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'
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.
**comprehend**

**Usage**

```r
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**

```r
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 information about them
- `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 sentiment, POSITIVE, NEUTRAL, MIXED, or NEGATIVE, in each one
- `batch_detect_syntax`: Inspects the text of a batch of documents for the syntax and part of speech of the words in the document and returns information about them
- `classify_document`: Creates a new document classification request to analyze a single document in real-time, using a previously created and trained custom model and an endpoint
- `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 trained custom model
- `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
comprehend

describe_entities_detection_job

describe_entity_recognizer

describe_events_detection_job

describe_key_phrases_detection_job

describe_pii_entities_detection_job

describe_sentiment_detection_job

describe_topics_detection_job

detect_dominant_language

detect_entities

detect_key_phrases

detect_pii_entities

detect_sentiment

detect_syntax

list_document_classification_jobs

list_document_classifiers

list_dominant_language_detection_jobs

list_endpoints

list_entities_detection_jobs

list_entity_recognizers

list_events_detection_jobs

list_key_phrases_detection_jobs

list_pii_entities_detection_jobs

list_sentiment_detection_jobs

list_tags_for_resource

list_topics_detection_jobs

start_document_classification_job

start_dominant_language_detection_job

start_entities_detection_job

start_events_detection_job

start_key_phrases_detection_job

start_pii_entities_detection_job

start_sentiment_detection_job

start_topics_detection_job

stop_dominant_language_detection_job

stop_entities_detection_job

stop_events_detection_job

stop_key_phrases_detection_job

stop_pii_entities_detection_job

stop_sentiment_detection_job

stop_training_document_classifier

stop_training_entity_recognizer

tag_resource

untag_resource

update_endpoint

Gets the properties associated with an entities detection job
Provides details about an entity recognizer including status, S3 buckets containing training data
Gets the status and details of an events detection job
Gets the properties associated with a key phrases detection job
Gets the properties associated with a PII entities detection job
Gets the properties associated with a sentiment detection job
Gets the properties associated with a topic detection job
Determines the dominant language of the input text
Inspects text for named entities, and returns information about them
Detects the key noun phrases found in the text
Inspects the input text for entities that contain personally identifiable information
Inspects text and returns an inference of the prevailing sentiment (POSITIVE, MIXED, NEGATIVE)
Inspects text and returns the part of speech of words in the document
Gets a list of the documentation classification jobs that you have submitted
Gets a list of the document classifiers that you have created
Gets a list of the dominant language detection jobs that you have submitted
Gets a list of all existing endpoints that you’ve created
Gets a list of the entity detection jobs that you have submitted
Gets a list of the properties of all entity recognizers that you created, including recognizers currently in training
Gets a list of the events detection jobs that you have submitted
Get a list of key phrase detection jobs that you have submitted
Get a list of PII entity detection jobs that you have submitted
Get a list of sentiment detection jobs that you have submitted
Lists all tags associated with a given Amazon Comprehend resource
Gets a list of the topic detection jobs that you have submitted
Starts an asynchronous document classification job
Starts an asynchronous dominant language detection job for a collection of documents
Starts an asynchronous entity detection job for a collection of documents
Starts an asynchronous event detection job for a collection of documents
Starts an asynchronous key phrase detection job for a collection of documents
Starts an asynchronous PII entity detection job for a collection of documents
Starts an asynchronous sentiment detection job for a collection of documents
Starts an asynchronous topic detection job
Starts an asynchronous topic detection job
Stops a dominant language detection job in progress
Stops an entities detection job in progress
Stops an events detection job in progress
Stops a key phrases detection job in progress
Stops a PII entities detection job in progress
Stops a sentiment detection job in progress
Stops a document classifier training job while in progress
Stops an entity recognizer training job while in progress
Associates a specific tag with an Amazon Comprehend resource
Removes a specific tag associated with an Amazon Comprehend resource
Updates information about the specified endpoint
comprehendmedical

Examples

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

## End(Not run)
```

### Description

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

### Usage

```r
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

```r
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
describe_icd10cm_inference_job
describe_phi_detection_job
describe_rx_norm_inference_job
detect_entities
detect_entities_v2
detect_phi
infer_icd10cm
infer_rx_norm
list_entities_detection_v2_jobs
list_icd10cm_inference_jobs
list_phi_detection_jobs
list_rx_norm_inference_jobs
start_entities_detection_v2_job
start_icd10cm_inference_job
start_phi_detection_job
start_rx_norm_inference_job
stop_entities_detection_v2_job
stop_icd10cm_inference_job
stop_phi_detection_job
stop_rx_norm_inference_job

- Gets the properties associated with a medical entities detection job
- Gets the properties associated with an InferICD10CM job
- Gets the properties associated with a protected health information (PHI) detection job
- Gets the properties associated with an InferRxNorm job
- The DetectEntities operation is deprecated
- Inspects the clinical text for a variety of medical entities and returns specific information about them such as entity category, location, and confidence score on that information
- Inspects the clinical text for protected health information (PHI) entities and returns the entity category, location, and confidence score for each entity
- InferICD10CM detects medical conditions as entities listed in a patient record and links those entities to standardized concept identifiers in the ICD-10-CM knowledge base from the Centers for Disease Control
- InferRxNorm detects medications as entities listed in a patient record and links them to the normalized concept identifiers in the RxNorm database from the National Library of Medicine
- Gets a list of medical entity detection jobs that you have submitted
- Gets a list of InferICD10CM jobs that you have submitted
- Gets a list of protected health information (PHI) detection jobs that you have submitted
- Gets a list of InferRxNorm jobs that you have submitted
- Starts an asynchronous medical entity detection job for a collection of documents
- Starts an asynchronous job to detect medical conditions and link them to the ICD-10-CM ontology
- Starts an asynchronous job to detect protected health information (PHI)
- Starts an asynchronous job to detect medication entities and link them to the RxNorm ontology
- Stops a medical entities detection job in progress
- Stops an InferICD10CM inference job in progress
- Stops a protected health information (PHI) detection job in progress
- Stops an InferRxNorm inference job in progress

Examples

