Title 'Amazon Web Services' Machine Learning Services

Version 0.4.0

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

License Apache License (>= 2.0)

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

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

Imports paws.common (>= 0.6.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.2.3

Collate 'augmentedairuntime_service.R'
 'augmentedairuntime_interfaces.R'
 'augmentedairuntime_operations.R'
 'comprehend_service.R'
 'comprehend_interfaces.R'
 'comprehend_operations.R'
 'comprehendmedical_service.R'
 'comprehendmedical_interfaces.R'
 'comprehendmedical_operations.R'
 'elasticinference_service.R'
 'elasticinference_interfaces.R'
 'elasticinference_operations.R'
 'forecastqueryservice_service.R'
 'forecastqueryservice_interfaces.R'
 'forecastqueryservice_operations.R'
 'forecastservice_service.R'
 'forecastservice_interfaces.R'
 'forecastservice_operations.R'
 'frauddetector_service.R'
 'frauddetector_interfaces.R'
 'frauddetector_operations.R'
 'lexmodelbuildingservice_service.R'
 'lexmodelbuildingservice_interfaces.R'
 'lexmodelbuildingservice_operations.R'
 'lexmodelsv2_service.R'
 'lexmodelsv2_interfaces.R'
 'lexmodelsv2_operations.R'
 'lexruntimeservice_service.R'
 'lexruntimeservice_interfaces.R'
 'lexruntimeservice_operations.R'
 'lexruntimev2_service.R'
R topics documented:

'lexruntimev2_interfaces.R' 'lexruntimev2_operations.R'
'lookoutequipment_service.R' 'lookoutequipment_interfaces.R'
'lookoutequipment_operations.R' 'lookoutmetrics_service.R'
'lookoutmetrics_interfaces.R' 'lookoutmetrics_operations.R'
'machinelearning_service.R' 'machinelearning_interfaces.R'
'machinelearning_operations.R' 'panorama_service.R'
'panorama_interfaces.R' 'panorama_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'
'reexports_paws.common.R' 'rekognition_service.R'
'rekognition_interfaces.R' 'rekognition_operations.R'
'sagemaker_service.R' 'sagemaker_interfaces.R'
'sagemaker_operations.R' 'sagemakeredgemanager_service.R'
'sagemakeredgemanager_interfaces.R' 'sagemakeredgemanager_operations.R'
'sagemakerfeaturestoreruntime_service.R'
'sagemakerfeaturestoreruntime_interfaces.R'
'sagemakerfeaturestoreruntime_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'
'voiceid_service.R' 'voiceid_interfaces.R'
'voiceid_operations.R'

NeedsCompilation no

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

Maintainer Dyfan Jones <dyfan.r.jones@gmail.com>

Repository CRAN

Date/Publication 2023-09-11 18:50:06 UTC

R topics documented:

augmentedairuntime ........................................... 3
comprehend .......................................................... 6
comprehendmedical ............................................... 10
elasticinference .................................................... 13
Description

Amazon Augmented AI (Amazon A2I) adds the benefit of human judgment to any machine learning application. When an AI application can’t evaluate data with a high degree of confidence, human reviewers can take over. This human review is called a human review workflow. To create and start a human review workflow, you need three resources: a worker task template, a flow definition, and a human loop.

For information about these resources and prerequisites for using Amazon A2I, see Get Started with Amazon Augmented AI in the Amazon SageMaker Developer Guide.

This API reference includes information about API actions and data types that you can use to interact with Amazon A2I programmatically. Use this guide to:

- Start a human loop with the `start_human_loop` operation when using Amazon A2I with a custom task type. To learn more about the difference between custom and built-in task types, see Use Task Types. To learn how to start a human loop using this API, see Create and Start a Human Loop for a Custom Task Type in the Amazon SageMaker Developer Guide.
• Manage your human loops. You can list all human loops that you have created, describe individual human loops, and stop and delete human loops. To learn more, see Monitor and Manage Your Human Loop in the Amazon SageMaker Developer Guide.

Amazon A2I integrates APIs from various AWS services to create and start human review workflows for those services. To learn how Amazon A2I uses these APIs, see Use APIs in Amazon A2I in the Amazon SageMaker Developer Guide.

Usage

```r
augmentedairuntime(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

- **config** Optional configuration of credentials, endpoint, and/or region.
  - **credentials:**
    - **creds:**
      - `access_key_id`: AWS access key ID
      - `secret_access_key`: AWS secret access key
      - `session_token`: AWS temporary session token
    - **profile**: The name of a profile to use. If not given, then the default profile is used.
    - **anonymous**: Set anonymous credentials.
    - **endpoint**: The complete URL to use for the constructed client.
    - **region**: The AWS Region used in instantiating the client.
  - **close_connection**: Immediately close all HTTP connections.
  - **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  - **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
  - **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy [https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html](https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html)

- **credentials** Optional credentials shorthand for the config parameter
  - **creds:**
    - `access_key_id`: AWS access key ID
    - `secret_access_key`: AWS secret access key
    - `session_token`: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
augmentedairuntime

endpoint  Optional shorthand for complete URL to use for the constructed client.
region    Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

```r
svc <- augmentedairuntime(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

- **delete_human_loop** Deletes the specified human loop for a flow definition
- **describe_human_loop** Returns information about the specified human loop
- **list_human_loops** Returns information about human loops, given the specified parameters
- **start_human_loop** Starts a human loop, provided that at least one activation condition is met
- **stop_human_loop** Stops the specified human loop
comprehend

Examples

```r
## Not run:
svc <- augmentedairuntime()
svc$delete_human_loop(
  Foo = 123
)

## End(Not run)
```

comprehend  
Amazon Comprehend

Description

Amazon Comprehend is an Amazon Web Services 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

```r
comprehend(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

- **config**: Optional configuration of credentials, endpoint, and/or region.
  - **credentials**: 
    - **creds**:
      - **access_key_id**: AWS access key ID
      - **secret_access_key**: AWS secret access key
      - **session_token**: AWS temporary session token
    - **profile**: The name of a profile to use. If not given, then the default profile is used.
    - **anonymous**: Set anonymous credentials.
    - **endpoint**: The complete URL to use for the constructed client.
    - **region**: The AWS Region used in instantiating the client.
  - **close_connection**: Immediately close all HTTP connections.
• **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.

• **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.

• **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy [https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html](https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html)

credentails

Optional credentials shorthand for the config parameter

• **creds**:
  – **access_key_id**: AWS access key ID
  – **secret_access_key**: AWS secret access key
  – **session_token**: AWS temporary session token

• **profile**: The name of a profile to use. If not given, then the default profile is used.

• **anonymous**: Set anonymous credentials.

documentation

Optional shorthand for complete URL to use for the constructed client.

region

Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

```r
svc <- comprehend(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  )
)
```
secret_access_key = "string",
session_token = "string"
),
profile = "string",
anonymous = "logical"
),
endpoint = "string",
region = "string"
)

Operations

batch_detect_dominant_language
batch_detect_entities
batch_detect_key_phrases
batch_detect_sentiment
batch_detect_syntax
batch_detect_targeted_sentiment
classify_document
contains_pii_entities
create_dataset
create_document_classifier
create_endpoint
create_entity_recognizer
create_flywheel
delete_document_classifier
delete_endpoint
delete_entity_recognizer
delete_flywheel
delete_resource_policy
describe_dataset
describe_document_classification_job
describe_document_classifier
describe_dominant_language_detection_job
describe_endpoint
describe_entities_detection_job
describe_entity_recognizer
describe_events_detection_job
describe_flywheel
describe_flywheel_iteration
describe_key_phrases_detection_job
describe_pii_entities_detection_job
describe_resource_policy
describe_sentiment_detection_job
describe_targeted_sentiment_detection_job
describe_topics_detection_job
detect_dominant_language
detect_entities

Determines the dominant language of the input text for a batch of documents
Inspects the text of a batch of documents for named entities and returns information
Dets the key noun phrases found in a batch of documents
Inspects a batch of documents and returns an inference of the prevailing sentiment
Inspects a batch of documents for the syntax and part of speech of the text
Creates a new document classification request to analyze a single document in real-time
Analyzes input text for the presence of personally identifiable information (PII)
Creates a model-specific endpoint for synchronous inference for a previously-trained custom model
Creates an entity recognizer using submitted files
A flywheel is an Amazon Web Services resource that orchestrates the ongoing training of a model
Deletes a previously created document classifier
Deletes a model-specific endpoint for a previously-trained custom model
Deletes an entity recognizer
Deletes a flywheel
Deletes a resource-based policy that is attached to a custom model
Retrieves information about the dataset that you specify
Gets the properties associated with a document classification job
Gets the properties associated with a document classifier
Gets the properties associated with a dominant language detection job
Gets the properties associated with a specific endpoint
Gets the properties associated with an entities detection job
Provides details about an entity recognizer including status, S3 buckets containing training data
Gets the status and details of an events detection job
Provides configuration information about the flywheel
Retrieves the configuration properties of a flywheel iteration
Gets the properties associated with a key phrases detection job
Gets the properties associated with a PII entities detection job
Gets the details of a resource-based policy that is attached to a custom model, including
Gets the properties associated with a sentiment detection job
Gets the properties associated with a targeted sentiment detection job
Gets the properties associated with a topic detection job
Determines the dominant language of the input text
Detects named entities in input text when you use the pre-trained model
detect_key_phrases
detect_pii_entities
detect_sentiment
detect_syntax
detect_targeted_sentiment
import_model
list_datasets
list_document_classification_jobs
list_document_classifiers
list_dominant_language_detection_jobs
list_endpoints
list_entities_detection_jobs
list_entity_recognizers
list_entity_recognizer_summaries
list_events_detection_jobs
list_flywheel_iteration_history
list_flywheels
list_key_phrases_detection_jobs
list_pii_entities_detection_jobs
list_sentiment_detection_jobs
list_tags_for_resource
list_targeted_sentiment_detection_jobs
list_topics_detection_jobs
put_resource_policy
start_document_classification_job
start_dominant_language_detection_job
start_entities_detection_job
start_events_detection_job
start_flywheel_iteration
start_key_phrases_detection_job
start_pii_entities_detection_job
start_sentiment_detection_job
start_targeted_sentiment_detection_job
start_topics_detection_job
stop_dominant_language_detection_job
stop_entities_detection_job
stop_events_detection_job
stop_key_phrases_detection_job
stop_pii_entities_detection_job
stop_sentiment_detection_job
stop_targeted_sentiment_detection_job
stop_training_document_classifier
stop_training_entity_recognizer
tag_resource
untag_resource
update_endpoint
update_flywheel

Detects the key noun phrases found in the text
Detects the PII entities found in the text
Detects the sentiment of the text
Inspects the syntax of the text
Detects the targeted sentiment of the text
Imports a custom model
Lists the datasets that have been configured
Lists the document classification jobs
Lists the document classifiers
Lists the dominant language detection jobs
Lists the endpoints
Lists the entities detection jobs
Lists the entity recognizers
Lists the entity recognizer summaries
Lists the events detection jobs
Lists the flywheel iteration history
Lists the flywheels
Lists the key phrases detection jobs
Lists the PII entities detection jobs
Lists the sentiment detection jobs
Lists the tags for a resource
Lists the targeted sentiment detection jobs
Lists the topics detection jobs
Attaches a resource policy to a custom model
Starts a document classification job
Starts a dominant language detection job
Starts an entities detection job
Starts an events detection job
Starts a flywheel iteration
Starts a key phrases detection job
Starts a PII entities detection job
Starts a sentiment detection job
Starts a targeted sentiment detection job
Starts a training document classifier
Starts a training entity recognizer
Attaches a specific tag to a resource
Removes a specific tag from a resource
Updates information about an endpoint
Updates the configuration information for an existing flywheel
comprehendmedical

AWS Comprehend Medical

Description

Amazon Comprehend Medical extracts structured information from unstructured clinical text. Use these actions to gain insight in your documents. Amazon Comprehend Medical only detects entities in English language texts. Amazon Comprehend Medical places limits on the sizes of files allowed for different API operations. To learn more, see Guidelines and quotas in the Amazon Comprehend Medical Developer Guide.

Usage

comprehendmedical(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)

Arguments

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

  * credentials:
    - creds:
      - access_key_id: AWS access key ID
      - secret_access_key: AWS secret access key
      - session_token: AWS temporary session token
    - profile: The name of a profile to use. If not given, then the default profile is used.
    - anonymous: Set anonymous credentials.
    - endpoint: The complete URL to use for the constructed client.
    - region: The AWS Region used in instantiating the client.
• **close_connection**: Immediately close all HTTP connections.

• **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.

• **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.

• **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy [https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html](https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html)

**credentials**

Optional credentials shorthand for the config parameter

• **creds**:
  – **access_key_id**: AWS access key ID
  – **secret_access_key**: AWS secret access key
  – **session_token**: AWS temporary session token

• **profile**: The name of a profile to use. If not given, then the default profile is used.

• **anonymous**: Set anonymous credentials.

**endpoint**

Optional shorthand for complete URL to use for the constructed client.

**region**

Optional shorthand for AWS Region used in instantiating the client.

**Value**

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

**Service syntax**

```r
svc <- comprehendmedical(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
```
creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
),
    profile = "string",
    anonymous = "logical"
),
    endpoint = "string",
    region = "string"
)

Operations

describe_entities_detection_v2_job
describe_icd10cm_inference_job
describe_phi_detection_job
describe_rx_norm_inference_job
describe_snomedct_inference_job
detect_entities
detect_entities_v2
detect_phi
infer_icd10cm
infer_rx_norm
infer_snomedct
list_entities_detection_v2_jobs
list_icd10cm_inference_jobs
list_phi_detection_jobs
list_rx_norm_inference_jobs
list_snomedct_inference_jobs
start_entities_detection_v2_job
start_icd10cm_inference_job
start_phi_detection_job
start_rx_norm_inference_job
start_snomedct_inference_job
stop_entities_detection_v2_job
stop_icd10cm_inference_job
stop_phi_detection_job
stop_rx_norm_inference_job
stop_snomedct_inference_job

get the properties associated with a medical entities detection job
get the properties associated with an InferICD10CM job
get the properties associated with a protected health information (PHI) detection job
get the properties associated with an InferRxNorm job
get the properties associated with an InferSNOMEDCT job
the DetectEntities operation is deprecated
inspects the clinical text for a variety of medical entities and returns specific information about them
inspects the clinical text for protected health information (PHI) entities and returns the entity category, location, and confidence score for each entity
InferICD10CM detects medical conditions as entities listed in a patient record and links them to the ICD-10-CM ontology
InferRxNorm detects medications as entities listed in a patient record and links them to the RxNorm database
InferSNOMEDCT detects possible medical concepts as entities and links them to codes in the SNOMED-CT ontology
gets a list of medical entity detection jobs that you have submitted
gets a list of InferICD10CM jobs that you have submitted
gets a list of protected health information (PHI) detection jobs you have submitted
gets a list of InferRxNorm jobs that you have submitted
gets a list of InferSNOMEDCT jobs a user has submitted
starts an asynchronous medical entity detection job for a collection of documents
starts an asynchronous job to detect medical conditions and link them to the ICD-10-CM ontology
starts an asynchronous job to detect protected health information (PHI)
starts an asynchronous job to detect medication entities and link them to the RxNorm ontology
starts an asynchronous job to detect medical concepts and link them to the SNOMED-CT ontology
stops a medical entities detection job in progress
stops an InferICD10CM inference job in progress
stops a protected health information (PHI) detection job in progress
stops an InferRxNorm inference job in progress
stops an InferSNOMEDCT inference job in progress

Examples

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

## End(Not run)

---

**elasticinference**  
*Amazon Elastic Inference*

**Description**

Elastic Inference public APIs.

February 15, 2023: Starting April 15, 2023, AWS will not onboard new customers to Amazon Elastic Inference (EI), and will help current customers migrate their workloads to options that offer better price and performance. After April 15, 2023, new customers will not be able to launch instances with Amazon EI accelerators in Amazon SageMaker, Amazon ECS, or Amazon EC2. However, customers who have used Amazon EI at least once during the past 30-day period are considered current customers and will be able to continue using the service.

**Usage**

```r
elasticinference(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

**Arguments**

- **config**: Optional configuration of credentials, endpoint, and/or region.
  - **credentials**:
    - **creds**:
      - `access_key_id`: AWS access key ID
      - `secret_access_key`: AWS secret access key
      - `session_token`: AWS temporary session token
    - **profile**: The name of a profile to use. If not given, then the default profile is used.
    - **anonymous**: Set anonymous credentials.
    - **endpoint**: The complete URL to use for the constructed client.
    - **region**: The AWS Region used in instantiating the client.
  - **close_connection**: Immediately close all HTTP connections.
  - **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
elasticinference

- **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy `https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html`

**credentials**
Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

**endpoint**
Optional shorthand for complete URL to use for the constructed client.

**region**
Optional shorthand for AWS Region used in instantiating the client.

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

**Service syntax**

```r
svc <- elasticinference(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    )
  )
)
```
forecastqueryservice

),
profile = "string",
anonymous = "logical"
),
endpoint = "string",
region = "string"
)

Operations

describe_accelerator_offerings Describes the locations in which a given accelerator type or set of types is present in a given region

describe_accelerators Describes information over a provided set of accelerators belonging to an account

describe_accelerator_types Describes the accelerator types available in a given region, as well as their characteristics, such as memory and throughput

list_tags_for_resource Returns all tags of an Elastic Inference Accelerator

tag_resource Adds the specified tags to an Elastic Inference Accelerator

untag_resource Removes the specified tags from an Elastic Inference Accelerator

Examples

## Not run:
svc <- elasticinference()
svc$describe_accelerator_offerings( Foo = 123 )

## End(Not run)

forecastqueryservice  Amazon Forecast Query Service

Description

Provides APIs for creating and managing Amazon Forecast resources.

