Package ‘paws.machine.learning’

August 23, 2021

Title 'Amazon Web Services' Machine Learning Services

Version 0.1.12

Description Interface to 'Amazon Web Services' machine learning services, including 'SageMaker' managed machine learning service, natural language processing, speech recognition, translation, and more <https://aws.amazon.com/machine-learning/>.

License Apache License (>= 2.0)

URL https://github.com/paws-r/paws

BugReports https://github.com/paws-r/paws/issues

Imports paws.common (>= 0.3.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.1.1

Collate 'comprehend_service.R' 'comprehend_interfaces.R' 'comprehend_operations.R' 'comprehendmedical_service.R' 'comprehendmedical_interfaces.R' 'comprehendmedical_operations.R' 'lexmodelbuildingservice_service.R' 'lexmodelbuildingservice_interfaces.R' 'lexmodelbuildingservice_operations.R' 'lexruntimeservice_service.R' 'lexruntimeservice_interfaces.R' 'lexruntimeservice_operations.R' 'machinelearning_service.R' 'machinelearning_interfaces.R' 'machinelearning_operations.R' 'personalize_service.R' 'personalize_interfaces.R' 'personalize_operations.R' 'personalizeevents_service.R' 'personalizeevents_interfaces.R' 'personalizeevents_operations.R' 'personalizeruntime_service.R' 'personalizeruntime_interfaces.R' 'personalizeruntime_operations.R' 'polly_service.R' 'polly_interfaces.R' 'polly_operations.R' 'rekognition_service.R' 'rekognition_interfaces.R' 'rekognition_operations.R' 'sagemaker_service.R'

'sagemaker_interfaces.R' 'sagemaker_operations.R'
'sagemakerruntime_service.R' 'sagemakerruntime_interfaces.R'
'sagemakerruntime_operations.R' 'textract_service.R'
'textract_interfaces.R' 'textract_operations.R'
'transcribeservice_service.R' 'transcribeservice_interfaces.R'
'transcribeservice_operations.R' 'translate_service.R'
'translate_interfaces.R' 'translate_operations.R'

NeedsCompilation no

Author David Kretch [aut, cre],
Adam Banker [aut],
Amazon.com, Inc. [cph]

Maintainer David Kretch <`david.kretch@gmail.com`>

Repository CRAN

Date/Publication 2021-08-23 07:10:38 UTC

R topics documented:

`comprehend` *Amazon Comprehend*

Description

Amazon Comprehend is an AWS service for gaining insight into the content of documents. Use these actions to determine the topics contained in your documents, the topics they discuss, the predominant sentiment expressed in them, the predominant language used, and more.

Usage

```
comprehend(config = list())
```

Arguments

`config`	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- comprehend(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

batch_detect_dominant_language	Determines the dominant language of the input text for a batch of documents
batch_detect_entities	Inspects the text of a batch of documents for named entities and returns informa
batch_detect_key_phrases	Detects the key noun phrases found in a batch of documents
batch_detect_sentiment	Inspects a batch of documents and returns an inference of the prevailing sentim
batch_detect_syntax	Inspects the text of a batch of documents for the syntax and part of speech of th
classify_document	Creates a new document classification request to analyze a single document in r
create_document_classifier	Creates a new document classifier that you can use to categorize documents
create_endpoint	Creates a model-specific endpoint for synchronous inference for a previously tr
create_entity_recognizer	Creates an entity recognizer using submitted files
delete_document_classifier	Deletes a previously created document classifier
delete_endpoint	Deletes a model-specific endpoint for a previously-trained custom model
delete_entity_recognizer	Deletes an entity recognizer
describe_document_classification_job	Gets the properties associated with a document classification job
describe_document_classifier	Gets the properties associated with a document classifier
describe_dominant_language_detection_job	Gets the properties associated with a dominant language detection job
describe_endpoint	Gets the properties associated with a specific endpoint

describe_entities_detection_job	Gets the properties associated with an entities detection job
describe_entity_recognizer	Provides details about an entity recognizer including status, S3 buckets contain
describe_events_detection_job	Gets the status and details of an events detection job
describe_key_phrases_detection_job	Gets the properties associated with a key phrases detection job
describe_pii_entities_detection_job	Gets the properties associated with a PII entities detection job
describe_sentiment_detection_job	Gets the properties associated with a sentiment detection job
describe_topics_detection_job	Gets the properties associated with a topic detection job
detect_dominant_language	Determines the dominant language of the input text
detect_entities	Inspects text for named entities, and returns information about them
detect_key_phrases	Detects the key noun phrases found in the text
detect_pii_entities	Inspects the input text for entities that contain personally identifiable informatio
detect_sentiment	Inspects text and returns an inference of the prevailing sentiment (POSITIVE, N
detect_syntax	Inspects text for syntax and the part of speech of words in the document
list_document_classification_jobs	Gets a list of the documentation classification jobs that you have submitted
list_document_classifiers	Gets a list of the document classifiers that you have created
list_dominant_language_detection_jobs	Gets a list of the dominant language detection jobs that you have submitted
list_endpoints	Gets a list of all existing endpoints that you've created
list_entities_detection_jobs	Gets a list of the entity detection jobs that you have submitted
list_entity_recognizers	Gets a list of the properties of all entity recognizers that you created, including
list_events_detection_jobs	Gets a list of the events detection jobs that you have submitted
list_key_phrases_detection_jobs	Get a list of key phrase detection jobs that you have submitted
list_pii_entities_detection_jobs	Gets a list of the PII entity detection jobs that you have submitted
list_sentiment_detection_jobs	Gets a list of sentiment detection jobs that you have submitted
list_tags_for_resource	Lists all tags associated with a given Amazon Comprehend resource
list_topics_detection_jobs	Gets a list of the topic detection jobs that you have submitted
start_document_classification_job	Starts an asynchronous document classification job
start_dominant_language_detection_job	Starts an asynchronous dominant language detection job for a collection of doc
start_entities_detection_job	Starts an asynchronous entity detection job for a collection of documents
start_events_detection_job	Starts an asynchronous event detection job for a collection of documents
start_key_phrases_detection_job	Starts an asynchronous key phrase detection job for a collection of documents
start_pii_entities_detection_job	Starts an asynchronous PII entity detection job for a collection of documents
start_sentiment_detection_job	Starts an asynchronous sentiment detection job for a collection of documents
start_topics_detection_job	Starts an asynchronous topic detection job
stop_dominant_language_detection_job	Stops a dominant language detection job in progress
stop_entities_detection_job	Stops an entities detection job in progress
stop_events_detection_job	Stops an events detection job in progress
stop_key_phrases_detection_job	Stops a key phrases detection job in progress
stop_pii_entities_detection_job	Stops a PII entities detection job in progress
stop_sentiment_detection_job	Stops a sentiment detection job in progress
stop_training_document_classifier	Stops a document classifier training job while in progress
stop_training_entity_recognizer	Stops an entity recognizer training job while in progress
tag_resource	Associates a specific tag with an Amazon Comprehend resource
untag_resource	Removes a specific tag associated with an Amazon Comprehend resource
update_endpoint	Updates information about the specified endpoint