```r
## 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

```r
csv <- lexmodelbuildingservice(
  config = list(
    credentials = list(
      cred = 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 type
- `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
- `deleteUtterances` 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 platform
get_bot_channel_associations
get_bots
get_bot_versions
get_builtin_intent
get_builtin_intents
get_builtin_slot_types
get_export
get_import
get_intent
get_intents
get_intent_versions
get_slot_type
get_slot_types
get_slot_type_versions
get_utterances_view
list_tags_for_resource
put_bot
put_bot_alias
put_intent
put_slot_type
start_import
tag_resource
untag_resource

Returns a list of all of the channels associated with the specified bot
Returns bot information as follows:
Gets information about all of the versions of a bot
Returns information about a built-in intent
Gets a list of built-in intents that meet the specified criteria
Gets a list of built-in slot types that meet the specified criteria
Exports the contents of an Amazon Lex resource in a specified format
Gets information about an import job started with the StartImport operation
Returns information about an intent
Gets information about all of the versions of an intent
Returns information about a specific version of a slot type
Gets information about all versions of a slot type
Use the GetUtterancesView operation to get information about the utterances that your users
gets a list of tags associated with the specified resource
Creates an Amazon Lex conversational bot or replaces an existing bot
Creates an alias for the specified version of the bot or replaces an alias for the specified bot
Creates an intent or replaces an existing intent
Creates a custom slot type or replaces an existing custom slot type
Starts a job to import a resource to Amazon Lex
Adds the specified tags to the specified 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)

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

cfg | 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

```r
## 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

```r
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

```r
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
### Examples

```r
## 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 Amazon Personalize dataset
create_event_tracker               Creates an event tracker that you use when sending event data to the specified dataset group
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 retraining 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
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>delete_solution</td>
<td>Deletes all versions of a solution and the Solution object itself.</td>
</tr>
<tr>
<td>describe_algorithm</td>
<td>Describes the given algorithm.</td>
</tr>
<tr>
<td>describe_batch_inference_job</td>
<td>Gets the properties of a batch inference job including name, Amazon Resource Name (ARN), and status.</td>
</tr>
<tr>
<td>describe_campaign</td>
<td>Describes the given campaign, including its status.</td>
</tr>
<tr>
<td>describe_dataset</td>
<td>Describes the given dataset.</td>
</tr>
<tr>
<td>describe_dataset_group</td>
<td>Describes the given dataset group.</td>
</tr>
<tr>
<td>describe_dataset_import_job</td>
<td>Describes the dataset import job created by CreateDatasetImportJob, including the import job ARN.</td>
</tr>
<tr>
<td>describe_event_tracker</td>
<td>Describes an event tracker.</td>
</tr>
<tr>
<td>describe_feature_transformation</td>
<td>Describes the given feature transformation.</td>
</tr>
<tr>
<td>describe_filter</td>
<td>Describes a filter’s properties.</td>
</tr>
<tr>
<td>describe_recipe</td>
<td>Describes a recipe.</td>
</tr>
<tr>
<td>describe_schema</td>
<td>Describes a schema.</td>
</tr>
<tr>
<td>describe_solution</td>
<td>Describes a solution.</td>
</tr>
<tr>
<td>describe_solution_version</td>
<td>Describes a specific version of a solution.</td>
</tr>
<tr>
<td>get_solution_metrics</td>
<td>Gets the metrics for the specified solution version.</td>
</tr>
<tr>
<td>list_batch_inference_jobs</td>
<td>Gets a list of the batch inference jobs that have been performed off of a solution version.</td>
</tr>
<tr>
<td>list_campaigns</td>
<td>Returns a list of campaigns that use the given solution.</td>
</tr>
<tr>
<td>list_dataset_groups</td>
<td>Returns a list of dataset groups.</td>
</tr>
<tr>
<td>list_dataset_import_jobs</td>
<td>Returns a list of dataset import jobs that use the given dataset.</td>
</tr>
<tr>
<td>list_datasets</td>
<td>Returns the list of datasets contained in the given dataset group.</td>
</tr>
<tr>
<td>list_event_trackers</td>
<td>Returns the list of event trackers associated with the account.</td>
</tr>
<tr>
<td>list_filters</td>
<td>Lists all filters that belong to a given dataset group.</td>
</tr>
<tr>
<td>list_recipes</td>
<td>Returns a list of available recipes.</td>
</tr>
<tr>
<td>list_schemas</td>
<td>Returns the list of schemas associated with the account.</td>
</tr>
<tr>
<td>list_solutions</td>
<td>Returns a list of solutions that use the given dataset group.</td>
</tr>
<tr>
<td>list_solution_versions</td>
<td>Returns a list of solution versions for the given solution.</td>
</tr>
<tr>
<td>update_campaign</td>
<td>Updates a campaign by either deploying a new solution or changing the value of the campaign’s minProvisionedTPS parameter.</td>
</tr>
</tbody>
</table>

**Examples**

```r
## Not run:
svc <- personalize()
svc$screate_batch_inference_job(
  Foo = 123
)
## End(Not run)
```

---

**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

```r
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

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

## End(Not run)
```
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
polly

Examples

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

## End(Not run)
```

---

**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"
  ),
```

---

**polly**

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.
rekognition

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

