

Package ‘paws.machine.learning’

September 11, 2023

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 'augmentedairruntime_service.R'
'augmentedairruntime_interfaces.R'
'augmentedairruntime_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'

'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:

augmentedairruntime	3
comprehend	6
comprehendmedical	10
elasticinference	13

forecastqueryservice	15
forecastservice	17
frauddetector	21
lexmodelbuildingservice	25
lexmodelsv2	28
lexruntimev2	32
lexruntimev2	35
lookoutequipment	37
lookoutmetrics	40
machinelearning	43
panorama	46
personalize	49
personalizeevents	53
personalizeruntime	55
polly	57
rekognition	60
sagemaker	66
sagemakeredgemanager	75
sagemakerfeaturestoreruntime	78
sagemakerruntime	80
extract	82
transcribeservice	85
translate	88
voiceid	91
Index	94

augmentedairuntime *Amazon Augmented AI Runtime*

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

```
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-endpoint.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 <- 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

Examples

```
## 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

```
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.

	<ul style="list-style-type: none"> • 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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"
),
endpoint = "string",
region = "string"
)

```

Operations

batch_detect_dominant_language	Determines the dominant language of the input text for a batch of documents
batch_detect_entities	Inspects the text of a batch of documents for named entities and returns information
batch_detect_key_phrases	Detects the key noun phrases found in a batch of documents
batch_detect_sentiment	Inspects a batch of documents and returns an inference of the prevailing sentiment
batch_detect_syntax	Inspects the text of a batch of documents for the syntax and part of speech of the
batch_detect_targeted_sentiment	Inspects a batch of documents and returns a sentiment analysis for each entity i
classify_document	Creates a new document classification request to analyze a single document in m
contains_pii_entities	Analyzes input text for the presence of personally identifiable information (PII)
create_dataset	Creates a dataset to upload training or test data for a model associated with a fly
create_document_classifier	Creates a new document classifier that you can use to categorize documents
create_endpoint	Creates a model-specific endpoint for synchronous inference for a previously tr
create_entity_recognizer	Creates an entity recognizer using submitted files
create_flywheel	A flywheel is an Amazon Web Services resource that orchestrates the ongoing tr
delete_document_classifier	Deletes a previously created document classifier
delete_endpoint	Deletes a model-specific endpoint for a previously-trained custom model
delete_entity_recognizer	Deletes an entity recognizer
delete_flywheel	Deletes a flywheel
delete_resource_policy	Deletes a resource-based policy that is attached to a custom model
describe_dataset	Returns information about the dataset that you specify
describe_document_classification_job	Gets the properties associated with a document classification job
describe_document_classifier	Gets the properties associated with a document classifier
describe_dominant_language_detection_job	Gets the properties associated with a dominant language detection job
describe_endpoint	Gets the properties associated with a specific endpoint
describe_entities_detection_job	Gets the properties associated with an entities detection job
describe_entity_recognizer	Provides details about an entity recognizer including status, S3 buckets contain
describe_events_detection_job	Gets the status and details of an events detection job
describe_flywheel	Provides configuration information about the flywheel
describe_flywheel_iteration	Retrieve the configuration properties of a flywheel iteration
describe_key_phrases_detection_job	Gets the properties associated with a key phrases detection job
describe_pii_entities_detection_job	Gets the properties associated with a PII entities detection job
describe_resource_policy	Gets the details of a resource-based policy that is attached to a custom model, i
describe_sentiment_detection_job	Gets the properties associated with a sentiment detection job
describe_targeted_sentiment_detection_job	Gets the properties associated with a targeted sentiment detection job
describe_topics_detection_job	Gets the properties associated with a topic detection job
detect_dominant_language	Determines the dominant language of the input text
detect_entities	Detects named entities in input text when you use the pre-trained model