Examples

```
## Not run:
svc <- comprehend()
svc$batch_detect_dominant_language(
  Foo = 123
)

## End(Not run)
```

comprehendmedical *AWS Comprehend Medical*

Description

Amazon Comprehend Medical extracts structured information from unstructured clinical text. Use these actions to gain insight in your documents.

Usage

```
comprehendmedical(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like svc$operation(...), where svc is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- comprehendmedical(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

describe_entities_detection_v2_job	Gets the properties associated with a medical entities detection job
describe_icd10cm_inference_job	Gets the properties associated with an InferICD10CM job
describe_phi_detection_job	Gets the properties associated with a protected health information (PHI) detection job
describe_rx_norm_inference_job	Gets the properties associated with an InferRxNorm job
detect_entities	The DetectEntities operation is deprecated
detect_entities_v2	Inspects the clinical text for a variety of medical entities and returns specific information
detect_phi	Inspects the clinical text for protected health information (PHI) entities and returns the en
infer_icd10cm	InferICD10CM detects medical conditions as entities listed in a patient record and links
infer_rx_norm	InferRxNorm detects medications as entities listed in a patient record and links to the no
list_entities_detection_v2_jobs	Gets a list of medical entity detection jobs that you have submitted
list_icd10cm_inference_jobs	Gets a list of InferICD10CM jobs that you have submitted
list_phi_detection_jobs	Gets a list of protected health information (PHI) detection jobs that you have submitted
list_rx_norm_inference_jobs	Gets a list of InferRxNorm jobs that you have submitted
start_entities_detection_v2_job	Starts an asynchronous medical entity detection job for a collection of documents
start_icd10cm_inference_job	Starts an asynchronous job to detect medical conditions and link them to the ICD-10-CM
start_phi_detection_job	Starts an asynchronous job to detect protected health information (PHI)
start_rx_norm_inference_job	Starts an asynchronous job to detect medication entities and link them to the RxNorm on
stop_entities_detection_v2_job	Stops a medical entities detection job in progress
stop_icd10cm_inference_job	Stops an InferICD10CM inference job in progress
stop_phi_detection_job	Stops a protected health information (PHI) detection job in progress
stop_rx_norm_inference_job	Stops an InferRxNorm inference job in progress

Examples

```
## Not run:
svc <- comprehendmedical()
svc$describe_entities_detection_v2_job(
  Foo = 123
)

## End(Not run)
```

`lexmodelbuildingservice`

Amazon Lex Model Building Service

Description

Amazon Lex Build-Time Actions

Amazon Lex is an AWS service for building conversational voice and text interfaces. Use these actions to create, update, and delete conversational bots for new and existing client applications.

Usage

```
lexmodelbuildingservice(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like svc$operation(...), where svc is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- lexmodelbuildingservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

create_bot_version	Creates a new version of the bot based on the $LATEST version
create_intent_version	Creates a new version of an intent based on the $LATEST version of the intent
create_slot_type_version	Creates a new version of a slot type based on the $LATEST version of the specified slot typ
delete_bot	Deletes all versions of the bot, including the $LATEST version
delete_bot_alias	Deletes an alias for the specified bot
delete_bot_channel_association	Deletes the association between an Amazon Lex bot and a messaging platform
delete_bot_version	Deletes a specific version of a bot
delete_intent	Deletes all versions of the intent, including the $LATEST version
delete_intent_version	Deletes a specific version of an intent
delete_slot_type	Deletes all versions of the slot type, including the $LATEST version
delete_slot_type_version	Deletes a specific version of a slot type
delete_utterances	Deletes stored utterances
get_bot	Returns metadata information for a specific bot
get_bot_alias	Returns information about an Amazon Lex bot alias
get_bot_aliases	Returns a list of aliases for a specified Amazon Lex bot
get_bot_channel_association	Returns information about the association between an Amazon Lex bot and a messaging pla