```r
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 target input image.
- **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 faces in a streaming video.
- **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 model.
- **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 Rekognition ID.
- **get_celebrity_recognition**: Gets the celebrity recognition results for a Amazon Rekognition Video analysis started by StartCelebrityRecognition.
- **get_content_moderation**: Gets the unsafe content analysis results for a Amazon Rekognition Video analysis started by StartContentModeration.
- **get_face_detection**: Gets face detection results for a Amazon Rekognition Video analysis started by StartFaceDetection.
get_face_search Gets the face search results for Amazon Rekognition Video face search started by StartFaceSearch
get_label_detection Gets the label detection results of a Amazon Rekognition Video analysis started by StartLabelDetection
get_person_tracking Gets the path tracking results of a Amazon Rekognition Video analysis started by StartPersonTracking
get_segment_detection Gets the segment detection results of a Amazon Rekognition Video analysis started by StartSegmentDetection
get_text_detection Gets the text detection results of a Amazon Rekognition Video analysis started by StartTextDetection
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 specified collection for matching faces
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 detected in a stored video
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

```r
## 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
- add_tags
- associate_trial_component
- create_action
- create_algorithm
- create_app
- create_app_image_config
- create_artifact
- create_auto_ml_job
- create_code_repository
- create_compilation_job
- create_context
- create_data_quality_job_definition
- create_device_fleet
- create_domain
- create_edge_packaging_job
- create_endpoint
- create_endpoint_config
- create_experiment
- create_feature_group
- create_flow_definition
- create_human_task_ui
- create_hyper_parameter_tuning_job
- create_image
- create_image_version
- create_labeling_job
- create_model
- create_model_bias_job_definition
- create_model_explainability_job_definition
- create_model_package
- create_model_package_group
- create_model_quality_job_definition
- create_monitoring_schedule
- create_notebook_instance
- create_notebook_instance_lifecycle_config
- create_pipeline
- create_presigned_domain_url
- create_presigned_notebook_instance_url
- create_processing_job
- create_project
- create_training_job
- create_transform_job
- create_trial
- create_trial_component
- create_user_profile
- create_workforce

Creates an association between the source and the destination
Adds or overwrites one or more tags for the specified Amazon SageMaker resource
Associates a trial component with a trial
Creates an action
Create a machine learning algorithm that you can use in Amazon SageMaker
Creates a configuration for running a SageMaker image as a KernelGateway app
Creates a device fleet
Creates an Autopilot job
Creates a Git repository as a resource in your Amazon SageMaker account
Starts a model compilation job
Creates a context
Creates a definition for a job that monitors data quality and drift
Creates a device fleet
Creates a Domain used by Amazon SageMaker Studio
Starts a SageMaker Edge Manager model packaging job
Creates an endpoint using the endpoint configuration specified in the request
Creates an endpoint configuration that Amazon SageMaker hosting services uses to deploy models
Starts a SageMaker experiment
Create a new FeatureGroup
Creates a flow definition
Defines the settings you will use for the human review workflow user interface
Starts a hyperparameter tuning job
Creates a custom SageMaker image
Creates a version of the SageMaker image specified by ImageName
Creates a job that uses workers to label the data objects in your input data
Creates a model in Amazon SageMaker
Creates the definition for a model bias job
Creates the definition for a model explainability job
Creates a model package that you can use to create Amazon SageMaker models
Creates a model group
Creates a definition for a job that monitors model quality and drift
Creates a schedule that regularly starts Amazon SageMaker Processing Jobs
Creates an Amazon SageMaker notebook instance
Creates a lifecycle configuration that you can associate with a notebook
Creates a pipeline using a JSON pipeline definition
Creates a URL for a specified UserProfile in a Domain
Returns a URL that you can use to connect to the Jupyter server from a notebook
Creates a processing job
Creates a machine learning (ML) project that can contain one or more templates that set up an ML pipeline from training to deploying an approved model
Starts a model training job
Starts a transform job
Creates an Amazon SageMaker trial
Creates a trial component, which is a stage of a machine learning trial
Creates a user profile
Use this operation to create a workforce
create_workteam
delete_action
delete_algorithm
delete_app
delete_app_image_config
delete_artifact
delete_association
delete_code_repository
delete_context
delete_data_quality_job_definition
delete_device_fleet
delete_domain
delete_endpoint
delete_endpoint_config
delete_experiment
delete_feature_group
delete_flow_definition
delete_human_task_ui
delete_image
delete_image_version
delete_model
delete_model_bias_job_definition
delete_model_explainability_job_definition
delete_model_package
delete_model_package_group
delete_model_package_group_policy
delete_model_quality_job_definition
delete_monitoring_schedule
delete_notebook_instance
delete_notebook_instance_lifecycle_config
delete_pipeline
delete_project
delete_tags
delete_trial
delete_trial_component
delete_user_profile
delete_workforce
delete_workteam
deregister_devices
describe_action
describe_algorithm
describe_app
describe_app_image_config
describe_artifact
describe_auto_ml_job
describe_code_repository
describe_compilation_job
describe_context

Creates a new work team for labeling your data
Deletes an action
Removes the specified algorithm from your account
Used to stop and delete an app
Deletes an AppImageConfig
Deletes an artifact
Deletes an association
Deletes the specified Git repository from your account
Deletes an context
Deletes a data quality monitoring job definition
Deletes a fleet
Used to delete a domain
Deletes an endpoint
Deletes an endpoint configuration
Deletes an Amazon SageMaker experiment
Delete the FeatureGroup and any data that was written to the OnlineStore
Volumes
Deletes the specified flow definition
Use this operation to delete a human task user interface (worker task template)
Deletes a SageMaker image and all versions of the image
Deletes a version of a SageMaker image
Deletes a model
Deletes an Amazon SageMaker model bias job definition
Deletes an Amazon SageMaker model explainability job definition
Deletes a model package
Deletes the specified model group
Deletes a model group resource policy
Deletes the specified model quality monitoring job definition
Deletes a monitoring schedule
Deletes an Amazon SageMaker notebook instance
Deletes a notebook instance lifecycle configuration
Deletes a pipeline if there are no in-progress executions
Delete the specified project
Deletes the specified tags from an Amazon SageMaker resource
Deletes the specified trial
Deletes the specified trial component
Deletes a user profile
Use this operation to delete a workforce
Deletes an existing work team
Deregisters the specified devices
Describes an action
Returns a description of the specified algorithm that is in your account
Describes the app
Describes an AppImageConfig
Describes an artifact
Returns information about an Amazon SageMaker job
Gets details about the specified Git repository
Returns information about a model compilation job
Describes a context
describe_data_quality_job_definition
describe_device
describe_device_fleet
describe_domain
describe_edge_packaging_job
describe_endpoint
describe_endpoint_config
describe_experiment
describe_feature_group
describe_flow_definition
describe_human_task_ui
describe_hyper_parameter_tuning_job
describe_image
describe_image_version
describe_labeling_job
describe_model
describe_model_bias_job_definition
describe_model_explainability_job_definition
describe_model_package
describe_model_package_group
describe_model_quality_job_definition
describe_monitoring_schedule
describe_notebook_instance
describe_notebook_instance_lifecycle_config
describe_pipeline
describe_pipeline_definition_for_execution
describe_pipeline_execution
describe_processing_job
describe_project
describe_subscribed_workteam
describe_training_job
describe_transform_job
describe_trial
describe_trial_component
describe_user_profile
describe_workforce
describe_workteam
disable_sagemaker_servicecatalog_portfolio
disassociate_trial_component
enable_sagemaker_servicecatalog_portfolio
get_device_fleet_report
get_model_package_group_policy
get_sagemaker_servicecatalog_portfolio_status
get_search_suggestions
list_actions
list_algorithms
list_app_image_configs
list_apps

Gets the details of a data quality monitoring job definition
Describes the device
A description of the fleet the device belongs to
The description of the domain
A description of edge packaging jobs
Returns the description of an endpoint
Returns the description of an endpoint configuration created using the CreateEndpointConfig API
Provides a list of an experiment’s properties
Use this operation to describe a FeatureGroup
Returns information about the specified flow definition
Returns information about the requested human task user interface (worker task template)
Gets a description of a hyperparameter tuning job
Describes a SageMaker image
Describes a version of a SageMaker image
Gets information about a labeling job