<code>detect_key_phrases</code>	Detects the key noun phrases found in the text
<code>detect_pii_entities</code>	Inspects the input text for entities that contain personally identifiable information
<code>detect_sentiment</code>	Inspects text and returns an inference of the prevailing sentiment (POSITIVE, NEUTRAL, NEGATIVE)
<code>detect_syntax</code>	Inspects text for syntax and the part of speech of words in the document
<code>detect_targeted_sentiment</code>	Inspects the input text and returns a sentiment analysis for each entity identified
<code>import_model</code>	Creates a new custom model that replicates a source custom model that you imported
<code>list_datasets</code>	List the datasets that you have configured in this Region
<code>list_document_classification_jobs</code>	Gets a list of the document classification jobs that you have submitted
<code>list_document_classifiers</code>	Gets a list of the document classifiers that you have created
<code>list_document_classifier_summaries</code>	Gets a list of summaries of the document classifiers that you have created
<code>list_dominant_language_detection_jobs</code>	Gets a list of the dominant language detection jobs that you have submitted
<code>list_endpoints</code>	Gets a list of all existing endpoints that you've created
<code>list_entities_detection_jobs</code>	Gets a list of the entity detection jobs that you have submitted
<code>list_entity_recognizers</code>	Gets a list of the properties of all entity recognizers that you created, including their supported languages
<code>list_entity_recognizer_summaries</code>	Gets a list of summaries for the entity recognizers that you have created
<code>list_events_detection_jobs</code>	Gets a list of the events detection jobs that you have submitted
<code>list_flywheel_iteration_history</code>	Information about the history of a flywheel iteration
<code>list_flywheels</code>	Gets a list of the flywheels that you have created
<code>list_key_phrases_detection_jobs</code>	Get a list of key phrase detection jobs that you have submitted
<code>list_pii_entities_detection_jobs</code>	Gets a list of the PII entity detection jobs that you have submitted
<code>list_sentiment_detection_jobs</code>	Gets a list of sentiment detection jobs that you have submitted
<code>list_tags_for_resource</code>	Lists all tags associated with a given Amazon Comprehend resource
<code>list_targeted_sentiment_detection_jobs</code>	Gets a list of targeted sentiment detection jobs that you have submitted
<code>list_topics_detection_jobs</code>	Gets a list of the topic detection jobs that you have submitted
<code>put_resource_policy</code>	Attaches a resource-based policy to a custom model
<code>start_document_classification_job</code>	Starts an asynchronous document classification job
<code>start_dominant_language_detection_job</code>	Starts an asynchronous dominant language detection job for a collection of documents
<code>start_entities_detection_job</code>	Starts an asynchronous entity detection job for a collection of documents
<code>start_events_detection_job</code>	Starts an asynchronous event detection job for a collection of documents
<code>start_flywheel_iteration</code>	Start the flywheel iteration
<code>start_key_phrases_detection_job</code>	Starts an asynchronous key phrase detection job for a collection of documents
<code>start_pii_entities_detection_job</code>	Starts an asynchronous PII entity detection job for a collection of documents
<code>start_sentiment_detection_job</code>	Starts an asynchronous sentiment detection job for a collection of documents
<code>start_targeted_sentiment_detection_job</code>	Starts an asynchronous targeted sentiment detection job for a collection of documents
<code>start_topics_detection_job</code>	Starts an asynchronous topic detection job
<code>stop_dominant_language_detection_job</code>	Stops a dominant language detection job in progress
<code>stop_entities_detection_job</code>	Stops an entities detection job in progress
<code>stop_events_detection_job</code>	Stops an events detection job in progress
<code>stop_key_phrases_detection_job</code>	Stops a key phrases detection job in progress
<code>stop_pii_entities_detection_job</code>	Stops a PII entities detection job in progress
<code>stop_sentiment_detection_job</code>	Stops a sentiment detection job in progress
<code>stop_targeted_sentiment_detection_job</code>	Stops a targeted sentiment detection job in progress
<code>stop_training_document_classifier</code>	Stops a document classifier training job while in progress
<code>stop_training_entity_recognizer</code>	Stops an entity recognizer training job while in progress
<code>tag_resource</code>	Associates a specific tag with an Amazon Comprehend resource
<code>untag_resource</code>	Removes a specific tag associated with an Amazon Comprehend resource
<code>update_endpoint</code>	Updates information about the specified endpoint
<code>update_flywheel</code>	Update the configuration information for an existing flywheel

Examples

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

## End(Not run)
```

comprehendmedical *AWS Comprehend Medical*

Description

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

	<ul style="list-style-type: none"> • 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	<p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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	Gets the properties associated with a medical entities detection job
describe_icd10cm_inference_job	Gets the properties associated with an InferICD10CM job
describe_phi_detection_job	Gets the properties associated with a protected health information (PHI) detection job
describe_rx_norm_inference_job	Gets the properties associated with an InferRxNorm job
describe_snomedct_inference_job	Gets the properties associated with an InferSNOMEDCT job
detect_entities	The DetectEntities operation is deprecated
detect_entities_v2	Inspects the clinical text for a variety of medical entities and returns specific information
detect_phi	Inspects the clinical text for protected health information (PHI) entities and returns the entities
infer_icd10cm	InferICD10CM detects medical conditions as entities listed in a patient record and links them to codes
infer_rx_norm	InferRxNorm detects medications as entities listed in a patient record and links them to the RxNorm
infer_snomedct	InferSNOMEDCT detects possible medical concepts as entities and links them to codes
list_entities_detection_v2_jobs	Gets a list of medical entity detection jobs that you have submitted
list_icd10cm_inference_jobs	Gets a list of InferICD10CM jobs that you have submitted
list_phi_detection_jobs	Gets a list of protected health information (PHI) detection jobs you have submitted
list_rx_norm_inference_jobs	Gets a list of InferRxNorm jobs that you have submitted
list_snomedct_inference_jobs	Gets a list of InferSNOMEDCT jobs a user has submitted
start_entities_detection_v2_job	Starts an asynchronous medical entity detection job for a collection of documents
start_icd10cm_inference_job	Starts an asynchronous job to detect medical conditions and link them to the ICD-10-CM
start_phi_detection_job	Starts an asynchronous job to detect protected health information (PHI)
start_rx_norm_inference_job	Starts an asynchronous job to detect medication entities and link them to the RxNorm
start_snomedct_inference_job	Starts an asynchronous job to detect medical concepts and link them to the SNOMED-C
stop_entities_detection_v2_job	Stops a medical entities detection job in progress
stop_icd10cm_inference_job	Stops an InferICD10CM inference job in progress
stop_phi_detection_job	Stops a protected health information (PHI) detection job in progress
stop_rx_norm_inference_job	Stops an InferRxNorm inference job in progress
stop_snomedct_inference_job	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

```

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.

