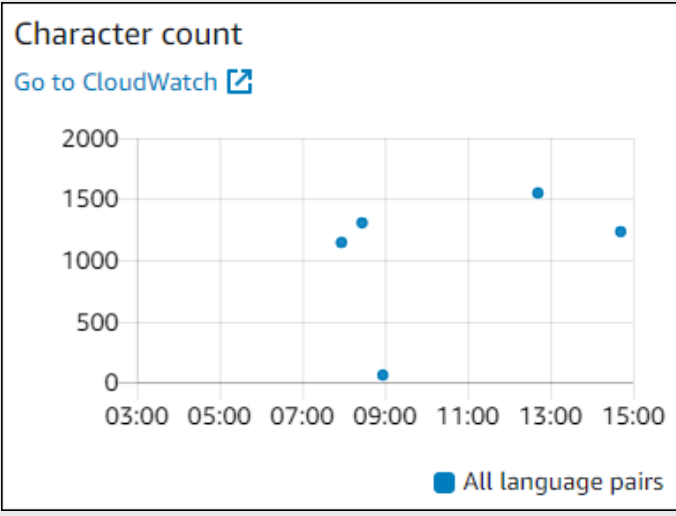
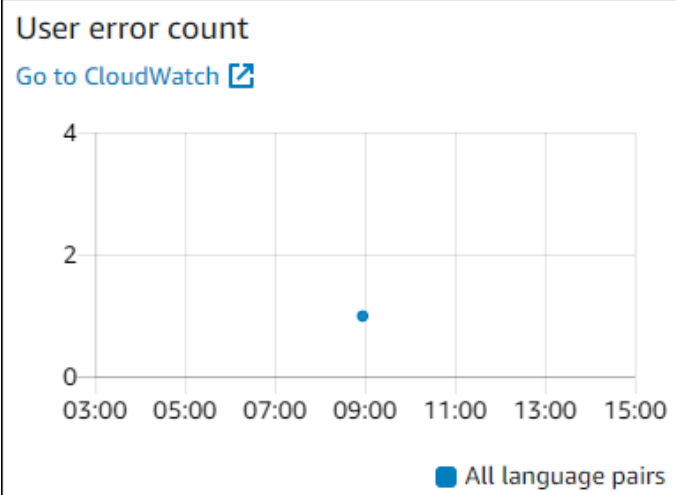
モニタリングは、Amazon Translate と ソリューションの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。は、Amazon Translate のモニタリングに使用できるさまざまなツール AWS を提供します。こうしたツールをいくつか設定することで、ソリューションのモニタリングをカスタマイズできます。モニタリングタスクをできるだけ自動化することをお勧めします。

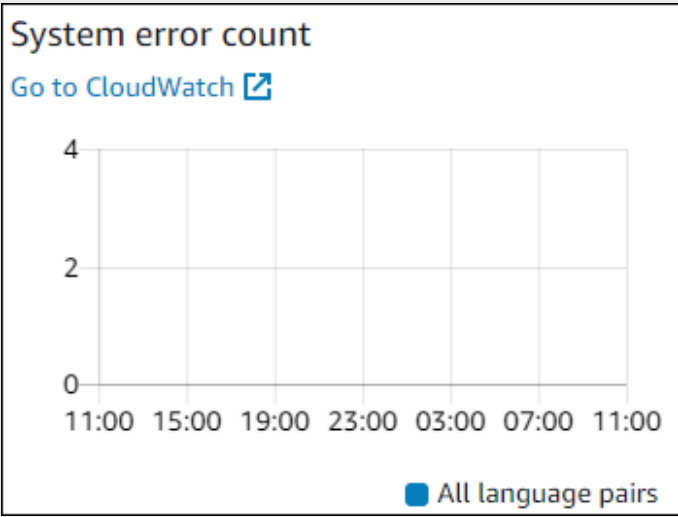
Amazon Translate はソリューションでもっとも重要なメトリクスを表示する事前設定済みのグラフを提供します。各グラフはソリューションのパフォーマンス概要を示しています。ソリューションパフォーマンスの経時的な違いを見るためには、グラフで表示する時間範囲を変更できます。

Amazon Translate をモニタリング CloudWatch するために Amazon を使用することもできます。Amazon Translate を使用すると CloudWatch、ソリューションの特定のメトリクスのモニタリングを自動化できます。メトリクスが設定したしきい値の外にある場合は通知が届きます。を使用して CloudWatch API、ニーズに適したカスタムモニタリングアプリケーションを作成することもできます。詳細については、[「Amazon ユーザーガイド」の「Amazon とは CloudWatch」](#) を参照してください。 CloudWatch

次の表は Amazon Translate が提供する事前設定済みのグラフをそれぞれ説明したものです。

グラフ	説明																
<p data-bbox="126 262 487 298">Successful request count</p> <p data-bbox="126 310 370 346">Go to CloudWatch</p>  <table border="1" data-bbox="175 367 779 745"><thead><tr><th>Time</th><th>Successful request count</th></tr></thead><tbody><tr><td>08:00</td><td>35</td></tr><tr><td>08:30</td><td>10</td></tr><tr><td>09:00</td><td>3</td></tr><tr><td>13:00</td><td>38</td></tr><tr><td>15:00</td><td>28</td></tr></tbody></table> <p data-bbox="548 709 779 745">■ All language pairs</p>	Time	Successful request count	08:00	35	08:30	10	09:00	3	13:00	38	15:00	28	<p data-bbox="831 247 1242 283">成功したリクエストカウント</p> <p data-bbox="831 325 1485 409">指定した期間中に成功した Amazon Translate へのリクエスト数です。</p>				
Time	Successful request count																
08:00	35																
08:30	10																
09:00	3																
13:00	38																
15:00	28																
<p data-bbox="126 823 506 858">Throttled request count</p> <p data-bbox="126 871 393 907">Go to CloudWatch</p>  <table border="1" data-bbox="175 934 779 1312"><thead><tr><th>Time</th><th>Throttled request count</th></tr></thead><tbody><tr><td>11:00</td><td>0</td></tr><tr><td>15:00</td><td>0</td></tr><tr><td>19:00</td><td>0</td></tr><tr><td>23:00</td><td>0</td></tr><tr><td>03:00</td><td>0</td></tr><tr><td>07:00</td><td>0</td></tr><tr><td>11:00</td><td>0</td></tr></tbody></table> <p data-bbox="516 1281 779 1316">■ All language pairs</p>	Time	Throttled request count	11:00	0	15:00	0	19:00	0	23:00	0	03:00	0	07:00	0	11:00	0	<p data-bbox="831 802 1242 837">調整済みリクエストカウント</p> <p data-bbox="831 879 1502 1060">指定した期間中に調整した Amazon Translate へのリクエスト数です。この情報を使用してアプリケーションが Amazon Translate にリクエストを送信するのが早すぎるか判断します。</p>
Time	Throttled request count																
11:00	0																
15:00	0																
19:00	0																
23:00	0																
03:00	0																
07:00	0																
11:00	0																