Usage

forecastqueryservice(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
Arguments

config  Optional configuration of credentials, endpoint, and/or region.
  • credentials:
    – creds:
      * access_key_id: AWS access key ID
      * secret_access_key: AWS secret access key
      * session_token: AWS temporary session token
    – profile: The name of a profile to use. If not given, then the default profile is used.
    – anonymous: Set anonymous credentials.
    – endpoint: The complete URL to use for the constructed client.
    – region: The AWS Region used in instantiating the client.
  • close_connection: Immediately close all HTTP connections.
  • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
  • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-ends.html

credentials  Optional credentials shorthand for the config parameter
  • creds:
    – access_key_id: AWS access key ID
    – secret_access_key: AWS secret access key
    – session_token: AWS temporary session token
  • profile: The name of a profile to use. If not given, then the default profile is used.
  • anonymous: Set anonymous credentials.

endpoint  Optional shorthand for complete URL to use for the constructed client.

region  Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

svc <- forecastqueryservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string",
      ),
      profile = "string",  # Optional
      anonymous = TRUE,  # Optional
    ),
    endpoint = "string",  # Optional
    region = "string",  # Optional
  ),
  close_connection = TRUE,  # Optional
  timeout = 60,  # Optional
  s3_force_path_style = TRUE,  # Optional
  sts_regional_endpoint = "region"  # Optional
)
secret_access_key = "string",
    session_token = "string"
),
    profile = "string",
    anonymous = "logical"
),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
),
    credentials = list(
        creds = list(
            access_key_id = "string",
            secret_access_key = "string",
            session_token = "string"
        ),
        profile = "string",
        anonymous = "logical"
    ),
    endpoint = "string",
    region = "string"
)

Operations

query_forecast  Retrieves a forecast for a single item, filtered by the supplied criteria
query_what_if_forecast  Retrieves a what-if forecast

Examples

## Not run:
svc <- forecastqueryservice()
svc$query_forecast(
    Foo = 123
)

## End(Not run)
Description

Provides APIs for creating and managing Amazon Forecast resources.

Usage

```r
forecastservice(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

- **config**: Optional configuration of credentials, endpoint, and/or region.
  - **credentials**:
    - **creds**:
      - **access_key_id**: AWS access key ID
      - **secret_access_key**: AWS secret access key
      - **session_token**: AWS temporary session token
    - **profile**: The name of a profile to use. If not given, then the default profile is used.
    - **anonymous**: Set anonymous credentials.
    - **endpoint**: The complete URL to use for the constructed client.
    - **region**: The AWS Region used in instantiating the client.
  - **close_connection**: Immediately close all HTTP connections.
  - **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  - **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
  - **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy [https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html](https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html)

- **credentials**: Optional credentials shorthand for the config parameter
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.

- **endpoint**: Optional shorthand for complete URL to use for the constructed client.

- **region**: Optional shorthand for AWS Region used in instantiating the client.
Value

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

Service syntax

```r
svc <- forecastservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

- `create_auto_predictor`: Creates an Amazon Forecast predictor
- `create_dataset`: Creates an Amazon Forecast dataset
- `create_dataset_group`: Creates a dataset group, which holds a collection of related datasets
- `create_dataset_import_job`: Imports your training data to an Amazon Forecast dataset
- `create_explainability`: Explainability is only available for Forecasts and Predictors generated from an AutoPredictor
- `create_explainability_export`: Exports an Explainability resource created by the CreateExplainability operation
- `create_forecast`: Creates a forecast for each item in the TARGET_TIME_SERIES dataset that was used to train the predictor
- `create_forecast_export_job`: Exports a forecast created by the CreateForecast operation to your Amazon Simple
create_monitor
create_predictor
create_predictor_backtest_export_job
create_what_if_analysis
create_what_if_forecast
delete_dataset
delete_dataset_group
delete_dataset_import_job
delete_explainability
delete_explainability_export
delete_forecast
delete_forecast_export_job
delete_monitor
delete_predictor
delete_predictor_backtest_export_job
delete_resource_tree
delete_what_if_analysis
delete_what_if_forecast
describe_auto_predictor
describe_dataset
describe_dataset_group
describe_dataset_import_job
describe_explainability
describe_explainability_export
describe_forecast
describe_forecast_export_job
describe_monitor
describe_predictor
describe_predictor_backtest_export_job
describe_what_if_analysis
describe_what_if_forecast
describe_what_if_forecast_export
get_accuracy_metrics
list_dataset_groups
list_dataset_import_jobs
list_datasets
list_explainabilities
list_explainability_exports
list_forecast_export_jobs
list_forecasts
list_monitor_evaluations
list_monitors
list_predictor_backtest_export_jobs
list_predictors
list_tags_for_resource
list_what_if_analyses

Creates a predictor monitor resource for an existing auto predictor
This operation creates a legacy predictor that does not include all the predictor functions
Exports backtest forecasts and accuracy metrics generated by the CreateAutoPredictor operation
What-if analysis is a scenario modeling technique where you make a hypothetical change to a time series
A what-if forecast is a forecast that is created from a modified version of the baseline time series
Exports a forecast created by the CreateWhatIfForecast operation to your Amazon S3 bucket
Deletes an Amazon Forecast dataset that was created using the CreateDataset operation
Deletes a dataset group created using the CreateDatasetGroup operation
Deletes a dataset import job created using the CreateDatasetImportJob operation
Deletes an Explainability resource
Deletes an Explainability export
Deletes a forecast created using the CreateForecast operation
Deletes a forecast export job created using the CreateForecastExportJob operation
Deletes a monitor resource
Deletes a predictor created using the DescribePredictor or CreatePredictor operations
Deletes a predictor backtest export job
Deletes an entire resource tree
Deletes a what-if analysis created using the CreateWhatIfAnalysis operation
Deletes a what-if forecast created using the CreateWhatIfForecast operation
Deletes a what-if forecast export created using the CreateWhatIfForecastExport operation
Describes a predictor created using the CreateAutoPredictor operation
Describes an Amazon Forecast dataset created using the CreateDataset operation
Describes a dataset group created using the CreateDatasetGroup operation
Describes a dataset import job created using the CreateDatasetImportJob operation
Describes an Explainability resource created using the CreateExplainability operation
Describes an Explainability export created using the CreateExplainabilityExport operation
Describes a forecast created using the CreateForecast operation
Describes a forecast export job created using the CreateForecastExportJob operation
Describes a monitor resource
This operation is only valid for legacy predictors created with CreatePredictor
Describes a predictor backtest export job created using the CreatePredictorBacktestExportJob operation
Describes the what-if analysis created using the CreateWhatIfAnalysis operation
Describes the what-if forecast created using the CreateWhatIfForecast operation
Describes the what-if forecast export created using the CreateWhatIfForecastExport operation
Provides metrics on the accuracy of the models that were trained by the CreatePredictorBacktestExportJob operation
Returns a list of dataset groups created using the CreateDatasetGroup operation
Returns a list of dataset import jobs created using the CreateDatasetImportJob operation
Returns a list of datasets created using the CreateDataset operation
Returns a list of Explainability resources created using the CreateExplainability operation
Returns a list of Explainability exports created using the CreateExplainabilityExport operation
Returns a list of forecast export jobs created using the CreateForecastExportJob operation
Returns a list of forecasts created using the CreateForecast operation
Returns a list of the monitoring evaluation results and predictor events collected by a monitor
Returns a list of monitors created with the CreateMonitor operation and CreateAutoPredictor
Returns a list of predictor backtest export jobs created using the CreatePredictorBacktestExportJob operation
Returns a list of predictors created using the CreateAutoPredictor or CreatePredictor operation
Lists the tags for an Amazon Forecast resource
Returns a list of what-if analyses created using the CreateWhatIfAnalysis operation
## Not run:
svc <- forecastservice()
svc$create_auto_predictor(
  Foo = 123
)
## End(Not run)

This is the Amazon Fraud Detector API Reference. This guide is for developers who need detailed information about Amazon Fraud Detector API actions, data types, and errors. For more information about Amazon Fraud Detector features, see the Amazon Fraud Detector User Guide.

We provide the Query API as well as AWS software development kits (SDK) for Amazon Fraud Detector in Java and Python programming languages.

The Amazon Fraud Detector Query API provides HTTPS requests that use the HTTP verb GET or POST and a Query parameter Action. AWS SDK provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS. These libraries provide basic functions that automatically take care of tasks such as cryptographically signing your requests, retrying requests, and handling error responses, so that it is easier for you to get started. For more information about the AWS SDKs, go to Tools to build on AWS page, scroll down to the SDK section, and choose plus (+) sign to expand the section.

```
frauddetector(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```
Arguments

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

  • credentials:
    – creds:
      * access_key_id: AWS access key ID
      * secret_access_key: AWS secret access key
      * session_token: AWS temporary session token
    – profile: The name of a profile to use. If not given, then the default profile is used.
    – anonymous: Set anonymous credentials.
    – endpoint: The complete URL to use for the constructed client.
    – region: The AWS Region used in instantiating the client.
  • close_connection: Immediately close all HTTP connections.
  • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
  • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-er.html

credentials Optional credentials shorthand for the config parameter

  • creds:
    – access_key_id: AWS access key ID
    – secret_access_key: AWS secret access key
    – session_token: AWS temporary session token
  • profile: The name of a profile to use. If not given, then the default profile is used.
  • anonymous: Set anonymous credentials.

endpoint Optional shorthand for complete URL to use for the constructed client.

region Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

```
svc <- frauddetector(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = FALSE,
      endpoint = "string",
      region = "string"
    )
  ),
  close_connection = FALSE,
  timeout = 60,
  s3_force_path_style = FALSE,
  sts_regional_endpoint = "string"
)
```

frauddetector
frauddetector

secret_access_key = "string",
session_token = "string"
),
profile = "string",
anonymous = "logical"
),
endpoint = "string",
region = "string",
close_connection = "logical",
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string"
),
credentials = list(
creds = list(
access_key_id = "string",
secret_access_key = "string",
session_token = "string"
),
profile = "string",
anonymous = "logical"
),
endpoint = "string",
region = "string"
)

Operations

batch_create_variable Creates a batch of variables
batch_get_variable Gets a batch of variables
cancel_batch_import_job Cancels an in-progress batch import job
cancel_batch_prediction_job Cancels the specified batch prediction job
create_batch_import_job Creates a batch import job
create_batch_prediction_job Creates a batch prediction job
create_detector_version Creates a detector version
create_list Creates a list
create_model Creates a model using the specified model type
create_model_version Creates a version of the model using the specified model type and model id
create_rule Creates a rule for use with the specified detector
create_variable Creates a variable
delete_batch_import_job Deletes the specified batch import job ID record
delete_batch_prediction_job Deletes a batch prediction job
delete_detector Deletes the detector
delete_detector_version Deletes the detector version
delete_entity_type Deletes an entity type
delete_event Deletes the specified event
delete_events_by_event_type Deletes all events of a particular event type
delete_event_type Deletes an event type
delete_external_model
delete_label
delete_list
delete_model
delete_model_version
delete_outcome
delete_rule
delete_variable
describe_detector
describe_model_versions
get_batch_import_jobs
get_batch_prediction_jobs
get_delete_events_by_event_type_status
get_detectors
get_detector_version
get_entity_types
get_event
get_event_prediction
get_event_prediction_metadata
get_event_types
get_external_models
get_kms_encryption_key
get_labels
get_list_elements
get_lists_metadata
get_models
get_model_version
get_outcomes
get_rules
get_variables
list_event_predictions
list_tags_for_resource
put_detector
put_entity_type
put_event_type
put_external_model
put_kms_encryption_key
put_label
put_outcome
send_event
tag_resource
untag_resource
update_detector_version
update_detector_version_metadata
update_detector_version_status
update_event_label
update_list
update_model

Remove a SageMaker model from Amazon Fraud Detector
Deletes a label
Deletes the list, provided it is not used in a rule
Deletes a model
Deletes a model version
Deletes an outcome
Deletes the rule
Deletes a variable
Gets all versions for a specified detector
Gets all of the model versions for the specified model type or for the specified model ID
Gets all batch import jobs or a specific job of the specified ID
Gets all batch prediction jobs or a specific job if you specify a job ID
Retrieves the status of a DeleteEventsByEventType action
Gets all detectors or a single detector if a detectorID is specified
Gets a particular detector version
Gets all entity types or a specific entity type if a name is specified
Retrieves details of events stored with Amazon Fraud Detector
Evaluates an event against a detector version
Gets details of the past fraud predictions for the specified event ID, event type, detector ID, and detector version ID
Gets all event types or a specific event type if name is provided
Gets the details for one or more Amazon SageMaker models that have been imported
Gets the encryption key if a KMS key has been specified to be used to encrypt content
Gets all labels or a specific label if name is provided
Gets all the elements in the specified list
Gets the metadata of either all the lists under the account or the specified list
Gets one or more models
Gets the details of the specified model version
Gets one or more outcomes
Get all rules for a detector (paginated) if ruleId and ruleVersion are not specified
Gets all of the variables or the specific variable
Gets a list of past predictions
Lists all tags associated with the resource
Creates or updates a detector
Creates or updates an entity type
Creates or updates an event type
Creates or updates an Amazon SageMaker model endpoint
Specifies the KMS key to be used to encrypt content in Amazon Fraud Detector
Creates or updates label
Creates or updates an outcome
Stores events in Amazon Fraud Detector without generating fraud predictions for the event
Assigns tags to a resource
Removes tags from a resource
Updates a detector version
Updates the detector version’s description
Updates the detector version’s status
Updates the specified event with a new label
Updates a list
Updates model 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(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

- **config**: Optional configuration of credentials, endpoint, and/or region.
  - **credentials**:
    - **creds**:
      - **access_key_id**: AWS access key ID
      - **secret_access_key**: AWS secret access key
      - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.

  * **close_connection**: Immediately close all HTTP connections.
  * **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  * **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.

**credentials**

Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token

- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

**endpoint**

Optional shorthand for complete URL to use for the constructed client.

**region**

Optional shorthand for AWS Region used in instantiating the client.

**Value**

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

**Service syntax**

```r
csv <- lexmodelbuildingservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
```
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string"
),
credentials = list(
creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
),
    profile = "string",
anonymous = "logical"
),
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:
get_bot_versions Gets information about all of the versions of a bot
get_builtin_intent Returns information about a built-in intent
get_builtin_intents Gets a list of built-in intents that meet the specified criteria
get_builtin_slot_types Gets a list of built-in slot types that meet the specified criteria
get_export Exports the contents of a Amazon Lex resource in a specified format
get_import Gets information about an import job started with the StartImport operation
get_intent Returns information about an intent
get_intents Returns intent information as follows:
get_intent_versions Gets information about all of the versions of an intent
get_migration Provides details about an ongoing or complete migration from an Amazon Lex V1 bot to an Amazon Lex V2 bot
get_migrations Gets a list of migrations between Amazon Lex V1 and Amazon Lex V2
get_slot_type | Returns information about a specific version of a slot type
get_slot_types | Returns slot type information as follows:
get_slot_type_versions | Gets information about all versions of a slot type
get_utterances_view | Use the GetUtterancesView operation to get information about the utterances that your user
list_tags_for_resource | Gets a list of tags associated with the specified resource
put_bot | Creates an Amazon Lex conversational bot or replaces an existing bot
put_bot_alias | Creates an alias for the specified version of the bot or replaces an alias for the specified bot
put_intent | Creates an intent or replaces an existing intent
put_slot_type | Creates a custom slot type or replaces an existing custom slot type
start_import | Starts a job to import a resource to Amazon Lex
start_migration | Starts migrating a bot from Amazon Lex V1 to Amazon Lex V2
tag_resource | Adds the specified tags to the specified resource
untag_resource | Removes tags from a bot, bot alias or bot channel

Examples

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

---

**Description**

Amazon Lex Model Building V2

**Usage**

