Package ‘paws.machine.learning’

April 26, 2020

Title  Amazon Web Services Machine Learning APIs

Version  0.1.8

Description  Interface to Amazon Web Services machine learning APIs, 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)

Imports  paws.common (>= 0.3.0)

Suggests  testthat

Encoding  UTF-8

LazyData  true

RoxygenNote  7.1.0

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'
comprehend

'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 2020-04-26 17:10:03 UTC

R topics documented:

comprehend .......................................................... 2
comprehendmedical ................................................. 4
lexmodelbuildingservice .......................................... 6
lexruntimeservice ................................................ 8
machinelearning .................................................. 9
personalize .......................................................... 10
personalizeevents .................................................. 12
personalizeruntime ............................................... 13
polly ................................................................. 14
rekognition .......................................................... 16
sagemaker ........................................................... 18
sagemakerruntime ............................................... 21
textract ............................................................ 22
transcribeservice ............................................... 24
translate ............................................................ 25

Index 27

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.

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 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
batch_detect_syntax Inspects the text of a batch of documents for the syntax and part of speech of the words in the document
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 Only those classifiers that are in terminated states (IN_ERROR, TRAINED) will be deleted
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 containing training data, recognizer metadata, metrics, and so on
describe_key_phrases_detection_job Gets the properties associated with a key phrases 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_sentiment Inspects text and returns an inference of the prevailing sentiment (POSITIVE, NEUTRAL, MIXED, or NEGATIVE)
detect_syntax Inspects text for syntax and the part of speech of words in the document
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.

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_phi_detection_job Gets the properties associated with a protected health information (PHI) detection 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 about them such as entity category, location, and confidence score on that information
detect_phi Inspects the clinical text for protected health information (PHI) entities and entity category, location, and confidence score on that information
infer_icd10cm InferICD10CM detects medical conditions as entities listed in a patient record and links those entities to normalized concept identifiers in the ICD-10-CM knowledge base from the Centers for Disease Control
infer_rx_norm InferRxNorm detects medications as entities listed in a patient record and links to the normalized concept identifiers in the RxNorm database from the National Library of Medicine
list_entities_detection_v2_jobs Gets a list of medical entity detection jobs that you have submitted
list_phi_detection_jobs Gets a list of protected health information (PHI) detection jobs that you have submitted
start_entities_detection_v2_job Starts an asynchronous medical entity detection job for a collection of documents
start_phi_detection_job Starts an asynchronous job to detect protected health information (PHI)
stop_entities_detection_v2_job Stops a medical entities detection job in progress
stop_phi_detection_job Stops a protected health information (PHI) detection job in progress

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

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

### Arguments

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

### Service syntax

```r
csvc <- 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 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
delete_bot_version
delete_intent
delete_intent_version
delete_slot_type
delete_slot_type_version
delete_utterances
get_bot
get_bot_alias
get_bot_aliases
get_bot_channel_association
get_bot_channel_associations
get_bots
get_bot_versions
get_built_in_intent
get_built_in_intents
get_built_in_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
put_bot
put_bot_alias
put_intent
put_slot_type
start_import

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 platform
get_bot_channel_associations Returns a list of all of the channels associated with the specified bot
get_bots Gets information about all of the versions of a bot
get_built_in_intent Returns information about a built-in intent
get_built_in_intents Gets a list of built-in intents that meet the specified criteria
get_built_in_slot_types Exports the contents of a Amazon Lex resource in a specified format
get_export Gets information about an import job started with the StartImport operation
get_import Returns information about an intent
get_intents Gets intent information as follows: - If you specify the nameContains field, returns the $LATEST version of all intents that contain the specified string
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: - If you specify the nameContains field, returns the $LATEST version of all slot types that contain the specified string
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 users have made to your bot
get_bot
put_bot
put_bot_alias
put_intent
put_slot_type
start_import

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)
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

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

Arguments

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

Service syntax

```python
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

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

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

- **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 information
- **get_data_source**
  - Returns a DataSource that includes metadata and data file information, as well as the current status of the DataSource
- **get_evaluation**
  - Returns an Evaluation that includes metadata as well as the current status of the Evaluation
- **get_ml_model**
  - Returns an MLModel that includes detailed metadata, data source information, and the current status of the MLModel
- **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

```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.

Service syntax
```r
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_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_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 (ARN), etc.
### 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 job status

### describe_event_tracker
Describes an event tracker

### describe_feature_transformation
Describes the given feature transformation

### 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_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 campaign’s minProvisionedTPS parameter

---

### Examples

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

## End(Not run)
```

---

### personalizeevents

**Amazon Personalize Events**

### Description

Amazon Personalize Events

### Usage

```r
personalizeevents(config = list())
```
personalizeruntime

Arguments

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

Service syntax

```r
csv <- 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

Examples

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

Description

Amazon Personalize Runtime

Usage

```r
personalizeruntime(config = list())
```
Arguments
config  Optional configuration of credentials, endpoint, and/or region.