Describes a model that you created using the CreateModel API
Returns a description of a model bias job definition
Returns a description of a model explainability job definition
Returns a description of the specified model package, which is used to create a SageMaker model
Gets a description for the specified model group
Describes the schedule for a monitoring job
Returns information about a notebook instance
Returns a description of a notebook instance lifecycle configuration
Describes the details of a pipeline
Describes the details of an execution’s pipeline definition
Describes the details of a pipeline execution
Returns a description of a processing job
Describes the details of a project
Gets information about a work team provided by a vendor
Returns information about a training job
Returns information about a transform job
Provides a list of a trial’s properties
Provides a list of a trials component’s properties
Describes a user profile
Lists private workforce information, including workforce name, Amazon Resource Name (ARN), and allowed IP address ranges
Gets information about a specific work team
Disables using Service Catalog in SageMaker
Disassociates a trial component from a trial
Enables using Service Catalog in SageMaker
Describes a fleet
Gets a resource policy that manages access for a model group
Gets the status of Service Catalog in SageMaker
An auto-complete API for the search functionality in the Amazon SageMaker console
Lists the actions in your account and their properties
Lists the machine learning algorithms that have been created
Lists the AppImageConfigs in your account and their properties
Lists apps
<table>
<thead>
<tr>
<th>Function Call</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_artifacts</td>
<td>Lists the artifacts in your account and their properties</td>
</tr>
<tr>
<td>list_associations</td>
<td>Lists the associations in your account and their properties</td>
</tr>
<tr>
<td>list_auto_ml_jobs</td>
<td>Request a list of jobs</td>
</tr>
<tr>
<td>list_candidates_for_auto_ml_job</td>
<td>List the Candidates created for the job</td>
</tr>
<tr>
<td>list_code_repositories</td>
<td>Gets a list of the Git repositories in your account</td>
</tr>
<tr>
<td>list_compilation_jobs</td>
<td>Lists model compilation jobs that satisfy various filters</td>
</tr>
<tr>
<td>list_contexts</td>
<td>Lists the contexts in your account and their properties</td>
</tr>
<tr>
<td>list_data_quality_job_definitions</td>
<td>Lists the data quality job definitions in your account</td>
</tr>
<tr>
<td>list_device_fleets</td>
<td>Returns a list of devices in the fleet</td>
</tr>
<tr>
<td>list_devices</td>
<td>A list of devices</td>
</tr>
<tr>
<td>list_domains</td>
<td>Lists the domains</td>
</tr>
<tr>
<td>list_edge_packaging_jobs</td>
<td>Returns a list of edge packaging jobs</td>
</tr>
<tr>
<td>list_endpoint_configs</td>
<td>Lists endpoint configurations</td>
</tr>
<tr>
<td>list_endpoints</td>
<td>Lists endpoints</td>
</tr>
<tr>
<td>list_experiments</td>
<td>Lists all the experiments in your account</td>
</tr>
<tr>
<td>list_feature_groups</td>
<td>List FeatureGroups based on given filter and order</td>
</tr>
<tr>
<td>list_flow_definitions</td>
<td>Returns information about the flow definitions in your account</td>
</tr>
<tr>
<td>list_human_task_uis</td>
<td>Returns information about the human task user interfaces in your account</td>
</tr>
<tr>
<td>list_hyper_parameter_tuning_jobs</td>
<td>Gets a list of HyperParameterTuningJobSummary objects that describe the hyperparameter tuning jobs launched in your account</td>
</tr>
<tr>
<td>list_images</td>
<td>Lists the images in your account and their properties</td>
</tr>
<tr>
<td>list_image_versions</td>
<td>Lists the versions of a specified image and their properties</td>
</tr>
<tr>
<td>list_labeling_jobs</td>
<td>Gets a list of labeling jobs</td>
</tr>
<tr>
<td>list_labeling_jobs_for_workteam</td>
<td>Gets a list of labeling jobs assigned to a specified work team</td>
</tr>
<tr>
<td>list_model_bias_job_definitions</td>
<td>Lists model bias jobs definitions that satisfy various filters</td>
</tr>
<tr>
<td>list_model_explainability_job_definitions</td>
<td>Lists model explainability job definitions that satisfy various filters</td>
</tr>
<tr>
<td>list_model_package_groups</td>
<td>Gets a list of the model groups in your AWS account</td>
</tr>
<tr>
<td>list_model_packages</td>
<td>Lists the model packages that have been created</td>
</tr>
<tr>
<td>list_model_quality_job_definitions</td>
<td>Gets a list of model quality monitoring job definitions in your account</td>
</tr>
<tr>
<td>list_models</td>
<td>Lists models created with the CreateModel API</td>
</tr>
<tr>
<td>list_monitoring_executions</td>
<td>Returns list of all monitoring job executions</td>
</tr>
<tr>
<td>list_monitoring_schedules</td>
<td>Returns list of all monitoring schedules</td>
</tr>
<tr>
<td>list_notebook_instance_lifecycle_configs</td>
<td>Lists notebook instance lifestyle configurations created with the CreateNotebookInstanceLifecycleConfig API</td>
</tr>
<tr>
<td>list_notebook_instances</td>
<td>Returns a list of the Amazon SageMaker notebook instances in the requester’s account in an AWS Region</td>
</tr>
<tr>
<td>list_pipeline_executions</td>
<td>Gets a list of the pipeline executions</td>
</tr>
<tr>
<td>list_pipeline_execution_steps</td>
<td>Gets a list of PipeLineExecutionStep objects</td>
</tr>
<tr>
<td>list_pipeline_parameters_for_execution</td>
<td>Gets a list of parameters for a pipeline execution</td>
</tr>
<tr>
<td>list_pipelines</td>
<td>Gets a list of pipelines</td>
</tr>
<tr>
<td>list_processing_jobs</td>
<td>Lists processing jobs that satisfy various filters</td>
</tr>
<tr>
<td>list_projects</td>
<td>Gets a list of the projects in an AWS account</td>
</tr>
<tr>
<td>list_subscribed_workteams</td>
<td>Gets a list of the work teams that you are subscribed to in the AWS Marketplace</td>
</tr>
<tr>
<td>list_tags</td>
<td>Returns the tags for the specified Amazon SageMaker resource</td>
</tr>
<tr>
<td>list_training_jobs</td>
<td>Lists training jobs</td>
</tr>
<tr>
<td>list_training_jobs_for_hyper_parameter_tuning_job</td>
<td>Gets a list of TrainingJobSummary objects that describe the training jobs launched by a hyperparameter tuning job</td>
</tr>
<tr>
<td>list_transform_jobs</td>
<td>Lists transform jobs</td>
</tr>
<tr>
<td>list_trial_components</td>
<td>Lists the trial components in your account</td>
</tr>
<tr>
<td>list_trials</td>
<td>Lists the trials in your account</td>
</tr>
<tr>
<td>list_user_profiles</td>
<td>Lists user profiles</td>
</tr>
<tr>
<td>list_workforces</td>
<td>Use this operation to list all private and vendor workforces in an AWS Region</td>
</tr>
</tbody>
</table>
list_workteams
put_model_package_group_policy
register_devices
render_ui_template
search
start_monitoring_schedule
start_notebook_instance
start_pipeline_execution
stop_auto_ml_job
stop_compilation_job
stop_edge_packaging_job
stop_hyper_parameter_tuning_job
stop_labeling_job
stop_monitoring_schedule
stop_notebook_instance
stop_pipeline_execution
stop_processing_job
stop_training_job
stop_transform_job
update_action
update_app_image_config
update_artifact
update_code_repository
update_context
update_device_fleet
update_devices
update_domain
update_endpoint
update_endpoint_weights_and_capacities
update_experiment
update_image
update_model_package
update_monitoring_schedule
update_notebook_instance
update_notebook_instance_lifecycle_config
update_pipeline
update_pipeline_execution
update_training_job
update_trial
update_trial_component
update_user_profile
update_workforce
update_workteam

Gets a list of private work teams that you have defined in a region
Adds a resource policy to control access to a model group
Register devices
Renders the UI template so that you can preview the worker’s experience
Finds Amazon SageMaker resources that match a search query
Starts a previously stopped monitoring schedule
Launches an ML compute instance with the latest version of the libraries
Starts a pipeline execution
A method for forcing the termination of a running job
Stops a model compilation job
Request to stop an edge packaging job
Stops a running hyperparameter tuning job and all running training jobs
Stops a running labeling job
Stops a previously started monitoring schedule
Terminates the ML compute instance
Starts a pipeline execution
Stops a processing job
Starts a training job
Stops a transform job
Updates an action
Updates the properties of an AppImageConfig
Updates an artifact
Updates the specified Git repository with the specified values
Updates a context
Updates a fleet of devices
Updates one or more devices in a fleet
Updates the default settings for new user profiles in the domain
Deploys the new EndpointConfig specified in the request, switches to using newly created endpoint, and then deletes resources provisioned for the endpoint using the previous EndpointConfig (there is no availability loss)
Updates variant weight of one or more variants associated with an existing endpoint, or capacity of one variant associated with an existing endpoint
Updates the description of an experiment
Updates the properties of a SageMaker image
Updates a versioned model
Updates a previously created schedule
Updates a notebook instance
Updates a notebook instance lifecycle configuration created with the CreateNotebookInstanceLifecycleConfig API
Updates a pipeline
Updates a pipeline execution
Update a model training job to request a new Debugger profiling configuration
Updates the display name of a trial
Updates one or more properties of a trial component
Updates a user profile
Use this operation to update your workforce
Updates an existing work team with new member definitions or description

Examples

## Not run:
sagemakerruntime <- sagemaker()
svc$add_association(
  Foo = 123
)

## End(Not run)

---

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