get_bot_channel_associations	Returns a list of all of the channels associated with the specified bot
get_bots	Returns bot information as follows:
get_bot_versions	Gets information about all of the versions of a bot
get_builtin_intent	Returns information about a built-in intent
get_builtin_intents	Gets a list of built-in intents that meet the specified criteria
get_builtin_slot_types	Gets a list of built-in slot types that meet the specified criteria
get_export	Exports the contents of a Amazon Lex resource in a specified format
get_import	Gets information about an import job started with the StartImport operation
get_intent	Returns information about an intent
get_intents	Returns intent information as follows:
get_intent_versions	Gets information about all of the versions of an intent
get_slot_type	Returns information about a specific version of a slot type
get_slot_types	Returns slot type information as follows:
get_slot_type_versions	Gets information about all versions of a slot type
get_utterances_view	Use the GetUtterancesView operation to get information about the utterances that your user
list_tags_for_resource	Gets a list of tags associated with the specified resource
put_bot	Creates an Amazon Lex conversational bot or replaces an existing bot
put_bot_alias	Creates an alias for the specified version of the bot or replaces an alias for the specified bot
put_intent	Creates an intent or replaces an existing intent
put_slot_type	Creates a custom slot type or replaces an existing custom slot type
start_import	Starts a job to import a resource to Amazon Lex
tag_resource	Adds the specified tags to the specified resource
untag_resource	Removes tags from a bot, bot alias or bot channel

Examples

```
## Not run:
svc <- lexmodelbuildingservice()
# This example shows how to get configuration information for a bot.
svc$get_bot(
  name = "DocOrderPizza",
  versionOrAlias = "$LATEST"
)

## End(Not run)
```

lexruntimeservice *Amazon Lex Runtime Service*

Description

Amazon Lex provides both build and runtime endpoints. Each endpoint provides a set of operations (API). Your conversational bot uses the runtime API to understand user utterances (user input text or voice). For example, suppose a user says "I want pizza", your bot sends this input to Amazon Lex using the runtime API. Amazon Lex recognizes that the user request is for the OrderPizza intent (one of the intents defined in the bot). Then Amazon Lex engages in user conversation on behalf of the bot to elicit required information (slot values, such as pizza size and crust type), and then performs fulfillment activity (that you configured when you created the bot). You use the build-time API to create and manage your Amazon Lex bot. For a list of build-time operations, see the build-time API, .

Usage

```
lexruntimeservice(config = list())
```

Arguments

`config`	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- lexruntimeservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

delete_session	Removes session information for a specified bot, alias, and user ID
get_session	Returns session information for a specified bot, alias, and user ID
post_content	Sends user input (text or speech) to Amazon Lex
post_text	Sends user input to Amazon Lex
put_session	Creates a new session or modifies an existing session with an Amazon Lex bot

Examples

```
## Not run:
svc <- lexruntimeservice()
svc$delete_session(
  Foo = 123
)

## End(Not run)
```

---

machinelearning *Amazon Machine Learning*

Description

Definition of the public APIs exposed by Amazon Machine Learning

Usage

```
machinelearning(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like svc$operation(...), where svc is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- machinelearning(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

add_tags	Adds one or more tags to an object, up to a limit of 10
create_batch_prediction	Generates predictions for a group of observations
create_data_source_from_rds	Creates a DataSource object from an Amazon Relational Database Service (Amazon RDS
create_data_source_from_redshift	Creates a DataSource from a database hosted on an Amazon Redshift cluster
create_data_source_from_s3	Creates a DataSource object
create_evaluation	Creates a new Evaluation of an MLModel
create_ml_model	Creates a new MLModel using the DataSource and the recipe as information sources
create_realtime_endpoint	Creates a real-time endpoint for the MLModel
delete_batch_prediction	Assigns the DELETED status to a BatchPrediction, rendering it unusable
delete_data_source	Assigns the DELETED status to a DataSource, rendering it unusable
delete_evaluation	Assigns the DELETED status to an Evaluation, rendering it unusable
delete_ml_model	Assigns the DELETED status to an MLModel, rendering it unusable
delete_realtime_endpoint	Deletes a real time endpoint of an MLModel
delete_tags	Deletes the specified tags associated with an ML object
describe_batch_predictions	Returns a list of BatchPrediction operations that match the search criteria in the request
describe_data_sources	Returns a list of DataSource that match the search criteria in the request
describe_evaluations	Returns a list of DescribeEvaluations that match the search criteria in the request
describe_ml_models	Returns a list of MLModel that match the search criteria in the request
describe_tags	Describes one or more of the tags for your Amazon ML object
get_batch_prediction	Returns a BatchPrediction that includes detailed metadata, status, and data file informatio
get_data_source	Returns a DataSource that includes metadata and data file information, as well as the curr
get_evaluation	Returns an Evaluation that includes metadata as well as the current status of the Evaluatio
get_ml_model	Returns an MLModel that includes detailed metadata, data source information, and the cu
predict	Generates a prediction for the observation using the specified ML Model
update_batch_prediction	Updates the BatchPredictionName of a BatchPrediction
update_data_source	Updates the DataSourceName of a DataSource
update_evaluation	Updates the EvaluationName of an Evaluation
update_ml_model	Updates the MLModelName and the ScoreThreshold of an MLModel

Examples

```
## Not run:
svc <- machinelearning()
svc$add_tags(
  Foo = 123
)

## End(Not run)
```

`personalize` *Amazon Personalize*

Description

Amazon Personalize is a machine learning service that makes it easy to add individualized recommendations to customers.

Usage

