

Package ‘paws.machine.learning’

January 14, 2020

Title Amazon Web Services Machine Learning APIs

Version 0.1.6

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 (>= 0.2.5)

Suggests testthat

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

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'
'lexruntime_service.R' 'lexruntime_interfaces.R'
'lexruntime_operations.R' 'machinelearning_service.R'
'machinelearning_interfaces.R' 'machinelearning_operations.R'
'personalize_service.R' 'personalize_interfaces.R'
'personalize_operations.R' 'personalizeevents_service.R'
'personalizeevents_interfaces.R'
'personalizeevents_operations.R' 'personalizeruntime_service.R'
'personalizeruntime_interfaces.R'
'personalizeruntime_operations.R' 'polly_service.R'
'polly_interfaces.R' 'polly_operations.R'
'rekognition_service.R' 'rekognition_interfaces.R'
'rekognition_operations.R' 'sagemaker_service.R'
'sagemaker_interfaces.R' 'sagemaker_operations.R'
'sagemakerruntime_service.R' 'sagemakerruntime_interfaces.R'

```
'sagemakerruntime_operations.R' 'textract_service.R'
'textract_interfaces.R' 'textract_operations.R'
'transcribeservice_service.R' 'transcribeservice_interfaces.R'
'transcribeservice_operations.R' 'translate_service.R'
'translate_interfaces.R' 'translate_operations.R'
```

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2020-01-14 20:20:02 UTC

R topics documented:

comprehend	2
comprehendmedical	4
lexmodelbuildingservice	6
lexruntime_service	7
machinelearning	9
personalize	10
personalizeevents	12
personalizeruntime	13
polly	14
rekognition	15
sagemaker	17
sagemakerruntime	21
textract	22
transcribeservice	23
translate	24
Index	26

comprehend	<i>Amazon Comprehend</i>
------------	--------------------------

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
batch_detect_syntax	Inspects the text of a batch of documents for the syntax and part of speech of the words
classify_document	Creates a new document classification request to analyze a single document in real time
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 the active state
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 files, and other information
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, NEGATIVE)
detect_syntax	Inspects text for syntax and the part of speech of words in the document

<code>list_document_classification_jobs</code>	Gets a list of the documentation classification jobs that you have submitted
<code>list_document_classifiers</code>	Gets a list of the document classifiers that you have created
<code>list_dominant_language_detection_jobs</code>	Gets a list of the dominant language detection jobs that you have submitted
<code>list_endpoints</code>	Gets a list of all existing endpoints that you've created
<code>list_entities_detection_jobs</code>	Gets a list of the entity detection jobs that you have submitted
<code>list_entity_recognizers</code>	Gets a list of the properties of all entity recognizers that you created, including
<code>list_key_phrases_detection_jobs</code>	Get a list of key phrase detection jobs that you have submitted
<code>list_sentiment_detection_jobs</code>	Gets a list of sentiment detection jobs that you have submitted
<code>list_tags_for_resource</code>	Lists all tags associated with a given Amazon Comprehend resource
<code>list_topics_detection_jobs</code>	Gets a list of the topic detection jobs that you have submitted
<code>start_document_classification_job</code>	Starts an asynchronous document classification job
<code>start_dominant_language_detection_job</code>	Starts an asynchronous dominant language detection job for a collection of doc
<code>start_entities_detection_job</code>	Starts an asynchronous entity detection job for a collection of documents
<code>start_key_phrases_detection_job</code>	Starts an asynchronous key phrase detection job for a collection of documents
<code>start_sentiment_detection_job</code>	Starts an asynchronous sentiment detection job for a collection of documents
<code>start_topics_detection_job</code>	Starts an asynchronous topic detection job
<code>stop_dominant_language_detection_job</code>	Stops a dominant language detection job in progress
<code>stop_entities_detection_job</code>	Stops an entities detection job in progress
<code>stop_key_phrases_detection_job</code>	Stops a key phrases detection job in progress
<code>stop_sentiment_detection_job</code>	Stops a sentiment detection job in progress
<code>stop_training_document_classifier</code>	Stops a document classifier training job while in progress
<code>stop_training_entity_recognizer</code>	Stops an entity recognizer training job while in progress
<code>tag_resource</code>	Associates a specific tag with an Amazon Comprehend resource
<code>untag_resource</code>	Removes a specific tag associated with an Amazon Comprehend resource
<code>update_endpoint</code>	Updates information about the specified endpoint

Examples

```
svc <- comprehend()
svc$batch_detect_dominant_language(
  Foo = 123
)
```

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
detect_phi	Inspects the clinical text for protected health information (PHI) entities and entity categories
infer_icd10cm	InferICD10CM detects medical conditions as entities listed in a patient record and links to the norm
infer_rx_norm	InferRxNorm detects medications as entities listed in a patient record and links to the norm
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

```
svc <- comprehendmedical()
svc$describe_entities_detection_v2_job(
  Foo = 123
)
```

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.