	<ul style="list-style-type: none"> • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	<p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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"
    )
  )
)
```

```

    ),
    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 their supported instance types
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. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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",
    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

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

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- forecasts-service(
  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](#)
[create_dataset](#)
[create_dataset_group](#)
[create_dataset_import_job](#)
[create_explainability](#)
[create_explainability_export](#)
[create_forecast](#)
[create_forecast_export_job](#)

Creates an Amazon Forecast predictor
 Creates an Amazon Forecast dataset
 Creates a dataset group, which holds a collection of related datasets
 Imports your training data to an Amazon Forecast dataset
 Explainability is only available for Forecasts and Predictors generated from an Auto
 Exports an Explainability resource created by the CreateExplainability operation
 Creates a forecast for each item in the TARGET_TIME_SERIES dataset that was u
 Exports a forecast created by the CreateForecast operation to your Amazon Simple

<code>create_monitor</code>	Creates a predictor monitor resource for an existing auto predictor
<code>create_predictor</code>	This operation creates a legacy predictor that does not include all the predictor functions
<code>create_predictor_backtest_export_job</code>	Exports backtest forecasts and accuracy metrics generated by the <code>CreateAutoPredictor</code> operation
<code>create_what_if_analysis</code>	What-if analysis is a scenario modeling technique where you make a hypothetical change to a forecast
<code>create_what_if_forecast</code>	A what-if forecast is a forecast that is created from a modified version of the baseline forecast
<code>create_what_if_forecast_export</code>	Exports a forecast created by the <code>CreateWhatIfForecast</code> operation to your Amazon Forecast console
<code>delete_dataset</code>	Deletes an Amazon Forecast dataset that was created using the <code>CreateDataset</code> operation
<code>delete_dataset_group</code>	Deletes a dataset group created using the <code>CreateDatasetGroup</code> operation
<code>delete_dataset_import_job</code>	Deletes a dataset import job created using the <code>CreateDatasetImportJob</code> operation
<code>delete_explainability</code>	Deletes an Explainability resource
<code>delete_explainability_export</code>	Deletes an Explainability export
<code>delete_forecast</code>	Deletes a forecast created using the <code>CreateForecast</code> operation
<code>delete_forecast_export_job</code>	Deletes a forecast export job created using the <code>CreateForecastExportJob</code> operation
<code>delete_monitor</code>	Deletes a monitor resource
<code>delete_predictor</code>	Deletes a predictor created using the <code>DescribePredictor</code> or <code>CreatePredictor</code> operation
<code>delete_predictor_backtest_export_job</code>	Deletes a predictor backtest export job
<code>delete_resource_tree</code>	Deletes an entire resource tree
<code>delete_what_if_analysis</code>	Deletes a what-if analysis created using the <code>CreateWhatIfAnalysis</code> operation
<code>delete_what_if_forecast</code>	Deletes a what-if forecast created using the <code>CreateWhatIfForecast</code> operation
<code>delete_what_if_forecast_export</code>	Deletes a what-if forecast export created using the <code>CreateWhatIfForecastExport</code> operation
<code>describe_auto_predictor</code>	Describes a predictor created using the <code>CreateAutoPredictor</code> operation
<code>describe_dataset</code>	Describes an Amazon Forecast dataset created using the <code>CreateDataset</code> operation
<code>describe_dataset_group</code>	Describes a dataset group created using the <code>CreateDatasetGroup</code> operation
<code>describe_dataset_import_job</code>	Describes a dataset import job created using the <code>CreateDatasetImportJob</code> operation
<code>describe_explainability</code>	Describes an Explainability resource created using the <code>CreateExplainability</code> operation
<code>describe_explainability_export</code>	Describes an Explainability export created using the <code>CreateExplainabilityExport</code> operation
<code>describe_forecast</code>	Describes a forecast created using the <code>CreateForecast</code> operation
<code>describe_forecast_export_job</code>	Describes a forecast export job created using the <code>CreateForecastExportJob</code> operation
<code>describe_monitor</code>	Describes a monitor resource
<code>describe_predictor</code>	This operation is only valid for legacy predictors created with <code>CreatePredictor</code>
<code>describe_predictor_backtest_export_job</code>	Describes a predictor backtest export job created using the <code>CreatePredictorBacktestExportJob</code> operation
<code>describe_what_if_analysis</code>	Describes the what-if analysis created using the <code>CreateWhatIfAnalysis</code> operation
<code>describe_what_if_forecast</code>	Describes the what-if forecast created using the <code>CreateWhatIfForecast</code> operation
<code>describe_what_if_forecast_export</code>	Describes the what-if forecast export created using the <code>CreateWhatIfForecastExport</code> operation
<code>get_accuracy_metrics</code>	Provides metrics on the accuracy of the models that were trained by the <code>CreatePredictor</code> operation
<code>list_dataset_groups</code>	Returns a list of dataset groups created using the <code>CreateDatasetGroup</code> operation
<code>list_dataset_import_jobs</code>	Returns a list of dataset import jobs created using the <code>CreateDatasetImportJob</code> operation
<code>list_datasets</code>	Returns a list of datasets created using the <code>CreateDataset</code> operation
<code>list_explainabilities</code>	Returns a list of Explainability resources created using the <code>CreateExplainability</code> operation
<code>list_explainability_exports</code>	Returns a list of Explainability exports created using the <code>CreateExplainabilityExport</code> operation
<code>list_forecast_export_jobs</code>	Returns a list of forecast export jobs created using the <code>CreateForecastExportJob</code> operation
<code>list_forecasts</code>	Returns a list of forecasts created using the <code>CreateForecast</code> operation
<code>list_monitor_evaluations</code>	Returns a list of the monitoring evaluation results and predictor events collected by the <code>CreateMonitor</code> operation
<code>list_monitors</code>	Returns a list of monitors created with the <code>CreateMonitor</code> operation and <code>CreateAutoPredictor</code>
<code>list_predictor_backtest_export_jobs</code>	Returns a list of predictor backtest export jobs created using the <code>CreatePredictorBacktestExportJob</code> operation
<code>list_predictors</code>	Returns a list of predictors created using the <code>CreateAutoPredictor</code> or <code>CreatePredictor</code> operation
<code>list_tags_for_resource</code>	Lists the tags for an Amazon Forecast resource
<code>list_what_if_analyses</code>	Returns a list of what-if analyses created using the <code>CreateWhatIfAnalysis</code> operation