```
personalize(config = list())
```

Arguments

`config`	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- personalize(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

create_batch_inference_job	Creates a batch inference job
create_campaign	Creates a campaign by deploying a solution version
create_dataset	Creates an empty dataset and adds it to the specified dataset group
create_dataset_group	Creates an empty dataset group
create_dataset_import_job	Creates a job that imports training data from your data source (an Amazon S3 bucket) to an
create_event_tracker	Creates an event tracker that you use when sending event data to the specified dataset grou
create_filter	Creates a recommendation filter
create_schema	Creates an Amazon Personalize schema from the specified schema string
create_solution	Creates the configuration for training a model
create_solution_version	Trains or retrains an active solution
delete_campaign	Removes a campaign by deleting the solution deployment
delete_dataset	Deletes a dataset
delete_dataset_group	Deletes a dataset group
delete_event_tracker	Deletes the event tracker
delete_filter	Deletes a filter
delete_schema	Deletes a schema

delete_solution	Deletes all versions of a solution and the Solution object itself
describe_algorithm	Describes the given algorithm
describe_batch_inference_job	Gets the properties of a batch inference job including name, Amazon Resource Name (ARI
describe_campaign	Describes the given campaign, including its status
describe_dataset	Describes the given dataset
describe_dataset_group	Describes the given dataset group
describe_dataset_import_job	Describes the dataset import job created by CreateDatasetImportJob, including the import j
describe_event_tracker	Describes an event tracker
describe_feature_transformation	Describes the given feature transformation
describe_filter	Describes a filter's properties
describe_recipe	Describes a recipe
describe_schema	Describes a schema
describe_solution	Describes a solution
describe_solution_version	Describes a specific version of a solution
get_solution_metrics	Gets the metrics for the specified solution version
list_batch_inference_jobs	Gets a list of the batch inference jobs that have been performed off of a solution version
list_campaigns	Returns a list of campaigns that use the given solution
list_dataset_groups	Returns a list of dataset groups
list_dataset_import_jobs	Returns a list of dataset import jobs that use the given dataset
list_datasets	Returns the list of datasets contained in the given dataset group
list_event_trackers	Returns the list of event trackers associated with the account
list_filters	Lists all filters that belong to a given dataset group
list_recipes	Returns a list of available recipes
list_schemas	Returns the list of schemas associated with the account
list_solutions	Returns a list of solutions that use the given dataset group
list_solution_versions	Returns a list of solution versions for the given solution
update_campaign	Updates a campaign by either deploying a new solution or changing the value of the campa

Examples

```
## Not run:
svc <- personalize()
svc$create_batch_inference_job(
  Foo = 123
)

## End(Not run)
```

---

personalizeevents *Amazon Personalize Events*

Description

Amazon Personalize can consume real-time user event data, such as *stream* or *click* data, and use it for model training either alone or combined with historical data. For more information see recording-events.

Usage

```
personalizeevents(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- personalizeevents(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

put_events	Records user interaction event data
put_items	Adds one or more items to an Items dataset
put_users	Adds one or more users to a Users dataset

Examples

```
## Not run:
svc <- personalizeevents()
svc$put_events(
  Foo = 123
)

## End(Not run)
```

personalizeruntime *Amazon Personalize Runtime*

Description

Amazon Personalize Runtime

Usage

```
personalizeruntime(config = list())
```

Arguments

`config`	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- personalizeruntime(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

get_personalized_ranking	Re-ranks a list of recommended items for the given user
get_recommendations	Returns a list of recommended items

Examples

```
## Not run:
svc <- personalizeruntime()
svc$get_personalized_ranking(
  Foo = 123
)

## End(Not run)
```

polly *Amazon Polly*

Description

Amazon Polly is a web service that makes it easy to synthesize speech from text.

The Amazon Polly service provides API operations for synthesizing high-quality speech from plain text and Speech Synthesis Markup Language (SSML), along with managing pronunciations lexicons that enable you to get the best results for your application domain.

Usage

```
polly(config = list())
```

Arguments

`config`	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- polly(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

delete_lexicon	Deletes the specified pronunciation lexicon stored in an AWS Region
describe_voices	Returns the list of voices that are available for use when requesting speech synthesis
get_lexicon	Returns the content of the specified pronunciation lexicon stored in an AWS Region
get_speech_synthesis_task	Retrieves a specific SpeechSynthesisTask object based on its TaskID
list_lexicons	Returns a list of pronunciation lexicons stored in an AWS Region
list_speech_synthesis_tasks	Returns a list of SpeechSynthesisTask objects ordered by their creation date
put_lexicon	Stores a pronunciation lexicon in an AWS Region
start_speech_synthesis_task	Allows the creation of an asynchronous synthesis task, by starting a new SpeechSynthesisTask
synthesize_speech	Synthesizes UTF-8 input, plain text or SSML, to a stream of bytes

Examples

```
## Not run:
svc <- polly()
# Deletes a specified pronunciation lexicon stored in an AWS Region.
svc$delete_lexicon(
  Name = "example"
)

## End(Not run)
```

rekognition *Amazon Rekognition*

Description

This is the Amazon Rekognition API reference.