Service syntax

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

Operations

create_bot_version	Creates a new version of the bot based on the \$LATEST version
create_intent_version	Creates a new version of an intent based on the \$LATEST version of the intent
create_slot_type_version	Creates a new version of a slot type based on the \$LATEST version of the specified slot 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
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	Returns bot information as follows: - If you provide the nameContains field, the response includes only bots whose names contain the specified value.
get_bot_versions	Gets information about all of the versions of a bot
get_builtin_intent	Returns information about a built-in intent
get_builtin_intents	Gets a list of built-in intents that meet the specified criteria
get_builtin_slot_types	Gets a list of built-in slot types that meet the specified criteria
get_export	Exports the contents of an Amazon Lex resource in a specified format
get_import	Gets information about an import job started with the StartImport operation
get_intent	Returns information about an intent
get_intents	Returns intent information as follows: - If you specify the nameContains field, returns the intents whose names contain the specified value.
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_type_version	Returns slot type information as follows: - If you specify the nameContains field, returns the slot types whose names contain the specified value.
get_slot_type_versions	Gets information about all versions of a slot type
get_utterances_view	Use the GetUtterancesView operation to get information about the utterances that your user has made to the specified Amazon Lex bot.
put_bot	Creates an Amazon Lex conversational bot or replaces an existing bot
put_bot_alias	Creates an alias for the specified version of the bot or replaces an alias for the specified bot
put_intent	Creates an intent or replaces an existing intent
put_slot_type	Creates a custom slot type or replaces an existing custom slot type
start_import	Starts a job to import a resource to Amazon Lex

Examples

```
# This example shows how to get configuration information for a bot.
svc <- lexmodelbuildingservice()
svc$get_bot(
  name = "DocOrderPizza",
  versionOrAlias = "$LATEST"
)
```

lexruntime-service

Amazon Lex Runtime Service

Description

Amazon Lex provides both build and runtime endpoints. Each endpoint provides a set of operations (API). Your conversational bot uses the runtime API to understand user utterances (user input text).

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

Usage

```
lexruntime-service(config = list())
```

Arguments

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

Service syntax

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

Operations

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

Examples

```
svc <- lexruntime-service()
svc$delete_session(
  Foo = 123
)
```

machinelearning	<i>Amazon Machine Learning</i>
-----------------	--------------------------------

Description

Definition of the public APIs exposed by Amazon Machine Learning

Usage

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

Arguments

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

Service syntax

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

Operations

add_tags	Adds one or more tags to an object, up to a limit of 10
create_batch_prediction	Generates predictions for a group of observations
create_data_source_from_rds	Creates a DataSource object from an Amazon Relational Database Service (Amazon RDS)
create_data_source_from_redshift	Creates a DataSource from a database hosted on an Amazon Redshift cluster
create_data_source_from_s3	Creates a DataSource object
create_evaluation	Creates a new Evaluation of an MLModel
create_ml_model	Creates a new MLModel using the DataSource and the recipe as information sources
create_realtime_endpoint	Creates a real-time endpoint for the MLModel
delete_batch_prediction	Assigns the DELETED status to a BatchPrediction, rendering it unusable
delete_data_source	Assigns the DELETED status to a DataSource, rendering it unusable
delete_evaluation	Assigns the DELETED status to an Evaluation, rendering it unusable