list_what_if_forecast_exports	Returns a list of what-if forecast exports created using the CreateWhatIfForecastEx
list_what_if_forecasts	Returns a list of what-if forecasts created using the CreateWhatIfForecast operation
resume_resource	Resumes a stopped monitor resource
stop_resource	Stops a resource
tag_resource	Associates the specified tags to a resource with the specified resourceArn
untag_resource	Deletes the specified tags from a resource
update_dataset_group	Replaces the datasets in a dataset group with the specified datasets

Examples

```
## Not run:
svc <- forecastservice()
svc$create_auto_predictor(
  Foo = 123
)

## End(Not run)
```

frauddetector

Amazon Fraud Detector

Description

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.

Usage

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

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 = "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

<code>delete_external_model</code>	Removes a SageMaker model from Amazon Fraud Detector
<code>delete_label</code>	Deletes a label
<code>delete_list</code>	Deletes the list, provided it is not used in a rule
<code>delete_model</code>	Deletes a model
<code>delete_model_version</code>	Deletes a model version
<code>delete_outcome</code>	Deletes an outcome
<code>delete_rule</code>	Deletes the rule
<code>delete_variable</code>	Deletes a variable
<code>describe_detector</code>	Gets all versions for a specified detector
<code>describe_model_versions</code>	Gets all of the model versions for the specified model type or for the specified model
<code>get_batch_import_jobs</code>	Gets all batch import jobs or a specific job of the specified ID
<code>get_batch_prediction_jobs</code>	Gets all batch prediction jobs or a specific job if you specify a job ID
<code>get_delete_events_by_event_type_status</code>	Retrieves the status of a DeleteEventsByEventType action
<code>get_detectors</code>	Gets all detectors or a single detector if a detectorId is specified
<code>get_detector_version</code>	Gets a particular detector version
<code>get_entity_types</code>	Gets all entity types or a specific entity type if a name is specified
<code>get_event</code>	Retrieves details of events stored with Amazon Fraud Detector
<code>get_event_prediction</code>	Evaluates an event against a detector version
<code>get_event_prediction_metadata</code>	Gets details of the past fraud predictions for the specified event ID, event type, detector version, and rule
<code>get_event_types</code>	Gets all event types or a specific event type if name is provided
<code>get_external_models</code>	Gets the details for one or more Amazon SageMaker models that have been imported
<code>get_kms_encryption_key</code>	Gets the encryption key if a KMS key has been specified to be used to encrypt content
<code>get_labels</code>	Gets all labels or a specific label if name is provided
<code>get_list_elements</code>	Gets all the elements in the specified list
<code>get_lists_metadata</code>	Gets the metadata of either all the lists under the account or the specified list
<code>get_models</code>	Gets one or more models
<code>get_model_version</code>	Gets the details of the specified model version
<code>get_outcomes</code>	Gets one or more outcomes
<code>get_rules</code>	Get all rules for a detector (paginated) if ruleId and ruleVersion are not specified
<code>get_variables</code>	Gets all of the variables or the specific variable
<code>list_event_predictions</code>	Gets a list of past predictions
<code>list_tags_for_resource</code>	Lists all tags associated with the resource
<code>put_detector</code>	Creates or updates a detector
<code>put_entity_type</code>	Creates or updates an entity type
<code>put_event_type</code>	Creates or updates an event type
<code>put_external_model</code>	Creates or updates an Amazon SageMaker model endpoint
<code>put_kms_encryption_key</code>	Specifies the KMS key to be used to encrypt content in Amazon Fraud Detector
<code>put_label</code>	Creates or updates label
<code>put_outcome</code>	Creates or updates an outcome
<code>send_event</code>	Stores events in Amazon Fraud Detector without generating fraud predictions for them
<code>tag_resource</code>	Assigns tags to a resource
<code>untag_resource</code>	Removes tags from a resource
<code>update_detector_version</code>	Updates a detector version
<code>update_detector_version_metadata</code>	Updates the detector version's description
<code>update_detector_version_status</code>	Updates the detector version's status
<code>update_event_label</code>	Updates the specified event with a new label
<code>update_list</code>	Updates a list
<code>update_model</code>	Updates model description