Service syntax

```r
csvc <- 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

```r
## 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.

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

delte_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)
Description

This is the Amazon Rekognition API reference.

Usage

rekognition(config = list())

Arguments

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

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 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.
delete_collection Deletes the specified collection.
delete_faces Deletes faces from a collection.
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
detect_labels
detect_moderation_labels
detect_text
get_celebrity_info
get_celebrity_recognition
get_content_moderation
get_face_detection
get_face_search
get_label_detection
get_person_tracking
index_faces
list_collections
list_faces
list_stream_processors
recognizeCelebrities
search_faces
search_faces_by_image
start_celebrity_recognition
start_content_moderation
start_face_detection
start_face_search
start_label_detection
start_person_tracking
start_project_version
start_stream_processor
stop_project_version
stop_stream_processor

Detects faces within an image that is provided as input
Detects instances of real-world entities within an image (JPEG or PNG) provided as input
Detects unsafe content in a specified JPEG or PNG format image
Detects text in the input image and converts it into machine-readable text
Gets the name and additional information about a celebrity based on his or her Amazon Rekognition ID
Gets the celebrity recognition results for a Amazon Rekognition Video analysis started by StartCelebrityRecognition
Gets the unsafe content analysis results for a Amazon Rekognition Video analysis started by StartContentModeration
Gets face detection results for a Amazon Rekognition Video analysis started by StartFaceDetection
Gets the face search results for Amazon Rekognition Video face search started by StartFaceSearch
Gets the label detection results of a Amazon Rekognition Video analysis started by StartLabelDetection
Gets the path tracking results of a Amazon Rekognition Video analysis started by StartPersonTracking
Detections faces in the input image and adds them to the specified collection
Returns list of collection IDs in your account
Returns metadata for faces in the specified collection
Gets a list of stream processors that you have created with CreateStreamProcessor
For a given input face ID, searches for matching faces in the collection the face belongs to
For a given input image, first detects the largest face in the image, and then searches the specified collection
Starts asynchronous recognition of celebrities in a stored video
Starts asynchronous detection of unsafe content in a stored video
Starts asynchronous detection of faces in a stored video
Starts the asynchronous search for faces in a collection that match the faces of persons detected in an input image
Starts asynchronous detection of labels in a stored video
Starts the asynchronous tracking of a person’s path in a stored video
Starts the running of the version of a model
Starts processing a stream processor
Stops a running model
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"
    )
  )
)
sagemaker Amazon SageMaker Service

Description
Provides APIs for creating and managing Amazon SageMaker resources.

Usage
sagemaker(config = list())

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

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_tags
  Adds or overwrites one or more tags for the specified Amazon SageMaker resource.

- associate_trial_component
  Associates a trial component with a trial.

- create_algorithm
  Create a machine learning algorithm that you can use in Amazon SageMaker.

- create_app
  Creates a running App for the specified UserProfile.

- create_auto_ml_job
  Creates an AutoPilot job.

- create_code_repository
  Creates a Git repository as a resource in your Amazon SageMaker account.

- create_compilation_job
  Starts a model compilation job.
create_domain
create_endpoint
create_endpoint_config
create_experiment
create_flow_definition
create_human_task_ui
create_hyper_parameter_tuning_job
create_labeling_job
create_model
create_model_package
create_monitoring_schedule
create_notebook_instance
create_notebook_instance_lifecycle_config
create_presigned_domain_url
create_presigned_notebook_instance_url
create_processing_job
create_training_job
create_transform_job
create_trial
create_trial_component
create_user_profile
create_workteam
delete_algorithm
delete_app
delete_code_repository
delete_domain
delete_endpoint
delete_endpoint_config
delete_experiment
delete_flow_definition
delete_model
delete_model_package
delete_monitoring_schedule
delete_notebook_instance
delete_notebook_instance_lifecycle_config
delete_tags
delete_trial
delete_trial_component
delete_user_profile
delete_workteam
describe_algorithm
describe_app
describe_auto_ml_job
describe_code_repository
describe_compilation_job
describe_domain
describe_endpoint
describe_endpoint_config