<code>delete_ml_model</code>	Assigns the DELETED status to an MLModel, rendering it unusable
<code>delete_realtime_endpoint</code>	Deletes a real time endpoint of an MLModel
<code>delete_tags</code>	Deletes the specified tags associated with an ML object
<code>describe_batch_predictions</code>	Returns a list of BatchPrediction operations that match the search criteria in the request
<code>describe_data_sources</code>	Returns a list of DataSource that match the search criteria in the request
<code>describe_evaluations</code>	Returns a list of DescribeEvaluations that match the search criteria in the request
<code>describe_ml_models</code>	Returns a list of MLModel that match the search criteria in the request
<code>describe_tags</code>	Describes one or more of the tags for your Amazon ML object
<code>get_batch_prediction</code>	Returns a BatchPrediction that includes detailed metadata, status, and data file information
<code>get_data_source</code>	Returns a DataSource that includes metadata and data file information, as well as the current status of the DataSource
<code>get_evaluation</code>	Returns an Evaluation that includes metadata as well as the current status of the Evaluation
<code>get_ml_model</code>	Returns an MLModel that includes detailed metadata, data source information, and the current status of the MLModel
<code>predict</code>	Generates a prediction for the observation using the specified ML Model
<code>update_batch_prediction</code>	Updates the BatchPredictionName of a BatchPrediction
<code>update_data_source</code>	Updates the DataSourceName of a DataSource
<code>update_evaluation</code>	Updates the EvaluationName of an Evaluation
<code>update_ml_model</code>	Updates the MLModelName and the ScoreThreshold of an MLModel

Examples

```
svc <- machinelearning()
svc$add_tags(
  Foo = 123
)
```

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

```

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), and creation time
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 name, Amazon Resource Name (ARN), and creation time
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

Examples

```
svc <- personalize()
svc$create_batch_inference_job(
  Foo = 123
)
```

personalizeevents *Amazon Personalize Events*

Description

Amazon Personalize Events

Usage

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

Arguments

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

Service syntax

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

```
    region = "string"  
  )  
)
```

Operations

[put_events](#) Records user interaction event data

Examples

```
svc <- personalizeevents()  
svc$put_events(  
  Foo = 123  
)
```

personalizeruntime *Amazon Personalize Runtime*

Description

Amazon Personalize Runtime

Usage

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

Arguments

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

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

<code>get_personalized_ranking</code>	Re-ranks a list of recommended items for the given user
<code>get_recommendations</code>	Returns a list of recommended items

Examples

```
svc <- personalizeruntime()
svc$get_personalized_ranking(
  Foo = 123
)
```

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

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

Examples

```
# Deletes a specified pronunciation lexicon stored in an AWS Region.
svc <- polly()
svc$delete_lexicon(
  Name = "example"
)
```

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.

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 <i>source</i> input image with each of the 100 largest faces detected in the <i>target</i> 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	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_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 Video analysis
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
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 the input image
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_stream_processor	Starts processing a stream processor

stop_project_version	Stops a running model
stop_stream_processor	Stops a running stream processor that was created by CreateStreamProcessor

Examples

```
# This operation compares the largest face detected in the source image
# with each face detected in the target image.
svc <- rekognition()
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

<code>add_tags</code>	Adds or overwrites one or more tags for the specified Amazon SageMaker resource
<code>associate_trial_component</code>	Associates a trial component with a trial
<code>create_algorithm</code>	Create a machine learning algorithm that you can use in Amazon SageMaker
<code>create_app</code>	Creates a running App for the specified UserProfile
<code>create_auto_ml_job</code>	Creates an AutoPilot job
<code>create_code_repository</code>	Creates a Git repository as a resource in your Amazon SageMaker account
<code>create_compilation_job</code>	Starts a model compilation job
<code>create_domain</code>	Creates a Domain for Amazon SageMaker Amazon SageMaker Studio
<code>create_endpoint</code>	Creates an endpoint using the endpoint configuration specified in the request
<code>create_endpoint_config</code>	Creates an endpoint configuration that Amazon SageMaker hosting service uses to serve the model
<code>create_experiment</code>	Creates an Amazon SageMaker <i>experiment</i>
<code>create_flow_definition</code>	Creates a flow definition
<code>create_human_task_ui</code>	Defines the settings you will use for the human review workflow user interface
<code>create_hyper_parameter_tuning_job</code>	Starts a hyperparameter tuning job
<code>create_labeling_job</code>	Creates a job that uses workers to label the data objects in your input dataset
<code>create_model</code>	Creates a model in Amazon SageMaker
<code>create_model_package</code>	Creates a model package that you can use to create Amazon SageMaker endpoints
<code>create_monitoring_schedule</code>	Creates a schedule that regularly starts Amazon SageMaker Processing Jobs
<code>create_notebook_instance</code>	Creates an Amazon SageMaker notebook instance
<code>create_notebook_instance_lifecycle_config</code>	Creates a lifecycle configuration that you can associate with a notebook instance
<code>create_presigned_domain_url</code>	Creates a URL for a specified UserProfile in a Domain
<code>create_presigned_notebook_instance_url</code>	Returns a URL that you can use to connect to the Jupyter server from a notebook instance
<code>create_processing_job</code>	Creates a processing job
<code>create_training_job</code>	Starts a model training job
<code>create_transform_job</code>	Starts a transform job
<code>create_trial</code>	Creates an Amazon SageMaker <i>trial</i>
<code>create_trial_component</code>	Creates a <i>trial component</i> , which is a stage of a machine learning <i>trial</i>
<code>create_user_profile</code>	Creates a new user profile
<code>create_workteam</code>	Creates a new work team for labeling your data
<code>delete_algorithm</code>	Removes the specified algorithm from your account
<code>delete_app</code>	Used to stop and delete an app
<code>delete_code_repository</code>	Deletes the specified Git repository from your account
<code>delete_domain</code>	Used to delete a domain
<code>delete_endpoint</code>	Deletes an endpoint
<code>delete_endpoint_config</code>	Deletes an endpoint configuration
<code>delete_experiment</code>	Deletes an Amazon SageMaker experiment