グラフ	説明
<p>Average response time (milliseconds) Go to CloudWatch</p>  <p>■ All language pairs</p>	<p>平均応答時間</p> <p>Amazon Translate が指定期間中にリクエストを処理した平均時間です。</p>
<p>Character count Go to CloudWatch</p>  <p>■ All language pairs</p>	<p>文字数カウント</p> <p>指定した指定期間中に Amazon Translate へ送信した文字数の合計です。この合計数が請求されます。</p>
<p>User error count Go to CloudWatch</p>  <p>■ All language pairs</p>	<p>ユーザーエラーカウント</p> <p>指定した期間中に発生したユーザーエラーの数です。ユーザーエラーはHTTPエラーコードの範囲 400～499 です。</p>

グラフ	説明
 <p>The graph, titled 'System error count', shows a line representing 'All language pairs'. The y-axis ranges from 0 to 4 with increments of 2. The x-axis shows time intervals: 11:00, 15:00, 19:00, 23:00, 03:00, 07:00, and 11:00. The line remains at 0 throughout the period. A link 'Go to CloudWatch' is visible at the top left of the graph area.</p>	<p>システムエラーカウント</p> <p>指定した期間中に発生したシステムエラーの数です。システムエラーは、HTTPエラーコードの範囲 500 ~ 599 です。</p>

Amazon Translate のモニタリング

Amazon では CloudWatch、個々の Amazon Translate オペレーションのメトリクスまたはアカウントのグローバル Amazon Translate メトリクスを取得できます。メトリクスを使用すると、Amazon Translate ソリューションの状態をトラッキングし、定義されているしきい値を 1 つ以上のメトリクスが超えた場合に通知するようにアラームをセットアップできます。たとえば、指定期間中に Amazon Translate に送信されたリクエスト数をモニタリングして、リクエストのレイテンシーを確認し、エラー数がしきい値を超えた場合はアラームを生成できます。

Amazon Translate の CloudWatch メトリクスについて

Amazon Translate オペレーションのメトリクスを取得するには、以下の情報を指定する必要があります。

- メトリクスディメンション。ディメンションは、メトリクスを識別するための名前と値のペアのセットです。Amazon Translate には 2 つのディメンションがあります。
 - Operation
 - Language pair
- メトリクス名 (SuccessfulRequestCount、RequestCharacters など)。メトリクスの一覧については、「[CloudWatch Amazon Translate のメトリクス](#)」を参照してください。

Amazon Translate のメトリクスは AWS Management Console、AWS CLI、またはを使用して取得できます CloudWatch API。は、Amazon AWS Software Development Kit (SDKs) のいずれかまたは CloudWatch API ツール API から使用できます CloudWatch。

次の表に、CloudWatch メトリクスの一般的な用途を示します。ここで紹介するのは使用開始するための提案事項であり、総括的な一覧ではありません。

どうすればよいか?	モニタリング対象のメトリクス
成功したリクエストの数をトラッキングします	SuccessfulRequestCount のメトリクスの sum 統計
アプリケーションが最大しきい値に到達した場合に通知します	ThrottledCount のメトリクスの sum 統計
アプリケーションの応答時間を検索します	ResponseTime のメトリクスの average 統計
アプリケーションのエラー数を見つけます	ServerErrorCount と UserErrorCount のメトリクスの sum 統計をモニタリングします。
請求可能な文字の数を検索します	CharacterCount のメトリクスの sum 統計

で Amazon Translate をモニタリングするには、適切な CloudWatch アクセス許可が必要です。詳細については、「Amazon CloudWatch CloudWatch ユーザーガイド」の「Amazon の [認証とアクセスコントロール CloudWatch](#)」を参照してください。

Amazon Translate メトリクスの表示

CloudWatch コンソールで Amazon Translate メトリクスを表示します。

メトリクスを表示するには (CloudWatch コンソール)

1. にサインイン AWS Management Console し、 で CloudWatch コンソールを開きます <https://console.aws.amazon.com/cloudwatch/>。
2. メトリクスを選択し、すべてのメトリクスを選択し、AWS/翻訳を選択します。
3. デイメンションを選択して、メトリクスの名前を選び [Add to graph] を選択します。

4. 日付範囲の値を選択します。指定した日付範囲のメトリクスカウントがグラフに表示されます。

を使用した Amazon Translate API呼び出しのログ記録 AWS CloudTrail

Amazon Translate は AWS CloudTrail、Amazon Translate のIAMエンティティまたはサービスによって実行されたアクションを記録する AWS サービスであると統合されています Amazon Translate 。はAmazon Translate のすべてのAPI呼び出しをイベントとして CloudTrail キャプチャします。これには、Amazon Translate コンソールからの呼び出しとAmazon Translate APIオペレーションへのコード呼び出しが含まれます。CloudTrail 証跡を作成する場合は、Amazon Translate の CloudTrail イベントを含むイベントの Amazon Simple Storage Service (Amazon S3) バケットへの継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールのイベント履歴で最新のイベントを表示できます。によって収集された情報を使用して CloudTrail 、Amazon Translate に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