create_domain
create_endpoint
create_endpoint_config
create_experiment
create_flow_definition
create_human_task_ui
create_hyper_parameter_tuning_job
create_labeling_job
create_model
create_model_package
create_monitoring_schedule
create_notebook_instance
create_notebook_instance_lifecycle_config
create_presigned_domain_url
create_presigned_notebook_instance_url
create_processing_job
create_training_job
create_transform_job
create_trial
create_trial_component
create_user_profile
create_workteam
delete_algorithm
delete_app
delete_code_repository
delete_domain
delete_endpoint
delete_endpoint_config
delete_experiment
delete_flow_definition
delete_model
delete_model_package
delete_monitoring_schedule
delete_notebook_instance
delete_notebook_instance_lifecycle_config
delete_tags
delete_trial
delete_trial_component
delete_user_profile
delete_workteam
describe_algorithm
describe_app
describe_auto_ml_job
describe_code_repository
describe_compilation_job
describe_domain
describe_endpoint
describe_endpoint_config

Create a Domain for Amazon SageMaker
Create an endpoint using the endpoint configuration specified in the request
Create an endpoint configuration that Amazon SageMaker hosting services use to deploy a model
Create an Amazon SageMaker experiment
Create a flow definition
Defines the settings you will use for the human review workflow user interface
Starts a hyperparameter tuning job
Creates a job that uses workers to label the data objects in your input dataset
Creates a model in Amazon SageMaker
Creates a model package that you can use to create Amazon SageMaker models
Create a schedule that regularly starts Amazon SageMaker Processing Jobs
Create an Amazon SageMaker notebook instance
Creates a lifecycle configuration that you can associate with a notebook instance
Creates a URL for a specified User Profile in a Domain
Returns a URL that you can use to connect to the Jupyter server from a notebook instance
Create a training job
Starts a model training job
Starts a transform job
Create a SageMaker trial
Creates a trial component, which is a stage of a machine learning trial
Creates a new user profile
Creates a new work team for labeling your data
Removes the specified algorithm from your account
Used to stop and delete an app
Deletes the specified Git repository from your account
Used to delete a domain
Deletes an endpoint
Deletes an endpoint configuration
Deletes an Amazon SageMaker experiment
Deletes the specified flow definition
Deletes a model
Deletes a model package
Deletes a monitoring schedule
Deletes an Amazon SageMaker notebook instance
Deletes a notebook instance lifecycle configuration
Deletes the specified tags from an Amazon SageMaker resource
Deletes a notebook instance lifecycle configuration
Deletes the specified trial
Deletes the specified trial component
Deletes a user profile
Deletes an existing work team
Returns a description of the specified algorithm that is in your account
Describes the app
Returns information about an Amazon SageMaker job
Gets details about the specified Git repository
Returns information about a model compilation job
The description of the domain
Returns the description of an endpoint
Returns the description of an endpoint configuration created using the CreateEndpointConfig API
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>describe_experiment</td>
<td>Provides a list of an experiment’s properties</td>
</tr>
<tr>
<td>describe_flow_definition</td>
<td>Returns information about the specified flow definition</td>
</tr>
<tr>
<td>describe_human_task_ui</td>
<td>Returns information about the requested human task user interface</td>
</tr>
<tr>
<td>describe_hyper_parameter_tuning_job</td>
<td>Gets a description of a hyperparameter tuning job</td>
</tr>
<tr>
<td>describe_labeling_job</td>
<td>Gets information about a labeling job</td>
</tr>
<tr>
<td>describe_model</td>
<td>Describes a model that you created using the CreateModel API</td>
</tr>
<tr>
<td>describe_model_package</td>
<td>Returns a description of the specified model package, which is used to create a model</td>
</tr>
<tr>
<td>describe_monitoring_schedule</td>
<td>Describes the schedule for a monitoring job</td>
</tr>
<tr>
<td>describe_notebook_instance</td>
<td>Returns information about a notebook instance</td>
</tr>
<tr>
<td>describe_notebook_instance_lifecycle_config</td>
<td>Provides a list of a notebook instance lifecycle configuration</td>
</tr>
<tr>
<td>describe_processing_job</td>
<td>Returns a description of a processing job</td>
</tr>
<tr>
<td>describe_subscribed_workteam</td>
<td>Gets information about a work team provided by a vendor</td>
</tr>
<tr>
<td>describe_training_job</td>
<td>Returns information about a training job</td>
</tr>
<tr>
<td>describe_transform_job</td>
<td>Returns information about a transform job</td>
</tr>
<tr>
<td>describe_trial</td>
<td>Provides a list of a trial’s properties</td>
</tr>
<tr>
<td>describe_trial_component</td>
<td>Provides a list of a trials component’s properties</td>
</tr>
<tr>
<td>describe_user_profile</td>
<td>Describes the user profile</td>
</tr>
<tr>
<td>describe_workteam</td>
<td>Gets information about a specific work team</td>
</tr>
<tr>
<td>disassociate_trial_component</td>
<td>Disassociates a trial component from a trial</td>
</tr>
<tr>
<td>get_search_suggestions</td>
<td>An auto-complete API for the search functionality in the Amazon SageMaker</td>
</tr>
<tr>
<td>list_algorithms</td>
<td>Lists the machine learning algorithms that have been created</td>
</tr>
<tr>
<td>list_apps</td>
<td>Lists apps</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_domains</td>
<td>Lists the domains</td>
</tr>
<tr>
<td>list_endpoints</td>
<td>Lists endpoint configurations</td>
</tr>
<tr>
<td>list_endpoint_configs</td>
<td>Lists endpoints</td>
</tr>
<tr>
<td>list_experiments</td>
<td>Lists all the experiments in your account</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</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_packages</td>
<td>Lists the model packages that have been created</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 CreateModel API</td>
</tr>
<tr>
<td>list_notebook_instances</td>
<td>Returns a list of the Amazon SageMaker notebook instances in the request</td>
</tr>
<tr>
<td>list_processing_jobs</td>
<td>Lists processing jobs that satisfy various filters</td>
</tr>
<tr>
<td>list_subscribed_workteams</td>
<td>Gets a list of the work teams that you are subscribed to in the AWS Market</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</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>
</tbody>
</table>
The Amazon SageMaker runtime API.