<code>delete_flow_definition</code>	Deletes the specified flow definition
<code>delete_model</code>	Deletes a model
<code>delete_model_package</code>	Deletes a model package
<code>delete_monitoring_schedule</code>	Deletes a monitoring schedule
<code>delete_notebook_instance</code>	Deletes an Amazon SageMaker notebook instance
<code>delete_notebook_instance_lifecycle_config</code>	Deletes a notebook instance lifecycle configuration
<code>delete_tags</code>	Deletes the specified tags from an Amazon SageMaker resource
<code>delete_trial</code>	Deletes the specified trial
<code>delete_trial_component</code>	Deletes the specified trial component
<code>delete_user_profile</code>	Deletes a user profile
<code>delete_workteam</code>	Deletes an existing work team
<code>describe_algorithm</code>	Returns a description of the specified algorithm that is in your account
<code>describe_app</code>	Describes the app
<code>describe_auto_ml_job</code>	Returns information about an Amazon SageMaker job
<code>describe_code_repository</code>	Gets details about the specified Git repository
<code>describe_compilation_job</code>	Returns information about a model compilation job
<code>describe_domain</code>	The description of the domain
<code>describe_endpoint</code>	Returns the description of an endpoint
<code>describe_endpoint_config</code>	Returns the description of an endpoint configuration created using the C
<code>describe_experiment</code>	Provides a list of an experiment's properties
<code>describe_flow_definition</code>	Returns information about the specified flow definition
<code>describe_human_task_ui</code>	Returns information about the requested human task user interface
<code>describe_hyper_parameter_tuning_job</code>	Gets a description of a hyperparameter tuning job
<code>describe_labeling_job</code>	Gets information about a labeling job
<code>describe_model</code>	Describes a model that you created using the CreateModel API
<code>describe_model_package</code>	Returns a description of the specified model package, which is used to c
<code>describe_monitoring_schedule</code>	Describes the schedule for a monitoring job
<code>describe_notebook_instance</code>	Returns information about a notebook instance
<code>describe_notebook_instance_lifecycle_config</code>	Returns a description of a notebook instance lifecycle configuration
<code>describe_processing_job</code>	Returns a description of a processing job
<code>describe_subscribed_workteam</code>	Gets information about a work team provided by a vendor
<code>describe_training_job</code>	Returns information about a training job
<code>describe_transform_job</code>	Returns information about a transform job
<code>describe_trial</code>	Provides a list of a trial's properties
<code>describe_trial_component</code>	Provides a list of a trials component's properties
<code>describe_user_profile</code>	Describes the user profile
<code>describe_workteam</code>	Gets information about a specific work team
<code>disassociate_trial_component</code>	Disassociates a trial component from a trial
<code>get_search_suggestions</code>	An auto-complete API for the search functionality in the Amazon Sage
<code>list_algorithms</code>	Lists the machine learning algorithms that have been created
<code>list_apps</code>	Lists apps
<code>list_auto_ml_jobs</code>	Request a list of jobs
<code>list_candidates_for_auto_ml_job</code>	List the Candidates created for the job
<code>list_code_repositories</code>	Gets a list of the Git repositories in your account
<code>list_compilation_jobs</code>	Lists model compilation jobs that satisfy various filters
<code>list_domains</code>	Lists the domains
<code>list_endpoint_configs</code>	Lists endpoint configurations
<code>list_endpoints</code>	Lists endpoints