<code>update_model_version</code>	Updates a model version
<code>update_model_version_status</code>	Updates the status of a model version
<code>update_rule_metadata</code>	Updates a rule's metadata
<code>update_rule_version</code>	Updates a rule version resulting in a new rule version
<code>update_variable</code>	Updates a variable

Examples

```
## Not run:
svc <- frauddetector()
svc$batch_create_variable(
  Foo = 123
)

## End(Not run)
```

lexmodelbuildingservice

Amazon Lex Model Building Service

Description

Amazon Lex Build-Time Actions

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

Usage

```
lexmodelbuildingservice(
  config = list(),
  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

	<ul style="list-style-type: none"> – 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. <ul style="list-style-type: none"> • 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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 an Amazon Lex resource in a specified format
get_import	Gets information about an import job started with the StartImport operation
get_intent	Returns information about an intent
get_intents	Returns intent information as follows:
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

```
## 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)
```

lexmodelsv2

Amazon Lex Model Building V2

Description

Amazon Lex Model Building V2

Usage

```
lexmodelsv2(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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",
    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
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	Removes a custom vocabulary from the specified locale in the specified bot
delete_export	Removes a previous export and the associated files stored in an S3 bucket
delete_import	Removes a previous import and the associated file stored in an S3 bucket
delete_intent	Removes the specified intent
delete_resource_policy	Removes an existing policy from a bot or bot alias
delete_resource_policy_statement	Deletes a policy statement from a resource policy
delete_slot	Deletes the specified slot from an intent
delete_slot_type	Deletes a slot type from a bot locale
delete_test_set	The action to delete the selected test set
delete utterances	Deletes stored utterances
describe_bot	Provides metadata information about a bot
describe_bot_alias	Get information about a specific bot alias
describe_bot_locale	Describes the settings that a bot has for a specific locale
describe_bot_recommendation	Provides metadata information about a bot recommendation
describe_bot_version	Provides metadata about a version of a bot
describe_custom_vocabulary_metadata	Provides metadata information about a custom vocabulary
describe_export	Gets information about a specific export
describe_import	Gets information about a specific import
describe_intent	Returns metadata about an intent
describe_resource_policy	Gets the resource policy and policy revision for a bot or bot alias
describe_slot	Gets metadata information about a slot
describe_slot_type	Gets metadata information about a slot type
describe_test_execution	Gets metadata information about the test execution
describe_test_set	Gets metadata information about the test set
describe_test_set_discrepancy_report	Gets metadata information about the test set discrepancy report
describe_test_set_generation	Gets metadata information about the test set generation
get_test_execution_artifacts_url	The pre-signed Amazon S3 URL to download the test execution result artifacts
list_aggregated_utterances	Provides a list of utterances that users have sent to the bot
list_bot_aliases	Gets a list of aliases for the specified bot
list_bot_locales	Gets a list of locales for the specified bot
list_bot_recommendations	Get a list of bot recommendations that meet the specified criteria
list_bots	Gets a list of available bots
list_bot_versions	Gets information about all of the versions of a bot
list_built_in_intents	Gets a list of built-in intents provided by Amazon Lex that you can use in your bot
list_built_in_slot_types	Gets a list of built-in slot types that meet the specified criteria
list_custom_vocabulary_items	Paginated list of custom vocabulary items for a given bot locale's custom vocabulary
list_exports	Lists the exports for a bot, bot locale, or custom vocabulary
list_imports	Lists the imports for a bot, bot locale, or custom vocabulary
list_intent_metrics	Retrieves summary metrics for the intents in your bot
list_intent_paths	Retrieves summary statistics for a path of intents that users take over sessions with your bot
list_intents	Get a list of intents that meet the specified criteria
list_intent_stage_metrics	Retrieves summary metrics for the stages within intents in your bot
list_recommended_intents	Gets a list of recommended intents provided by the bot recommendation that you can use in your bot
list_session_analytics_data	Retrieves a list of metadata for individual user sessions with your bot
list_session_metrics	Retrieves summary metrics for the user sessions with your bot
list_slots	Gets a list of slots that match the specified criteria
list_slot_types	Gets a list of slot types that match the specified criteria
list_tags_for_resource	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 Lis
list_utterance_metrics	To use this API operation, your IAM role must have permissions to perform the Lis
search_associated_transcripts	Search for associated transcripts that meet the specified criteria
start_bot_recommendation	Use this to provide your transcript data, and to start the bot recommendation proces
start_import	Starts importing a bot, bot locale, or custom vocabulary from a zip archive that you
start_test_execution	The action to start test set execution
start_test_set_generation	The action to start the generation of test set
stop_bot_recommendation	Stop an already running Bot Recommendation request
tag_resource	Adds the specified tags to the specified resource
untag_resource	Removes tags from a bot, bot alias, or bot channel
update_bot	Updates the configuration of an existing bot
update_bot_alias	Updates the configuration of an existing bot alias
update_bot_locale	Updates the settings that a bot has for a specific locale
update_bot_recommendation	Updates an existing bot recommendation request
update_export	Updates the password used to protect an export zip archive
update_intent	Updates the settings for an intent
update_resource_policy	Replaces the existing resource policy for a bot or bot alias with a new one
update_slot	Updates the settings for a slot
update_slot_type	Updates the configuration of an existing slot type
update_test_set	The action to update the test set