```r
lexmodelsv2{
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
}
```
Arguments

config Optional configuration of credentials, endpoint, and/or region.
  • credentials:
    – creds:
      • access_key_id: AWS access key ID
      • secret_access_key: AWS secret access key
      • session_token: AWS temporary session token
    – profile: The name of a profile to use. If not given, then the default profile is used.
    – anonymous: Set anonymous credentials.
    – endpoint: The complete URL to use for the constructed client.
    – region: The AWS Region used in instantiating the client.
  • close_connection: Immediately close all HTTP connections.
  • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
  • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html

credentials Optional credentials shorthand for the config parameter
  • creds:
    – access_key_id: AWS access key ID
    – secret_access_key: AWS secret access key
    – session_token: AWS temporary session token
  • profile: The name of a profile to use. If not given, then the default profile is used.
  • anonymous: Set anonymous credentials.

endpoint Optional shorthand for complete URL to use for the constructed client.

region Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

svc <- lexmodelsv2(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  ),
  timeout = 60,
  close_connection = TRUE
)
secret_access_key = "string",
    session_token = "string"
),
    profile = "string",
    anonymous = "logical"
),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
),
credentials = list(
    creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
),
    endpoint = "string",
    region = "string"
)

Operations

batch_create_custom_vocabulary_item  Create a batch of custom vocabulary items for a given bot locale’s custom vocabulary
batch_delete_custom_vocabulary_item  Delete a batch of custom vocabulary items for a given bot locale’s custom vocabulary
batch_update_custom_vocabulary_item  Update a batch of custom vocabulary items for a given bot locale’s custom vocabulary
build_bot_locale                      Builds a bot, its intents, and its slot types into a specific locale
create_bot                           Creates an Amazon Lex conversational bot
create_bot_alias                     Creates an alias for the specified version of a bot
create_bot_locale                    Creates a locale in the bot
create_bot_version                   Creates a new version of the bot based on the DRAFT version
create_export                        Creates a zip archive containing the contents of a bot or a bot locale
create_intent                        Creates an intent
create_resource_policy               Creates a new resource policy with the specified policy statements
create_resource_policy_statement     Adds a new resource policy statement to a bot or bot alias
create_slot                          Creates a slot in an intent
create_slot_type                     Creates a custom slot type
create_test_set_discrepancy_report   Create a report that describes the differences between the bot and the test set
create_upload_url                    Gets a pre-signed S3 write URL that you use to upload the zip archive when importing a bot
delete_bot                           Deletes all versions of a bot, including the Draft version
delete_bot_alias                     Deletes the specified bot alias
delete_bot_locale                    Removes a locale from a bot
delete_bot_version                   Deletes a specific version of a bot
delete_custom_vocabulary
delete_export
delete_import
delete_intent
delete_resource_policy
delete_resource_policy_statement
delete_slot
delete_slot_type
delete_test_set
delete utterances
describe_bot
describe_bot_alias
describe_bot_locale
describe_bot_recommendation
describe_bot_version
describe_custom_vocabulary_metadata
describe_export
describe_import
describe_intent
describe_resource_policy
describe_slot
describe_slot_type
describe_test_execution
describe_test_set
describe_test_set_discrepancy_report
describe_test_set_generation
get_test_execution_artifacts_url
list_aggregated_utterances
list_bot_aliases
list_bot_locales
list_bot_recommendations
list_bots
list_bot_versions
list_built_in_intents
list_built_in_slot_types
list_custom_vocabulary_items
list_exports
list_imports
list_intent_metrics
list_intent_paths
list_intents
list_intent_stage_metrics
list_recommended_intents
list_session_analytics_data
list_session_metrics
list_slots
list_slot_types
list_tags_for_resource

Removes a custom vocabulary from the specified locale in the specified bot
Removes a previous export and the associated files stored in an S3 bucket
Removes a previous import and the associated file stored in an S3 bucket
Removes the specified intent
Removes an existing policy from a bot or bot alias
Deletes a policy statement from a resource policy
Deletes the specified slot from an intent
Deletes a slot type from a bot locale
The action to delete the selected test set
Deletes stored utterances
Provides metadata information about a bot
Get information about a specific bot alias
Describes the settings that a bot has for a specific locale
Provides metadata information about a bot recommendation
Provides metadata about a version of a bot
Provides metadata information about a custom vocabulary
Gets information about a specific export
Gets information about a specific import
Returns metadata about an intent
Gets the resource policy and policy revision for a bot or bot alias
Gets metadata information about a slot
Gets metadata information about a slot type
Gets metadata information about the test execution
Gets metadata information about the test set
Gets metadata information about the test set discrepancy report
Gets metadata information about the test set generation
The pre-signed Amazon S3 URL to download the test execution result artifacts
Provides a list of utterances that users have sent to the bot
Gets a list of aliases for the specified bot
Gets a list of locales for the specified bot
Get a list of bot recommendations that meet the specified criteria
Gets a list of available bots
Gets a list of built-in intents provided by Amazon Lex that you can use in your bot
Gets a list of built-in slot types that meet the specified criteria
Paginated list of custom vocabulary items for a given bot locale’s custom vocabulary
Lists the exports for a bot, bot locale, or custom vocabulary
Lists the imports for a bot, bot locale, or custom vocabulary
Retrieves summary metrics for the intents in your bot
Retrieves summary statistics for a path of intents that users take over sessions with your bot
Get a list of intents that meet the specified criteria
Retrieves summary metrics for the stages within intents in your bot
Gets a list of recommended intents provided by the bot recommendation that you can use in your bot
Retrieves a list of metadata for individual user sessions with your bot
Retrieves summary metrics for the user sessions with your bot
Gets a list of slots that match the specified criteria
Gets a list of slot types that match the specified criteria
Gets a list of tags associated with a resource
list_test_execution_result_items  Gets a list of test execution result items
list_test_executions     The list of test set executions
list_test_set_records  The list of test set records
list_test_sets           The list of the test sets
list_utterance_analytics_data
                         To use this API operation, your IAM role must have permissions to perform the ListAggregatedUtterances operation, which provides access to utterance-related analytics
list_utterance_metrics  To use this API operation, your IAM role must have permissions to perform the ListAggregatedUtterances operation, which provides access to utterance-related analytics
search_associated_transcripts
start_bot_recommendation  Use this to provide your transcript data, and to start the bot recommendation process
start_import
start_test_execution
start_test_set_generation
stop_bot_recommendation
Tag_resource
untag_resource
update_bot
update_bot_alias
update_bot_locale
update_bot_recommendation
update_export
update_intent
update_resource_policy
update_slot
update_slot_type
update_test_set

Examples

## Not run:
svc <- lexmodelsv2()
svc$batch_create_custom_vocabulary_item(
   Foo = 123
)
## End(Not run)

lexruntimeservice  Amazon Lex Runtime Service

Description

Amazon Lex provides both build and runtime endpoints. Each endpoint provides a set of operations (API). Your conversational bot uses the runtime API to understand user utterances (user input text or voice). For example, suppose a user says “I want pizza”, your bot sends this input to Amazon Lex using the runtime API. Amazon Lex recognizes that the user request is for the OrderPizza intent
lexruntimeservice (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

```r
lexruntimeservice(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

- **config**: Optional configuration of credentials, endpoint, and/or region.
  - **credentials**:
    - `creds`:
      - `access_key_id`: AWS access key ID
      - `secret_access_key`: AWS secret access key
      - `session_token`: AWS temporary session token
    - `profile`: The name of a profile to use. If not given, then the default profile is used.
    - `anonymous`: Set anonymous credentials.
    - `endpoint`: The complete URL to use for the constructed client.
    - `region`: The AWS Region used in instantiating the client.
  - `close_connection`: Immediately close all HTTP connections.
  - `timeout`: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  - `s3_force_path_style`: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.

- **credentials**: Optional credentials shorthand for the config parameter
  - `creds`:
    - `access_key_id`: AWS access key ID
    - `secret_access_key`: AWS secret access key
    - `session_token`: AWS temporary session token
  - `profile`: The name of a profile to use. If not given, then the default profile is used.
  - `anonymous`: Set anonymous credentials.

- **endpoint**: Optional shorthand for complete URL to use for the constructed client.

- **region**: Optional shorthand for AWS Region used in instantiating the client.
Value

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

Service syntax

```r
csvc <- lexruntimeservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  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)
```

Description

This section contains documentation for the Amazon Lex V2 Runtime V2 API operations.

Usage

```r
lexruntimev2(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

- **config**: Optional configuration of credentials, endpoint, and/or region.
  - **credentials**:
    - **creds**:
      - `access_key_id`: AWS access key ID
      - `secret_access_key`: AWS secret access key
      - `session_token`: AWS temporary session token
    - **profile**: The name of a profile to use. If not given, then the default profile is used.
    - **anonymous**: Set anonymous credentials.
    - **endpoint**: The complete URL to use for the constructed client.
    - **region**: The AWS Region used in instantiating the client.
  - **close_connection**: Immediately close all HTTP connections.
  - **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  - **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.  

**sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy [https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html](https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html)

**credentials**: Optional credentials shorthand for the config parameter

- **creds**:
  - `access_key_id`: AWS access key ID
  - `secret_access_key`: AWS secret access key
  - `session_token`: AWS temporary session token

- **profile**: The name of a profile to use. If not given, then the default profile is used.

- **anonymous**: Set anonymous credentials.

**endpoint**: Optional shorthand for complete URL to use for the constructed client.

**region**: Optional shorthand for AWS Region used in instantiating the client.

### Value

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

### Service syntax

```r
svc <- lexruntimev2(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
```
lookoutequipment Amazon Lookout for Equipment

Description

Amazon Lookout for Equipment is a machine learning service that uses advanced analytics to identify anomalies in machines from sensor data for use in predictive maintenance.

Usage

lookoutequipment(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
Arguments

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

- credentials:
  - creds:
    - access_key_id: AWS access key ID
    - secret_access_key: AWS secret access key
    - session_token: AWS temporary session token
  - profile: The name of a profile to use. If not given, then the default profile is used.
  - anonymous: Set anonymous credentials.
  - endpoint: The complete URL to use for the constructed client.
  - region: The AWS Region used in instantiating the client.
- close_connection: Immediately close all HTTP connections.
- timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
- sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html

credentials Optional credentials shorthand for the config parameter

- creds:
  - access_key_id: AWS access key ID
  - secret_access_key: AWS secret access key
  - session_token: AWS temporary session token
  - profile: The name of a profile to use. If not given, then the default profile is used.
  - anonymous: Set anonymous credentials.

default

endpoint Optional shorthand for complete URL to use for the constructed client.

region Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

svc <- lookoutequipment(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
      )
    )
  )
)
secret_access_key = "string",
session_token = "string"
),
profile = "string",
anonymous = "logical"
),
endpoint = "string",
region = "string",
close_connection = "logical",
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string"
),
credentials = list(
creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
),
    profile = "string",
anonymous = "logical"
),
endpoint = "string",
region = "string"
)

Operations

create_dataset
create_inference_scheduler
create_label
create_label_group
create_model
delete_dataset
delete_inference_scheduler
delete_label
delete_label_group
delete_model
delete_resource_policy
describe_data_ingestion_job
describe_dataset
describe_inference_scheduler
describe_label
describe_label_group
describe_model
describe_model_version
describe_resource_policy
import_dataset

Creates a container for a collection of data being ingested for analysis
Creates a scheduled inference
Creates a label for an event
Creates a group of labels
Creates an ML model for data inference
Deletes a dataset and associated artifacts
Deletes an inference scheduler that has been set up
Deletes a label
Deletes a group of labels
Deletes an ML model currently available for Amazon Lookout for Equipment
Deletes the resource policy attached to the resource
Provides information on a specific data ingestion job such as creation time, dataset ARN, and status
Provides a JSON description of the data in each time series dataset, including names, column names, data types, and timestamps
Specifies information about the inference scheduler being used, including name, model, status, and associated metadata
Returns the name of the label
Returns information about the label group
Provides a JSON containing the overall information about a specific ML model, including model name, ARN, dataset, training and evaluation information, status, and so on
Retrieves information about a specific machine learning model version
Provides the details of a resource policy attached to a resource
Imports a dataset
import_model_version | Imports a model that has been trained successfully
list_data_ingestion_jobs | Provides a list of all data ingestion jobs, including dataset name and ARN, S3 location of the
list_datasets | Lists all datasets currently available in your account, filtering on the dataset name
list_inference_events | Lists all inference events that have been found for the specified inference scheduler
list_inference_executions | Lists all inference executions that have been performed by the specified inference scheduler
list_inference_schedulers | Retrieves a list of all inference schedulers currently available for your account
list_label_groups | Returns a list of the label groups
list_labels | Provides a list of labels
list_models | Generates a list of all models in the account, including model name and ARN, dataset, and status
list_model_versions | Generates a list of all model versions for a given model, including the model version, model version ARN, and status
list_sensor_statistics | Lists statistics about the data collected for each of the sensors that have been successfully ingested
list_tags_for_resource | Lists all the tags for a specified resource, including key and value
put_resource_policy | Creates a resource control policy for a given resource
start_data_ingestion_job | Starts a data ingestion job
start_inference_scheduler | Starts an inference scheduler
stop_inference_scheduler | Stops an inference scheduler
tag_resource | Associates a given tag to a resource in your account
untag_resource | Removes a specific tag from a given resource
update_active_model_version | Sets the active model version for a given machine learning model
update_inference_scheduler | Updates an inference scheduler
update_label_group | Updates the label group

Examples

## Not run:
svc <- lookoutequipment()
svc$create_dataset(
  Foo = 123
)
## End(Not run)

lookoutmetrics

Amazon Lookout for Metrics

Description

This is the Amazon Lookout for Metrics API Reference. For an introduction to the service with tutorials for getting started, visit Amazon Lookout for Metrics Developer Guide.

Usage

lookoutmetrics(
  config = list(),
)
lookoutmetrics

```python
credentials = list(),
endpoint = NULL,
region = NULL
)
```

## Arguments

### config

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

- **credentials**: 
  - **creds**: 
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
  - **endpoint**: The complete URL to use for the constructed client.
  - **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy [https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-ends.html](https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-ends.html)

### credentials

Optional credentials shorthand for the config parameter

- **creds**: 
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

### endpoint

Optional shorthand for complete URL to use for the constructed client.

### region

Optional shorthand for AWS Region used in instantiating the client.

## Value

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

```r
svc <- lookoutmetrics(
    config = list(
        credentials = list(
            creds = list(
                access_key_id = "string",
                secret_access_key = "string",
                session_token = "string"
            ),
            profile = "string",
            anonymous = "logical"
        ),
        endpoint = "string",
        region = "string",
        close_connection = "logical",
        timeout = "numeric",
        s3_force_path_style = "logical",
        sts_regional_endpoint = "string"
    ),
    credentials = list(
        creds = list(
            access_key_id = "string",
            secret_access_key = "string",
            session_token = "string"
        ),
        profile = "string",
        anonymous = "logical"
    ),
    endpoint = "string",
    region = "string"
)
```

Operations

- `activate_anomaly_detector`: Activates an anomaly detector
- `back_test_anomaly_detector`: Runs a backtest for anomaly detection for the specified resource
- `create_alert`: Creates an alert for an anomaly detector
- `create_anomaly_detector`: Creates an anomaly detector
- `create_metric_set`: Creates a dataset
- `deactivate_anomaly_detector`: Deactivates an anomaly detector
- `delete_alert`: Deletes an alert
- `delete_anomaly_detector`: Deletes a detector
- `describe_alert`: Describes an alert
- `describe_anomaly_detection_executions`: Returns information about the status of the specified anomaly detection jobs
- `describe_anomaly_detector`: Describes a detector
- `describe_metric_set`: Describes a dataset
- `detect_metric_set_config`:Detects an Amazon S3 dataset’s file format, interval, and offset
- `get_anomaly_group`: Returns details about a group of anomalous metrics
machinelearning

get_data_quality_metrics
get_feedback
get_sample_data
list_alerts
list_anomaly_detectors
list_anomaly_group_related_metrics
list_anomaly_group_summaries
list_anomaly_group_time_series
list_metric_sets
list_tags_for_resource
put_feedback
tag_resource
untag_resource
update_alert
update_anomaly_detector
update_metric_set

Returns details about the requested data quality metrics
Get feedback for an anomaly group
Returns a selection of sample records from an Amazon S3 datasource
Lists the alerts attached to a detector
Lists the detectors in the current AWS Region
Returns a list of measures that are potential causes or effects of an anomaly group
Returns a list of anomaly groups
Gets a list of anomalous metrics for a measure in an anomaly group
Lists the datasets in the current AWS Region
Gets a list of tags for a detector, dataset, or alert
Add feedback for an anomalous metric
Adds tags to a detector, dataset, or alert
Removes tags from a detector, dataset, or alert
Make changes to an existing alert
Updates a detector
Updates a dataset

Examples

```r
## Not run:
svc <- lookoutmetrics()
svc$activate_anomaly_detector(
  Foo = 123
)

## End(Not run)
```

---

**Description**

Definition of the public APIs exposed by Amazon Machine Learning

**Usage**

```r
machinelearning(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```
Arguments

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

- **credentials**:
  - **creds**:
    - *access_key_id*: AWS access key ID
    - *secret_access_key*: AWS secret access key
    - *session_token*: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
  - **endpoint**: The complete URL to use for the constructed client.
  - **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy

**credentials**  
Optional credentials shorthand for the config parameter

- **creds**:
  - *access_key_id*: AWS access key ID
  - *secret_access_key*: AWS secret access key
  - *session_token*: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

**endpoint**  
Optional shorthand for complete URL to use for the constructed client.

**region**  
Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

secret_access_key = "string",
session_token = "string"
),
profile = "string",
anonymous = "logical"
),
endpoint = "string",
region = "string",
close_connection = "logical",
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)

Operations

add_tags
create_batch_prediction
create_data_source_from_rds
create_data_source_from_redshift
create_data_source_from_s3
create_evaluation
create_ml_model
create realtime_endpoint
delete_batch_prediction
delete_data_source
delete_evaluation
delete_ml_model
delete_realtime_endpoint
delete_tags
describe_batch_predictions
describe_data_sources
describe_evaluations
describe_ml_models
describe_tags
get_batch_prediction

Adds one or more tags to an object, up to a limit of 10
Generates predictions for a group of observations
Creates a DataSource object from an Amazon Relational Database Service (Amazon RDS)
Creates a DataSource from a database hosted on an Amazon Redshift cluster
Creates a DataSource object
Creates a new Evaluation of an MLModel
Creates a new MLModel using the DataSource and the recipe as information sources
Assigns the DELETED status to a BatchPrediction, rendering it unusable
Assigns the DELETED status to a DataSource, rendering it unusable
Assigns the DELETED status to an Evaluation, rendering it unusable
Assigns the DELETED status to an MLModel, rendering it unusable
Deletes a real time endpoint of an MLModel
Deletes the specified tags associated with an ML object
Returns a list of BatchPrediction operations that match the search criteria in the request
Returns a list of DataSource that match the search criteria in the request
Returns a list of DescribeEvaluations that match the search criteria in the request
Returns a list of MLModel that match the search criteria in the request
Describes one or more of the tags for your Amazon ML object
Returns a BatchPrediction that includes detailed metadata, status, and data file information
get_data_source
get_evaluation
get_ml_model
predict
update_batch_prediction
update_data_source
update_evaluation
update_ml_model

Returns a DataSource that includes metadata and data file information, as well as the current status of the DataSource.

Returns an Evaluation that includes metadata as well as the current status of the Evaluation.

Returns an MLModel that includes detailed metadata, data source information, and the current status of the MLModel.

Generates a prediction for the observation using the specified ML Model.

Updates the BatchPredictionName of a BatchPrediction.

Updates the DataSourceName of a DataSource.

Updates the EvaluationName of an Evaluation.

Updates the MLModelName and the ScoreThreshold of an MLModel.

Examples

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

Description

Overview

This is the AWS Panorama API Reference. For an introduction to the service, see What is AWS Panorama? in the AWS Panorama Developer Guide.

Usage

```
panorama(config = list(), credentials = list(), endpoint = NULL, region = NULL)
```

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>
<tr>
<td>credentials</td>
<td>• credentials:</td>
</tr>
<tr>
<td></td>
<td>• creds:</td>
</tr>
<tr>
<td></td>
<td>• access_key_id: AWS access key ID</td>
</tr>
<tr>
<td></td>
<td>• secret_access_key: AWS secret access key</td>
</tr>
<tr>
<td></td>
<td>• session_token: AWS temporary session token</td>
</tr>
<tr>
<td></td>
<td>• profile: The name of a profile to use. If not given, then the default</td>
</tr>
<tr>
<td></td>
<td>profile is used.</td>
</tr>
<tr>
<td></td>
<td>• anonymous: Set anonymous credentials.</td>
</tr>
</tbody>
</table>
• **endpoint**: The complete URL to use for the constructed client.
• **region**: The AWS Region used in instantiating the client.
  