<code>list_experiments</code>	Lists all the experiments in your account
<code>list_flow_definitions</code>	Returns information about the flow definitions in your account
<code>list_human_task_uis</code>	Returns information about the human task user interfaces in your account
<code>list_hyper_parameter_tuning_jobs</code>	Gets a list of HyperParameterTuningJobSummary objects that describe the tuning jobs
<code>list_labeling_jobs</code>	Gets a list of labeling jobs
<code>list_labeling_jobs_for_workteam</code>	Gets a list of labeling jobs assigned to a specified work team
<code>list_model_packages</code>	Lists the model packages that have been created
<code>list_models</code>	Lists models created with the CreateModel API
<code>list_monitoring_executions</code>	Returns list of all monitoring job executions
<code>list_monitoring_schedules</code>	Returns list of all monitoring schedules
<code>list_notebook_instance_lifecycle_configs</code>	Lists notebook instance lifecycle configurations created with the CreateNotebookInstanceLifecycleConfig API
<code>list_notebook_instances</code>	Returns a list of the Amazon SageMaker notebook instances in the requested region
<code>list_processing_jobs</code>	Lists processing jobs that satisfy various filters
<code>list_subscribed_workteams</code>	Gets a list of the work teams that you are subscribed to in the AWS Marketplace
<code>list_tags</code>	Returns the tags for the specified Amazon SageMaker resource
<code>list_training_jobs</code>	Lists training jobs
<code>list_training_jobs_for_hyper_parameter_tuning_job</code>	Gets a list of TrainingJobSummary objects that describe the training jobs
<code>list_transform_jobs</code>	Lists transform jobs
<code>list_trial_components</code>	Lists the trial components in your account
<code>list_trials</code>	Lists the trials in your account
<code>list_user_profiles</code>	Lists user profiles
<code>list_workteams</code>	Gets a list of work teams that you have defined in a region
<code>render_ui_template</code>	Renders the UI template so that you can preview the worker's experience
<code>search</code>	Finds Amazon SageMaker resources that match a search query
<code>start_monitoring_schedule</code>	Starts a previously stopped monitoring schedule
<code>start_notebook_instance</code>	Launches an ML compute instance with the latest version of the libraries
<code>stop_auto_ml_job</code>	A method for forcing the termination of a running job
<code>stop_compilation_job</code>	Stops a model compilation job
<code>stop_hyper_parameter_tuning_job</code>	Stops a running hyperparameter tuning job and all running training jobs
<code>stop_labeling_job</code>	Stops a running labeling job
<code>stop_monitoring_schedule</code>	Stops a previously started monitoring schedule
<code>stop_notebook_instance</code>	Terminates the ML compute instance
<code>stop_processing_job</code>	Stops a processing job
<code>stop_training_job</code>	Stops a training job
<code>stop_transform_job</code>	Stops a transform job
<code>update_code_repository</code>	Updates the specified Git repository with the specified values
<code>update_domain</code>	Updates a domain
<code>update_endpoint</code>	Deploys the new EndpointConfig specified in the request, switches to use the new EndpointConfig
<code>update_endpoint_weights_and_capacities</code>	Updates variant weight of one or more variants associated with an existing endpoint
<code>update_experiment</code>	Adds, updates, or removes the description of an experiment
<code>update_monitoring_schedule</code>	Updates a previously created schedule
<code>update_notebook_instance</code>	Updates a notebook instance
<code>update_notebook_instance_lifecycle_config</code>	Updates a notebook instance lifecycle configuration created with the CreateNotebookInstanceLifecycleConfig API
<code>update_trial</code>	Updates the display name of a trial
<code>update_trial_component</code>	Updates one or more properties of a trial component
<code>update_user_profile</code>	Updates a user profile
<code>update_workteam</code>	Updates an existing work team with new member definitions or descriptions

Examples

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

sagemakerruntime

Amazon SageMaker Runtime

Description

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 application

Examples

```
svc <- sagemakerruntime()
svc$invoke_endpoint(
  Foo = 123
)
```

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

Operations

analyze_document	Analyzes an input document for relationships between detected items
detect_document_text	Detects text in the input document
get_document_analysis	Gets the results for an Amazon Textract asynchronous operation that analyzes text in a 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

```

svc <- textract()
svc$analyze_document(
  Foo = 123
)

```

transcribeservice *Amazon Transcribe Service*

Description

Operations and objects for transcribing speech to text.