Examples

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

## End(Not run)
```

lexruntime-service

Amazon Lex Runtime Service

Description

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

(one of the intents defined in the bot). Then Amazon Lex engages in user conversation on behalf of the bot to elicit required information (slot values, such as pizza size and crust type), and then performs fulfillment activity (that you configured when you created the bot). You use the build-time API to create and manage your Amazon Lex bot. For a list of build-time operations, see the build-time API, .

Usage

```
lexruntime-service(
  config = list(),
  credentials = list(),
  endpoint = NULL,
  region = NULL
)
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- lexruntime-service(
  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

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

Examples

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

## End(Not run)
```

lexruntimev2

Amazon Lex Runtime V2

Description

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

Usage

```
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`.

	<ul style="list-style-type: none"> • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	<p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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",
```

```

        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
put_session	Creates a new session or modifies an existing session with an Amazon Lex V2 bot
recognize_text	Sends user input to Amazon Lex V2
recognize_utterance	Sends user input to Amazon Lex V2

Examples

```

## Not run:
svc <- lexruntimev2()
svc$delete_session(
  Foo = 123
)

## End(Not run)

```

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. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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	Creates a container for a collection of data being ingested for analysis
create_inference_scheduler	Creates a scheduled inference
create_label	Creates a label for an event
create_label_group	Creates a group of labels
create_model	Creates an ML model for data inference
delete_dataset	Deletes a dataset and associated artifacts
delete_inference_scheduler	Deletes an inference scheduler that has been set up
delete_label	Deletes a label
delete_label_group	Deletes a group of labels
delete_model	Deletes an ML model currently available for Amazon Lookout for Equipment
delete_resource_policy	Deletes the resource policy attached to the resource
describe_data_ingestion_job	Provides information on a specific data ingestion job such as creation time, dataset ARN, and
describe_dataset	Provides a JSON description of the data in each time series dataset, including names, column
describe_inference_scheduler	Specifies information about the inference scheduler being used, including name, model, status
describe_label	Returns the name of the label
describe_label_group	Returns information about the label group
describe_model	Provides a JSON containing the overall information about a specific ML model, including mo
describe_model_version	Retrieves information about a specific machine learning model version
describe_resource_policy	Provides the details of a resource policy attached to a resource
import_dataset	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 sta
list_model_versions	Generates a list of all model versions for a given model, including the model version, model v
list_sensor_statistics	Lists statistics about the data collected for each of the sensors that have been successfully ing
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(),
```



```

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

```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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

get_data_quality_metrics	Returns details about the requested data quality metrics
get_feedback	Get feedback for an anomaly group
get_sample_data	Returns a selection of sample records from an Amazon S3 datasource
list_alerts	Lists the alerts attached to a detector
list_anomaly_detectors	Lists the detectors in the current AWS Region
list_anomaly_group_related_metrics	Returns a list of measures that are potential causes or effects of an anomaly group
list_anomaly_group_summaries	Returns a list of anomaly groups
list_anomaly_group_time_series	Gets a list of anomalous metrics for a measure in an anomaly group
list_metric_sets	Lists the datasets in the current AWS Region
list_tags_for_resource	Gets a list of tags for a detector, dataset, or alert
put_feedback	Add feedback for an anomalous metric
tag_resource	Adds tags to a detector, dataset, or alert
untag_resource	Removes tags from a detector, dataset, or alert
update_alert	Make changes to an existing alert
update_anomaly_detector	Updates a detector
update_metric_set	Updates a dataset

Examples

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

## End(Not run)
```

machinelearning

Amazon Machine Learning

Description

Definition of the public APIs exposed by Amazon Machine Learning

Usage

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

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- machinelearning(
  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_tags	Adds one or more tags to an object, up to a limit of 10
create_batch_prediction	Generates predictions for a group of observations
create_data_source_from_rds	Creates a DataSource object from an Amazon Relational Database Service (Amazon RDS)
create_data_source_from_redshift	Creates a DataSource from a database hosted on an Amazon Redshift cluster
create_data_source_from_s3	Creates a DataSource object
create_evaluation	Creates a new Evaluation of an MLModel
create_ml_model	Creates a new MLModel using the DataSource and the recipe as information sources
create_realtime_endpoint	Creates a real-time endpoint for the MLModel
delete_batch_prediction	Assigns the DELETED status to a BatchPrediction, rendering it unusable
delete_data_source	Assigns the DELETED status to a DataSource, rendering it unusable
delete_evaluation	Assigns the DELETED status to an Evaluation, rendering it unusable
delete_ml_model	Assigns the DELETED status to an MLModel, rendering it unusable
delete_realtime_endpoint	Deletes a real time endpoint of an MLModel
delete_tags	Deletes the specified tags associated with an ML object
describe_batch_predictions	Returns a list of BatchPrediction operations that match the search criteria in the request
describe_data_sources	Returns a list of DataSource that match the search criteria in the request
describe_evaluations	Returns a list of DescribeEvaluations that match the search criteria in the request
describe_ml_models	Returns a list of MLModel that match the search criteria in the request
describe_tags	Describes one or more of the tags for your Amazon ML object
get_batch_prediction	Returns a BatchPrediction that includes detailed metadata, status, and data file information

get_data_source	Returns a DataSource that includes metadata and data file information, as well as the current status of the DataSource
get_evaluation	Returns an Evaluation that includes metadata as well as the current status of the Evaluation
get_ml_model	Returns an MLModel that includes detailed metadata, data source information, and the current status of the MLModel
predict	Generates a prediction for the observation using the specified ML Model
update_batch_prediction	Updates the BatchPredictionName of a BatchPrediction
update_data_source	Updates the DataSourceName of a DataSource
update_evaluation	Updates the EvaluationName of an Evaluation
update_ml_model	Updates the MLModelName and the ScoreThreshold of an MLModel