  • **close_connection**: Immediately close all HTTP connections.
  • **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  • **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
  • **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy [https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html](https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html)

**credentials**

Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
  
  • **profile**: The name of a profile to use. If not given, then the default profile is used.
  • **anonymous**: Set anonymous credentials.

**endpoint**

Optional shorthand for complete URL to use for the constructed client.

**region**

Optional shorthand for AWS Region used in instantiating the client.

**Value**

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

**Service syntax**

```r
svc <- panorama(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  )
)```

credentials = list(
    creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
),
endpoint = "string",
region = "string"
)

Operations

create_application_instance Creates an application instance and deploys it to a device
create_job_for_devices Creates a job to run on a device
create_node_from_template_job Creates a camera stream node
create_package Creates a package and storage location in an Amazon S3 access point
create_package_import_job Imports a node package
delete_device Deletes a device
delete_package Deletes a package
deregister_package_version Deregisters a package version
describe_application_instance Returns information about an application instance on a device
describe_application_instance_details Returns information about an application instance’s configuration manifest
describe_device Returns information about a device
describe_device_job Returns information about a device job
describe_node Returns information about a node
describe_node_from_template_job Returns information about a job to create a camera stream node
describe_package Returns information about a package
describe_package_import_job Returns information about a package import job
describe_package_version Returns information about a package version
list_application_instance_dependencies Returns a list of application instance dependencies
list_application_instance_node_instances Returns a list of application node instances
list_application_instances Returns a list of application instances
list_devices Returns a list of devices
list_devices_jobs Returns a list of jobs
list_node_from_template_jobs Returns a list of camera stream node jobs
list_nodes Returns a list of nodes
list_package_import_jobs Returns a list of package import jobs
list_packages Returns a list of packages
list_tags_for_resource Returns a list of tags for a resource
provision_device Creates a device and returns a configuration archive
register_package_version Registers a package version
remove_application_instance Removes an application instance
signal_application_instance_node_instances Signal camera nodes to stop or resume
tag_resource Tags a resource
untag_resource
update_device_metadata

Examples

```r
## Not run:
svc <- panorama()
svc$create_application_instance(
    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

```r
personalize(
    config = list(),
    credentials = list(),
    endpoint = NULL,
    region = NULL
)
```

Arguments

- `config`: Optional configuration of credentials, endpoint, and/or region.
  - `credentials`:
    - `creds`:
      - `access_key_id`: AWS access key ID
      - `secret_access_key`: AWS secret access key
      - `session_token`: AWS temporary session token
    - `profile`: The name of a profile to use. If not given, then the default profile is used.
    - `anonymous`: Set anonymous credentials.
    - `endpoint`: The complete URL to use for the constructed client.
    - `region`: The AWS Region used in instantiating the client.
personalize

- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.

**credentials**
Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token
- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.

**endpoint**
Optional shorthand for complete URL to use for the constructed client.

**region**
Optional shorthand for AWS Region used in instantiating the client.

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

**Service syntax**

```r
svc <- personalize(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
```
creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
),
profile = "string",
anonymous = "logical"
),
endpoint = "string",
region = "string"
)

Operations

create_batch_inference_job  Creates a batch inference job
create_batch_segment_job    Creates a batch segment job
create_campaign            Creates a campaign that deploys a solution version
create_dataset             Creates an empty dataset and adds it to the specified dataset group
create_dataset_export_job  Creates a job that exports data from your dataset to an Amazon S3 bucket
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 adding event data to a specified dataset group using the PutEvents API
create_filter              Creates a recommendation filter
create_metric_attribution  Creates a metric attribution
create_recommender         Creates a recommender with the recipe (a Domain dataset group use case) you specify
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 retracts an active solution in a Custom dataset group
delete_campaign            Removes a campaign by deleting the solution deployment
delete_dataset             Deletes a dataset
delete_dataset_group       Deletes a dataset group
delete_event_tracker       Deletes the event tracker
delete_filter              Deletes a filter
delete_metric_attribution  Deletes a metric attribution
delete_recommender         Deactivates and removes a recommender
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), status, input and output configurations, and the ARN of the solution version used to generate the recommendations
describe_batch_segment_job Gets the properties of a batch segment job including name, Amazon Resource Name (ARN), status, input and output configurations, and the ARN of the solution version used to generate segments
describe_campaign          Describes the given campaign, including its status
describe_dataset           Describes the given dataset
describe_dataset_export_job Describes the dataset export job created by CreateDatasetExportJob, including the export job name, Amazon Resource Name (ARN), status, input and output configurations
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), status, input and output configurations
describe_event_tracker     Describes an event tracker
describe_feature_transformation Describes a filter’s properties
## personalize

describe_metric_attribution  
Describes a metric attribution

describe_recipe  
Describes a recipe

describe_recommender  
Describes the given recommender, including its status

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_batch_segment_jobs  
Gets a list of the batch segment jobs that have been performed off of a solution version that

list_campaigns  
Returns a list of campaigns that use the given solution

list_dataset_export_jobs  
Returns a list of dataset export jobs that use the given dataset

list_dataset_groups  
Returns a list of dataset groups

list_dataset_import_jobs  
Returns a list of dataset import jobs that use the given dataset

list_datasets  
Returns the list of datasets contained in the given dataset group

list_event_trackers  
Returns the list of event trackers associated with the account

list_filters  
Lists all filters that belong to a given dataset group

list_metric_attribution_metrics  
Lists the metrics for the metric attribution

list_metric_attributions  
Lists metric attributions

list_recipes  
Returns a list of available recipes

list_recommenders  
Returns a list of recommenders in a given Domain dataset group

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

list_tags_for_resource  
Get a list of tags attached to a resource

start_recommender  
Starts a recommender that is INACTIVE

stop_recommender  
Stops a recommender that is ACTIVE

stop_solution_version_creation  
Stops creating a solution version that is in a state of CREATE_PENDING or CREATE_IN_PROGRESS

tag_resource  
Add a list of tags to a resource

untag_resource  
Remove tags that are attached to a resource

update_campaign  
Updates a campaign by either deploying a new solution or changing the value of the campaign's minProvisionedTPS parameter

update_dataset  
Update a dataset to replace its schema with a new or existing one

update_metric_attribution  
Updates a metric attribution

update_recommender  
Updates the recommender to modify the recommender configuration

### Examples

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

## End(Not run)
personalizeevents  Amazon Personalize Events

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

Usage
personalizeevents(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)

Arguments
config Optional configuration of credentials, endpoint, and/or region.
  • credentials:
    – creds:
      * access_key_id: AWS access key ID
      * secret_access_key: AWS secret access key
      * session_token: AWS temporary session token
    – profile: The name of a profile to use. If not given, then the default profile is used.
    – anonymous: Set anonymous credentials.
    – endpoint: The complete URL to use for the constructed client.
    – region: The AWS Region used in instantiating the client.
  • close_connection: Immediately close all HTTP connections.
  • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
  • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html

credentials Optional credentials shorthand for the config parameter
  • creds:
    – access_key_id: AWS access key ID
    – secret_access_key: AWS secret access key
    – session_token: AWS temporary session token
• **profile**: The name of a profile to use. If not given, then the default profile is used.

• **anonymous**: Set anonymous credentials.

endpoint Optional shorthand for complete URL to use for the constructed client.
region Optional shorthand for AWS Region used in instantiating the client.

**Value**

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

**Service syntax**

```r
svc <- personalizeevents(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  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(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

### Arguments

- **config**: Optional configuration of credentials, endpoint, and/or region.
  - **credentials**:
    - **creds**:
      - **access_key_id**: AWS access key ID
      - **secret_access_key**: AWS secret access key
      - **session_token**: AWS temporary session token
    - **profile**: The name of a profile to use. If not given, then the default profile is used.
    - **anonymous**: Set anonymous credentials.
    - **endpoint**: The complete URL to use for the constructed client.
    - **region**: The AWS Region used in instantiating the client.
  - **close_connection**: Immediately close all HTTP connections.
personalizeruntime

• **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.

• **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.

• **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-ehml

credentials Optional credentials shorthand for the config parameter

• **creds**:
  – **access_key_id**: AWS access key ID
  – **secret_access_key**: AWS secret access key
  – **session_token**: AWS temporary session token

• **profile**: The name of a profile to use. If not given, then the default profile is used.

• **anonymous**: Set anonymous credentials.

documentation Optional shorthand for complete URL to use for the constructed client.

region Optional shorthand for AWS Region used in instantiating the client.

**Value**

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

**Service syntax**

```
svc <- personalizeruntime(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
    ),
    profile = "string",
    anonymous = "logical"
  )
)
```
polly

```
secret_access_key = "string",
session_token = "string"
),
profile = "string",
anonymous = "logical"
),
endpoint = "string",
region = "string"
)

Operations

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

Examples

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

polly

Amazon Polly

Description

Amazon Polly is a web service that makes it easy to synthesize speech from text.
The Amazon Polly service provides API operations for synthesizing high-quality speech from plain

text and Speech Synthesis Markup Language (SSML), along with managing pronunciations lexicons
that enable you to get the best results for your application domain.

Usage

polly(config = list(), credentials = list(), endpoint = NULL, region = NULL)

Arguments

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

  • credentials:
- **creds:**
  - **access_key_id:** AWS access key ID
  - **secret_access_key:** AWS secret access key
  - **session_token:** AWS temporary session token
- **profile:** The name of a profile to use. If not given, then the default profile is used.
- **anonymous:** Set anonymous credentials.
- **endpoint:** The complete URL to use for the constructed client.
- **region:** The AWS Region used in instantiating the client.

  - **close_connection:** Immediately close all HTTP connections.
  - **timeout:** The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  - **s3_force_path_style:** Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
  - **sts_regional_endpoint:** Set sts regional endpoint resolver to regional or legacy [https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html](https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html)

**credentials**
Optional credentials shorthand for the config parameter

- **creds:**
  - **access_key_id:** AWS access key ID
  - **secret_access_key:** AWS secret access key
  - **session_token:** AWS temporary session token
- **profile:** The name of a profile to use. If not given, then the default profile is used.
- **anonymous:** Set anonymous credentials.

**endpoint**
Optional shorthand for complete URL to use for the constructed client.

**region**
Optional shorthand for AWS Region used in instantiating the client.

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

**Service syntax**

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


```
polly

anonymous = "logical"
),
endpoint = "string",
region = "string",
close_connection = "logical",
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string"
),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)

Operations

delete_lexicon Deletes the specified pronunciation lexicon stored in an Amazon Web Services 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 Amazon Web Services 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 Amazon Web Services Region
list_speech_synthesis_tasks Returns a list of SpeechSynthesisTask objects ordered by their creation date
put_lexicon Stores a pronunciation lexicon in an Amazon Web Services 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 API Reference for Amazon Rekognition Image, Amazon Rekognition Custom Labels, Amazon Rekognition Stored Video, Amazon Rekognition Streaming Video. It provides descriptions of actions, data types, common parameters, and common errors.

Amazon Rekognition Image

- associate_faces
- compare_faces
- create_collection
- create_user
- delete_collection
- delete_faces
- delete_user
- describe_collection
- detect_faces
- detect_labels
- detect_moderation_labels
- detect_protective_equipment
- detect_text
- disassociate_faces
- get_celebrity_info
- index_faces
- list_collections
- list_faces
- list_users
- recognize_celebrities
- search_faces
- search_faces_by_image
- search_users
- search_users_by_image

Amazon Rekognition Custom Labels

- copy_project_version
- create_dataset
• create_project
• create_project_version
• delete_dataset
• delete_project
• delete_project_policy
• delete_project_version
• describe_dataset
• describe_projects
• describe_project_versions
• detect_custom_labels
• distribute_dataset_entries
• list_dataset_entries
• list_dataset_labels
• list_project_policies
• put_project_policy
• start_project_version
• stop_project_version
• update_dataset_entries

Amazon Rekognition Video Stored Video

• get_celebrity_recognition
• get_content_moderation
• get_face_detection
• get_face_search
• get_label_detection
• get_person_tracking
• get_segment_detection
• get_text_detection
• start_celebrity_recognition
• start_content_moderation
• start_face_detection
• start_face_search
• start_label_detection
• start_person_tracking
• start_segment_detection
• start_text_detection

Amazon Rekognition Video Streaming Video
• `create_stream_processor`
• `delete_stream_processor`
• `describe_stream_processor`
• `list_stream_processors`
• `start_stream_processor`
• `stop_stream_processor`
• `update_stream_processor`

Usage

rekognition(
    config = list(),
    credentials = list(),
    endpoint = NULL,
    region = NULL
)

Arguments

cfg                  Optional configuration of credentials, endpoint, and/or region.
    • `credentials`
        – `creds`:
            • access_key_id: AWS access key ID
            • secret_access_key: AWS secret access key
            • session_token: AWS temporary session token
        – profile: The name of a profile to use. If not given, then the default profile is used.
        – anonymous: Set anonymous credentials.
        – endpoint: The complete URL to use for the constructed client.
        – region: The AWS Region used in instantiating the client.
    • `close_connection`: Immediately close all HTTP connections.
    • `timeout`: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
    • `s3_force_path_style`: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
    • `sts_regional_endpoint`: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html

credentials         Optional credentials shorthand for the config parameter
    • `creds`:
        – access_key_id: AWS access key ID
        – secret_access_key: AWS secret access key
        – session_token: AWS temporary session token
• **profile**: The name of a profile to use. If not given, then the default profile is used.

• **anonymous**: Set anonymous credentials.

**endpoint**
Optional shorthand for complete URL to use for the constructed client.

**region**
Optional shorthand for AWS Region used in instantiating the client.

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

**Service syntax**
```
svc <- rekognition(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

**Operations**

* **associate_faces** Associates one or more faces with an existing UserID*
compare_faces
copy_project_version
create_collection
create_dataset
create_face_liveness_session
create_project
create_project_version
create_stream_processor
create_user
delete_collection
delete_dataset
delete_faces
delete_project
delete_project_policy
delete_project_version
delete_stream_processor
delete_user
describe_collection
describe_dataset
describe_projects
describe_project_versions
describe_stream_processor
detect_custom_labels
detect_faces
detect_labels
detect_moderation_labels
detect_protective_equipment
detect_text
disassociate_faces
distribute_dataset_entries
get_celebrity_info
get_celebrity_recognition
get_content_moderation
get_face_detection
get_face_liveness_session_results
get_face_search
get_label_detection
get_person_tracking
get_segment_detection
get_text_detection
index_faces
list_collections
list_dataset_entries
list_dataset_labels
list_faces
list_project_policies
list_stream_processors
list_tags_for_resource

compares a face in the source input image with each of the 100 largest faces detected in the target input image.
Copies a version of an Amazon Rekognition Custom Labels model from a source project to a destination project.
Creates a collection in an AWS Region.
Creates a new Amazon Rekognition Custom Labels dataset.
This API operation initiates a Face Liveness session.
Creates a new Amazon Rekognition Custom Labels project.
Creates an Amazon Rekognition stream processor that you can use to detect and recognize faces.
Creates a new User within a collection specified by CollectionId.
Deletes the specified collection.
Deletes an existing Amazon Rekognition Custom Labels dataset.
Deletes faces from a collection.
Deletes an Amazon Rekognition Custom Labels project.
Deletes an existing project policy.
Deletes an Amazon Rekognition Custom Labels model.
Deletes the stream processor identified by Name.
Deletes the specified UserID within the collection.
Describes the specified collection.
Describes an Amazon Rekognition Custom Labels dataset.
Gets information about your Amazon Rekognition Custom Labels projects.
Lists and describes the versions of a model in an Amazon Rekognition Custom Labels project.
Provides information about a stream processor created by CreateStreamProcessor.
Detects custom labels in a supplied image by using an Amazon Rekognition Custom Labels model.
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 Personal Protective Equipment (PPE) worn by people detected in an image.
Detects text in the input image and converts it into machine-readable text.
Removes the association between a Face supplied in an array of FaceIds and the User.
Distributes the entries (images) in a training dataset across the training dataset and the test dataset.
Gets the name and additional information about a celebrity based on their Amazon Rekognition ID.
Gets the celebrity recognition results for a Amazon Rekognition Video analysis started by StartCelebrityRecognition.
Gets the inappropriate, unwanted, or offensive 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.
Retrieves the results of a specific Face Liveness session.
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.
Gets the segment detection results of a Amazon Rekognition Video analysis started by StartSegmentDetection.
Gets the text detection results of a Amazon Rekognition Video analysis started by StartTextDetection.
Detects faces in the input image and adds them to the specified collection.
Returns list of collection IDs in your account.
Lists the entries (images) within a dataset.
Lists the labels in a dataset.
Returns metadata for faces in the specified collection.
Gets a list of the project policies attached to a project.
Gets a list of stream processors that you have created with CreateStreamProcessor.
Returns a list of tags in an Amazon Rekognition collection, stream processor, or Custom Labels model.
list_users
put_project_policy
recognize_celebrities
search_faces
search_faces_by_image
search_users
search_users_by_image
start_celebrity_recognition
start_content_moderation
start_face_detection
start_face_search
start_label_detection
start_person_tracking
start_project_version
start_segment_detection
start_stream_processor
start_text_detection
stop_project_version
stop_stream_processor
tag_resource
untag_resource
update_dataset_entries
update_stream_processor

Returns metadata of the User such as UserID in the specified collection
Attaches a project policy to a Amazon Rekognition Custom Labels project in a trusting AWS account
Returns an array of celebrities recognized in the input image
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
Searches for UserIDs within a collection based on a FaceId or UserId
Searches for UserIDs using a supplied image
Starts asynchronous recognition of celebrities in a stored video
Starts asynchronous detection of inappropriate, unwanted, or offensive 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 a video
Starts the 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 asynchronous detection of segment detection in a video
Starts processing a stream processor
Starts asynchronous detection of text in a stored video
Stops a running model
Starts a running stream processor that was created by CreateStreamProcessor
Adds one or more key-value tags to an Amazon Rekognition collection, stream processor, or model
Removes one or more tags from an Amazon Rekognition collection, stream processor, or model
Adds or updates one or more entries (images) in a dataset
Allows you to update a stream processor

**Examples**

```r
## Not run:
svc <- rekognition()

# This operation associates one or more faces with an existing UserID.
svc$associate_faces(
  ClientRequestToken = "550e8400-e29b-41d4-a716-446655440002",
  CollectionId = "MyCollection",
  FaceIds = list(
    "f5817d37-94f6-4335-bfee-6cf79a3d006e",
    "851cb847-dccc-4fea-9309-9f4805967855",
    "35ebbb41-7f67-4263-908d-dd0ecba05ab9"
  ),
  UserId = "DemoUser",
  UserMatchThreshold = 70L
)

## End(Not run)
```
**Amazon SageMaker Service**

**Description**

Provides APIs for creating and managing SageMaker resources.