- List the trials in your account
- Lists user profiles
- Gets a list of work teams that you have defined in a region
- 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
- A method for forcing the termination of a running job
- Stops a model compilation 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
- Stops a processing job
- Stops a training job
- Stops a transform job
- Updates the specified Git repository with the specified values
- Updates a 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
- Updates a previously created schedule
- Updates a notebook instance lifecycle configuration created with the CreateNotebookInstanceLifecycleConfig API
- Updates a notebook instance lifecycle configuration created with the CreateNotebookInstanceLifecycleConfig API
- Updates the display name of a trial
- Updates one or more properties of a trial component
- Updates a user profile
- Updates an existing work team with new member definitions or descriptions

Examples

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

## End(Not run)
```

---

**Amazon SageMaker Runtime**

The Amazon SageMaker runtime API.
Usage
sagemakerruntime(config = list())

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

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 applications use this API to get inferences from the model hosted at the specified endpoint

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.

Service syntax

svc <- extract(
  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

gt_document_analysis Gets the results for an Amazon Textract asynchronous operation that analyzes text in a document

gt_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

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

## End(Not run)
### Amazon Transcribe Service

**Description**

Operations and objects for transcribing speech to text.

**Usage**

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

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Optional configuration of credentials, endpoint, and/or region.</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>create_vocabulary</td>
<td>Creates a new custom vocabulary that you can use to change the way Amazon Transcribe handles transcription.</td>
</tr>
<tr>
<td>create_vocabulary_filter</td>
<td>Creates a new vocabulary filter that you can use to filter words, such as profane words, from the output.</td>
</tr>
<tr>
<td>delete_transcription_job</td>
<td>Deletes a previously submitted transcription job along with any other generated results such as the transcription.</td>
</tr>
<tr>
<td>delete_vocabulary</td>
<td>Deletes a vocabulary from Amazon Transcribe.</td>
</tr>
<tr>
<td>delete_vocabulary_filter</td>
<td>Removes a vocabulary filter.</td>
</tr>
<tr>
<td>get_transcription_job</td>
<td>Returns information about a transcription job.</td>
</tr>
<tr>
<td>get_vocabulary</td>
<td>Gets information about a vocabulary.</td>
</tr>
<tr>
<td>get_vocabulary_filter</td>
<td>Returns information about a vocabulary filter.</td>
</tr>
<tr>
<td>list_transcription_jobs</td>
<td>Lists transcription jobs with the specified status.</td>
</tr>
<tr>
<td>list_vocabulary</td>
<td>Returns a list of vocabularies that match the specified criteria.</td>
</tr>
<tr>
<td>list_vocabulary_filters</td>
<td>Gets information about vocabulary filters.</td>
</tr>
<tr>
<td>start_transcription_job</td>
<td>Starts an asynchronous job to transcribe speech to text.</td>
</tr>
<tr>
<td>update_vocabulary</td>
<td>Updates an existing vocabulary with new values.</td>
</tr>
<tr>
<td>update_vocabulary_filter</td>
<td>Updates a vocabulary filter with a new list of filtered words.</td>
</tr>
</tbody>
</table>
## Examples

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

# End(Not run)
```