```r
cvc <- 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 applications use this API to get inferences from the model hosted at the specified endpoint.

## Examples

```r
## 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**

```r
csvc <- textract(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    )
  )
)```
transcribeservice

```
),
    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 document)
- **get_document_text_detection** (gets the results for an Amazon Textract asynchronous operation that detects text in a document)
- **start_document_analysis** (starts the asynchronous analysis of an input document for relationships between detected items)
- **start_document_text_detection** (starts the asynchronous detection of text in a document)

**Examples**

```r
## 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**

```r
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

```r
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 Medical transcribes your audio file.
- `create_vocabulary`: Creates a new custom vocabulary that you can use to change the way Amazon Transcribe handles transcription of an audio file.
- `create_vocabulary_filter`: Creates a new vocabulary filter that you can use to filter words, such as profane words, from the output of a transcription job.
- `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 information.
- `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 such as the transcription, models, and so on.
- `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 name.
- `list_medical_vocabularies`: Returns a list of vocabularies that match the specified criteria.
- `list_transcription_jobs`: Lists transcription jobs with the specified status.
- `list_vocabulary`: 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 you used to create the vocabulary.
- `update_vocabulary`: Updates an existing vocabulary with new values.
- `update_vocabulary_filter`: Updates a vocabulary filter with a new list of filtered words.
translate

Examples

```r
## 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**

```r
csvc <- 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
translate

create_parallel_data
delete_parallel_data
delete_terminology
describe_text_translation_job
get_parallel_data
get_terminology
import_terminology
list_parallel_data
list_terminologies
list_text_translation_jobs
start_text_translation_job
stop_text_translation_job
translate_text
update_parallel_data

Creates a parallel data resource in Amazon Translate by importing an input file from Amazon S3
Deletes a parallel data resource in Amazon Translate
A synchronous action that deletes a custom terminology
Gets the properties associated with an asynchronous batch translation job including name, ID, status, source and target languages, input/output S3 buckets, and so on
Provides information about a parallel data resource
Retrieves a custom terminology
Creates or updates a custom terminology, depending on whether or not one already exists for the terminology name
Provides a list of your parallel data resources in Amazon Translate
Provides a list of custom terminologies associated with your account
Gets a list of the batch translation jobs that you have submitted
Starts an asynchronous batch translation job
 Stops an asynchronous batch translation job that is in progress
 Translates input text from the source language to the target language
 Updates a previously created parallel data resource by importing a new input file from Amazon S3