Other Resources:

- SageMaker Developer Guide
- Amazon Augmented AI Runtime API Reference

**Usage**

```r
sagemaker(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

**Arguments**

- **config**: Optional configuration of credentials, endpoint, and/or region.
  - **credentials**:
    - **creds**:
      - **access_key_id**: AWS access key ID
      - **secret_access_key**: AWS secret access key
      - **session_token**: AWS temporary session token
    - **profile**: The name of a profile to use. If not given, then the default profile is used.
    - **anonymous**: Set anonymous credentials.
    - **endpoint**: The complete URL to use for the constructed client.
    - **region**: The AWS Region used in instantiating the client.
  - **close_connection**: Immediately close all HTTP connections.
  - **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  - **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
  - **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy.

- **credentials**: Optional credentials shorthand for the config parameter
  - **creds**:
    - **access_key_id**: AWS access key ID
- **secret_access_key**: AWS secret access key
- **session_token**: AWS temporary session token
  
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  
  - **anonymous**: Set anonymous credentials.

  **endpoint**
  
  Optional shorthand for complete URL to use for the constructed client.

  **region**
  
  Optional shorthand for AWS Region used in instantiating the client.

**Value**

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

**Service syntax**

```r
svc <- sagemaker(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```
Operations
add_association
add_tags
associate_trial_component
batch_describe_model_package
create_action
create_algorithm
create_app
create_app_image_config
create_artifact
create_auto_ml_job
create_auto_ml_job_v2
create_code_repository
create_compilation_job
create_context
create_data_quality_job_definition
create_device_fleet
create_domain
create_edge_deployment_plan
create_edge_deployment_stage
create_edge_packaging_job
create_endpoint
create_endpoint_config
create_experiment
create_feature_group
create_flow_definition
create_hub
create_human_task_ui
create_hyper_parameter_tuning_job
create_image
create_image_version
create_inference_experiment
create_inference_recommendations_job
create_labeling_job
create_model
create_model_bias_job_definition
create_model_card
create_model_card_export_job
create_model_explainability_job_definition
create_model_package
create_model_package_group
create_model_quality_job_definition
create_monitoring_schedule
create_notebook_instance
create_notebook_instance_lifecycle_config
create_preliminary_diff
create_presigned_domain_url
create_presigned_notebook_instance_url
create_processing_job

Creates an association between the source and the destination
Adds or overwrites one or more tags for the specified SageMaker resource
Associates a trial component with a trial
This action batch describes a list of versioned model packages
Creates an action
Create a machine learning algorithm that you can use in SageMaker and list in the Amazon Web Services Marketplace
Creates a configuration for running a SageMaker image as a KernelGateway app
Creates an artifact
Creates an Autopilot job also referred to as Autopilot experiment or AutoML job
Creates an Autopilot job also referred to as Autopilot experiment or AutoML job V2
Creates a Git repository as a resource in your SageMaker account
Starts a model compilation job
Creates a context
Creates a definition for a job that monitors data quality and drift
Creates a device fleet
Creates a Domain used by Amazon SageMaker Studio
Creates an edge deployment plan, consisting of multiple stages
Creates a new stage in an existing edge deployment plan
Starts a SageMaker Edge Manager model packaging job
Creates an endpoint using the endpoint configuration specified in the request
Creates an endpoint configuration that SageMaker hosting services uses
Starts a SageMaker experiment
Create a new FeatureGroup
Creates a flow definition
Create a hub
Defines the settings you will use for the human review workflow user interface
Starts a hyperparameter tuning job
Creates a custom SageMaker image
Creates a version of the SageMaker image specified by ImageName
Creates an inference experiment using the configurations specified in the request
Starts a recommendation job
Creates a job that uses workers to label the data objects in your input dataset
Creates a model in SageMaker
Creates the definition for a model bias job
Creates an Amazon SageMaker Model Card
Creates an Amazon SageMaker Model Card export job
Creates the definition for a model explainability job
Creates a model package that you can use to create SageMaker models
Creates a model group
Creates a definition for a job that monitors model quality and drift
Creates a schedule that regularly starts Amazon SageMaker Processing jobs
Creates an SageMaker notebook instance
Creates a lifecycle configuration that you can associate with a notebook
Creates a pipeline using a JSON pipeline definition
Creates a URL for a specified UserProfile in a Domain
Returns a URL that you can use to connect to the Jupyter server from a notebook
 Creates a processing job
create_project
create_space
create_studio_lifecycle_config
create_training_job
create_transform_job
create_trial
create_trial_component
create_user_profile
create_workforce
delete_action
delete_algorithm
delete_app
delete_app_image_config
delete_artifact
delete_association
delete_code_repository
delete_context
delete_data_quality_job_definition
delete_device_fleet
delete_domain
delete_edge_deployment_plan
delete_edge_deployment_stage
delete_endpoint
delete_endpoint_config
delete_experiment
delete_feature_group
delete_flow_definition
delete_hub
delete_hub_content
delete_human_task_ui
delete_image
delete_image_version
delete_inference_experiment
delete_model
delete_model_bias_job_definition
delete_model_card
create a machine learning (ML) project that can contain one or more templates that set up an ML pipeline from training to deploying an approved model
create a space used for real time collaboration in a Domain
create a new Studio Lifecycle Configuration
starts a model training job
starts a transform job
create an SageMaker trial
create a trial component, which is a stage of a machine learning trial
create a user profile
use this operation to create a workforce
create a new work team for labeling your data
delete an action
removes the specified algorithm from your account
used to stop and delete an app
delete an AppImageConfig
deletes an artifact
deletes an association
deletes the specified Git repository from your account
deletes a context
deletes a data quality monitoring job definition
deletes a fleet
used to delete a domain
deletes an edge deployment plan if (and only if) all the stages in the plan are inactive or there are no stages in the plan
deletes a stage in an edge deployment plan if (and only if) the stage is inactive
deletes an endpoint
deletes an endpoint configuration
deletes an SageMaker experiment
delete the FeatureGroup and any data that was written to the OnlineStore of the FeatureGroup
deletes the specified flow definition
delete a hub
delete the contents of a hub
use this operation to delete a human task user interface (worker task template)
deletes a SageMaker image and all versions of the image
deletes a version of a SageMaker image
deletes an inference experiment
deletes a model
deletes an Amazon SageMaker model bias job definition
deletes an Amazon SageMaker Model Card
deletes an Amazon SageMaker model explainability job definition
deletes a model package
deletes the specified model group
deletes a model group resource policy
deletes the specified model quality monitoring job definition
deletes a monitoring schedule
deletes an SageMaker notebook instance
deletes a notebook instance lifecycle configuration
deletes a pipeline if there are no running instances of the pipeline
deletes the specified project
used to delete a space
delete_studio_lifecycle_config
delete_tags
delete_trial
delete_trial_component
delete_user_profile
delete_workforce
delete_workteam
deregister_devices
describe_action
describe_algorithm
describe_app
describe_app_image_config
describe_artifact
describe_auto_ml_job
describe_auto_ml_job_v2
describe_code_repository
describe_compilation_job
describe_context
describe_data_quality_job_definition
describe_device
describe_device_fleet
describe_domain
describe_edge_deployment_plan
describe_edge_packaging_job
describe_endpoint
describe_endpoint_config
describe_experiment
describe_feature_group
describe_feature_metadata
describe_flow_definition
describe_hub
describe_hub_content
describe_human_task_ui
describe_hyper_parameter_tuning_job
describe_image
describe_image_version
describe_inference_experiment
describe_inference_recommendations_job
describe_labeling_job
describe_lineage_group
describe_model
describe_model_bias_job_definition
describe_model_card
describe_model_card_export_job
describe_model_explainability_job_definition
describe_model_package
describe_model_package_group
describe_model_quality_job_definition

Deletes the Studio Lifecycle Configuration
Deletes the specified tags from an SageMaker resource
Deletes the specified trial
Deletes the specified trial component
Deletes a user profile
Use this operation to delete a workforce
Deletes an existing work team
Deregisters the specified devices
Describes an action
Returns a description of the specified algorithm that is in your account
Describes the app
Describes an AppImageConfig
Describes an artifact
Returns information about an AutoML job created by calling CreateAutoMLJob
Returns information about an AutoML job created by calling CreateAutoMLJobV2
Gets details about the specified Git repository
Returns information about a model compilation job
Describes a context
Gets the details of a data quality monitoring job definition
Describes the device
A description of the fleet the device belongs to
The description of the domain
Describes an edge deployment plan with deployment status per stage
A description of edge packaging jobs
Returns the description of an endpoint
Returns the description of an endpoint configuration created using the CreateEndpointConfig API
Provides a list of an experiment’s properties
Use this operation to describe a FeatureGroup
Shows the metadata for a feature within a feature group
Returns information about the specified flow definition
Describe a hub
Describe the content of a hub
Returns information about the requested human task user interface (worker task template)
Returns a description of a hyperparameter tuning job, depending on the fields selected
Describes a SageMaker image
Describes a version of a SageMaker image
Returns details about an inference experiment
Provides the results of the Inference Recommender job
Gets information about a labeling job
Provides a list of properties for the requested lineage group
Describes a model that you created using the CreateModel API
Returns a description of a model bias job definition
Describes the content, creation time, and security configuration of an Amazon SageMaker Model Card
Returns a description of a model explainability job definition
Returns a description of the specified model package, which is used to create SageMaker models
Gets a description for the specified model group
Returns a description of a model quality job definition
describe_monitoring_schedule
describe_notebook_instance
describe_notebook_instance_lifecycle_config
describe_pipeline
describe_pipeline_definition_for_execution
describe_pipeline_execution
describe_processing_job
describe_project
describe_space
describe_studio_lifecycle_config
describe_subscribed_workteam
describe_training_job
describe_transform_job
describe_trial
describe_trial_component
describe_user_profile
describe_workforce
describe_workteam
disable_sagemaker_servicecatalog_portfolio
disassociate_trial_component
enable_sagemaker_servicecatalog_portfolio
get_device_fleet_report
get_lineage_group_policy
get_model_package_group_policy
get_sagemaker_servicecatalog_portfolio_status
get_scaling_configuration_recommendation
get_search_suggestions
import_hub_content
list_actions
list_algorithms
list_aliases
list_app_image_configs
list_apps
list_artifacts
list_associations
list_auto_ml_jobs
list_candidates_for_auto_ml_job
list_code_repositories
list_compilation_jobs
list_contexts
list_data_quality_job_definitions
list_device_fleets
list_devices
list_domains
list_edge_deployment_plans
list_edge_packaging_jobs
list_endpoint_configs
list_endpoints

Describes the schedule for a monitoring job
Returns information about a notebook instance
Returns a description of a notebook instance lifecycle configuration
Describes the details of a pipeline
Describes the details of an execution’s pipeline definition
Describes the details of a pipeline execution
Returns a description of a processing job
Describes the details of a project
Describes the space
Describes the Studio Lifecycle Configuration
Gets information about a work team provided by a vendor
Returns information about a training job
Returns information about a transform job
Provides a list of a trial’s properties
Provides a list of a trials component’s properties
Describes a user profile
Lists private workforce information, including workforce name, Amazon Resource Name (ARN), and, if applicable, allowed IP address ranges (CIDRs)
Disables using Service Catalog in SageMaker
Disassociates a trial component from a trial
Enables using Service Catalog in SageMaker
Describes a fleet
The resource policy for the lineage group
Gets a resource policy that manages access for a model group
Gets the status of Service Catalog in SageMaker
Starts an Amazon SageMaker Inference Recommender autoscaling recommendation job
An auto-complete API for the search functionality in the SageMaker console
Import hub content
Lists the actions in your account and their properties
Lists the machine learning algorithms that have been created
Lists the aliases of a specified image or image version
Lists the AppImageConfigs in your account and their properties
Lists apps
Lists the artifacts in your account and their properties
Lists the associations in your account and their properties
Request a list of jobs
List the candidates created for the job
Gets a list of the Git repositories in your account
Lists model compilation jobs that satisfy various filters
Lists the contexts in your account and their properties
Lists the data quality job definitions in your account
Returns a list of devices in the fleet
A list of devices
Lists the domains
Lists all edge deployment plans
Returns a list of edge packaging jobs
Lists endpoint configurations
Lists endpoints
list_experiments
list_feature_groups
list_flow_definitions
list_hub_contents
list_hub_content_versions
list_hubs
list_human_task_uis
list_hyper_parameter_tuning_jobs
list_images
list_image_versions
list_inference_experiments
list_inference_recommendations_jobs
list_inference_recommendations_job_steps
list_labeling_jobs
list_labeling_jobs_for_workteam
list_lineage_groups
list_model_bias_job_definitions
list_model_card_export_jobs
list_model_cards
list_model_card_versions
list_model_explainability_job_definitions
list_model_metadata
list_model_package_groups
list_model_packages
list_model_quality_job_definitions
list_models
list_monitoring_alert_history
list_monitoring_alerts
list_monitoring_executions
list_monitoring_schedules
list_notebook_instance_lifecycle_configs
list_notebook_instances
list_pipeline_executions
list_pipeline_execution_steps
list_pipeline_parameters_for_execution
list_pipelines
list_processing_jobs
list_projects
list_resource_catalogs
list_spaces
list_stage_devices
list_studio_lifecycle_configs
list_subscribed_workteams
list_tags
list_training_jobs
list_training_jobs_for_hyper_parameter_tuning_job
list_transform_jobs
list_trial_components

Lists all the experiments in your account
List FeatureGroups based on given filter and order
Returns information about the flow definitions in your account
List the contents of a hub
List hub content versions
List all existing hubs
Returns information about the human task user interfaces in your account
Gets a list of HyperParameterTuningJobSummary objects that describe
Lists the images in your account and their properties
Lists the versions of a specified image and their properties
Returns the list of all inference experiments
Lists recommendation jobs that satisfy various filters
Returns a list of the subtasks for an Inference Recommender job
Gets a list of labeling jobs
Gets a list of labeling jobs assigned to a specified work team
A list of lineage groups shared with your Amazon Web Services account
Lists model bias jobs definitions that satisfy various filters
List the export jobs for the Amazon SageMaker Model Card
List existing model cards
List existing versions of an Amazon SageMaker Model Card
Lists model explainability job definitions that satisfy various filters
Lists the domain, framework, task, and model name of standard machine
Gets a list of the model groups in your Amazon Web Services account
Lists the model packages that have been created
Gets a list of model quality monitoring job definitions in your account
Lists models created with the CreateModel API
Gets a list of past alerts in a model monitoring schedule
Gets the alerts for a single monitoring schedule
Returns list of all monitoring job executions
Returns list of all monitoring schedules
Lists notebook instance lifestyle configurations created with the CreateNote
Lists the SageMaker notebook instances in the requester’s account
Gets a list of the pipeline executions
Gets a list of PipelineExecutionStep objects
Gets a list of parameters for a pipeline execution
Gets a list of pipelines
Lists processing jobs that satisfy various filters
Gets a list of the projects in an Amazon Web Services account
Lists Amazon SageMaker Catalogs based on given filters and orders
Lists spaces
Lists devices allocated to the stage, containing detailed device information
Lists the Studio Lifecycle Configurations in your Amazon Web Services
Gets a list of the work teams that you are subscribed to in the Amazon Web
Returns the tags for the specified SageMaker resource
Lists training jobs
Gets a list of TrainingJobSummary objects that describe the training job
Lists transform jobs
Lists the trial components in your account
list_trials
list_user_profiles
list_workforces
list_workteams
put_model_package_group_policy
query_lineage
register_devices
render_ui_template
retry_pipeline_execution
search
send_pipeline_execution_step_failure
send_pipeline_execution_step_success
start_edge_deployment_stage
start_inference_experiment
start_monitoring_schedule
start_notebook_instance
start_pipeline_execution
stop_auto_ml_job
stop_compilation_job
stop_edge_deployment_stage
stop_edge_packaging_job
stop_hyper_parameter_tuning_job
stop_inference_experiment
stop_inference_recommendations_job
stop_labeling_job
stop_monitoring_schedule
stop_notebook_instance
stop_pipeline_execution
stop_processing_job
stop_training_job
stop_transform_job
update_action
update_app_image_config
update_artifact
update_code_repository
update_context
update_device_fleet
update_devices
update_domain
update_endpoint
update_endpoint_weights_and_capacities
update_experiment
update_feature_group
update_feature_metadata
update_hub
update_image
update_image_version
update_inference_experiment