## Description

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

## Usage

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

## Arguments

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

## Service syntax

```r
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

- **delete_terminology**: A synchronous action that deletes a custom terminology
- **describe_text_translation_job**: Gets the properties associated with an asynchronous batch translation job including name, ID, status, and so on
- **get_terminology**: Retrieves a custom terminology
- **import_terminology**: Creates or updates a custom terminology, depending on whether or not one already exists for the given terminology name
- **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

Examples

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

## End(Not run)
```
Index

add_tags, 10, 18
analyze_document, 23
associate_trial_component, 18

batch_detect_dominant_language, 3
batch_detect_entities, 3
batch_detect_key_phrases, 3
batch_detect_sentiment, 3
batch_detect_syntax, 3

classify_document, 3
comprehend, 2
comprehendmedical, 4
create_algorithm, 18
create_app, 18
create_auto_ml_job, 18
create_batch_inference_job, 11
create_batch_prediction, 10
create_bot_version, 6
create_campaign, 11
create_code_repository, 18
create_collection, 16
create_compilation_job, 18
create_data_source_from_rds, 10
create_data_source_from_redshift, 10
create_data_source_from_s3, 10
create_dataset, 11
create_dataset_group, 11
create_dataset_import_job, 11
create_document_classifier, 3
create_domain, 19
create_endpoint, 3, 19
create_endpoint_config, 19
create_entity_recognizer, 3
create_evaluation, 10
create_event_tracker, 11
create_experiment, 19
create_flow_definition, 19
create_human_task_ui, 19
create_hyper_parameter_tuning_job, 19
create_intent_version, 6
create_labeling_job, 19
create_ml_model, 10
create_model, 19
create_model_package, 19
create_monitoring_schedule, 19
create_notebook_instance, 19
create_notebook_instance_lifecycle_config, 19
create_presigned_domain_url, 19
create_presigned_notebook_instance_url, 19
create_processing_job, 19
create_project, 16
create_project_version, 16
create_realtime_endpoint, 10
create_schema, 11
create_slot_type_version, 6
create_solution, 11
create_solution_version, 11
create_stream_processor, 16
create_training_job, 19
create_transform_job, 19
create_trial, 19
create_trial_component, 19
create_user_profile, 19
create_vocabulary, 24
create_vocabulary_filter, 24
create_workteam, 19
delete_algorithm, 19
delete_app, 19
delete_batch_prediction, 10
delete_bot, 6
delete_bot_alias, 6
delete_bot_channel_association, 7
delete_bot_version, 7
delete_campaign, 11
delete_code_repository, 19
delete_collection, 16
delete_data_source, 10
delete_dataset, 11
delete_dataset_group, 11
delete_document_classifier, 3
delete_domain, 19
delete_endpoint, 3, 19
delete_endpoint_config, 19
delete_entity_recognizer, 3
delete_evaluation, 10
delete_event_tracker, 11
delete_experiment, 19
delete_faces, 16
delete_flow_definition, 19
delete_intent, 7
delete_intent_version, 7
delete_lexicon, 15
delete_ml_model, 10
delete_model, 19
delete_model_package, 19
delete_monitoring_schedule, 19
delete_notebook_instance, 19
delete_notebook_instance_lifecycle_config, 19
delete_realtime_endpoint, 10
delete_schema, 11
delete_session, 8
delete_slot_type, 7
delete_slot_type_version, 7
delete_solution, 11
delete_stream_processor, 16
delete_tags, 10, 19
delete_lexicon, 15
delete_terminology, 26
delete_transcription_job, 24
delete_trial, 19
delete_trial_component, 19
delete_user_profile, 19
delete_utterances, 7
delete_vocabulary, 24
delete_vocabulary_filter, 24