Examples

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

## End(Not run)
```
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_association</td>
<td>22</td>
</tr>
<tr>
<td>add_tags</td>
<td>12, 22</td>
</tr>
<tr>
<td>analyze_document</td>
<td>29</td>
</tr>
<tr>
<td>associate_trial_component</td>
<td>22</td>
</tr>
<tr>
<td>batch_detect_dominant_language</td>
<td>3</td>
</tr>
<tr>
<td>batch_detect_entities</td>
<td>3</td>
</tr>
<tr>
<td>batch_detect_key_phrases</td>
<td>3</td>
</tr>
<tr>
<td>batch_detect_sentiment</td>
<td>3</td>
</tr>
<tr>
<td>batch_detect_syntax</td>
<td>3</td>
</tr>
<tr>
<td>classify_document</td>
<td>3</td>
</tr>
<tr>
<td>compare_faces</td>
<td>19</td>
</tr>
<tr>
<td>comprehend</td>
<td>2</td>
</tr>
<tr>
<td>comprehendmedical</td>
<td>5</td>
</tr>
<tr>
<td>create_action</td>
<td>22</td>
</tr>
<tr>
<td>create_algorithm</td>
<td>22</td>
</tr>
<tr>
<td>create_app</td>
<td>22</td>
</tr>
<tr>
<td>create_app_image_config</td>
<td>22</td>
</tr>
<tr>
<td>create_artifact</td>
<td>22</td>
</tr>
<tr>
<td>create_auto_ml_job</td>
<td>22</td>
</tr>
<tr>
<td>create_batch_inference_job</td>
<td>13</td>
</tr>
<tr>
<td>create_batch_prediction</td>
<td>12</td>
</tr>
<tr>
<td>create_bot_version</td>
<td>7</td>
</tr>
<tr>
<td>create_campaign</td>
<td>13</td>
</tr>
<tr>
<td>create_code_repository</td>
<td>22</td>
</tr>
<tr>
<td>create_collection</td>
<td>19</td>
</tr>
<tr>
<td>create_collection_job</td>
<td>22</td>
</tr>
<tr>
<td>create_context</td>
<td>22</td>
</tr>
<tr>
<td>create_data_quality_job_definition</td>
<td>22</td>
</tr>
<tr>
<td>create_data_source_from_rds</td>
<td>12</td>
</tr>
<tr>
<td>create_data_source_from_redshift</td>
<td>12</td>
</tr>
<tr>
<td>create_data_source_from_s3</td>
<td>12</td>
</tr>
<tr>
<td>create_dataset</td>
<td>13</td>
</tr>
<tr>
<td>create_dataset_group</td>
<td>13</td>
</tr>
<tr>
<td>create_dataset_import_job</td>
<td>13</td>
</tr>
<tr>
<td>create_device_fleet</td>
<td>22</td>
</tr>
<tr>
<td>create_document_classifier</td>
<td>3</td>
</tr>
<tr>
<td>create_domain</td>
<td>22</td>
</tr>
<tr>
<td>create_edge_packaging_job</td>
<td>22</td>
</tr>
<tr>
<td>create_endpoint</td>
<td>3, 22</td>
</tr>
<tr>
<td>create_endpoint_config</td>
<td>22</td>
</tr>
<tr>
<td>create_entity_recognizer</td>
<td>3</td>
</tr>
<tr>
<td>create_evaluation</td>
<td>12</td>
</tr>
<tr>
<td>create_event_tracker</td>
<td>13</td>
</tr>
<tr>
<td>create_experiment</td>
<td>22</td>
</tr>
<tr>
<td>create_feature_group</td>
<td>22</td>
</tr>
<tr>
<td>create_filter</td>
<td>13</td>
</tr>
<tr>
<td>create_flow_definition</td>
<td>22</td>
</tr>
<tr>
<td>create_human_task_ui</td>
<td>22</td>
</tr>
<tr>
<td>create_hyper_parameter_tuning_job</td>
<td>22</td>
</tr>
<tr>
<td>create_image</td>
<td>22</td>
</tr>
<tr>
<td>create_image_version</td>
<td>22</td>
</tr>
<tr>
<td>create_intent_version</td>
<td>7</td>
</tr>
<tr>
<td>create_labeling_job</td>
<td>22</td>
</tr>
<tr>
<td>create_language_model</td>
<td>30</td>
</tr>
<tr>
<td>create_medical_vocabulary</td>
<td>30</td>
</tr>
<tr>
<td>create_ml_model</td>
<td>12</td>
</tr>
<tr>
<td>create_model</td>
<td>22</td>
</tr>
<tr>
<td>create_model_bias_job_definition</td>
<td>22</td>
</tr>
<tr>
<td>create_model_explainability_job_definition</td>
<td>22</td>
</tr>
<tr>
<td>create_model_package</td>
<td>22</td>
</tr>
<tr>
<td>create_model_package_group</td>
<td>22</td>
</tr>
<tr>
<td>create_model_quality_job_definition</td>
<td>22</td>
</tr>
<tr>
<td>create_monitoring_schedule</td>
<td>22</td>
</tr>
<tr>
<td>create_notebook_instance</td>
<td>22</td>
</tr>
<tr>
<td>create_notebook_instance_lifecycle_config</td>
<td>22</td>
</tr>
<tr>
<td>create_parallel_data</td>
<td>33</td>
</tr>
<tr>
<td>create_pipeline</td>
<td>22</td>
</tr>
<tr>
<td>create_presigned_domain_url</td>
<td>22</td>
</tr>
<tr>
<td>create_presigned_notebook_instance_url</td>
<td>22</td>
</tr>
<tr>
<td>create_processing_job</td>
<td>22</td>
</tr>
<tr>
<td>create_project</td>
<td>19, 22</td>
</tr>
<tr>
<td>create_project_version</td>
<td>19</td>
</tr>
<tr>
<td>create_realtime_endpoint</td>
<td>12</td>
</tr>
</tbody>
</table>
create_schema, 13
create_slot_type_version, 7
create_solution, 13
create_solution_version, 13
create_stream_processor, 19
create_training_job, 22
create_transform_job, 22
create_trial, 22
create_trial_component, 22
create_user_profile, 22
create_vocabulary, 30
create_vocabulary_filter, 30
create_workforce, 22
create_workteam, 23
delete_action, 23
delete_algorithm, 23
delete_app, 23
delete_app_image_config, 23
delete_artifact, 23
delete_association, 23
delete_batch_prediction, 12
delete_bot, 7
delete_bot_alias, 7
delete_bot_channel_association, 7
delete_bot_version, 7
delete_campaign, 13
delete_code_repository, 23
delete_collection, 19
delete_context, 23
delete_data_quality_job_definition, 23
delete_data_source, 12
delete_dataset, 13
delete_dataset_group, 13
delete_device_fleet, 23
delete_document_classifier, 3
delete_domain, 23
delete_endpoint, 3, 23
delete_endpoint_config, 23
delete_entity_recognizer, 3
delete_evaluation, 12
delete_event_tracker, 13
delete_experiment, 23
delete_faces, 19
delete_feature_group, 23
delete_filter, 13
delete_flow_definition, 23