SageMaker API operations:
- **list_trials**: Lists the trials in your account.
- **list_user_profiles**: Lists user profiles.
- **list_workforces** and **list_workteams**: Use this operation to list all private and vendor workforces in an Amazon Web Services Region.
- **list_monitoring_schedules**: Gets a list of private work teams that you have defined in a region.
- **list_workforces**: Adds a resource policy to control access to a model group.
- **register_devices**: Use this action to inspect your lineage and discover relationships between entities.
- **render_ui_template**: Register devices.
- **retry_pipeline_execution**: Renders the UI template so that you can preview the worker’s experience.
- **send_pipeline_execution_step_failure**: Retry the execution of the pipeline.
- **send_pipeline_execution_step_success**: Finds SageMaker resources that match a search query.
- **start_edge_deployment_stage**: Notifies the pipeline that the execution of a callback step failed, along with a message describing why.
- **start_inference_experiment**: Notifies the pipeline that the execution of a callback step succeeded and provides a list of the step’s output parameters.
- **start_notebook_instance**: Starts a stage in an edge deployment plan.
- **start_pipeline_execution**: Starts an inference experiment.
- **stop_auto_ml_job**: Starts a previously stopped monitoring schedule.
- **stop_compilation_job**: Launches an ML compute instance with the latest version of the libraries and attaches your ML storage volume.
- **stop_edge_deployment_stage**: Starts a pipeline execution.
- **stop_edge_packaging_job**: A method for forcing a running job to shut down.
- **stop_hyper_parameter_tuning_job**: Stops a model compilation job.
- **stop_inference_experiment**: Stops a stage in an edge deployment plan.
- **stop_inference_recommendations_job**: Request to stop an edge packaging job.
- **stop_labeling_job**: Stops a running hyperparameter tuning job and all running training jobs.
- **stop_monitoring_schedule**: Starts a pipeline execution.
- **stop_notebook_instance**: Stops an inference experiment.
- **stop_pipeline_execution**: Stops an Inference Recommender job.
- **stop_processing_job**: Stops a running labeling job.
- **stop_training_job**: Terminates the ML compute instance.
- **stop_transform_job**: Stops a previously started monitoring schedule.
- **update_action**: Stops a pipeline execution.
- **update_app_image_config**: Stops a processing job.
- **update_artifact**: Stops a training job.
- **update_code_repository**: Stops a batch transform job.
- **update_context**: Updates the properties of an AppImageConfig.
- **update_device_fleet**: Updates an action.
- **update_devices**: Updates the properties of an running hyperparameter tuning job.
- **update_domain**: Adds a variant weight of one or more variants associated with an existing experiment.
- **update_endpoint**: Removes the description of an experiment.
- **update_endpoint_weights_and_capacities**: Updates the feature group by either adding features or updating the online store configuration.
- **update_experiment**: Updates the description and parameters of the feature group.
- **update_feature_group**: Update a hub.
- **update_feature_metadata**: Updates the properties of a SageMaker image.
- **update_hub**: Updates the properties of a SageMaker image version.
- **update_image**: Updates an inference experiment that you created.
update_model_card
update_model_package
update_monitoring_alert
update_monitoring_schedule
update_notebook_instance
update_notebook_instance_lifecycle_config
update_pipeline
update_pipeline_execution
update_project
update_space
update_training_job
update_trial
update_trial_component
update_user_profile
update_workforce
update_workteam

Update an Amazon SageMaker Model Card
Updates a versioned model
Update the parameters of a model monitor alert
Updates a previously created schedule
Updates a notebook instance
Updates a notebook instance lifecycle configuration created with the Create_notebook_instance_lifecycle_config API
Updates a pipeline
Updates a pipeline execution
Updates a machine learning (ML) project that is created from a template
Updates the settings of a space
Update a model training job to request a new Debugger profiling configuration
Updates the display name of a trial
Updates one or more properties of a trial component
Updates a user profile
Use this operation to update your workforce
Updates an existing work team with new member definitions or descriptions

Examples

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

## End(Not run)
```

---

**sagemakeredgemanager**  
**Amazon Sagemaker Edge Manager**

**Description**

SageMaker Edge Manager dataplane service for communicating with active agents.

**Usage**

```r
sagemakeredgemanager(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```
Arguments

config Optional configuration of credentials, endpoint, and/or region.
  • credentials:
    – creds:
      * access_key_id: AWS access key ID
      * secret_access_key: AWS secret access key
      * session_token: AWS temporary session token
    – profile: The name of a profile to use. If not given, then the default profile is used.
    – anonymous: Set anonymous credentials.
    – endpoint: The complete URL to use for the constructed client.
    – region: The AWS Region used in instantiating the client.
  • close_connection: Immediately close all HTTP connections.
  • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
  • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html

credentials Optional credentials shorthand for the config parameter
  • creds:
    – access_key_id: AWS access key ID
    – secret_access_key: AWS secret access key
    – session_token: AWS temporary session token
  • profile: The name of a profile to use. If not given, then the default profile is used.
  • anonymous: Set anonymous credentials.

endpoint Optional shorthand for complete URL to use for the constructed client.

region Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

svc <- sagemakeredgemanager(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
    )
  )
)
Operations

get_deployments Use to get the active deployments from a device
get_device_registration Use to check if a device is registered with SageMaker Edge Manager
send_heartbeat Use to get the current status of devices registered on SageMaker Edge Manager

Examples

```r
## Not run:
svc <- sagemakeredgemanager()
svc$get_deployments(
  Foo = 123
)
```

## End(Not run)
Amazon SageMaker Feature Store Runtime

Description

Contains all data plane API operations and data types for the Amazon SageMaker Feature Store. Use this API to put, delete, and retrieve (get) features from a feature store.

Use the following operations to configure your OnlineStore and OfflineStore features, and to create and manage feature groups:

- CreateFeatureGroup
- DeleteFeatureGroup
- DescribeFeatureGroup
- ListFeatureGroups

Usage

sagemakerfeaturestoreruntime(
config = list(),
credentials = list(),
endpoint = NULL,
region = NULL
)

Arguments

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

- credentials:
  - creds:
    - access_key_id: AWS access key ID
    - secret_access_key: AWS secret access key
    - session_token: AWS temporary session token
  - profile: The name of a profile to use. If not given, then the default profile is used.
  - anonymous: Set anonymous credentials.
  - endpoint: The complete URL to use for the constructed client.
  - region: The AWS Region used in instantiating the client.
- close_connection: Immediately close all HTTP connections.
- timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
sagemakerfeaturestorerruntime

- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy [https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html](https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html)

**credentials** Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token

- **profile**: The name of a profile to use. If not given, then the default profile is used.

- **anonymous**: Set anonymous credentials.

**endpoint** Optional shorthand for complete URL to use for the constructed client.

**region** Optional shorthand for AWS Region used in instantiating the client.

**Value**

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

**Service syntax**

```r
svc <- sagemakerfeaturestorerruntime(
    config = list(
        credentials = list(
            creds = list(
                access_key_id = "string",
                secret_access_key = "string",
                session_token = "string"
            ),
            profile = "string",
            anonymous = "logical"
        ),
        endpoint = "string",
        region = "string",
        close_connection = "logical",
        timeout = "numeric",
        s3_force_path_style = "logical",
        sts_regional_endpoint = "string"
    ),
    credentials = list(
        creds = list(
            access_key_id = "string",
            secret_access_key = "string",
            session_token = "string"
        ),
        profile = "string",
```
sagemakerruntime

```r
anonymous = "logical",
endpoint = "string",
region = "string"
```

**Operations**

- **batch_get_record** Retrieves a batch of Records from a FeatureGroup
- **delete_record** Deletes a Record from a FeatureGroup in the OnlineStore
- **get_record** Use for OnlineStore serving from a FeatureStore
- **put_record** The PutRecord API is used to ingest a list of Records into your feature group

**Examples**

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

---

**sagemakerruntime**  
*Amazon SageMaker Runtime*

**Description**

The Amazon SageMaker runtime API.

**Usage**

```r
sagemakerruntime(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

**Arguments**

- **config** Optional configuration of credentials, endpoint, and/or region.
  - **credentials**
    - **creds**
* **access_key_id**: AWS access key ID
* **secret_access_key**: AWS secret access key
* **session_token**: AWS temporary session token

- **profile**: The name of a profile to use. If not given, then the default profile is used.
- **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.

- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endo.html

**credentials**

Optional credentials shorthand for the config parameter

- **creds**:
  - **access_key_id**: AWS access key ID
  - **secret_access_key**: AWS secret access key
  - **session_token**: AWS temporary session token

- **profile**: The name of a profile to use. If not given, then the default profile is used.

- **anonymous**: Set anonymous credentials.

**endpoint**

Optional shorthand for complete URL to use for the constructed client.

**region**

Optional shorthand for AWS Region used in instantiating the client.

**Value**

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

**Service syntax**

```r
svc <- sagemakerruntime(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    profile = "string",
    anonymous = "logical"
  )
)```

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(), credentials = list(), endpoint = NULL, region = NULL)

Arguments

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

  • credentials:
    – creds:
      – access_key_id: AWS access key ID
      – secret_access_key: AWS secret access key
      – session_token: AWS temporary session token
    – profile: The name of a profile to use. If not given, then the default profile is used.
    – anonymous: Set anonymous credentials.
    – endpoint: The complete URL to use for the constructed client.
    – region: The AWS Region used in instantiating the client.
  • close_connection: Immediately close all HTTP connections.
  • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
  • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-ends.html

credentials Optional credentials shorthand for the config parameter

  • creds:
    – access_key_id: AWS access key ID
    – secret_access_key: AWS secret access key
    – session_token: AWS temporary session token
  • profile: The name of a profile to use. If not given, then the default profile is used.
  • anonymous: Set anonymous credentials.

endpoint Optional shorthand for complete URL to use for the constructed client.

region Optional shorthand for AWS Region used in instantiating the client.

Value

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

```r
text <- textract(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

- **analyze_document**: Analyzes an input document for relationships between detected items
- **analyze_expense**: AnalyzeExpense synchronously analyzes an input document for financially related relationships between text
- **analyze_id**: Analyzes identity documents for relevant information
- **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
- **get_expense_analysis**: Gets the results for an Amazon Textract asynchronous operation that analyzes invoices and receipts
- **get_lending_analysis**: Gets summarized results for the StartLendingAnalysis operation, which analyzes text in a lending document
- **get_lending_analysis_summary**: Gets summarized results for the StartLendingAnalysis operation, which analyzes text in a lending 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
- **start_expense_analysis**: Starts the asynchronous analysis of invoices or receipts for data like contact information, items purchased, vendor names
- **start_lending_analysis**: Starts the classification and analysis of an input document
transcribeservice

Examples

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

## End(Not run)
```

---

transcribeservice  Amazon Transcribe Service

---

Description

Amazon Transcribe offers three main types of batch transcription: **Standard**, **Medical**, and **Call Analytics**.

- **Standard transcriptions** are the most common option. Refer to for details.
- **Medical transcriptions** are tailored to medical professionals and incorporate medical terms. A common use case for this service is transcribing doctor-patient dialogue into after-visit notes. Refer to for details.
- **Call Analytics transcriptions** are designed for use with call center audio on two different channels; if you’re looking for insight into customer service calls, use this option. Refer to for details.

Usage

```r
transcribeservice(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

- **config**: Optional configuration of credentials, endpoint, and/or region.
  - **credentials**: 
    - **creds**:
      - **access_key_id**: AWS access key ID
      - **secret_access_key**: AWS secret access key
      - **session_token**: AWS temporary session token
– **profile**: The name of a profile to use. If not given, then the default profile is used.
– **anonymous**: Set anonymous credentials.
– **endpoint**: The complete URL to use for the constructed client.
– **region**: The AWS Region used in instantiating the client.

• **close_connection**: Immediately close all HTTP connections.
• **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
• **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
• **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy [https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html](https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html)

**credentials**

Optional credentials shorthand for the config parameter

• **creds**:
  – **access_key_id**: AWS access key ID
  – **secret_access_key**: AWS secret access key
  – **session_token**: AWS temporary session token

• **profile**: The name of a profile to use. If not given, then the default profile is used.
• **anonymous**: Set anonymous credentials.

**endpoint**

Optional shorthand for complete URL to use for the constructed client.

**region**

Optional shorthand for AWS Region used in instantiating the client.

**Value**

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

**Service syntax**

```r
svc <- transcribeservice(  
  config = list(  
    credentials = list(  
      creds = list(  
        access_key_id = "string",  
        secret_access_key = "string",  
        session_token = "string"  
      ),  
      profile = "string",  
      anonymous = "logical"  
    ),  
    endpoint = "string",  
    region = "string",  
    close_connection = "logical",  
    )
)  
```
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string" ),
credentials = list(
  creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
  ),
  profile = "string",
  anonymous = "logical"
),
endpoint = "string",
region = "string"
)

Operations

create_call_analytics_category Creates a new Call Analytics category
create_language_model Creates a new custom language model
create_medical_vocabulary Creates a new custom medical vocabulary
create_vocabulary Creates a new custom vocabulary
create_vocabulary_filter Creates a new custom vocabulary filter
delete_call_analytics_category Deletes a Call Analytics category
delete_call_analytics_job Deletes a Call Analytics job
delete_language_model Deletes a custom language model
delete_medical_transcription_job Deletes a medical transcription job
delete_medical_vocabulary Deletes a custom medical vocabulary
delete_transcription_job Deletes a transcription job
delete_vocabulary Deletes a custom vocabulary
delete_vocabulary_filter Deletes a custom vocabulary filter
describe_language_model Provides information about the specified custom language model
get_call_analytics_category Provides information about the specified Call Analytics category
get_call_analytics_job Provides information about the specified Call Analytics job
get_medical_transcription_job Provides information about the specified medical transcription job
get_medical_vocabulary Provides information about the specified custom medical vocabulary
get_transcription_job Provides information about the specified transcription job
get_vocabulary Provides information about the specified custom vocabulary
get_vocabulary_filter Provides information about the specified custom vocabulary filter
list_call_analytics_categories Provides a list of Call Analytics categories, including all rules that make up each category
list_call_analytics_jobs Provides a list of Call Analytics jobs that match the specified criteria
list_language_models Provides a list of custom language models that match the specified criteria
list_medical_transcription_jobs Provides a list of medical transcription jobs that match the specified criteria
list_medical_vocabularies Provides a list of custom medical vocabularies that match the specified criteria
list_tags_for_resource Lists all tags associated with the specified transcription job, vocabulary, model, or resource
list_transcription_jobs Provides a list of transcription jobs that match the specified criteria
list_vocabularies Provides a list of custom vocabularies that match the specified criteria
list_vocabulary_filters
start_call_analytics_job
start_medical_transcription_job
start_transcription_job
tag_resource
untag_resource
update_call_analytics_category
update_medical_vocabulary
update_vocabulary
update_vocabulary_filter

Provides a list of custom vocabulary filters that match the specified criteria.
Transcribes the audio from a customer service call and applies any additional Request Parameters you choose to include in your request.
Transcribes the audio from a medical dictation or conversation and applies any additional Request Parameters you choose to include in your request.
Transcribes the audio from a media file and applies any additional Request Parameters you choose to include in your request.
Adds one or more custom tags, each in the form of a key:value pair, to the specified resource.
Removes the specified tags from the specified Amazon Transcribe resource.
Updates the specified Call Analytics category with new rules.
Updates an existing custom medical vocabulary with new values.
Updates an existing custom vocabulary with new values.
Updates an existing custom vocabulary filter with a new list of words.

Examples

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

## End(Not run)

---

**translate**

Amazon Translate

**Description**

Provides translation of the input content from the source language to the target language.

**Usage**

```r
translate(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

**Arguments**

- `config`: Optional configuration of credentials, endpoint, and/or region.
  - `credentials`
    - `creds`:
      - `access_key_id`: AWS access key ID
      - `secret_access_key`: AWS secret access key
* session_token: AWS temporary session token
  
  - profile: The name of a profile to use. If not given, then the default profile is used.
  
  - anonymous: Set anonymous credentials.
  
  - endpoint: The complete URL to use for the constructed client.
  
  - region: The AWS Region used in instantiating the client.

• close_connection: Immediately close all HTTP connections.

• timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.

• s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.

• sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html

credentials Optional credentials shorthand for the config parameter

  • creds:
    
    – access_key_id: AWS access key ID
    
    – secret_access_key: AWS secret access key
    
    – session_token: AWS temporary session token

  • profile: The name of a profile to use. If not given, then the default profile is used.

  • anonymous: Set anonymous credentials.

endpoint Optional shorthand for complete URL to use for the constructed client.

region Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

svc <- translate(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string"
  )
)
close_connection = "logical",
timeout = "numeric",
s3_force_path_style = "logical",
sts_regional_endpoint = "string"
),
credentials = list(
creds = list(
    access_key_id = "string",
    secret_access_key = "string",
    session_token = "string"
),
    profile = "string",
anonymous = "logical"
),
endpoint = "string",
region = "string"
)

Operations

create_parallel_data Creates a parallel data resource in Amazon Translate by importing an input file from Amazon

delete_parallel_data Deletes a parallel data resource in Amazon Translate

delete_terminology A synchronous action that deletes a custom terminology

describe_text_translation_job Gets the properties associated with an asynchronous batch translation job including name, ID, and status

get_parallel_data Provides information about a parallel data resource

get_terminology Retrieves a custom terminology

import_terminology Creates or updates a custom terminology, depending on whether one already exists for the provided terminology name

list_languages Provides a list of languages (RFC-5646 codes and names) that Amazon Translate supports

list_parallel_data Provides a list of your parallel data resources in Amazon Translate

list_tags_for_resource Lists all tags associated with a given Amazon Translate resource

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

tag_resource Associates a specific tag with a resource

translate_document Translates the input document from the source language to the target language

translate_text Translates input text from the source language to the target language

untag_resource Removes a specific tag associated with an Amazon Translate resource

update_parallel_data Updates a previously created parallel data resource by importing a new input file from Amazon

Examples

## Not run:
svc <- translate()
svc$create_parallel_data(
    Foo = 123
Amazon Voice ID

Description

Amazon Connect Voice ID provides real-time caller authentication and fraud risk detection, which make voice interactions in contact centers more secure and efficient.