delete_workteam, 19
describe_algorithm, 11, 19
describe_app, 19
describe_auto_ml_job, 19
describe_batch_inference_job, 11
describe_batch_predictions, 10
describe_campaign, 12
describe_code_repository, 19
describe_collection, 16
describe_compilation_job, 19
describe_data_sources, 10
describe_dataset, 12
describe_dataset_group, 12
describe_dataset_import_job, 12
describe_document_classification_job, 3
describe_document_classifier, 3
describe_domain, 19
describe_dominant_language_detection_job, 3
describe_endpoint, 3, 19
describe_endpoint_config, 19
describe_entities_detection_job, 19
describe_entities_detection_v2_job, 5
describe_entity_recognizer, 3
describe_evaluations, 10
describe_event_tracker, 12
describe_experiment, 20
describe_feature_transformation, 12
describe_flow_definition, 20
describe_flow_definition, 20
describe_human_task_ui, 20
describe_hyper_parameter_tuning_job, 20
describe_key_phrases_detection_job, 3
describe_labeling_job, 20
describe_ml_models, 10
describe_model, 20
describe_model_package, 20
describe_monitoring_schedule, 20
describe_notebook_instance, 20
describe_notebook_instance_lifecycle_config, 20
describe_phi_detection_job, 5
describe_processing_job, 20
describe_project_versions, 16
describe_projects, 16
describe_recipe, 12
describe_schema, 12
describe_sentiment_detection_job, 3
describe_solution, 12
describe_solution_version, 12
describe_stream_processor, 16
describe_subscribed_workteam, 20
describe_tags, 10
describe_text_translation_job, 26
describe_topics_detection_job, 3
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>describe_training_job</td>
<td>20</td>
</tr>
<tr>
<td>describe_transform_job</td>
<td>20</td>
</tr>
<tr>
<td>describe_trial</td>
<td>20</td>
</tr>
<tr>
<td>describe_trial_component</td>
<td>20</td>
</tr>
<tr>
<td>describe_user_profile</td>
<td>20</td>
</tr>
<tr>
<td>describe_voices</td>
<td>15</td>
</tr>
<tr>
<td>describe_workteam</td>
<td>20</td>
</tr>
<tr>
<td>describe_document_text</td>
<td>23</td>
</tr>
<tr>
<td>detect_custom_labels</td>
<td>16</td>
</tr>
<tr>
<td>detect_dominant_language</td>
<td>3</td>
</tr>
<tr>
<td>detect_entities</td>
<td>3</td>
</tr>
<tr>
<td>detect_entities_v2</td>
<td>5</td>
</tr>
<tr>
<td>detect_faces</td>
<td>17</td>
</tr>
<tr>
<td>detect_key_phrases</td>
<td>3</td>
</tr>
<tr>
<td>detect_labels</td>
<td>17</td>
</tr>
<tr>
<td>detect_moderation_labels</td>
<td>17</td>
</tr>
<tr>
<td>detect_phi</td>
<td>5</td>
</tr>
<tr>
<td>detect_sentiment</td>
<td>3</td>
</tr>
<tr>
<td>detect_syntax</td>
<td>3</td>
</tr>
<tr>
<td>detect_text</td>
<td>17</td>
</tr>
<tr>
<td>disassociate_trial_component</td>
<td>20</td>
</tr>
<tr>
<td>get_batch_prediction</td>
<td>10</td>
</tr>
<tr>
<td>get_bot</td>
<td>7</td>
</tr>
<tr>
<td>get_bot_alias</td>
<td>7</td>
</tr>
<tr>
<td>get_bot_aliases</td>
<td>7</td>
</tr>
<tr>
<td>get_bot_channel_association</td>
<td>7</td>
</tr>
<tr>
<td>get_bot_channel_associations</td>
<td>7</td>
</tr>
<tr>
<td>get_bot_versions</td>
<td>7</td>
</tr>
<tr>
<td>get_bots</td>
<td>7</td>
</tr>
<tr>
<td>get_builtin_intent</td>
<td>7</td>
</tr>
<tr>
<td>get_builtin_intents</td>
<td>7</td>
</tr>
<tr>
<td>get_builtin_slot_types</td>
<td>7</td>
</tr>
<tr>
<td>get_terminology</td>
<td>26</td>
</tr>
<tr>
<td>get_lexicon</td>
<td>15</td>
</tr>
<tr>
<td>get_ml_model</td>
<td>10</td>
</tr>
<tr>
<td>get_person_tracking</td>
<td>17</td>
</tr>
<tr>
<td>get_personalized_ranking</td>
<td>14</td>
</tr>
<tr>
<td>get_recommendations</td>
<td>14</td>
</tr>
<tr>
<td>get_search_suggestions</td>
<td>20</td>
</tr>
<tr>
<td>get_session</td>
<td>8</td>
</tr>
<tr>
<td>get_slot_type</td>
<td>7</td>
</tr>
<tr>
<td>get_slot_type_versions</td>
<td>7</td>
</tr>
<tr>