詳細については CloudTrail、[「AWS CloudTrail ユーザーガイド」](#) を参照してください。

トピック

- [の Amazon Translate 情報 CloudTrail](#)
- [Amazon Translate ログファイルエントリの概要](#)

の Amazon Translate 情報 CloudTrail

CloudTrail AWS アカウントを作成すると、 がアカウントで有効になります。Amazon Translate でアクティビティが発生すると、そのアクティビティは CloudTrail イベント履歴の他の AWS サービス イベントとともにイベントに記録されます。AWS アカウントで最近のイベントを表示、検索、ダウンロードできます。詳細については、[「イベント履歴を使用した CloudTrail イベントの表示」](#) を参照してください。

Amazon Translate のイベントなど、AWS アカウントのイベントの継続的な記録については、証跡を作成します。証跡により、 はログファイル CloudTrail を Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成すると、証跡はすべてのAWSリージョンに適用されます。証跡では、AWS パーティションのすべてのリージョンからのイベントがログに記録され、指定した S3 バケットにログファイルが配信されます。CloudTrail ログで収集されたイベントデータをより詳細に分析し、それに基づいて行動するように、他の AWS サービスを設定できます。詳細については、次を参照してください:

- [追跡を作成するための概要](#)
- [CloudTrail サポートされているサービスと統合](#)
- [の Amazon SNS通知の設定 CloudTrail](#)
- [複数のリージョンからの CloudTrail ログファイルの受信と複数のアカウントからの CloudTrail ログファイルの受信](#)

すべての Amazon Translate アクションは、[APIリファレンス セクション](#)によってログに記録され、[APIリファレンス セクション](#)に記載されています。例えば、DeleteTerminology、および TranslateText アクションを呼び出す ImportTerminology と、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。この情報は以下のことを確認するのに役立ちます:

- リクエストが、ルートユーザーの認証情報で行われたかどうか
- リクエストが、ロールとフェデレーテッドユーザーのどちらの一時的なセキュリティ認証情報を使用して送信されたか
- リクエストが別の AWS サービスによって行われたかどうか

詳細については、[CloudTrail userIdentity要素](#)を参照してください。

Amazon Translate ログファイルエントリの概要

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信できるようにする設定です。CloudTrail ログファイルには 1 つ以上のログエントリが含まれます。イベントは任意のソースからの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。CloudTrail ログファイルはパブリックAPIコールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次の例は、TranslateText アクションを示す CloudTrail ログエントリを示しています。

```
{
  "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.metal1.x86_64 boto3/1.12.197",
  "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"
}

```

CloudWatch Amazon Translate の メトリクスとディメンション

ソリューションのパフォーマンスをモニタリングするには、Amazon Translate の Amazon CloudWatch メトリクスとディメンションを使用します。

CloudWatch Amazon Translate のメトリクス

メトリクス	説明
CharacterCount	<p>リクエスト内の請求可能な文字の数を検索します。</p> <p>有効なディメンション: 言語ペア、オペレーション</p> <p>有効な統計: 平均、最大、平均、合計</p> <p>単位: 数</p>

メトリクス	説明
ResponseTime	<p>リクエストに回答するまでにかかった時間です。</p> <p>有効なディメンション: 言語ペア、オペレーション</p> <p>有効な統計: データサンプル、平均</p> <p>単位: データサンプルの場合は個。平均統計の場合はミリ秒。</p>
ServerErrorCount	<p>サーバーエラーの数。サーバーエラーのHTTPレスポンスコードの範囲は 500 ~ 599 です。</p> <p>有効なディメンション: オペレーション</p> <p>有効な統計: 合計、平均</p> <p>単位: 数</p>
SuccessfulRequestCount	<p>成功した翻訳リクエストの数です。成功したリクエストのレスポンスコードは 200 ~ 299 です。</p> <p>有効なディメンション: オペレーション</p> <p>有効な統計: 合計、平均</p> <p>単位: 数</p>
ThrottledCount	<p>スロットリングリクエスト件名の数です。ThrottledCount を使用して、アプリケーションによる Amazon Translate へのリクエストの送信速度が、アカウントでの受信設定よりも早いかどうかを判断します。詳細は「Amazon Translate Limits」を「Amazon Web Services 全般のリファレンス」でご覧ください。</p> <p>有効なディメンション: オペレーション</p> <p>有効な統計: 合計、平均</p> <p>単位: 数</p>

メトリクス	説明
UserErrorCount	<p>発生したユーザーエラーの数です。ユーザーエラーのHTTPレスポンスコードの範囲は 400~499 です。</p> <p>有効なディメンション: オペレーション</p> <p>有効な統計: 合計、平均</p> <p>単位: 数</p>

CloudWatch Amazon Translate のディメンション

次のディメンションを使用して Amazon Translate メトリクスをフィルターすることができます。メトリクスはソース言語とターゲット言語でグループ化されます。

ディメンション	説明
LanguagePair	特定の言語だけを含むようにメトリクスを制限します。
Operation	特定のオペレーションを使用するものだけにメトリクスを制限します。

Amazon による Amazon Translate イベントのモニタリング EventBridge

Amazon Translate は Amazon と統合 EventBridge され、翻訳ジョブや並列データリソースに影響する変更について通知します。AWS サービスからのイベントは、ほぼリアルタイムで EventBridge に配信されます。簡単なルールを記述して、注目するイベントと、イベントがルールに一致した場合に自動的に実行するアクションを指定できます。たとえば、自動的に開始できるアクションには以下のようなものがあります。