Usage

voiceid(config = list(), credentials = list(), endpoint = NULL, region = NULL)

Arguments

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

- credentials:
  - creds:
    - access_key_id: AWS access key ID
    - secret_access_key: AWS secret access key
    - session_token: AWS temporary session token
  - profile: The name of a profile to use. If not given, then the default profile is used.
  - anonymous: Set anonymous credentials.
  - endpoint: The complete URL to use for the constructed client.
  - region: The AWS Region used in instantiating the client.
- close_connection: Immediately close all HTTP connections.
- timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. http://s3.amazonaws.com/BUCKET/KEY.
- sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoints.html

credentials  Optional credentials shorthand for the config parameter

- creds:
  - access_key_id: AWS access key ID
  - secret_access_key: AWS secret access key
  - session_token: AWS temporary session token
profile: The name of a profile to use. If not given, then the default profile is used.

anonymous: Set anonymous credentials.

endpoint: Optional shorthand for complete URL to use for the constructed client.

region: Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

```
svc <- voiceid(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

associate_fraudster Associates the fraudsters with the watchlist specified in the same domain
create_domain
create_watchlist
delete_domain
delete_fraudster
delete_speaker
delete_watchlist
describe_domain
describe_fraudster
describe_fraudster_registration_job
describe_speaker
describe_speaker_enrollment_job
describe_watchlist
disassociate_fraudster
evaluate_session
list_domains
list_fraudster_registration_jobs
list_fraudsters
list_speaker_enrollment_jobs
list_speakers
list_tags_for_resource
list_watchlists
opt_out_speaker
start_fraudster_registration_job
start_speaker_enrollment_job
tag_resource
untag_resource
update_domain
update_watchlist

Examples

## Not run:
svc <- voiceid()
svc$associate_fraudster(
  Foo = 123
)

## End(Not run)
Index

activate_anomaly_detector, 42
add_association, 69
add_tags, 45, 69
analyze_document, 84
analyzeExpense, 84
analyze_id, 84
associate_faces, 60, 63
associate_fraudster, 92
associate_trial_component, 69
augmented_airuntime, 3
back_test_anomaly_detector, 42
batch_create_custom_vocabulary_item, 30
batch_create_variable, 23
batch_delete_custom_vocabulary_item, 30
batch_describe_model_package, 69
batch_detect_dominant_language, 8
batch_detect_entities, 8
batch_detect_key_phrases, 8
batch_detect_sentiment, 8
batch_detect_syntax, 8
batch_detect_targeted_sentiment, 8
batch_get_record, 80
batch_get_variable, 23
batch_update_custom_vocabulary_item, 30
build_bot_locale, 30
cancel_batch_import_job, 23
cancel_batch_prediction_job, 23
classify_document, 8
compare_faces, 60, 64
comprehend, 6
comprehendmedical, 10
contains pii_entities, 8
copy_project_version, 60, 64
create_action, 69
create_alert, 42
create_algorithm, 69
create_anomaly_detector, 42
create_app, 69
create_app_image_config, 69
create_application_instance, 48
create_artifact, 69
create_auto_ml_job, 69
create_auto_ml_job_v2, 69
create_auto_predictor, 19
create_batch_import_job, 23
create_batch_inference_job, 51
create_batch_prediction, 45
create_batch_prediction_job, 23
create_batch_segment_job, 51
create_bot, 30
create_bot_alias, 30
create_bot_locale, 30
create_bot_version, 27, 30
create_call_analytics_category, 87
create_campaign, 51
create_code_repository, 69
create_collection, 60, 64
create_compilation_job, 69
create_context, 69
create_data_quality_job_definition, 69
create_data_source_from_rds, 45
create_data_source_from_redshift, 45
create_data_source_from_s3, 45
create_dataset, 8, 19, 39, 51, 60, 64
create_dataset_export_job, 51
create_dataset_group, 19, 51
create_dataset_import_job, 19, 51
create_detector_version, 23
create_device_fleet, 69
create_document_classifier, 8
create_domain, 69, 93
create_edge_deployment_plan, 69
create_edge_deployment_stage, 69
create_edge_packaging_job, 69
create_endpoint, 8, 69
create_endpoint_config, 69
create_entity_recognizer, 8
create_evaluation, 45
create_event_tracker, 51
create_experiment, 69
create_explainability, 19
create_explainability_export, 19
create_export, 30
create_face_liveness_session, 64
create_feature_group, 69
create_filter, 51
create_flow_definition, 69
create_flywheel, 8
create_forecast, 19
create_forecast_export_job, 19
create_hub, 69
create_human_task_ui, 69
create_hyper_parameter_tuning_job, 69
create_image, 69
create_image_version, 69
create_inference_experiment, 69
create_inference_recommendations_job, 69
create_inference_scheduler, 39
create_intent, 30
create_intent_version, 27
create_job_for_devices, 48
create_label, 39
create_label_group, 39
create_labeling_job, 69
create_language_model, 87
create_list, 23
create_medical_vocabulary, 87
create_metric_attribution, 51
create_metric_set, 42
create_ml_model, 45
create_model, 23, 39, 69
create_model_bias_job_definition, 69
create_model_card, 69
create_model_card_export_job, 69
create_model_explainability_job_definition, 69
create_model_package, 69
create_model_package_group, 69
create_model_quality_job_definition, 69
create_model_version, 23
create_monitor, 20
create_monitoring_schedule, 69
create_node_from_template_job, 48
create_notebook_instance, 69
create_notebook_instance_lifecycle_config, 69
create_package, 48
create_package_import_job, 48
create_parallel_data, 90
create_pipeline, 69
create_predictor, 20
create_predictor_backtest_export_job, 20
create_presigned_domain_url, 69
create_presigned_notebook_instance_url, 69
create_processing_job, 69
create_project, 61, 64, 70
create_project_version, 61, 64
create_realtime_endpoint, 45
create_recommender, 51
create_resource_policy, 30
create_resource_policy_statement, 30
create_rule, 23
create_schema, 51
create_slot, 30
create_slot_type, 30
create_slot_type_version, 27
create_solution, 51
create_solution_version, 51
create_space, 70
create_stream_processor, 62, 64
create_studio_lifecycle_config, 70
create_test_set_discrepancy_report, 30
create_training_job, 70
create_transform_job, 70
create_trial, 70
create_trial_component, 70
create_upload_url, 30
create_user, 60, 64
create_user_profile, 70
create_variable, 23
create_vocabulary, 87
create_vocabulary_filter, 87
create_watchlist, 93
create_what_if_analysis, 20
create_what_if_forecast, 20
create_what_if_forecast_export, 20
create_workforce, 70
create_workteam, 70
deactivate_anomaly_detector, 42
delete_action, 70
delete_alert, 42
delete_algorithm, 70
delete_anomaly_detector, 42
delete_app, 70
delete_app_image_config, 70
delete_artifact, 70
delete_association, 70
delete_batch_import_job, 23
delete_batch_prediction, 45
delete_batch_prediction_job, 23
delete_bot, 27, 30
delete_bot_alias, 27, 30
delete_bot_channel_association, 27
delete_bot_locale, 30
delete_bot_version, 27, 30
delete_call_analytics_category, 87
delete_call_analytics_job, 87
delete_campaign, 51
delete_code_repository, 70
delete_collection, 60, 64
defterminate, 70
deflete_context, 70
deflect_custom_vocabulary, 31
deflete_data_quality_job_definition, 70
deflete_data_source, 45
deflete_dataset, 20, 39, 51, 61, 64
deflete_dataset_group, 20, 51
deflete_dataset_import_job, 20
deflete_detector, 23
deflete_detector_version, 23
deflete_device, 48
deflete_device_fleet, 70
deflete_document_classifier, 8
deflete_domain, 70, 93
deflete_edge_deployment_plan, 70
deflete_edge_deployment_stage, 70
deflete_endpoint, 8, 70
deflete_endpoint_config, 70
deflect_entity_recognizer, 8
deflect_entity_type, 23
defdelete_evaluation, 45
deflete_event, 23
deflete_event_tracker, 51
deflete_event_type, 23
deflect_events_by_event_type, 23
deflete_experiment, 70
deflect_explainability, 20
deflect_explainability_export, 20
deflect_export, 31
deflect_external_model, 24
deflect_faces, 60, 64
deflect_feature_group, 70
deflete_filter, 51
deflete_flow_definition, 70
deflect_flywheel, 8
deflete_forecast, 20
deflete_forecast_export_job, 20
deflete_fraudster, 93
deflete_hub, 70
deflete_hub_content, 70
deflect_human_loop, 5
deflect_human_task_ui, 70
deflect_image, 70
deflect_image_version, 70
deflect_import, 31
deflect_inference_experiment, 70
deflect_inference_scheduler, 39
deflete_intent, 27, 31
deflect_intent_version, 27
deflete_label, 24, 39
deflect_label_group, 39
deflect_language_model, 87
deflect_lexicon, 59
deflete_list, 24
deflect_medical_transcription_job, 87
deflect_medical_vocabulary, 87
deflect_metric_attribution, 51
deflect_ml_model, 45
deflete_model, 24, 39, 70
deflect_model_bias_job_definition, 70
deflect_model_card, 70
deflect_model_explainability_job_definition, 70
deflect_model_package, 70
deflect_model_package_group, 70
deflect_model_package_group_policy, 70
deflect_model_quality_job_definition, 70
deflete_model_version, 24
deflete_monitor, 20
deflect_monitoring_schedule, 70
deflect_notebook_instance, 70
deflect_notebook_instance_lifecycle_config,
INDEX

70
delete_outcome, 24
delete_package, 48
delete_parallel_data, 90
delete_pipeline, 70
delete_predictor, 20
delete_predictor_backtest_export_job, 20
delete_project, 61, 64, 70
delete_project_policy, 61, 64
delete_project_version, 61, 64
delete_realtime_endpoint, 45
delete_recommender, 51
delete_record, 80
delete_resource_policy, 8, 31, 39
delete_resource_policy_statement, 31
delete_resource_tree, 20
delete_rule, 24
delete_schema, 51
delete_session, 34, 37
delete_slot, 31
delete_slot_type, 27, 31
delete_slot_type_version, 27
delete_solution, 51
delete_space, 70
delete_speaker, 93
delete_stream_processor, 62, 64
delete_studio_lifecycle_config, 71
delete_tags, 45, 71
delete_terminology, 90
delete_test_set, 31
delete_transcription_job, 87
delete_trial, 71
delete_trial_component, 71
delete_user, 60, 64
delete_user_profile, 71
delete_utterances, 27, 31
delete_variable, 24
delete_vocabulary, 87
delete_vocabulary_filter, 87
delete_watchlist, 93
delete_what_if_analysis, 20
delete_what_if_forecast, 20
delete_what_if_forecast_export, 20
delete_workforce, 71
delete_workteam, 71
deregister_devices, 71
deregister_package_version, 48
describe_accelerator_offerings, 15
describe_accelerator_types, 15
describe_accelerators, 15
describe_action, 71
describe_alert, 42
describe_algorithm, 51, 71
describe_anomaly_detection_executions, 42
describe_anomaly_detector, 42
describe_app, 71
describe_app_image_config, 71
describe_application_instance, 48
describe_application_instance_details, 48
describe_artifact, 71
describe_auto_ml_job, 71
describe_auto_ml_job_v2, 71
describe_auto_predictor, 20
describe_batch_inference_job, 51
describe_batch_predictions, 45
describe_batch_segment_job, 51
describe_bot, 31
describe_bot_alias, 31
describe_bot_locale, 31
describe_bot_recommendation, 31
describe_bot_version, 31
describe_campaign, 51
describe_code_repository, 71
describe_collection, 60, 64
describe_compilation_job, 71
describe_context, 71
describe_custom_vocabulary_metadata, 31
describe_data_ingestion_job, 39
describe_data_quality_job_definition, 71
describe_data_sources, 45
describe_dataset, 8, 20, 39, 51, 61, 64
describe_dataset_export_job, 51
describe_dataset_group, 20, 51
describe_dataset_import_job, 20, 51
describe_detector, 24
describe_device, 48, 71
describe_device_fleet, 71
describe_device_job, 48
describe_document_classification_job, 8
describe_document_classifier, 8
describe_domain, 71, 93
describe_dominant_language_detection_job, 8
describe_edge_deployment_plan, 71
describe_endpoint, 8, 71
describe_endpoint_config, 71
describe_entities_detection_job, 8
describe_entities_detection_v2_job, 12
describe_entity_recognizer, 8
describe_evaluations, 45
describe_event_tracker, 51
describe_events_detection_job, 8
describe_experiment, 71
describe_explainability, 20
describe_explainability_export, 20
describe_export, 31
describe_feature_group, 71
describe_feature_metadata, 71
describe_feature_transformation, 51
describe_filter, 51
describe_flow_definition, 71
describe_flywheel, 8
describe_flywheel_iteration, 8
describe_forecast, 20
describe_forecast_export_job, 20
describe_fraudster, 93
describe_fraudster_registration_job, 93
describe_hub, 71
describe_hub_content, 71
describe_human_loop, 5
describe_human_task_ui, 71
describe_hyper_parameter_tuning_job, 71
describe_icd10cm_inference_job, 12
describe_image, 71
describe_image_version, 71
describe_import, 31
describe_inference_experiment, 71
describe_inference_recommendations_job, 71
describe_inference_scheduler, 39
describe_intent, 31
describe_key_phrases_detection_job, 8
describe_label, 39
describe_label_group, 39
describe_labeling_job, 71
describe_language_model, 87
describe_lineage_group, 71
describe_metric_attribution, 52
describe_metric_set, 42
describe_ml_models, 45
describe_model, 39, 71
describe_model_bias_job_definition, 71
describe_model_card, 71
describe_model_card_export_job, 71
describe_model_explainability_job_definition, 71
describe_model_package, 71
describe_model_package_group, 71
describe_model_quality_job_definition, 71
describe_model_version, 39
describe_model_versions, 24
describe_monitor, 20
describe_monitoring_schedule, 72
describe_node, 48
describe_node_from_template_job, 48
describe_notebook_instance, 72
describe_notebook_instance_lifecycle_config, 72
describe_package, 48
describe_package_import_job, 48
describe_package_version, 48
describe_phi_detection_job, 12
describe_pii_entities_detection_job, 8
describe_pipeline, 72
describe_pipeline_definition_for_execution, 72
describe_pipeline_execution, 72
describe_predictor, 20
describe_predictor_backtest_export_job, 20
describe_processing_job, 72
describe_project, 72
describe_project_versions, 61, 64
describe_projects, 61, 64
describe_recipe, 52
describe_recommender, 52
describe_resource_policy, 8, 31, 39
describe_rx_norm_inference_job, 12
describe_schema, 52
describe_sentiment_detection_job, 8
describe_slot, 31
describe_slot_type, 31
describe_snomedct_inference_job, 12
describe_solution, 52
describe_solution_version, 52
describe_space, 72
describe_speaker, 93
describe_speaker_enrollment_job, 93
describe_stream_processor, 62, 64
describe_studio_lifecycle_config, 72
describe_subscribed_workteam, 72
describe_tags, 45
describe_speaker_enrollment_job, 93
describe_stream_processor, 62, 64
describe_studio_lifecycle_config, 72
describe_subscribed_workteam, 72
describe_tags, 45
describe_targeted_sentiment_detection_job, 8
describe_test_execution, 31
describe_test_set, 31
describe_test_set_discrepancy_report, 31
describe_test_set_generation, 31
describe_text_translation_job, 90
describe_topics_detection_job, 8
describe_training_job, 72
describe_transform_job, 72
describe_trial, 72
describe_trial_component, 72
describe_user_profile, 72
describe_voices, 59
describe_watchlist, 93
describe_what_if_analysis, 20
describe_what_if_forecast, 20
describe_what_if_forecast_export, 20
describe_workforce, 72
describe_workteam, 72
detect_custom_labels, 61, 64
detect_document_text, 84
detect_dominant_language, 8
detect_entities, 8, 12
detect_entities_v2, 12
detect_faces, 60, 64
detect_key_phrases, 9
detect_labels, 60, 64
detect_metric_set_config, 42
detect_moderation_labels, 60, 64
detect_phi, 12
detect_pii_entities, 9
detect_protective_equipment, 60, 64
detect_sentiment, 9
detect_syntax, 9
detect_targeted_sentiment, 9
detect_text, 60, 64
disable_sagemaker_servicecatalog_portfolio, 72
disassociate_faces, 60, 64
disassociate_fraudster, 93
disassociate_trial_component, 72
distribute_dataset_entries, 61, 64
elasticinference, 13
enable_sagemaker_servicecatalog_portfolio, 72
evaluate_session, 93
forecastqueryservice, 15
forecastservice, 17
frauddetector, 21
get_accuracy_metrics, 20
get_anomaly_group, 42
get_batch_import_jobs, 24
get_batch_prediction, 45
get_batch_prediction_jobs, 24
get_bot, 27
get_bot_alias, 27
get_bot_aliases, 27
get_bot_channel_association, 27
get_bot_channel_associations, 27
get_bot_versions, 27
get_bots, 27
get_builtin_intent, 27
get_builtin_intents, 27
get_builtin_slot_types, 27
get_call_analytics_category, 87