Usage

```
rekognition(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- rekognition(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

compare_faces	Compares a face in the source input image with each of the 100 largest faces detected in the ta
create_collection	Creates a collection in an AWS Region
create_project	Creates a new Amazon Rekognition Custom Labels project
create_project_version	Creates a new version of a model and begins training
create_stream_processor	Creates an Amazon Rekognition stream processor that you can use to detect and recognize fac
delete_collection	Deletes the specified collection
delete_faces	Deletes faces from a collection
delete_project	Deletes an Amazon Rekognition Custom Labels project
delete_project_version	Deletes an Amazon Rekognition Custom Labels model
delete_stream_processor	Deletes the stream processor identified by Name
describe_collection	Describes the specified collection
describe_projects	Lists and gets information about your Amazon Rekognition Custom Labels projects
describe_project_versions	Lists and describes the models in an Amazon Rekognition Custom Labels project
describe_stream_processor	Provides information about a stream processor created by CreateStreamProcessor
detect_custom_labels	Detects custom labels in a supplied image by using an Amazon Rekognition Custom Labels m
detect_faces	Detects faces within an image that is provided as input
detect_labels	Detects instances of real-world entities within an image (JPEG or PNG) provided as input
detect_moderation_labels	Detects unsafe content in a specified JPEG or PNG format image
detect_protective_equipment	Detects Personal Protective Equipment (PPE) worn by people detected in an image
detect_text	Detects text in the input image and converts it into machine-readable text
get_celebrity_info	Gets the name and additional information about a celebrity based on his or her Amazon Rekog
get_celebrity_recognition	Gets the celebrity recognition results for a Amazon Rekognition Video analysis started by Star
get_content_moderation	Gets the unsafe content analysis results for a Amazon Rekognition Video analysis started by S
get_face_detection	Gets face detection results for a Amazon Rekognition Video analysis started by StartFaceDete

get_face_search	Gets the face search results for Amazon Rekognition Video face search started by StartFaceSea
get_label_detection	Gets the label detection results of a Amazon Rekognition Video analysis started by StartLabelI
get_person_tracking	Gets the path tracking results of a Amazon Rekognition Video analysis started by StartPersonT
get_segment_detection	Gets the segment detection results of a Amazon Rekognition Video analysis started by StartSeg
get_text_detection	Gets the text detection results of a Amazon Rekognition Video analysis started by StartTextDe
index_faces	Detects faces in the input image and adds them to the specified collection
list_collections	Returns list of collection IDs in your account
list_faces	Returns metadata for faces in the specified collection
list_stream_processors	Gets a list of stream processors that you have created with CreateStreamProcessor
recognize_celebrities	Returns an array of celebrities recognized in the input image
search_faces	For a given input face ID, searches for matching faces in the collection the face belongs to
search_faces_by_image	For a given input image, first detects the largest face in the image, and then searches the specif
start_celebrity_recognition	Starts asynchronous recognition of celebrities in a stored video
start_content_moderation	Starts asynchronous detection of unsafe content in a stored video
start_face_detection	Starts asynchronous detection of faces in a stored video
start_face_search	Starts the asynchronous search for faces in a collection that match the faces of persons detecte
start_label_detection	Starts asynchronous detection of labels in a stored video
start_person_tracking	Starts the asynchronous tracking of a person's path in a stored video
start_project_version	Starts the running of the version of a model
start_segment_detection	Starts asynchronous detection of segment detection in a stored video
start_stream_processor	Starts processing a stream processor
start_text_detection	Starts asynchronous detection of text in a stored video
stop_project_version	Stops a running model
stop_stream_processor	Stops a running stream processor that was created by CreateStreamProcessor

Examples

```
## Not run:
svc <- rekognition()
# This operation compares the largest face detected in the source image
# with each face detected in the target image.
svc$compare_faces(
  SimilarityThreshold = 90L,
  SourceImage = list(
    S3Object = list(
      Bucket = "mybucket",
      Name = "mysourceimage"
    )
  ),
  TargetImage = list(
    S3Object = list(
      Bucket = "mybucket",
      Name = "mytargetimage"
    )
  )
)

## End(Not run)
```

sagemaker *Amazon SageMaker Service*

Description

Provides APIs for creating and managing Amazon SageMaker resources.

Other Resources:

- Amazon SageMaker Developer Guide
- Amazon Augmented AI Runtime API Reference

Usage