- AWS Lambda 関数の呼び出し
- AWS Systems Manager Run Command の呼び出し
- Amazon Kinesis Data Streams へのイベントの中継
- AWS Step Functions ステートマシンのアクティブ化
- Amazon SNSトピックまたは Amazon SQSキューの通知

詳細については、「[Amazon ユーザーガイド](#)」の「[イベントに反応する Amazon EventBridge ルールの作成](#)」を参照してください。 EventBridge

Amazon Translate イベント

以下に、Amazon Translate のイベントの例を示します。

バッチ翻訳ジョブのイベント

バッチ翻訳ジョブを実行するには、Amazon Translate コンソールまたは [StartTextTranslationJob](#) オペレーションを使用します。Amazon Translate では、これらのジョブが正常に完了した時または失敗した時にイベントが送信されます。これらのイベントは次の例のようになります。

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "Translate TextTranslationJob State Change",
  "source": "aws.translate",
  "account": "111122223333",
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "jobId": "01234567-0123-0123-0123-012345678901",
    "jobStatus": "STATUS"
  }
}
```

jobStatus 属性の値は、Amazon Translate から送信されたイベントのジョブの状態によって異なります。jobStatus 値は次のとおりです。

- COMPLETED — ジョブが正常に完了し、出力が使用可能になりました。
- COMPLETED_WITH_ERROR — ジョブの完了時にエラーが発生しました。エラーは、ジョブの出力で分析できます。
- STOPPED — ジョブが停止されました。
- FAILED — ジョブが完了しませんでした。詳細を取得するには、[DescribeTextTranslationJob](#) オペレーションを使用します。

並列データリソースのイベント

Amazon Translate を使用して並列データリソースを作成または更新すると、オペレーションが成功したか失敗したかを示すイベントが送信されます。

Amazon Translate コンソールまたは [CreateParallelData](#) オペレーションを使用して、並列データリソースを作成します。これを行うと、Amazon Translate により次のようなイベントが送信されます。

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "Translate Parallel Data State Change",
  "source": "aws.translate",
  "account": "111122223333",
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [arn:aws:translate:us-east-1:111122223333:parallel-data/
ExampleParallelData],
  "detail": {
    "operation": "CreateParallelData",
    "name": "ExampleParallelData",
    "status": "STATUS"
  }
}
```

status 属性の値は次の通りです。

- ACTIVE — CreateParallelData オペレーションが成功し、リソースを使用する準備が整いました。
- FAILED – CreateParallelData オペレーションは失敗しました。

Amazon Translate コンソールまたは [UpdateParallelData](#) オペレーションを使用して、並列データリソースを更新します。これを行うと、Amazon Translate により次のようなイベントが送信されます。

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "Translate Parallel Data State Change",
```