get_call_analytics_job, 87
get_celebrity_info, 60, 64
get_celebrity_recognition, 61, 64
get_content_moderation, 61, 64
get_data_quality_metrics, 43
get_data_source, 46
get_delete_events_by_event_type_status, 24
get_deployments, 77
get_detector_version, 24
get_detectors, 24
get_device_fleet_report, 72
get_device_registration, 77
get_document_analysis, 84
get_document_text_detection, 84
get_entity_types, 24
get_evaluation, 46
get_event, 24
get_event_prediction, 24
get_event_prediction_metadata, 24
get_event_types, 24
get_expense_analysis, 84
get_export, 27
get_external_models, 24
get_face_detection, 61, 64
get_face_liveness_session_results, 64
get_face_search, 61, 64
get_feedback, 43
get_import, 27
get_intent, 27
get_intent_versions, 27
get_intents, 27
get_kms_encryption_key, 24
get_label_detection, 61, 64
get_labels, 24
get_lending_analysis, 84
get_lending_analysis_summary, 84
get_lexicon, 59
get_lineage_group_policy, 72
get_list_elements, 24
get_lists_metadata, 24
get_medical_transcription_job, 87
get_medical_vocabulary, 87
get_migration, 27
get_migrations, 27
get_ml_model, 46
get_model_package_group_policy, 72
get_model_version, 24
get_models, 24
get_outcomes, 24
get_parallel_data, 90
get_person_tracking, 61, 64
get_personized_ranking, 57
get_recommendations, 57
get_record, 80
get_rules, 24
get_sagemaker_servicecatalog_portfolio_status, 72
get_sample_data, 43
get_scaling_configuration_recommendation, 72
get_search_suggestions, 72
get_segment_detection, 61, 64
get_session, 34, 37
get_slot_type, 28
get_slot_type_versions, 28
get_slot_types, 28
get_solution_metrics, 52
get_speech_synthesis_task, 59
get_terminology, 90
get_test_execution_artifacts_url, 31
get_text_detection, 61, 64
get_transcription_job, 87
get_utterances_view, 28
get_variables, 24
get_vocabulary, 87
get_vocabulary_filter, 87
import_dataset, 39
import_hub_content, 72
import_model, 9
import_model_version, 40
import_terminology, 90
index_faces, 60, 64
infer_icd10cm, 12
infer_rx_norm, 12
infer_snomedct, 12
invoke_endpoint, 82
invoke_endpoint_async, 82
invoke_endpoint_with_response_stream, 82
lexmodelbuildingservice, 25
lexmodelsv2, 28
lexruntimeservice, 32
lexruntimev2, 35
list_actions, 72
list_aggregated_utterances, 31
list_alerts, 43
list_algorithms, 72
list_aliases, 72
list_anomaly_detectors, 43
list_anomaly_group_related_metrics, 43
list_anomaly_group_summaries, 43
list_anomaly_group_time_series, 43
list_app_image_configs, 72
list_application_instance_dependencies, 48
list_application_instance_node_instances, 48
list_application_instances, 48
list_apps, 72
list_artifacts, 72
list_associations, 72
INDEX

list_auto_ml_jobs, 72
list_batch_inference_jobs, 52
list_batch_segment_jobs, 52
list_bot_alias, 31
list_bot_locales, 31
list_bot_recommendations, 31
list_bot_versions, 31
list_bots, 31
list_built_in_intents, 31
list_built_in_slot_types, 31
list_call_analytics_categories, 87
list_call_analytics_jobs, 87
list_campaigns, 52
list_candidates_for_auto_ml_job, 72
list_code_repositories, 52
list_collections, 60, 64
list_compilation_jobs, 72
list_contexts, 72
list_custom_vocabulary_items, 31
list_data_ingestion_jobs, 40
list_data_quality_job_definitions, 72
list_dataset_entries, 61, 64
list_dataset_export_jobs, 52
list_dataset_groups, 20, 52
list_dataset_import_jobs, 20, 52
list_dataset_labels, 61, 64
list_datasets, 9, 20, 40, 52
list_device_fleets, 72
list_devices, 48, 72
list_devices_jobs, 48
list_document_classification_jobs, 9
list_document_classifier_summaries, 9
list_document_classifiers, 9
list_domains, 72, 93
list_dominant_language_detection_jobs, 9
list_edge_deployment_plans, 72
list_edge_packaging_jobs, 72
list_endpoint_configs, 72
list_endpoints, 9, 72
list_entities_detection_jobs, 9
list_entities_detection_v2_jobs, 12
list_entity_recognizer_summaries, 9
list_entity_recognizers, 9
list_event_predictions, 24
list_event_trackers, 52
list_events_detection_jobs, 9
list_experiments, 73
list_explainabilities, 20
list_explainability_exports, 20
list_exports, 31
list_faces, 60, 64
list_feature_groups, 73
list_filters, 52
list_flow_definitions, 73
list_flywheel_iteration_history, 9
list_flywheels, 9
list_forecast_export_jobs, 20
list_forecasts, 20
list_fraudster_registration_jobs, 93
list_fraudsters, 93
list_hub_content_versions, 73
list_hub_contents, 73
list_hubs, 73
list_human_loops, 5
list_human_task_ui, 73
list_hypper_parameter_tuning_jobs, 73
list_icd10cm_inference_jobs, 12
list_image_versions, 73
list_images, 73
list_imports, 31
list_inference_events, 40
list_inference_executions, 40
list_inference_experiments, 73
list_inference_recommendations_job_steps, 73
list_inference_recommendations_jobs, 73
list_inference_schedulers, 40
list_intent_metrics, 31
list_intent_paths, 31
list_intent_stage_metrics, 31
list_intents, 31
list_key_phrases_detection_jobs, 9
list_label_groups, 40
list_labeling_jobs, 73
list_labeling_jobs_for_workteam, 73
list_labels, 40
list_language_models, 87
list_languages, 90
list_lexicons, 59
list_lineage_groups, 73
list_medical_transcription_jobs, 87
list_medical_vocabularies, 87
list_metric_attribution_metrics, 52
list_metric_attributions, 52
<table>
<thead>
<tr>
<th>list_metric_sets</th>
<th>43</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_model_bias_job_definitions</td>
<td>73</td>
</tr>
<tr>
<td>list_model_card_export_jobs</td>
<td>73</td>
</tr>
<tr>
<td>list_model_card_versions</td>
<td>73</td>
</tr>
<tr>
<td>list_model_cards</td>
<td>73</td>
</tr>
<tr>
<td>list_model_explainability_job_definitions</td>
<td>73</td>
</tr>
<tr>
<td>list_model_metadata</td>
<td>73</td>
</tr>
<tr>
<td>list_model_package_groups</td>
<td>73</td>
</tr>
<tr>
<td>list_model_packages</td>
<td>73</td>
</tr>
<tr>
<td>list_model_quality_job_definitions</td>
<td>73</td>
</tr>
<tr>
<td>list_model_versions</td>
<td>40</td>
</tr>
<tr>
<td>list_models</td>
<td>40,73</td>
</tr>
<tr>
<td>list_monitor_evaluations</td>
<td>20</td>
</tr>
<tr>
<td>list_monitoring_alert_history</td>
<td>73</td>
</tr>
<tr>
<td>list_monitoring_alerts</td>
<td>73</td>
</tr>
<tr>
<td>list_monitoring_executions</td>
<td>73</td>
</tr>
<tr>
<td>list_monitoring_schedules</td>
<td>73</td>
</tr>
<tr>
<td>list_monitors</td>
<td>20</td>
</tr>
<tr>
<td>list_node_from_template_jobs</td>
<td>48</td>
</tr>
<tr>
<td>list_nodes</td>
<td>48</td>
</tr>
<tr>
<td>list_notebook_instance_lifecycle_configs</td>
<td>73</td>
</tr>
<tr>
<td>list_notebook_instances</td>
<td>73</td>
</tr>
<tr>
<td>list_package_import_jobs</td>
<td>48</td>
</tr>
<tr>
<td>list_packages</td>
<td>48</td>
</tr>
<tr>
<td>list_parallel_data</td>
<td>90</td>
</tr>
<tr>
<td>list_phi_detection_jobs</td>
<td>12</td>
</tr>
<tr>
<td>list_pii_entities_detection_jobs</td>
<td>9</td>
</tr>
<tr>
<td>list_pipeline_execution_steps</td>
<td>73</td>
</tr>
<tr>
<td>list_pipeline_executions</td>
<td>73</td>
</tr>
<tr>
<td>list_pipeline_parameters_for_execution</td>
<td>73</td>
</tr>
<tr>
<td>list_pipelines</td>
<td>73</td>
</tr>
<tr>
<td>list_predictor_backtest_export_jobs</td>
<td>20</td>
</tr>
<tr>
<td>list_predictors</td>
<td>20</td>
</tr>
<tr>
<td>list_processing_jobs</td>
<td>73</td>
</tr>
<tr>
<td>list_project_policies</td>
<td>61,64</td>
</tr>
<tr>
<td>list_projects</td>
<td>73</td>
</tr>
<tr>
<td>list_recipes</td>
<td>52</td>
</tr>
<tr>
<td>list_recommended_intents</td>
<td>31</td>
</tr>
<tr>
<td>list_recommenders</td>
<td>52</td>
</tr>
<tr>
<td>list_resource_catalogs</td>
<td>73</td>
</tr>
<tr>
<td>list_rx_norm_inference_jobs</td>
<td>12</td>
</tr>
<tr>
<td>list_schemas</td>
<td>52</td>
</tr>
<tr>
<td>list_sensor_statistics</td>
<td>40</td>
</tr>
<tr>
<td>list_sentiment_detection_jobs</td>
<td>9</td>
</tr>
<tr>
<td>list_session_analytics_data</td>
<td>31</td>
</tr>
<tr>
<td>list_session_metrics</td>
<td>31</td>
</tr>
<tr>
<td>list_slot_metrics</td>
<td>31</td>
</tr>
<tr>
<td>list_slots</td>
<td>31</td>
</tr>
<tr>
<td>list_snomedct_inference_jobs</td>
<td>12</td>
</tr>
<tr>
<td>list_solution_versions</td>
<td>52</td>
</tr>
<tr>
<td>list_solutions</td>
<td>52</td>
</tr>
<tr>
<td>list_spaces</td>
<td>73</td>
</tr>
<tr>
<td>list_speaker_enrollment_jobs</td>
<td>93</td>
</tr>
<tr>
<td>list_speakers</td>
<td>93</td>
</tr>
<tr>
<td>list_speech_synthesis_tasks</td>
<td>59</td>
</tr>
<tr>
<td>list_stage_devices</td>
<td>73</td>
</tr>
<tr>
<td>list_stream_processors</td>
<td>62,64</td>
</tr>
<tr>
<td>list_studio_lifecycle_configs</td>
<td>73</td>
</tr>
<tr>
<td>list_subscribed_workteams</td>
<td>73</td>
</tr>
<tr>
<td>list_tags</td>
<td>73</td>
</tr>
<tr>
<td>list_tags_for_resource</td>
<td>9,15,20,24,28,31,40,43,48,52,64,87,90,93</td>
</tr>
<tr>
<td>list_targeted_sentiment_detection_jobs</td>
<td>9</td>
</tr>
<tr>
<td>list_terminologies</td>
<td>90</td>
</tr>
<tr>
<td>list_test_execution_result_items</td>
<td>32</td>
</tr>
<tr>
<td>list_test_executions</td>
<td>32</td>
</tr>
<tr>
<td>list_test_set_records</td>
<td>32</td>
</tr>
<tr>
<td>list_test_sets</td>
<td>32</td>
</tr>
<tr>
<td>list_text_translation_jobs</td>
<td>90</td>
</tr>
<tr>
<td>list_topics_detection_jobs</td>
<td>9</td>
</tr>
<tr>
<td>list_training_jobs</td>
<td>73</td>
</tr>
<tr>
<td>list_training_jobs_for_hyper_parameter_tuning_job</td>
<td>73</td>
</tr>
<tr>
<td>list_transcription_jobs</td>
<td>87</td>
</tr>
<tr>
<td>list_transform_jobs</td>
<td>73</td>
</tr>
<tr>
<td>list_trial_components</td>
<td>73</td>
</tr>
<tr>
<td>list_trials</td>
<td>74</td>
</tr>
<tr>
<td>list_user_profiles</td>
<td>74</td>
</tr>
<tr>
<td>list_users</td>
<td>60,65</td>
</tr>
<tr>
<td>list_utterance_analytics_data</td>
<td>32</td>
</tr>
<tr>
<td>list_utterance_metrics</td>
<td>32</td>
</tr>
<tr>
<td>list_vocularies</td>
<td>87</td>
</tr>
<tr>
<td>list_vocabulary_filters</td>
<td>88</td>
</tr>
<tr>
<td>list_watchlists</td>
<td>93</td>
</tr>
<tr>
<td>list_what_if_analyses</td>
<td>20</td>
</tr>
<tr>
<td>list_what_if_forecast_exports</td>
<td>21</td>
</tr>
<tr>
<td>list_what_if_forecasts</td>
<td>21</td>
</tr>
<tr>
<td>list_workforces</td>
<td>74</td>
</tr>
<tr>
<td>list_workteams</td>
<td>74</td>
</tr>
<tr>
<td>lookoutequipment</td>
<td>37</td>
</tr>
<tr>
<td>lookoutmetrics</td>
<td>40</td>
</tr>
</tbody>
</table>
machinelearning, 43
opt_out_speaker, 93
panorama, 46
personalize, 49
personalizeevents, 53
personalizeruntime, 55
polly, 57
post_content, 34
post_text, 34
predict, 46
provision_device, 48
put_bot, 28
put_bot_alias, 28
put_detector, 24
put_entity_type, 24
put_event_type, 24
put_events, 54
put_external_model, 24
put_feedback, 43
put_intent, 28
put_items, 55
put_kms_encryption_key, 24
put_label, 24
put_lexicon, 59
put_model_package_group_policy, 74
put_outcome, 24
put_project_policy, 61, 65
put_record, 80
put_resource_policy, 9, 40
put_session, 34, 37
put_slot_type, 28
put_users, 55
query_forecast, 17
query_lineage, 74
query_what_if_forecast, 17
recognize_celebrities, 60, 65
recognize_text, 37
recognize_utterance, 37
register_devices, 74
register_package_version, 48
rekognition, 60
remove_application_instance, 48
render_ui_template, 74
resume_resource, 21
retry_pipeline_execution, 74
sagemaker, 66
sagemakeredgemanager, 75
sagemakerfeaturestoreruntime, 78
sagemakerruntime, 80
search, 74
search_associated_transcripts, 32
search_faces, 60, 65
search_faces_by_image, 60, 65
search_users, 60, 65
send_event, 24
send_heartbeat, 77
send_pipeline_execution_step_failure, 74
send_pipeline_execution_step_success, 74
signal_application_instance_node_instances, 48
start_bot_recommendation, 32
start_call_analytics_job, 88
start_celebrity_recognition, 61, 65
start_content_moderation, 61, 65
start_data_ingestion_job, 40
start_document_analysis, 84
start_document_classification_job, 9
start_document_text_detection, 84
start_edge_deployment_stage, 74
start_entities_detection_job, 9
start_entities_detection_v2_job, 12
start_events_detection_job, 9
start_fraudsterRegistration_job, 93
start_fraudster_registration_job, 93
start_humantask_job, 3, 5
start_icd10cm_inference_job, 12
start_import, 28, 32
start_inference_experiment, 74
start_inference_scheduler, 40
start_key_phrases_detection_job, 9
start_label_detection, 61, 65
start_lending_analysis, 84
start_medical_transcription_job, 88
start_migration, 28
start_monitoring_schedule, 74
start_notebook_instance, 74
start_person_tracking, 61, 65
start_phi_detection_job, 12
start_pii_entities_detection_job, 9
start_pipeline_execution, 74
start_project_version, 61, 65
start_recommender, 52
start_rx_norm_inference_job, 12
start_segment_detection, 61, 65
start_sentiment_detection_job, 9
start_snomedct_inference_job, 12
start_speaker_enrollment_job, 93
start_speech_synthesis_task, 59
start_stream_processor, 62, 65
start_targeted_sentiment_detection_job, 9
start_test_execution, 32
start_test_set_generation, 32
start_text_detection, 61, 65
start_text_translation_job, 90
start_topics_detection_job, 9
start_transcription_job, 88
stop_auto_ml_job, 74
stop_bot_recommendation, 32
stop_compilation_job, 74
stop_edge_deployment_stage, 74
stop_edge_packaging_job, 74
stop_entities_detection_job, 9
stop_entities_detection_v2_job, 12
stop_events_detection_job, 9
stop_human_loop, 5
stop_hyperparameter_tuning_job, 74
stop_icd10cm_inference_job, 12
stop_inference_experiment, 74
stop_inference_recommendations_job, 74
stop_key_phrases_detection_job, 40
stop_labeling_job, 74
stop_monitoring_schedule, 74
stop_notebook_instance, 74
stop_phi_detection_job, 12
stop_pii_entities_detection_job, 9
stop_pipeline_execution, 74
stop_processing_job, 74
stop_project_version, 61, 65
stop_recommender, 52
stop_resource, 21
stop_rx_norm_inference_job, 12
stop_sentiment_detection_job, 9
stop_snomedct_inference_job, 12
stop_solution_version_creation, 52
stop_stream_processor, 62, 65
stop_targeted_sentiment_detection_job, 9
stop_text_translation_job, 90
stop_training_document_classifier, 9
stop_training_entity_recognizer, 9
stop_training_job, 74
stop_transform_job, 74
synthesize_speech, 59
tag_resource, 9, 15, 21, 24, 28, 32, 40, 43, 48, 52, 65, 88, 90, 93
textract, 82
transcribeservice, 85
translate, 88
translate_document, 90
translate_text, 90
untag_resource, 9, 15, 21, 24, 28, 32, 40, 43, 49, 52, 65, 88, 90, 93
update_action, 74
update_active_model_version, 40
update_alert, 43
update_anomaly_detector, 43
update_app_image_config, 74
update_artifact, 74
update_batch_prediction, 46
update_bot, 32
update_bot_alias, 32
update_bot_locale, 32
update_bot_recommendation, 32
update_call_analytics_category, 88
update_campaign, 52
update_code_repository, 74
update_context, 74
update_data_source, 46
update_dataset, 52
update_dataset_entries, 61, 65
update_dataset_group, 21
update_detector_version, 24
update_detector_version_metadata, 24
update_detector_version_status, 24
update_device_fleet, 74
update_device_metadata, 49
INDEX

update_devices, 74
update_domain, 74, 93
update_endpoint, 9, 74
update_endpoint_weights_and_capacities, 74
update_evaluation, 46
update_event_label, 24
update_experiment, 74
update_export, 32
update_feature_group, 74
update_feature_metadata, 74
update_flywheel, 9
update_hub, 74
update_image, 74
update_image_version, 74
update_inference_experiment, 74
update_inference_scheduler, 40
update_intent, 32
update_label_group, 40
update_list, 24
update_medical_vocabulary, 88
update_metric_attribution, 52
update_metric_set, 43
update_ml_model, 46
update_model, 24
update_model_card, 75
update_model_package, 75
update_model_version, 25
update_model_version_status, 25
update_monitoring_alert, 75
update_monitoring_schedule, 75
update_notebook_instance, 75
update_notebook_instance_lifecycle_config, 75
update_parallel_data, 90
update_pipeline, 75
update_pipeline_execution, 75
update_project, 75
update_recommender, 52
update_resource_policy, 32
update_rule_metadata, 25
update_rule_version, 25
update_slot, 32
update_slot_type, 32
update_space, 75
update_stream_processor, 62, 65
update_test_set, 32
update_training_job, 75
update_trial, 75
update_trial_component, 75
update_user_profile, 75
update_variable, 25
update_vocabulary, 88
update_vocabulary_filter, 88
update_watchlist, 93
update_workforce, 75
update_workteam, 75
voiceid, 91