```
sagemaker(config = list())
```

Arguments

`config`	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- sagemaker(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

add_association	Creates an association between the source and the destination
add_tags	Adds or overwrites one or more tags for the specified Amazon SageMal
associate_trial_component	Associates a trial component with a trial
create_action	Creates an action
create_algorithm	Create a machine learning algorithm that you can use in Amazon SageM
create_app	Creates a running App for the specified UserProfile
create_app_image_config	Creates a configuration for running a SageMaker image as a KernelGate
create_artifact	Creates an artifact
create_auto_ml_job	Creates an Autopilot job
create_code_repository	Creates a Git repository as a resource in your Amazon SageMaker acco
create_compilation_job	Starts a model compilation job
create_context	Creates a context
create_data_quality_job_definition	Creates a definition for a job that monitors data quality and drift
create_device_fleet	Creates a device fleet
create_domain	Creates a Domain used by Amazon SageMaker Studio
create_edge_packaging_job	Starts a SageMaker Edge Manager model packaging job
create_endpoint	Creates an endpoint using the endpoint configuration specified in the re
create_endpoint_config	Creates an endpoint configuration that Amazon SageMaker hosting serv
create_experiment	Creates an SageMaker experiment
create_feature_group	Create a new FeatureGroup
create_flow_definition	Creates a flow definition
create_human_task_ui	Defines the settings you will use for the human review workflow user in
create_hyper_parameter_tuning_job	Starts a hyperparameter tuning job
create_image	Creates a custom SageMaker image
create_image_version	Creates a version of the SageMaker image specified by ImageName
create_labeling_job	Creates a job that uses workers to label the data objects in your input da
create_model	Creates a model in Amazon SageMaker
create_model_bias_job_definition	Creates the definition for a model bias job
create_model_explainability_job_definition	Creates the definition for a model explainability job
create_model_package	Creates a model package that you can use to create Amazon SageMaker
create_model_package_group	Creates a model group
create_model_quality_job_definition	Creates a definition for a job that monitors model quality and drift
create_monitoring_schedule	Creates a schedule that regularly starts Amazon SageMaker Processing
create_notebook_instance	Creates an Amazon SageMaker notebook instance
create_notebook_instance_lifecycle_config	Creates a lifecycle configuration that you can associate with a notebook
create_pipeline	Creates a pipeline using a JSON pipeline definition
create_presigned_domain_url	Creates a URL for a specified UserProfile in a Domain
create_presigned_notebook_instance_url	Returns a URL that you can use to connect to the Jupyter server from a
create_processing_job	Creates a processing job
create_project	Creates a machine learning (ML) project that can contain one or more t
create_training_job	Starts a model training job
create_transform_job	Starts a transform job
create_trial	Creates an Amazon SageMaker trial
create_trial_component	Creates a trial component, which is a stage of a machine learning trial
create_user_profile	Creates a user profile
create_workforce	Use this operation to create a workforce

create_workteam	Creates a new work team for labeling your data
delete_action	Deletes an action
delete_algorithm	Removes the specified algorithm from your account
delete_app	Used to stop and delete an app
delete_app_image_config	Deletes an AppImageConfig
delete_artifact	Deletes an artifact
delete_association	Deletes an association
delete_code_repository	Deletes the specified Git repository from your account
delete_context	Deletes an context
delete_data_quality_job_definition	Deletes a data quality monitoring job definition
delete_device_fleet	Deletes a fleet
delete_domain	Used to delete a domain
delete_endpoint	Deletes an endpoint
delete_endpoint_config	Deletes an endpoint configuration
delete_experiment	Deletes an Amazon SageMaker experiment
delete_feature_group	Delete the FeatureGroup and any data that was written to the OnlineSto
delete_flow_definition	Deletes the specified flow definition
delete_human_task_ui	Use this operation to delete a human task user interface (worker task ter
delete_image	Deletes a SageMaker image and all versions of the image
delete_image_version	Deletes a version of a SageMaker image
delete_model	Deletes a model
delete_model_bias_job_definition	Deletes an Amazon SageMaker model bias job definition
delete_model_explainability_job_definition	Deletes an Amazon SageMaker model explainability job definition
delete_model_package	Deletes a model package
delete_model_package_group	Deletes the specified model group
delete_model_package_group_policy	Deletes a model group resource policy
delete_model_quality_job_definition	Deletes the secified model quality monitoring job definition
delete_monitoring_schedule	Deletes a monitoring schedule
delete_notebook_instance	Deletes an Amazon SageMaker notebook instance
delete_notebook_instance_lifecycle_config	Deletes a notebook instance lifecycle configuration
delete_pipeline	Deletes a pipeline if there are no in-progress executions
delete_project	Delete the specified project
delete_tags	Deletes the specified tags from an Amazon SageMaker resource
delete_trial	Deletes the specified trial
delete_trial_component	Deletes the specified trial component
delete_user_profile	Deletes a user profile
delete_workforce	Use this operation to delete a workforce
delete_workteam	Deletes an existing work team
deregister_devices	Deregisters the specified devices
describe_action	Describes an action
describe_algorithm	Returns a description of the specified algorithm that is in your account
describe_app	Describes the app
describe_app_image_config	Describes an AppImageConfig
describe_artifact	Describes an artifact
describe_auto_ml_job	Returns information about an Amazon SageMaker job
describe_code_repository	Gets details about the specified Git repository
describe_compilation_job	Returns information about a model compilation job
describe_context	Describes a context

describe_data_quality_job_definition	Gets the details of a data quality monitoring job definition
describe_device	Describes the device
describe_device_fleet	A description of the fleet the device belongs to
describe_domain	The description of the domain
describe_edge_packaging_job	A description of edge packaging jobs
describe_endpoint	Returns the description of an endpoint
describe_endpoint_config	Returns the description of an endpoint configuration created using the C
describe_experiment	Provides a list of an experiment's properties
describe_feature_group	Use this operation to describe a FeatureGroup
describe_flow_definition	Returns information about the specified flow definition
describe_human_task_ui	Returns information about the requested human task user interface (wor
describe_hyper_parameter_tuning_job	Gets a description of a hyperparameter tuning job
describe_image	Describes a SageMaker image
describe_image_version	Describes a version of a SageMaker image
describe_labeling_job	Gets information about a labeling job
describe_model	Describes a model that you created using the CreateModel API
describe_model_bias_job_definition	Returns a description of a model bias job definition
describe_model_explainability_job_definition	Returns a description of a model explainability job definition
describe_model_package	Returns a description of the specified model package, which is used to c
describe_model_package_group	Gets a description for the specified model group
describe_model_quality_job_definition	Returns a description of a model quality job definition
describe_monitoring_schedule	Describes the schedule for a monitoring job
describe_notebook_instance	Returns information about a notebook instance