delete_human_task_ui, 23
delete_image, 23
delete_image_version, 23
delete_intent, 7
delete_intent_version, 7
delete_language_model, 30
delete_lexicon, 18
delete_medical_transcription_job, 30
delete_medical_vocabulary, 30
delete_ml_model, 12
delete_model, 23
delete_model_bias_job_definition, 23
delete_model_explainability_job_definition, 23
delete_model_package, 23
delete_model_package_group, 23
delete_model_package_group_policy, 23
delete_model_quality_job_definition, 23
delete_monitoring_schedule, 23
delete_notebook_instance, 23
delete_notebook_instance_lifecycle_config, 23
delete_parallel_data, 33
delete_pipeline, 23
delete_project, 19, 23
delete_project_version, 19
delete_realtime_endpoint, 12
delete_schema, 13
delete_session, 9
delete_slot_type, 7
delete_slot_type_version, 7
delete_solution, 14
delete_stream_processor, 19
delete_tags, 12, 23
delete_terminology, 33
delete_transcription_job, 30
delete_trial, 23
delete_trial_component, 23
delete_user_profile, 23
delete_utterances, 7
delete_vocabulary, 30
delete_vocabulary_filter, 30
delete_workforce, 23
delete_workteam, 23
deregister_devices, 23
describe_action, 23
describe_algorithm, 14, 23
describe_app, 23
describe_app_image_config, 23
describe_artifact, 23
describe_auto_ml_job, 23
describe_batch_inference_job, 14
describe_batch_predictions, 12
describe_campaign, 14
describe_code_repository, 23
describe_collection, 19
describe_compilation_job, 23
describe_context, 23
describe_data_quality_job_definition, 24
describe_data_sources, 12
describe_dataset, 14
describe_dataset_group, 14
describe_dataset_import_job, 14
describe_device, 24
describe_device_fleet, 24
describe_document_classification_job, 3
describe_document_classifier, 3
describe_domain, 24
describe_dominant_language_detection_job, 3
describe_edge_packaging_job, 24
describe_endpoint, 3, 24
describe_endpoint_config, 24
describe_entities_detection_job, 4
describe_entities_detection_v2_job, 6
describe_entity_recognizer, 4
describe_evaluations, 12
describe_event_tracker, 14
describe_events_detection_job, 4
describe_experiment, 24
describe_feature_group, 24
describe_feature_transformation, 14
describe_filter, 14
describe_flow_definition, 24
describe_human_task_ui, 24
describe_hyper_parameter_tuning_job, 24
describe_icd10cm_inference_job, 6
describe_image, 24
describe_image_version, 24
describe_key_phrases_detection_job, 4
describe_labeling_job, 24
describe_language_model, 30
describe_ml_models, 12
describe_model, 24
describe_model_bias_job_definition, 24
describe_model_explainability_job_definition, 24
describe_model_package, 24
describe_model_package_group, 24
describe_model_quality_job_definition, 24
describe_monitoring_schedule, 24
describe_notebook_instance, 24
describe_notebook_instance_lifecycle_config, 24
describe_phi_detection_job, 6
describe_pii_entities_detection_job, 4
describe_pipeline, 24
describe_pipeline_definition_for_execution, 24
describe_pipeline_execution, 24
describe_processing_job, 24
describe_project, 24
describe_project_versions, 19
describe_projects, 19
describe_recipe, 14
describe_rx_norm_inference_job, 6
describe_schema, 14
describe_sentiment_detection_job, 4
describe_solution, 14
describe_solution_version, 14
describe_stream_processor, 19
describe_subscribed_workteam, 24
describe_tags, 12
describe_text_translation_job, 33
describe_topics_detection_job, 4
describe_training_job, 24
describe_transform_job, 24
describe_trial, 24
describe_trial_component, 24
describe_user_profile, 24
describe_voices, 18
describe_workforce, 24
describe_workteam, 24
detect_custom_labels, 19
detect_document_text, 29
detect_dominant_language, 4
detect_entities, 4, 6
detect_entities_v2, 6
detect_faces, 19
detect_key_phrases, 4
detect_labels, 19
detect_moderation_labels, 19
detect_phi, 6
detect_pii_entities, 4
detect_protective_equipment, 19
detect_sentiment, 4
detect_syntax, 4
detect_text, 19
disable_sagemaker_servicecatalog_portfolio, 24
disassociate_trial_component, 24
enable_sagemaker_servicecatalog_portfolio, 24
get_batch_prediction, 12
get_bot, 7
get_bot_alias, 7
get_bot_aliases, 7
get_bot_channel_association, 7
get_bot_channelAssociations, 8
get_bot_versions, 8
get_bots, 8
get_builtin_intent, 8
get_builtin_intents, 8
get_builtin_slot_types, 8
get_celebrity_info, 19
get_celebrity_recognition, 19
get_content_moderation, 19
get_data_source, 12
get_device_fleet_report, 24
get_document_analysis, 29
get_document_text_detection, 29
get_evaluation, 12
get_export, 8
get_face_detection, 19
get_face_search, 20
get_import, 8
get_intent, 8
get_intent_versions, 8
get_intents, 8
get_label_detection, 20
get_lexicon, 18
get_medical_transcription_job, 30
get_medical_vocabulary, 30
get_ml_model, 12
get_model_package_group_policy, 24
get_parallel_data, 33
get_person_tracking, 20
get_personalized_ranking, 16
get_recommendations, 16
get_sagemaker_servicecatalog_portfolio_status, 24
get_search_suggestions, 24
get_segment_detection, 20
get_session, 9
get_slot_type, 8
get_slot_type_versions, 8
get_slot_types, 8
get_solution_metrics, 14
get_speech_synthesis_task, 18
get_terminology, 33
get_text_detection, 20
get_transcription_job, 30
get_utterances_view, 8
get_vocabulary, 30