<td>get_slot_types</td>
<td>7</td>
</tr>
<tr>
<td>get_solution_metrics</td>
<td>12</td>
</tr>
<tr>
<td>get_speech_synthesis_task</td>
<td>15</td>
</tr>
<tr>
<td>get_utterances_view</td>
<td>7</td>
</tr>
<tr>
<td>get_vocabulary</td>
<td>24</td>
</tr>
<tr>
<td>get_vocabulary_filter</td>
<td>24</td>
</tr>
<tr>
<td>get_celebrity_info</td>
<td>17</td>
</tr>
<tr>
<td>get_celebrity_recognition</td>
<td>17</td>
</tr>
<tr>
<td>get_content_moderation</td>
<td>17</td>
</tr>
<tr>
<td>get_data_source</td>
<td>10</td>
</tr>
<tr>
<td>get_document_analysis</td>
<td>23</td>
</tr>
<tr>
<td>get_document_text_detection</td>
<td>23</td>
</tr>
<tr>
<td>get_evaluation</td>
<td>10</td>
</tr>
<tr>
<td>get_export</td>
<td>7</td>
</tr>
<tr>
<td>get_face_detection</td>
<td>17</td>
</tr>
<tr>
<td>get_face_search</td>
<td>17</td>
</tr>
<tr>
<td>get_import</td>
<td>7</td>
</tr>
<tr>
<td>get_intent</td>
<td>7</td>
</tr>
<tr>
<td>get_intent_versions</td>
<td>7</td>
</tr>
<tr>
<td>get_intents</td>
<td>7</td>
</tr>
<tr>
<td>get_label_detection</td>
<td>17</td>
</tr>
<tr>
<td>get_lexicon</td>
<td>15</td>
</tr>
<tr>
<td>import_terminology</td>
<td>26</td>
</tr>
<tr>
<td>index_faces</td>
<td>17</td>
</tr>
<tr>
<td>infer_icd10cm</td>
<td>5</td>
</tr>
<tr>
<td>infer_rx_norm</td>
<td>5</td>
</tr>
<tr>
<td>invoke_endpoint</td>
<td>22</td>
</tr>
<tr>
<td>lexmodelbuildingservice</td>
<td>6</td>
</tr>
<tr>
<td>lexruntimeservice</td>
<td>8</td>
</tr>
<tr>
<td>list_algorithms</td>
<td>20</td>
</tr>
<tr>
<td>list_apps</td>
<td>20</td>
</tr>
<tr>
<td>list_auto_ml_jobs</td>
<td>20</td>
</tr>
<tr>
<td>list_batch_inference_jobs</td>
<td>12</td>
</tr>
<tr>
<td>list_campaigns</td>
<td>12</td>
</tr>
<tr>
<td>list_candidates_for_auto_ml_job</td>
<td>20</td>
</tr>
<tr>
<td>list_code_repositories</td>
<td>20</td>
</tr>
<tr>
<td>list_collections</td>
<td>17</td>
</tr>
<tr>
<td>list_compilation_jobs</td>
<td>20</td>
</tr>
<tr>
<td>list_dataset_groups</td>
<td>12</td>
</tr>
<tr>
<td>list_dataset_import_jobs</td>
<td>12</td>
</tr>
<tr>
<td>list_datasets</td>
<td>12</td>
</tr>
<tr>
<td>list_document_classification_jobs</td>
<td>4</td>
</tr>
<tr>
<td>list_document_classifiers</td>
<td>4</td>
</tr>
<tr>
<td>list_domains</td>
<td>20</td>
</tr>
<tr>
<td>list_domainalanguage_detection_jobs</td>
<td>4</td>
</tr>
<tr>
<td>list_endpoint_configs</td>
<td>20</td>
</tr>
<tr>
<td>list_endpoints</td>
<td>4, 20</td>
</tr>
<tr>
<td>list_entities_detection_jobs</td>
<td>4</td>
</tr>
<tr>
<td>list_entities_detection_v2_jobs</td>
<td>5</td>
</tr>
<tr>
<td>list_entity_recognizers</td>
<td>4</td>
</tr>
<tr>
<td>list_event_trackers</td>
<td>12</td>
</tr>
<tr>
<td>list_experiments</td>
<td>20</td>
</tr>
</tbody>
</table>
list_faces, 17
list_flow_definitions, 20
list_human_task_uis, 20
list_hyper_parameter_tuning_jobs, 20
list_key_phrases_detection_jobs, 4
list_labeling_jobs, 20
list_labeling_jobs_for_workteam, 20
list_lexicons, 15
list_model_packages, 20
list_models, 20
list_monitoring_executions, 20
list_monitoring_schedules, 20
list_notebook_instance_lifecycle_configs, 20
list_notebook_instances, 20
list_phi_detection_jobs, 5
list_processing_jobs, 20
list_recipes, 12
list_schemas, 12
list_sentiment_detection_jobs, 4
list_solution_versions, 12
list_solutions, 12
list_speech_synthesis_tasks, 15
list_stream_processors, 17
list_subscribed_workteams, 20
list_tags, 20
list_tags_for_resource, 4
list_terminologies, 26
list_text_translation_jobs, 26
list_topics_detection_jobs, 4
list_training_jobs, 20
list_training_jobs_for_hyper_parameter_tuning_job, 20
list_transcription_jobs, 24
list_transform_jobs, 20
list_trial_components, 20
list_trials, 21
list_user_profiles, 21
list_vocabulary_filters, 24
list_workteams, 21