describe_notebook_instance_lifecycle_config	Returns a description of a notebook instance lifecycle configuration
describe_pipeline	Describes the details of a pipeline
describe_pipeline_definition_for_execution	Describes the details of an execution's pipeline definition
describe_pipeline_execution	Describes the details of a pipeline execution
describe_processing_job	Returns a description of a processing job
describe_project	Describes the details of a project
describe_subscribed_workteam	Gets information about a work team provided by a vendor
describe_training_job	Returns information about a training job
describe_transform_job	Returns information about a transform job
describe_trial	Provides a list of a trial's properties
describe_trial_component	Provides a list of a trials component's properties
describe_user_profile	Describes a user profile
describe_workforce	Lists private workforce information, including workforce name, Amazo
describe_workteam	Gets information about a specific work team
disable_sagemaker_servicecatalog_portfolio	Disables using Service Catalog in SageMaker
disassociate_trial_component	Disassociates a trial component from a trial
enable_sagemaker_servicecatalog_portfolio	Enables using Service Catalog in SageMaker
get_device_fleet_report	Describes a fleet
get_model_package_group_policy	Gets a resource policy that manages access for a model group
get_sagemaker_servicecatalog_portfolio_status	Gets the status of Service Catalog in SageMaker
get_search_suggestions	An auto-complete API for the search functionality in the Amazon Sage
list_actions	Lists the actions in your account and their properties
list_algorithms	Lists the machine learning algorithms that have been created
list_app_image_configs	Lists the AppImageConfigs in your account and their properties
list_apps	Lists apps

list_artifacts	Lists the artifacts in your account and their properties
list_associations	Lists the associations in your account and their properties
list_auto_ml_jobs	Request a list of jobs
list_candidates_for_auto_ml_job	List the Candidates created for the job
list_code_repositories	Gets a list of the Git repositories in your account
list_compilation_jobs	Lists model compilation jobs that satisfy various filters
list_contexts	Lists the contexts in your account and their properties
list_data_quality_job_definitions	Lists the data quality job definitions in your account
list_device_fleets	Returns a list of devices in the fleet
list_devices	A list of devices
list_domains	Lists the domains
list_edge_packaging_jobs	Returns a list of edge packaging jobs
list_endpoint_configs	Lists endpoint configurations
list_endpoints	Lists endpoints
list_experiments	Lists all the experiments in your account
list_feature_groups	List FeatureGroups based on given filter and order
list_flow_definitions	Returns information about the flow definitions in your account
list_human_task_uis	Returns information about the human task user interfaces in your accou
list_hyper_parameter_tuning_jobs	Gets a list of HyperParameterTuningJobSummary objects that describe
list_images	Lists the images in your account and their properties
list_image_versions	Lists the versions of a specified image and their properties
list_labeling_jobs	Gets a list of labeling jobs
list_labeling_jobs_for_workteam	Gets a list of labeling jobs assigned to a specified work team
list_model_bias_job_definitions	Lists model bias jobs definitions that satisfy various filters
list_model_explainability_job_definitions	Lists model explainability job definitions that satisfy various filters
list_model_package_groups	Gets a list of the model groups in your AWS account
list_model_packages	Lists the model packages that have been created
list_model_quality_job_definitions	Gets a list of model quality monitoring job definitions in your account
list_models	Lists models created with the CreateModel API
list_monitoring_executions	Returns list of all monitoring job executions
list_monitoring_schedules	Returns list of all monitoring schedules
list_notebook_instance_lifecycle_configs	Lists notebook instance lifestyle configurations created with the Create
list_notebook_instances	Returns a list of the Amazon SageMaker notebook instances in the requ
list_pipeline_executions	Gets a list of the pipeline executions
list_pipeline_execution_steps	Gets a list of PipeLineExecutionStep objects
list_pipeline_parameters_for_execution	Gets a list of parameters for a pipeline execution
list_pipelines	Gets a list of pipelines
list_processing_jobs	Lists processing jobs that satisfy various filters
list_projects	Gets a list of the projects in an AWS account
list_subscribed_workteams	Gets a list of the work teams that you are subscribed to in the AWS Mar
list_tags	Returns the tags for the specified Amazon SageMaker resource
list_training_jobs	Lists training jobs
list_training_jobs_for_hyper_parameter_tuning_job	Gets a list of TrainingJobSummary objects that describe the training job
list_transform_jobs	Lists transform jobs
list_trial_components	Lists the trial components in your account
list_trials	Lists the trials in your account
list_user_profiles	Lists user profiles
list_workforces	Use this operation to list all private and vendor workforces in an AWS

list_workteams	Gets a list of private work teams that you have defined in a region
put_model_package_group_policy	Adds a resouce policy to control access to a model group
register_devices	Register devices
render_ui_template	Renders the UI template so that you can preview the worker's experienc
search	Finds Amazon SageMaker resources that match a search query
start_monitoring_schedule	Starts a previously stopped monitoring schedule
start_notebook_instance	Launches an ML compute instance with the latest version of the librarie
start_pipeline_execution	Starts a pipeline execution
stop_auto_ml_job	A method for forcing the termination of a running job
stop_compilation_job	Stops a model compilation job
stop_edge_packaging_job	Request to stop an edge packaging job
stop_hyper_parameter_tuning_job	Stops a running hyperparameter tuning job and all running training jobs
stop_labeling_job	Stops a running labeling job
stop_monitoring_schedule	Stops a previously started monitoring schedule
stop_notebook_instance	Terminates the ML compute instance
stop_pipeline_execution	Stops a pipeline execution
stop_processing_job	Stops a processing job
stop_training_job	Stops a training job
stop_transform_job	Stops a transform job
update_action	Updates an action
update_app_image_config	Updates the properties of an AppImageConfig
update_artifact	Updates an artifact
update_code_repository	Updates the specified Git repository with the specified values
update_context	Updates a context
update_device_fleet	Updates a fleet of devices
update_devices	Updates one or more devices in a fleet
update_domain	Updates the default settings for new user profiles in the domain
update_endpoint	Deploys the new EndpointConfig specified in the request, switches to u
update_endpoint_weights_and_capacities	Updates variant weight of one or more variants associated with an exist
update_experiment	Adds, updates, or removes the description of an experiment
update_image	Updates the properties of a SageMaker image
update_model_package	Updates a versioned model
update_monitoring_schedule	Updates a previously created schedule
update_notebook_instance	Updates a notebook instance
update_notebook_instance_lifecycle_config	Updates a notebook instance lifecycle configuration created with the Cr
update_pipeline	Updates a pipeline
update_pipeline_execution	Updates a pipeline execution
update_training_job	Update a model training job to request a new Debugger profiling config
update_trial	Updates the display name of a trial
update_trial_component	Updates one or more properties of a trial component
update_user_profile	Updates a user profile
update_workforce	Use this operation to update your workforce
update_workteam	Updates an existing work team with new member definitions or descrip

Examples