Usage

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

Arguments

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

Service syntax

```

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

```

Operations

create_vocabulary	Creates a new custom vocabulary that you can use to change the way Amazon Transcribe handles
create_vocabulary_filter	Creates a new vocabulary filter that you can use to filter words, such as profane words, from the ou
delete_transcription_job	Deletes a previously submitted transcription job along with any other generated results such as the
delete_vocabulary	Deletes a vocabulary from Amazon Transcribe

delete_vocabulary_filter	Removes a vocabulary filter
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_transcription_jobs	Lists transcription jobs with the specified status
list_vocabularies	Returns a list of vocabularies that match the specified criteria
list_vocabulary_filters	Gets information about vocabulary filters
start_transcription_job	Starts an asynchronous job to transcribe speech to text
update_vocabulary	Updates an existing vocabulary with new values
update_vocabulary_filter	Updates a vocabulary filter with a new list of filtered words

Examples

```
svc <- transcribeservice()
svc$create_vocabulary(
  Foo = 123
)
```

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.

Service syntax

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



```
    endpoint = "string",  
    region = "string"  
  )  
)
```

Operations

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,
get_terminology	Retrieves a custom terminology
import_terminology	Creates or updates a custom terminology, depending on whether or not one already exists for t
list_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

```
svc <- translate()  
svc$delete_terminology(  
  Foo = 123  
)
```

Index

add_tags, [9](#), [18](#)
analyze_document, [22](#)
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](#)
compare_faces, [16](#)
comprehend, [2](#)
comprehendmedical, [4](#)
create_algorithm, [18](#)
create_app, [18](#)
create_auto_ml_job, [18](#)
create_batch_inference_job, [11](#)
create_batch_prediction, [9](#)
create_bot_version, [6](#)
create_campaign, [11](#)
create_code_repository, [18](#)
create_collection, [16](#)
create_compilation_job, [18](#)
create_data_source_from_rds, [9](#)
create_data_source_from_redshift, [9](#)
create_data_source_from_s3, [9](#)
create_dataset, [11](#)
create_dataset_group, [11](#)
create_dataset_import_job, [11](#)
create_document_classifier, [3](#)
create_domain, [18](#)
create_endpoint, [3](#), [18](#)
create_endpoint_config, [18](#)
create_entity_recognizer, [3](#)
create_evaluation, [9](#)
create_event_tracker, [11](#)
create_experiment, [18](#)
create_flow_definition, [18](#)
create_human_task_ui, [18](#)
create_hyper_parameter_tuning_job, [18](#)
create_intent_version, [6](#)
create_labeling_job, [18](#)
create_ml_model, [9](#)
create_model, [18](#)
create_model_package, [18](#)
create_monitoring_schedule, [18](#)
create_notebook_instance, [18](#)
create_notebook_instance_lifecycle_config, [18](#)
create_presigned_domain_url, [18](#)
create_presigned_notebook_instance_url, [18](#)
create_processing_job, [18](#)
create_project, [16](#)
create_project_version, [16](#)
create_realtime_endpoint, [9](#)
create_schema, [11](#)
create_slot_type_version, [6](#)
create_solution, [11](#)
create_solution_version, [11](#)
create_stream_processor, [16](#)
create_training_job, [18](#)
create_transform_job, [18](#)
create_trial, [18](#)
create_trial_component, [18](#)
create_user_profile, [18](#)
create_vocabulary, [23](#)
create_vocabulary_filter, [23](#)
create_workteam, [18](#)

delete_algorithm, [18](#)
delete_app, [18](#)
delete_batch_prediction, [9](#)
delete_bot, [6](#)
delete_bot_alias, [6](#)
delete_bot_channel_association, [6](#)
delete_bot_version, [6](#)
delete_campaign, [11](#)
delete_code_repository, [18](#)