```
"source": "aws.translate",
"account": "111122223333",
"time": "2017-04-22T03:31:47Z",
"region": "us-east-1",
"resources": [arn:aws:translate:us-east-1:111122223333:parallel-data/
ExampleParallelData],
"detail": {
  "operation": "UpdateParallelData",
  "name": "ExampleParallelData",
  "status": "STATUS",
  "latestUpdateAttemptStatus": "STATUS",
  "latestUpdateAttemptAt": "2017-04-22T03:31:47Z"
}
}
```

status 属性は、更新により置き換えられる旧バージョンの並列データリソースのステータスを示します。値は次のとおりです。

- ACTIVE — 旧バージョンは正常に作成または更新されました。
- FAILED — 旧バージョンの作成または更新に失敗しました。

latestUpdateAttemptStatus 属性は、更新により作成される新バージョンの並列データリソースのステータスを示します。値は次のとおりです。

- ACTIVE — UpdateParallelData オペレーションは成功し、更新されたリソースを使用する準備が整いました。
- FAILED – UpdateParallelData オペレーションは失敗しました。

Amazon Translate のコンプライアンス検証

サードパーティーの監査者は、複数のコンプライアンスプログラムの一環として Amazon Translate のセキュリティと AWS コンプライアンスを評価します。これには、PCI、FedRAMP、HIPAAなどが含まれます。を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「[AWS Artifact のレポートのダウンロード](#)」を参照してください。

Amazon Translate を使用する際のお客様のコンプライアンス責任は、お客様のデータの機密性や貴社のコンプライアンス目的、適用される法律および規制によって決まります。AWS では、コンプライアンスに役立つ以下のリソースを提供しています。

- [セキュリティとコンプライアンスのクイックスタートガイド](#) — これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、機密性とコンプライアンスに焦点を当てたベースライン環境を AWS にデプロイするためのステップを提供します。
- [HIPAA セキュリティとコンプライアンスのアーキテクチャに関するホワイトペーパー](#) — このホワイトペーパーでは、企業が AWS を使用して HIPAA 準拠のアプリケーションを作成する方法について説明します。
- [AWS コンプライアンスリソース](#) — このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- [AWS Config](#) — この AWS サービスは、リソース設定が社内プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。
- [AWS Security Hub](#) — この AWS サービスは、内のセキュリティ状態を包括的に把握し、セキュリティ業界標準とベストプラクティスへの準拠を確認するのに役立ちます。

特定のコンプライアンスプログラムの対象となる AWS サービスのリストについては、[AWS 「コンプライアンスプログラムによる対象範囲内のサービス」](#) を参照してください。一般的な情報については、「[AWS コンプライアンスプログラム](#)」を参照してください。

Amazon Translate の耐障害性

AWS グローバルインフラストラクチャは、AWS リージョンとアベイラビリティゾーンを中心に構築されています。AWS リージョンは、低レイテンシー、高スループット、および高度に冗長なネットワークで接続された、物理的に分離および分離された複数のアベイラビリティゾーンを提供します。アベイラビリティゾーンでは、アベイラビリティゾーン間で中断せずに、自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、およびスケーラビリティが優れています。

AWS リージョンとアベイラビリティゾーンの詳細については、[AWS 「グローバルインフラストラクチャ」](#) を参照してください。

Amazon Translate のインフラストラクチャセキュリティ

マネージドサービスである Amazon Translate は、ホワイトペーパー「[Amazon Web Services: セキュリティプロセスの概要](#)」に記載されている AWS グローバルネットワークセキュリティの手順で保護されています。

ネットワーク経由で Amazon Translate にアクセスするには、AWS が公開したAPI呼び出しを使用します。クライアントは TLS 1.2 以降をサポートしている必要があります。クライアントは、一時 Diffie-Hellman (PFS) や楕円曲線一時 Diffie-Hellman (DHE) など、完全な前方秘匿性を持つ暗号スイートもサポートする必要がありますECDHE。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

さらに、リクエストは、AWS Identity and Access Management (IAM) プリンシパルに関連付けられたアクセスキー ID とシークレットアクセスキーを使用して署名する必要があります。または、[AWS Security Token Service](#) (AWS STS) を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

Amazon Translate とインターフェイスVPCエンドポイント (AWS PrivateLink)

インターフェイスVPCエンドポイントを作成することで、VPCと Amazon Translate 間のプライベート接続を確立できます。インターフェイスエンドポイントは、インターネットゲートウェイ [AWS PrivateLink](#)、NATデバイス、VPN接続、または AWS Direct Connect 接続APIなしで Amazon Translate にプライベートにアクセスできるテクノロジーである [AWS PrivateLink](#) を利用しています。のインスタンスは、パブリック IP アドレスがVPCなくても Amazon Translate と通信できますAPIs。VPC と Amazon Translate 間のトラフィックは、Amazon ネットワークを離れません。

各インターフェイスエンドポイントは、サブネット内の 1 つ以上の [Elastic Network Interface](#) によって表されます。

詳細については、「Amazon VPCユーザーガイド」の「[インターフェイスVPCエンドポイント \(AWS PrivateLink \)](#)」を参照してください。

Amazon Translate VPCエンドポイントに関する考慮事項

Amazon Translate のインターフェイスVPCエンドポイントを設定する前に、「Amazon VPCユーザーガイド」の「[インターフェイスエンドポイントのプロパティと制限](#)」を確認してください。

Amazon Translate は、からのすべてのAPIアクションの呼び出しをサポートしていますVPC。

Amazon Translate のインターフェイスVPCエンドポイントの作成

Amazon Translate サービスのVPCエンドポイントは、Amazon VPCコンソールまたは AWS Command Line Interface () を使用して作成できますAWS CLI。詳細については、「Amazon VPC [ユーザーガイド](#)」の「[インターフェイスエンドポイントの作成](#)」を参照してください。

次のサービス名を使用して Amazon Translate のVPCエンドポイントを作成します。

- `com.amazonaws.region.translate`

エンドポイントDNSのプライベートを有効にすると、などのリージョンのデフォルトDNS名を使用して Amazon Translate にAPIリクエストを行うことができます `translate.us-east-1.amazonaws.com`。

詳細については、「[Amazon VPCユーザーガイド](#)」の「[インターフェイスエンドポイントを介したサービスへのアクセス](#)」を参照してください。

Amazon Translate のVPCエンドポイントポリシーの作成

Amazon Translate へのアクセスを制御するエンドポイントポリシーをVPCエンドポイントにアタッチできます。このポリシーでは、以下の情報を指定します。

- アクションを実行できるプリンシパル。
- 実行可能なアクション。
- アクションを実行できるリソース。

詳細については、「[Amazon VPCユーザーガイド](#)」のVPC「[エンドポイントを使用した サービスへのアクセスの制御](#)」を参照してください。

例: Amazon Translate リアルタイム翻訳アクションのVPCエンドポイントポリシー

以下は、Amazon Translate でリアルタイム翻訳を行うためのエンドポイントポリシーの例です。このポリシーは、エンドポイントにアタッチされると、すべてのリソースのすべてのプリンシパルに対して、登録されている Amazon Translate アクションへのアクセスを許可します。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "translate:TranslateText",
      ],
      "Resource": "*"
    }
  ]
}
```

```
}
```