machinelearning, 9

personalize, 10
personalizeevents, 12
personalizeruntime, 13
polly, 14
post_content, 8

post_text, 8
predict, 10
put_bot, 7
put_bot_alias, 7
put_events, 13
put_intent, 7
put_lexicon, 15
put_session, 8
put_slot_type, 7
recognize_celebrities, 17
rekognition, 16
render_ui_template, 21
sagemaker, 18
sagemakerruntime, 21
search, 21
search_faces, 17
search_faces_by_image, 17
start_celebrity_recognition, 17
start_content_moderation, 17
start_document_analysis, 23
start_document_classification_job, 4
start_document_text_detection, 23
start_dominant_language_detection_job, 4
start_entities_detection_job, 4
start_entities_detection_v2_job, 5
start_face_detection, 17
start_face_search, 17
start_import, 7
start_key_phrases_detection_job, 4
start_label_detection, 17
start_monitoring_schedule, 21
start_notebook_instance, 21
start_person_tracking, 17
start_phi_detection_job, 5
start_project_version, 17
start_sentiment_detection_job, 4
start_speech_synthesis_task, 15
start_stream_processor, 17
start_text_translation_job, 26
start_topics_detection_job, 4
start_transcription_job, 24
stop_auto_ml_job, 21
stop_compilation_job, 21
stop_dominant_language_detection_job, 4
stop_entities_detection_job, 4
stop_entities_detection_v2_job, 5
stop_hyper_parameter_tuning_job, 21
stop_key_phrases_detection_job, 4
stop_labeling_job, 21
stop_monitoring_schedule, 21
stop_notebook_instance, 21
stop_phi_detection_job, 5
stop_processing_job, 21
stop_project_version, 17
stop_sentiment_detection_job, 4
stop_stream_processor, 17
stop_text_translation_job, 26
stop_training_document_classifier, 4
stop_training_entity_recognizer, 4
stop_training_job, 21
stop_transform_job, 21
synthesize_speech, 15
tag_resource, 4
textract, 22
transcribeservice, 24
translate, 25
translate_text, 26
untag_resource, 4
update_batch_prediction, 10
update_campaign, 12
update_code_repository, 21
update_data_source, 10
update_domain, 21
update_endpoint, 4, 21
update_endpoint_weights_and_capacities, 21
update_evaluation, 10
update_experiment, 21
update_ml_model, 10
update_monitoring_schedule, 21
update_notebook_instance, 21
update_notebook_instance_lifecycle_config, 21
update_trial, 21
update_trial_component, 21
update_user_profile, 21
update_vocabulary, 24
update_vocabulary_filter, 24
update_workteam, 21