- delete_collection, [16](#)
- delete_data_source, [9](#)
- delete_dataset, [11](#)
- delete_dataset_group, [11](#)
- delete_document_classifier, [3](#)
- delete_domain, [18](#)
- delete_endpoint, [3](#), [18](#)
- delete_endpoint_config, [18](#)
- delete_entity_recognizer, [3](#)
- delete_evaluation, [9](#)
- delete_event_tracker, [11](#)
- delete_experiment, [18](#)
- delete_faces, [16](#)
- delete_flow_definition, [19](#)
- delete_intent, [6](#)
- delete_intent_version, [6](#)
- 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_terminology, [25](#)
- delete_transcription_job, [23](#)
- delete_trial, [19](#)
- delete_trial_component, [19](#)
- delete_user_profile, [19](#)
- delete_utterances, [7](#)
- delete_vocabulary, [23](#)
- 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, [11](#)
- describe_code_repository, [19](#)
- describe_collection, [16](#)
- describe_compilation_job, [19](#)
- describe_data_sources, [10](#)
- describe_dataset, [11](#)
- describe_dataset_group, [11](#)
- describe_dataset_import_job, [11](#)
- 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, [3](#)
- describe_entities_detection_v2_job, [5](#)
- describe_entity_recognizer, [3](#)
- describe_evaluations, [10](#)
- describe_event_tracker, [11](#)
- describe_experiment, [19](#)
- describe_feature_transformation, [11](#)
- describe_flow_definition, [19](#)
- describe_human_task_ui, [19](#)
- describe_hyper_parameter_tuning_job, [19](#)
- describe_key_phrases_detection_job, [3](#)
- describe_labeling_job, [19](#)
- describe_ml_models, [10](#)
- describe_model, [19](#)
- describe_model_package, [19](#)
- describe_monitoring_schedule, [19](#)
- describe_notebook_instance, [19](#)
- describe_notebook_instance_lifecycle_config, [19](#)
- describe_phi_detection_job, [5](#)
- describe_processing_job, [19](#)
- describe_project_versions, [16](#)
- describe_projects, [16](#)
- describe_recipe, [11](#)
- describe_schema, [11](#)
- describe_sentiment_detection_job, [3](#)
- describe_solution, [11](#)
- describe_solution_version, [11](#)
- describe_stream_processor, [16](#)
- describe_subscribed_workteam, [19](#)
- describe_tags, [10](#)
- describe_text_translation_job, [25](#)
- describe_topics_detection_job, [3](#)

- describe_training_job, [19](#)
- describe_transform_job, [19](#)
- describe_trial, [19](#)
- describe_trial_component, [19](#)
- describe_user_profile, [19](#)
- describe_voices, [15](#)
- describe_workteam, [19](#)
- detect_custom_labels, [16](#)
- detect_document_text, [22](#)
- detect_dominant_language, [3](#)
- detect_entities, [3, 5](#)
- detect_entities_v2, [5](#)
- detect_faces, [16](#)
- detect_key_phrases, [3](#)
- detect_labels, [16](#)
- detect_moderation_labels, [16](#)
- detect_phi, [5](#)
- detect_sentiment, [3](#)
- detect_syntax, [3](#)
- detect_text, [16](#)
- disassociate_trial_component, [19](#)

- get_batch_prediction, [10](#)
- get_bot, [7](#)
- get_bot_alias, [7](#)
- get_bot_aliases, [7](#)
- get_bot_channel_association, [7](#)
- get_bot_channel_associations, [7](#)
- get_bot_versions, [7](#)
- get_bots, [7](#)
- get_builtin_intent, [7](#)
- get_builtin_intents, [7](#)
- get_builtin_slot_types, [7](#)
- get_celebrity_info, [16](#)
- get_celebrity_recognition, [16](#)
- get_content_moderation, [16](#)
- get_data_source, [10](#)
- get_document_analysis, [22](#)
- get_document_text_detection, [22](#)
- get_evaluation, [10](#)
- get_export, [7](#)
- get_face_detection, [16](#)
- get_face_search, [16](#)
- get_import, [7](#)
- get_intent, [7](#)
- get_intent_versions, [7](#)
- get_intents, [7](#)
- get_label_detection, [16](#)
- get_lexicon, [15](#)

- get_ml_model, [10](#)
- get_person_tracking, [16](#)
- get_personalized_ranking, [14](#)
- get_recommendations, [14](#)
- get_search_suggestions, [19](#)
- get_session, [8](#)
- get_slot_type, [7](#)
- get_slot_type_versions, [7](#)
- get_slot_types, [7](#)
- get_solution_metrics, [11](#)
- get_speech_synthesis_task, [15](#)
- get_terminology, [25](#)
- get_transcription_job, [24](#)
- get_utterances_view, [7](#)
- get_vocabulary, [24](#)
- get_vocabulary_filter, [24](#)

- import_terminology, [25](#)
- index_faces, [16](#)
- infer_icd10cm, [5](#)
- infer_rx_norm, [5](#)
- invoke_endpoint, [21](#)