例: Amazon Translate バッチ翻訳アクションのVPCエンドポイントポリシー

Amazon Translateのバッチ翻訳のためのエンドポイントポリシーの例を次に示します。このポリシーは、エンドポイントにアタッチされると、すべてのリソースのすべてのプリンシパルに対して、登録されている Amazon Translate アクションへのアクセスを許可します。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "translate:StartTextTranslationJob",
        "iam:PassRole"
      ],
      "Resource": "*"
    }
  ]
}
```

ガイドラインとクォータ

次のセクションには、Amazon Translate のガイドラインとクォータに関する情報が含まれています。

トピック

- [サポートされている AWS リージョン](#)
- [コンプライアンス](#)
- [スロットリング](#)
- [ガイドライン](#)
- [Service Quotas](#)

サポートされている AWS リージョン

Amazon Translate でサポートされる AWS リージョンのリストについては、「AWS 全般リファレンス」の「[Amazon Translate エンドポイントとクォータ](#)」を参照してください。

コンプライアンス

Amazon Translate コンプライアンスプログラムの詳細については、「[AWS コンプライアンス](#)」、「[AWS コンプライアンスプログラム](#)」、および「[AWS コンプライアンスプログラムによる対象範囲内のサービス](#)」を参照してください。

スロットリング

Amazon Translate は、お客様の運用トラフィックに対応するようにスケールされます。継続的なスロットリングが発生した場合は、[AWS Support](#)にお問い合わせください。

ガイドライン

Amazon Translate では、分析モデルの品質を継続的に向上させるため、お客様のデータを保存する場合があります。詳細については、「[Amazon Translate に関するよくある質問](#)」を参照してください。

[AWS Support にお問い合わせ](#)いただくと、お客様のデータを削除するとともに、お客様のアカウントに関連付けられているデータが今後保存されないようリクエストできます。ただし、お客様のデー

タを削除すると、翻訳の改善に役立つ独自のトレーニングデータも削除される可能性があるため、翻訳の品質が低下することがあります。

Service Quotas

Amazon Translate には次のサービスガイドラインとクォータがあります。

同期リアルタイム翻訳のクォータ

説明	制限
文字エンコーディング	UTF-8
最大入力テキスト	10,000 バイト
ドキュメントあたりの最大文字数	100,000
最大ドキュメントサイズ	100,000 bytes

非同期バッチ翻訳のクォータ

説明	制限
文字エンコーディング	UTF-8
ドキュメントあたりの最大文字数	1,000,000
ドキュメントあたりの最大サイズ	20 MB
1つのドキュメント内の翻訳可能なテキストの最大サイズ	1 MB
バッチジョブリクエストのターゲット言語の最大数	10
バッチ内のドキュメントの最大数	1,000,000
バッチ内の合計ドキュメントの最大サイズ	5 GB
同時バッチ翻訳ジョブの最大数	10
キューに入れられたバッチ変換ジョブの最大数	1,000

説明	制限
StartTextTranslationJob API アクションの 1 秒あたりのトランザクション	5
DescribeTextTranslationJob API アクションの 1 秒あたりのトランザクション	10
ListTextTranslationJobs API アクションの 1 秒あたりのトランザクション	10
StopTextTranslationJob API アクションのトランザクション/秒 (TPS)	5

カスタム用語のクォータ

説明	制限
最大カスタム用語ファイルサイズ	10 MB
各 AWS リージョンの AWS アカウントあたりのカスタム用語の最大数	100
カスタム用語ファイルあたりのターゲット言語の最大数	10
1 カスタム用語あたりのソーステキストとターゲットテキストの最大長	200 バイト
TranslateText または StartTextTranslationJob リクエストあたりの用語ファイルの最大数。	1
ImportTerminology API アクションの 1 秒あたりのトランザクション	5
GetTerminology API アクションの 1 秒あたりのトランザクション	10
ListTerminologies API アクションの 1 秒あたりのトランザクション	10

説明	制限
DeleteTerminology API アクションの 1 秒あたりのトランザクション	5

並列データクォータ

説明	制限
各リージョンの AWS アカウントあたりの並列データリソースの最大数	1,000
並列データ入力ファイルの最大サイズ	5 GB
並列データリソース内のソース言語の最大数	1
並列データ入力ファイル内の 1 つのセグメントまたはレコードの最大サイズ	1000 バイト
並列データリソースの作成または更新の同時操作の最大数	1
CreateParallelData API アクションの 1 秒あたりのトランザクション	5
GetParallelData API アクションの 1 秒あたりのトランザクション	10
ListParallelData API アクションの 1 秒あたりのトランザクション	10
UpdateParallelData API アクションの 1 秒あたりのトランザクション	5
DeleteParallelData API アクションの 1 秒あたりのトランザクション	5

Amazon Translate のドキュメント履歴

以下の表は、Amazon Translate の今回のリリースのドキュメント内容をまとめたものです。

変更	説明	日付
新機能:簡潔さ	Translate は、リアルタイムのテキスト翻訳の簡潔さをサポートするようになりました。簡潔にすると (簡潔でない翻訳出力に比べて) ほとんどの翻訳の翻訳出力の長さが短くなります。詳細については、「 Amazon Translate での簡潔さの使用 」を参照してください。	2023 年 10 月 31 日
リアルタイム翻訳へドキュメントを入力する際の言語の自動検出	リアルタイム翻訳 (コンソールまたは API) にドキュメントを入力するときに、言語の自動検出を使用できるようになりました。詳細については、「 リアルタイム翻訳 」を参照してください。	2023 年 8 月 3 日
リアルタイム翻訳への入力としての Word (.docx) ファイル	テキストファイルと HTML ファイルに加えて、.docx ファイルをリアルタイム翻訳 (コンソールまたは API) への入力として使用できるようになりました。詳細については、「 リアルタイム翻訳 」を参照してください。	2023 年 7 月 17 日
カスタム用語の強化	Translate は、翻訳の流畅さと正確さを向上させるカスタム用語機能の強化をサポートす	2023 年 6 月 30 日

リアルタイム翻訳への入力としてのテキストまたは HTML ファイル	テキストファイルまたは HTML ファイルをリアルタイム翻訳 (コンソールまたは API) への入力として使用できるようになりました。詳細については、「 カスタム用語による翻訳のカスタマイズ 」を参照してください。	2023 年 5 月 23 日
TranslateReadOnly ポリシーでは新しいアクションが許可されています。	Amazon Translate では、TranslateDocument TranslateReadOnly 管理ポリシー内のアクションが許可されるようになりました。詳細については、「 AWS 管理ポリシー: 」を参照してください TranslateReadOnly。	2023 年 5 月 23 日