Examples

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

panorama

AWS Panorama

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

- `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-endpoint.html>

credentials	Optional credentials shorthand for the config parameter
	<ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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	Removes tags from a resource
update_device_metadata	Updates a device's metadata

Examples

```
## Not run:
svc <- panorama()
svc$create_application_instance(
  Foo = 123
)

## End(Not run)
```

personalize	<i>Amazon Personalize</i>
-------------	---------------------------

Description

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

Usage

```
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.

	<ul style="list-style-type: none"> • 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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
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 retrains 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)
describe_batch_segment_job	Gets the properties of a batch segment job including name, Amazon Resource Name (ARN)
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
describe_dataset_group	Describes the given dataset group
describe_dataset_import_job	Describes the dataset import job created by CreateDatasetImportJob, including the import job
describe_event_tracker	Describes an event tracker
describe_feature_transformation	Describes the given feature transformation
describe_filter	Describes a filter's properties

<code>describe_metric_attribution</code>	Describes a metric attribution
<code>describe_recipe</code>	Describes a recipe
<code>describe_recommender</code>	Describes the given recommender, including its status
<code>describe_schema</code>	Describes a schema
<code>describe_solution</code>	Describes a solution
<code>describe_solution_version</code>	Describes a specific version of a solution
<code>get_solution_metrics</code>	Gets the metrics for the specified solution version
<code>list_batch_inference_jobs</code>	Gets a list of the batch inference jobs that have been performed off of a solution version
<code>list_batch_segment_jobs</code>	Gets a list of the batch segment jobs that have been performed off of a solution version that
<code>list_campaigns</code>	Returns a list of campaigns that use the given solution
<code>list_dataset_export_jobs</code>	Returns a list of dataset export jobs that use the given dataset
<code>list_dataset_groups</code>	Returns a list of dataset groups
<code>list_dataset_import_jobs</code>	Returns a list of dataset import jobs that use the given dataset
<code>list_datasets</code>	Returns the list of datasets contained in the given dataset group
<code>list_event_trackers</code>	Returns the list of event trackers associated with the account
<code>list_filters</code>	Lists all filters that belong to a given dataset group
<code>list_metric_attribution_metrics</code>	Lists the metrics for the metric attribution
<code>list_metric_attributions</code>	Lists metric attributions
<code>list_recipes</code>	Returns a list of available recipes
<code>list_recommenders</code>	Returns a list of recommenders in a given Domain dataset group
<code>list_schemas</code>	Returns the list of schemas associated with the account
<code>list_solutions</code>	Returns a list of solutions that use the given dataset group
<code>list_solution_versions</code>	Returns a list of solution versions for the given solution
<code>list_tags_for_resource</code>	Get a list of tags attached to a resource
<code>start_recommender</code>	Starts a recommender that is INACTIVE
<code>stop_recommender</code>	Stops a recommender that is ACTIVE
<code>stop_solution_version_creation</code>	Stops creating a solution version that is in a state of CREATE_PENDING or CREATE IN
<code>tag_resource</code>	Add a list of tags to a resource
<code>untag_resource</code>	Remove tags that are attached to a resource
<code>update_campaign</code>	Updates a campaign by either deploying a new solution or changing the value of the campa
<code>update_dataset</code>	Update a dataset to replace its schema with a new or existing one
<code>update_metric_attribution</code>	Updates a metric attribution
<code>update_recommender</code>	Updates the recommender to modify the recommender configuration

Examples

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

## End(Not run)
```

personalizeevents *Amazon Personalize Events*

Description

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

Usage

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

Arguments

config	<p>Optional configuration of credentials, endpoint, and/or region.</p> <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	<p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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

`put_items` Adds one or more items to an Items dataset
`put_users` Adds one or more users to a Users dataset