```
## Not run:
```

```
svc <- sagemaker()
svc$add_association(
  Foo = 123
)

## End(Not run)
```

sagemakerruntime *Amazon SageMaker Runtime*

Description

The Amazon SageMaker runtime API.

Usage

```
sagemakerruntime(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like svc$operation(...), where svc is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- sagemakerruntime(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

invoke_endpoint	After you deploy a model into production using Amazon SageMaker hosting services, your client applicatio

Examples

```
## Not run:
svc <- sagemakerruntime()
svc$invoke_endpoint(
  Foo = 123
)

## End(Not run)
```

textract *Amazon Textract*

Description

Amazon Textract detects and analyzes text in documents and converts it into machine-readable text. This is the API reference documentation for Amazon Textract.

Usage

```
textract(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like svc$operation(...), where svc is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- textract(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

analyze_document	Analyzes an input document for relationships between detected items
detect_document_text	Detects text in the input document
get_document_analysis	Gets the results for an Amazon Textract asynchronous operation that analyzes text in a docu
get_document_text_detection	Gets the results for an Amazon Textract asynchronous operation that detects text in a docum
start_document_analysis	Starts the asynchronous analysis of an input document for relationships between detected ite
start_document_text_detection	Starts the asynchronous detection of text in a document

Examples

```
## Not run:
svc <- textract()
svc$analyze_document(
  Foo = 123
)

## End(Not run)
```

transcribeservice *Amazon Transcribe Service*

Description

Operations and objects for transcribing speech to text.

Usage

```
transcribeservice(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- transcribeservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

create_language_model	Creates a new custom language model
create_medical_vocabulary	Creates a new custom vocabulary that you can use to change how Amazon Transcribe Me
create_vocabulary	Creates a new custom vocabulary that you can use to change the way Amazon Transcribe
create_vocabulary_filter	Creates a new vocabulary filter that you can use to filter words, such as profane words, fro
delete_language_model	Deletes a custom language model using its name
delete_medical_transcription_job	Deletes a transcription job generated by Amazon Transcribe Medical and any related infor
delete_medical_vocabulary	Deletes a vocabulary from Amazon Transcribe Medical
delete_transcription_job	Deletes a previously submitted transcription job along with any other generated results su
delete_vocabulary	Deletes a vocabulary from Amazon Transcribe
delete_vocabulary_filter	Removes a vocabulary filter
describe_language_model	Gets information about a single custom language model
get_medical_transcription_job	Returns information about a transcription job from Amazon Transcribe Medical
get_medical_vocabulary	Retrieves information about a medical vocabulary
get_transcription_job	Returns information about a transcription job
get_vocabulary	Gets information about a vocabulary
get_vocabulary_filter	Returns information about a vocabulary filter
list_language_models	Provides more information about the custom language models you've created
list_medical_transcription_jobs	Lists medical transcription jobs with a specified status or substring that matches their nam
list_medical_vocabularies	Returns a list of vocabularies that match the specified criteria
list_transcription_jobs	Lists transcription jobs with the specified status
list_vocabularies	Returns a list of vocabularies that match the specified criteria
list_vocabulary_filters	Gets information about vocabulary filters
start_medical_transcription_job	Starts a batch job to transcribe medical speech to text
start_transcription_job	Starts an asynchronous job to transcribe speech to text
update_medical_vocabulary	Updates a vocabulary with new values that you provide in a different text file from the one
update_vocabulary	Updates an existing vocabulary with new values
update_vocabulary_filter	Updates a vocabulary filter with a new list of filtered words

Examples

```
## Not run:
svc <- transcribeservice()
svc$create_language_model(
  Foo = 123
)

## End(Not run)
```

translate *Amazon Translate*

Description

Provides translation between one source language and another of the same set of languages.

Usage

```
translate(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- translate(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

create_parallel_data	Creates a parallel data resource in Amazon Translate by importing an input file from Amazon
delete_parallel_data	Deletes a parallel data resource in Amazon Translate
delete_terminology	A synchronous action that deletes a custom terminology
describe_text_translation_job	Gets the properties associated with an asycnhronous batch translation job including name, ID,
get_parallel_data	Provides information about a parallel data resource
get_terminology	Retrieves a custom terminology
import_terminology	Creates or updates a custom terminology, depending on whether or not one already exists for t
list_parallel_data	Provides a list of your parallel data resources in Amazon Translate
list_terminologies	Provides a list of custom terminologies associated with your account
list_text_translation_jobs	Gets a list of the batch translation jobs that you have submitted
start_text_translation_job	Starts an asynchronous batch translation job
stop_text_translation_job	Stops an asynchronous batch translation job that is in progress
translate_text	Translates input text from the source language to the target language
update_parallel_data	Updates a previously created parallel data resource by importing a new input file from Amazo

Examples

```
## Not run:
svc <- translate()
svc$create_parallel_data(
  Foo = 123
)

## End(Not run)
```

Index