Translate が、非同期バッチ処理で追加のリージョンをサポートするようになりました。	Translate が、非同期バッチ処理で追加のリージョンをサポートするようになりました。詳細については、「 Amazon Translate による非同期バッチ処理 」を参照してください。	2023 年 3 月 28 日
リアルタイム翻訳の入力サイズが増えました	リアルタイム翻訳で最大 10,000 文字を入力できるようになりました。詳細については、「 Amazon Translate の開始方法 (コンソール) 」を参照してください。	2022 年 12 月 16 日

[バッチモードにおけるネストされた入力フォルダのサポート](#)

ネストされた入力フォルダをバッチ翻訳ジョブに提供できるようになりました。詳細については、Amazon Translate での「[バッチ翻訳ジョブの実行](#)」を参照してください。

2022 年 11 月 18 日

[バッチモードにおける言語の自動検出のサポート](#)

バッチ翻訳ジョブのソース言語を自動検出できるようになりました。その結果、バッチ翻訳ジョブに異なるソース言語の文書を入力することが可能になりました。詳細については、Amazon Translate での「[バッチ翻訳ジョブの実行](#)」を参照してください。

2022 年 11 月 18 日

[複数のターゲット言語のサポート](#)

バッチ翻訳ジョブで複数のターゲット言語を指定できるようになりました。詳細については、Amazon Translate での「[バッチ翻訳ジョブの実行](#)」を参照してください。

2022 年 10 月 10 日

[タグのサポート](#)

Amazon Translate ParallelData でリソースにタグを付けたりカスタム用語集を作成したりできるようになりました。詳細については、「[Amazon Translate のリソースにタグを付ける](#)」を参照してください。

2022 年 10 月 6 日

[さらなる言語での形式のサポート](#)

Amazon Translate では、オランダ語、韓国語、メキシコスペイン語の翻訳形式レベルを設定できるようになりました。詳細については「[Amazon Translate の形式設定](#)」を参照してください。

2022 年 10 月 5 日

[API リファレンスを分離](#)

「Amazon Translate API リファレンス」は、「デベロッパーガイド」とは別のドキュメントになりました。詳細については、「[Amazon Translate API リファレンス](#)」を参照してください。

2022 年 8 月 25 日

[新機能](#)

翻訳出力の形式レベルを設定できるようになりました。詳細については「[Amazon Translate の形式設定](#)」を参照してください。

2022 年 2 月 22 日

[新機能](#)

翻訳出力内の不適切な単語やフレーズをマスクできるようになりました。詳細については「[Amazon Translate における不適切な言葉やフレーズのマスキング](#)」を参照してください。

2021 年 11 月 24 日

[AWS PrivateLink](#) のサポート

AWS PrivateLink を使用して VPC と Amazon Translate との間でプライベート接続を確立できるようになりました。詳細については、「[Amazon Translate とインターフェイス VPC エンドポイント \(AWS PrivateLink\)](#)」を参照してください。

2021 年 11 月 24 日

[並列データ更新](#)

Amazon Translate でサポートされている言語を使用した並列データリソースを作成できるようになりました。言語の 1 つとして英語を使用する必要はなくなりました。並列データの詳細については、「[並列データによる翻訳のカスタマイズ \(Active Custom Translation\)](#)」を参照してください。

2021 年 11 月 15 日

[カスタム用語の方向性](#)

任意の言語をソース言語またはターゲット言語とする多方向の用語集を作成できます。詳細については、「[カスタム用語の作成](#)」を参照してください。

2021 年 11 月 11 日

新しい言語

Amazon Translate では、アイルランド語、マラーティー語、ポルトガル語 (ポルトガル)、パンジャブ語がサポートされるようになりました。Amazon Translate でサポートされているすべての言語については、「[サポートされている言語と言語コード](#)」を参照してください。

2021 年 11 月 10 日

新しいカスタム暗号化設定

AWS Key Management Service で管理する独自のカスタマーマネージドキーを使用して、翻訳出力を暗号化できるようになりました。詳細については、「[バッチ翻訳ジョブの実行](#)」を参照してください。

2021 年 11 月 5 日

新たなファイル形式サポート

Amazon Translate では、非同期バッチ処理用として XML Localization Interchange File Format (XLIFF) ファイルがサポートされるようになりました。サポートされる形式については、「[サポートされるファイル形式](#)」を参照してください。

2021 年 6 月 9 日

[EventBridge インテグレーション](#)

Amazon Translate は、翻訳ジョブと parallel データリソースに影響する変更を通知するイベントを Amazon EventBridge に送信するようになりました。詳細については、「Amazon による [Amazon Translate イベントのモニタリング](#)」を参照してください EventBridge。

2021 年 6 月 4 日

[新しいクォータ](#)

Amazon Translate では、キューに入れられた最大 1000 個のバッチ翻訳ジョブがサポートされるようになりました。Amazon Translate のすべてのクォータについては、「[ガイドラインと制限](#)」を参照してください。

2021 年 4 月 23 日

[クォータの増加](#)

並列データ入力ファイルの最大サイズが 1 MB から 5 MB に増加しました。Amazon Translate のすべてのクォータについては、「[ガイドラインと制限](#)」を参照してください。

2021 年 3 月 31 日

新しい言語

Amazon Translate では現在、アルメニア語、カタロニア語、ペルシア語 (ペルシア語)、フィリピン語タガログ語、グジャラート語、ハイチクレオール語、アイスランド語、カンナダ語、カザフ語、リトアニア語、マケドニア語、マラヤーラム語、マルタ語、モンゴル語、シンハラ語、テルグ語、ウズベク語、ウェールズ語がサポートされています。Amazon Translate でサポートされているすべての言語については、「[サポートされている言語と言語コード](#)」を参照してください。

2020 年 11 月 23 日

新機能

並列データを使用してバッチ翻訳ジョブをカスタマイズできるようになりました。並列データはソーステキストの例とその翻訳例で構成されています。並列データを使用するジョブは、Active Custom Translation (アクティブカスタム翻訳) ジョブと呼ばれます。これらのジョブの実行中、Amazon Translate では並列データ内の例が反映されるように翻訳出力が調整されません。詳細については、「[並列データによる翻訳のカスタマイズ \(Active Custom Translation\)](#)」を参照してください。

2020 年 11 月 23 日

新たなファイル形式サポート