- lexmodelbuildingservice, [6](#)
- lexruntime, [7](#)
- list_algorithms, [19](#)
- list_apps, [19](#)
- list_auto_ml_jobs, [19](#)
- list_batch_inference_jobs, [11](#)
- list_campaigns, [11](#)
- list_candidates_for_auto_ml_job, [19](#)
- list_code_repositories, [19](#)
- list_collections, [16](#)
- list_compilation_jobs, [19](#)
- list_dataset_groups, [12](#)
- list_dataset_import_jobs, [12](#)
- list_datasets, [12](#)
- list_document_classification_jobs, [4](#)
- list_document_classifiers, [4](#)
- list_domains, [19](#)
- list_dominant_language_detection_jobs, [4](#)
- list_endpoint_configs, [19](#)
- list_endpoints, [4, 19](#)
- list_entities_detection_jobs, [4](#)
- list_entities_detection_v2_jobs, [5](#)
- list_entity_recognizers, [4](#)
- list_event_trackers, [12](#)
- list_experiments, [20](#)

list_faces, [16](#)
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, [16](#)
list_subscribed_workteams, [20](#)
list_tags, [20](#)
list_tags_for_resource, [4](#)
list_terminologies, [25](#)
list_text_translation_jobs, [25](#)
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, [20](#)
list_user_profiles, [20](#)
list_vocabularies, [24](#)
list_vocabulary_filters, [24](#)
list_workteams, [20](#)

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, [16](#)
rekognition, [15](#)
render_ui_template, [20](#)

sagemaker, [17](#)
sagemakerruntime, [21](#)
search, [20](#)
search_faces, [16](#)
search_faces_by_image, [16](#)
start_celebrity_recognition, [16](#)
start_content_moderation, [16](#)
start_document_analysis, [22](#)
start_document_classification_job, [4](#)
start_document_text_detection, [22](#)
start_dominant_language_detection_job, [4](#)
start_entities_detection_job, [4](#)
start_entities_detection_v2_job, [5](#)
start_face_detection, [16](#)
start_face_search, [16](#)
start_import, [7](#)
start_key_phrases_detection_job, [4](#)
start_label_detection, [16](#)
start_monitoring_schedule, [20](#)
start_notebook_instance, [20](#)
start_person_tracking, [16](#)
start_phi_detection_job, [5](#)
start_project_version, [16](#)
start_sentiment_detection_job, [4](#)
start_speech_synthesis_task, [15](#)
start_stream_processor, [16](#)
start_text_translation_job, [25](#)
start_topics_detection_job, [4](#)
start_transcription_job, [24](#)
stop_auto_ml_job, [20](#)
stop_compilation_job, [20](#)
stop_dominant_language_detection_job, [4](#)
stop_entities_detection_job, [4](#)

stop_entities_detection_v2_job, [5](#)
stop_hyper_parameter_tuning_job, [20](#)
stop_key_phrases_detection_job, [4](#)
stop_labeling_job, [20](#)
stop_monitoring_schedule, [20](#)
stop_notebook_instance, [20](#)
stop_phi_detection_job, [5](#)
stop_processing_job, [20](#)
stop_project_version, [17](#)
stop_sentiment_detection_job, [4](#)
stop_stream_processor, [17](#)
stop_text_translation_job, [25](#)
stop_training_document_classifier, [4](#)
stop_training_entity_recognizer, [4](#)
stop_training_job, [20](#)
stop_transform_job, [20](#)
synthesize_speech, [15](#)

tag_resource, [4](#)
textract, [22](#)
transcribeservice, [23](#)
translate, [24](#)
translate_text, [25](#)

untag_resource, [4](#)
update_batch_prediction, [10](#)
update_campaign, [12](#)
update_code_repository, [20](#)
update_data_source, [10](#)
update_domain, [20](#)
update_endpoint, [4](#), [20](#)
update_endpoint_weights_and_capacities,
[20](#)
update_evaluation, [10](#)
update_experiment, [20](#)
update_ml_model, [10](#)
update_monitoring_schedule, [20](#)
update_notebook_instance, [20](#)
update_notebook_instance_lifecycle_config,
[20](#)
update_trial, [20](#)
update_trial_component, [20](#)
update_user_profile, [20](#)
update_vocabulary, [24](#)
update_vocabulary_filter, [24](#)
update_workteam, [20](#)