get_vocabulary_filter, 30
import_terminology, 33
index_faces, 20
infer_icd10cm, 6
infer_rx_norm, 6
invoke_endpoint, 28
lexmodelbuildingservice, 6
lexruntimeservice, 8
list_actions, 24
list_algorithms, 24
list_app_image_configs, 24
list_apps, 24
list_artifacts, 25
list_associations, 25
list_auto_ml_jobs, 25
list_batch_inference_jobs, 14
list_campaigns, 14
list_candidates_for_auto_ml_job, 25
list_codeRepositories, 25
list_collections, 20
list_compilation_jobs, 25
list_contexts, 25
list_data_quality_job_definitions, 25
list_dataset_groups, 14
list_dataset_import_jobs, 14
list_datasets, 14
list_device_fleets, 25
list_devices, 25
list_document_classification_jobs, 4
list_document_classifiers, 4
list_domains, 25
list_dominant_language_detection_jobs, 4
list_edge_packaging_jobs, 25
list_endpoint_configs, 25
list_endpoints, 4, 25
list_entities_detection_jobs, 4
list_entities_detection_v2_jobs, 6
list_entity_recognizers, 4
list_event_trackers, 14
list_events_detection_jobs, 4
list_experiments, 25
list_faces, 20
list_feature_groups, 25
list_filters, 14
list_flow_definitions, 25
list_human_task_uis, 25
list_hyper_parameter_tuning_jobs, 25
list_icd10cm_inference_jobs, 6
list_image_versions, 25
list_images, 25
list_key_phrases_detection_jobs, 4
list_labeling_jobs, 25
list_labeling_jobs_for_workteam, 25
list_language_models, 30
list_lexicons, 18
list_medical_transcription_jobs, 30
list_medical_vocabulary, 30
list_model_bias_job_definitions, 25
list_model_explainability_job_definitions, 25
list_model_package_groups, 25
list_model_packages, 25
list_model_quality_job_definitions, 25
list_models, 25
list_monitoring_executions, 25
list_monitoring_schedules, 25
list_notebook_instance_lifecycle_configs, 25
list_notebook_instances, 25
list_parallel_data, 33
list_phi_detection_jobs, 6
list pii_entities_detection_jobs, 4
list_pipeline_execution_steps, 25
list_pipeline_executions, 25
list_pipeline_parameters_for_execution, 25
list_pipelines, 25
list_processing_jobs, 25
list_projects, 25
list_recipes, 14
list_rx_norm_inference_jobs, 6
list_schemas, 14
list_sentiment_detection_jobs, 4
list_solution_versions, 14
list_solutions, 14
list_speech_synthesis_tasks, 18
list_stream_processors, 20
list_subscribed_workteams, 25
list_tags, 25
list_tags_for_resource, 4, 8
list_terminologies, 33
list_text_translation_jobs, 33
list_topics_detection_jobs, 4
list_training_jobs, 25
list_training_jobs_for_hyper_parameter_tuning_job, 25
list_transcription_jobs, 30
list_transform_jobs, 25
list_trial_components, 25
list_trials, 25
list_user_profiles, 25
list_vocabulary, 30
list_vocabulary_filter, 30
list_workforces, 25
list_workteams, 26
machinelearning, 10
personalize, 12
personalizeevents, 14
personalizeruntime, 16
polly, 17
post_content, 9
post_text, 9
predict, 12
put_bot, 8
put_bot_alias, 8
put_events, 15
put_intent, 8
put_items, 15
put_lexicon, 18
put_model_package_group_policy, 26
put_session, 9
put_slot_type, 8
put_users, 15
recognize celebrities, 20
register_devices, 26
rekognition, 18
render_ui_template, 26
sagemaker, 21
sagemakerruntime, 27
search, 26
search_faces, 20
search_faces_by_image, 20
start_celebrity_recognition, 20
start_content_moderation, 20
start_document_classification_job, 4
start_document_text_detection, 29
start_dominant_language_detection_job, 4
start_entities_detection_job, 4
start_entities_detection_v2_job, 6
start_face_detection, 20
start_face_search, 20
start_icd10cm_inference_job, 6
start_import, 8
start_key_phrases_detection_job, 4
start_label_detection, 20
start_medical_transcription_job, 30
start_monitoring_schedule, 26
start_notebook_instance, 26
start_person_tracking, 20
start_phi_detection_job, 6
start_pii_entities_detection_job, 4
stop_events_detection_job, 4
stop_hyper_parameter_tuning_job, 26
stop_icd10cm_inference_job, 6
stop_key_phrases_detection_job, 4
stop_labeling_job, 26
stop_monitoring_schedule, 26
stop_notebook_instance, 26
stop_phi_detection_job, 6
stop_pii_entities_detection_job, 4
stop_pipeline_execution, 26
stop_processing_job, 26
stop_project_version, 20
stop_rx_norm_inference_job, 6
stop_sentiment_detection_job, 4
stop_stream_processor, 20
stop_text_translation_job, 33
stop_training_document_classifier, 4
stop_training_entity_recognizer, 4
stop_training_job, 26
stop_transform_job, 26
synthesize_speech, 18
tag_resource, 4, 8
textract, 28
transcribeservice, 29
translate, 31
translate_text, 33
untag_resource, 4, 8
update_action, 26
update_app_image_config, 26
update_artifact, 26
update_campaign, 14
update_code_repository, 26
update_context, 26
update_data_source, 12
update_device_fleet, 26
update_devices, 26
update_domain, 26
update_endpoint, 4, 26
update_endpoints_weights_and_capacities, 26
update_evaluation, 12
update_experiment, 26
update_image, 26
update_medical_vocabulary, 30
update_ml_model, 12
update_model_package, 26
update_monitoring_schedule, 26
update_notebook_instance, 26
update_notebook_instance_lifecycle_config, 26
update_parallel_data, 33
update_pipeline, 26
update_pipeline_execution, 26
update_training_job, 26
update_trial, 26
update_trial_component, 26
update_user_profile, 26
update_vocabulary, 30
update_vocabulary_filter, 30
update_workforce, 26
update_workteam, 26