Amazon Translate では、非同期バッチ処理の入力として、Word 文書 (.docx)、PowerPoint プレゼンテーション (.pptx)、Excel ワークブック (.xlsx) の Office オープン XML ファイル形式がサポートされるようになりました。詳細については、「[バッチ翻訳ジョブの開始](#)」を参照してください。

2020 年 7 月 29 日

新しい言語

Amazon Translate で、スペイン語 (メキシコ) の翻訳がサポートされるようになりました。サポートされているすべての言語については、「[サポートされている言語と言語コード](#)」を参照してください。

2020 年 4 月 30 日

新しいリージョン

Amazon Translate では 欧州 (ロンドン) リージョンにおける非同期バッチ処理がサポートされています。非同期バッチ処理を利用できるすべての AWS リージョンについては、「[利用可能なリージョン](#)」を参照してください。

2020 年 4 月 20 日

新機能

Amazon Translate に非同期バッチ翻訳機能が追加されました。詳細については、「[非同期バッチ処理](#)」を参照してください。

2019 年 12 月 23 日

新しいリージョン

Amazon Translate では、アジアパシフィック (香港)、アジアパシフィック (シドニー)、欧州 (ロンドン)、欧州 (パリ)、欧州 (ストックホルム)、米国西部 (北カリフォルニア) の各リージョンのサポートが追加されました。Amazon Translate でサポートされている AWS リージョンの完全なリストについては、Amazon Web Services 全般リファレンスの「[AWS リージョン表](#)」または「[AWS のリージョンとエンドポイント](#)」を参照してください。

2019 年 11 月 25 日

新しい言語

Amazon Translate では、翻訳対象の新しい言語として、アフリカーンス語、アルバニア語、アムハラ語、アゼルバイジャン語、ベンガル語、ボスニア語、ブルガリア語、カナダフランス語、クロアチア語、ダリ語、エストニア語、グルジア語、ハウサ語、ラトビア語、パシュト語、セルビア語、スロバキア語、スロベニア語、ソマリア語、スワヒリ語、タガログ語、タミル語が追加されました。Amazon Translate で直接翻訳できる言語の組み合わせの一覧については、「[サポートされている言語](#)」を参照してください。

2019 年 11 月 25 日

新しい言語

Amazon Translate では、翻訳可能な言語として、ギリシャ語、ハンガリー語、ルーマニア語、タイ語、ウクライナ語、ウルドゥー語、およびベトナム語が追加されました。Amazon Translate で直接翻訳できる言語の組み合わせの一覧については、「[サポートされている言語](#)」を参照してください。

2019 年 10 月 3 日

新機能

Amazon Translate に [FedRAMP コンプライアンス](#) が追加されました。詳細については、「[コンプライアンス](#)」を参照してください。

2019 年 7 月 31 日

新機能

Amazon Translate に [SOC コンプライアンス](#) が追加されました。詳細については、「[コンプライアンス](#)」を参照してください。

2019 年 5 月 30 日

新しいリージョン

Amazon Translate では、アジアパシフィック (ムンバイ)、アジアパシフィック (シンガポール)、アジアパシフィック (東京)、およびカナダ (中部) の各リージョンのサポートが追加されました。Amazon Translate でサポートされている AWS リージョンの完全なリストについては、Amazon Web Services 全般リファレンスの「[AWS リージョン表](#)」または「[AWS のリージョンとエンドポイント](#)」を参照してください。

2019 年 5 月 8 日

新しい言語

Amazon Translate では、翻訳対象の新しい言語として、ヒンディー語、マレー語、ノルウェー語、ペルシャ語が追加されました。Amazon Translate で直接翻訳できる言語の組み合わせの一覧については、「[サポートされている言語](#)」を参照してください。

2019 年 5 月 6 日

新しいリージョン

Amazon Translate では、欧州 (フランクフルト) およびアジアパシフィック (ソウル) リージョンのサポートが追加されました。Amazon Translate でサポートされている AWS リージョンの完全なリストについては、Amazon Web Services 全般リファレンスの「[AWS リージョン表](#)」または「[AWS のリージョンとエンドポイント](#)」を参照してください。

2019 年 2 月 28 日

新機能

Amazon Translate に [PCI コンプライアンス](#) が追加されました。詳細については、「[コンプライアンス](#)」を参照してください。

2018 年 12 月 12 日

新機能

Amazon Translate では、翻訳中により高度な制御を可能にするため、4 つの新しい API とカスタム用語機能が追加されました。翻訳リクエストでカスタム用語を使用すると、ブランド名、キャラクター名、モデル名、およびその他の独自コンテンツを、標準的な翻訳やコンテキストとは無関係に、毎回目的どおりに翻訳することができます。詳細については、「[カスタム用語](#)」を参照してください。

2018 年 11 月 27 日

新しい言語

Amazon Translate では、デンマーク語、オランダ語、フィンランド語、ヘブライ語、インドネシア語、韓国語、ポーランド語、およびスウェーデン語でドキュメントを翻訳できるようになりました。Amazon Translate ではサポートされていない言語ペアの数を大幅に減らすことで直接翻訳を継続的に強化しています。Amazon Translate で直接翻訳できる言語の組み合わせについては、「[サポートされている言語](#)」を参照してください。

2018 年 11 月 20 日

新機能

Amazon Translate に英語以外のサポート対象言語間の直接翻訳が追加されました。Amazon Translate で直接翻訳できる言語の組み合わせについては、「[サポートされている言語](#)」を参照してください。

2018 年 10 月 29 日

新機能

Amazon Translate に [HIPAA](#) コンプライアンスが追加されました。詳細については、「[コンプライアンス](#)」を参照してください。

2018 年 10 月 25 日

新機能

Amazon Translate では、新たな翻訳言語として、繁体字中国語、チェコ語、イタリア語、日本語、ロシア語、トルコ語が追加されました。Amazon Translate でサポートされている言語のリストについては、「[サポートされている言語](#)」を参照してください。

2018 年 7 月 17 日

新機能

Amazon Translate にソース言語の自動検出へのサポートが追加されました。詳細については、「[Amazon Translate の仕組み](#)」を参照してください。

2018 年 4 月 4 日

新しいガイド

Amazon Translate デベロッパーガイドの初回リリースです。

2017 年 11 月 29 日

API リファレンス

Amazon Translate APIリファレンスが別のドキュメントになりました。詳細については、[Amazon Translate APIリファレンス](#)を参照してください。

AWS 用語集

AWS の最新の用語については、「AWS の用語集リファレンス」の「[AWS 用語集](#)」を参照してください。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。