Examples

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

personalizeruntime *Amazon Personalize Runtime*

Description

Amazon Personalize Runtime

Usage

```
personalizeruntime(  
  config = list(),  
  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.

	<ul style="list-style-type: none"> • 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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",
```



```

        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:**

	<ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- polly(
  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_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

`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-endpoint.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

<code>compare_faces</code>	Compares a face in the source input image with each of the 100 largest faces detected in the target image
<code>copy_project_version</code>	Copies a version of an Amazon Rekognition Custom Labels model from a source project to a target project
<code>create_collection</code>	Creates a collection in an AWS Region
<code>create_dataset</code>	Creates a new Amazon Rekognition Custom Labels dataset
<code>create_face_liveness_session</code>	This API operation initiates a Face Liveness session
<code>create_project</code>	Creates a new Amazon Rekognition Custom Labels project
<code>create_project_version</code>	Creates a new version of a model and begins training
<code>create_stream_processor</code>	Creates an Amazon Rekognition stream processor that you can use to detect and recognize faces in a video stream
<code>create_user</code>	Creates a new User within a collection specified by CollectionId
<code>delete_collection</code>	Deletes the specified collection
<code>delete_dataset</code>	Deletes an existing Amazon Rekognition Custom Labels dataset
<code>delete_faces</code>	Deletes faces from a collection
<code>delete_project</code>	Deletes an Amazon Rekognition Custom Labels project
<code>delete_project_policy</code>	Deletes an existing project policy
<code>delete_project_version</code>	Deletes an Amazon Rekognition Custom Labels model
<code>delete_stream_processor</code>	Deletes the stream processor identified by Name
<code>delete_user</code>	Deletes the specified UserID within the collection
<code>describe_collection</code>	Describes the specified collection
<code>describe_dataset</code>	Describes an Amazon Rekognition Custom Labels dataset
<code>describe_projects</code>	Gets information about your Amazon Rekognition Custom Labels projects
<code>describe_project_versions</code>	Lists and describes the versions of a model in an Amazon Rekognition Custom Labels project
<code>describe_stream_processor</code>	Provides information about a stream processor created by CreateStreamProcessor
<code>detect_custom_labels</code>	Detects custom labels in a supplied image by using an Amazon Rekognition Custom Labels model
<code>detect_faces</code>	Detects faces within an image that is provided as input
<code>detect_labels</code>	Detects instances of real-world entities within an image (JPEG or PNG) provided as input
<code>detect_moderation_labels</code>	Detects unsafe content in a specified JPEG or PNG format image
<code>detect_protective_equipment</code>	Detects Personal Protective Equipment (PPE) worn by people detected in an image
<code>detect_text</code>	Detects text in the input image and converts it into machine-readable text
<code>disassociate_faces</code>	Removes the association between a Face supplied in an array of FaceIds and the User
<code>distribute_dataset_entries</code>	Distributes the entries (images) in a training dataset across the training dataset and the test dataset
<code>get_celebrity_info</code>	Gets the name and additional information about a celebrity based on their Amazon Rekognition Custom Labels model
<code>get_celebrity_recognition</code>	Gets the celebrity recognition results for a Amazon Rekognition Video analysis started by StartFaceSearch
<code>get_content_moderation</code>	Gets the inappropriate, unwanted, or offensive content analysis results for a Amazon Rekognition Video analysis started by StartFaceSearch
<code>get_face_detection</code>	Gets face detection results for a Amazon Rekognition Video analysis started by StartFaceSearch
<code>get_face_liveness_session_results</code>	Retrieves the results of a specific Face Liveness session
<code>get_face_search</code>	Gets the face search results for Amazon Rekognition Video face search started by StartFaceSearch
<code>get_label_detection</code>	Gets the label detection results of a Amazon Rekognition Video analysis started by StartFaceSearch
<code>get_person_tracking</code>	Gets the path tracking results of a Amazon Rekognition Video analysis started by StartFaceSearch
<code>get_segment_detection</code>	Gets the segment detection results of a Amazon Rekognition Video analysis started by StartFaceSearch
<code>get_text_detection</code>	Gets the text detection results of a Amazon Rekognition Video analysis started by StartFaceSearch
<code>index_faces</code>	Detects faces in the input image and adds them to the specified collection
<code>list_collections</code>	Returns list of collection IDs in your account
<code>list_dataset_entries</code>	Lists the entries (images) within a dataset
<code>list_dataset_labels</code>	Lists the labels in a dataset
<code>list_faces</code>	Returns metadata for faces in the specified collection
<code>list_project_policies</code>	Gets a list of the project policies attached to a project
<code>list_stream_processors</code>	Gets a list of stream processors that you have created with CreateStreamProcessor
<code>list_tags_for_resource</code>	Returns a list of tags in an Amazon Rekognition collection, stream processor, or Custom Labels model

<code>list_users</code>	Returns metadata of the User such as UserID in the specified collection
<code>put_project_policy</code>	Attaches a project policy to a Amazon Rekognition Custom Labels project in a trusting A
<code>recognize_celebrities</code>	Returns an array of celebrities recognized in the input image
<code>search_faces</code>	For a given input face ID, searches for matching faces in the collection the face belongs to
<code>search_faces_by_image</code>	For a given input image, first detects the largest face in the image, and then searches the s
<code>search_users</code>	Searches for UserIDs within a collection based on a FaceId or UserId
<code>search_users_by_image</code>	Searches for UserIDs using a supplied image
<code>start_celebrity_recognition</code>	Starts asynchronous recognition of celebrities in a stored video
<code>start_content_moderation</code>	Starts asynchronous detection of inappropriate, unwanted, or offensive content in a stored
<code>start_face_detection</code>	Starts asynchronous detection of faces in a stored video
<code>start_face_search</code>	Starts the asynchronous search for faces in a collection that match the faces of persons de
<code>start_label_detection</code>	Starts asynchronous detection of labels in a stored video
<code>start_person_tracking</code>	Starts the asynchronous tracking of a person's path in a stored video
<code>start_project_version</code>	Starts the running of the version of a model
<code>start_segment_detection</code>	Starts asynchronous detection of segment detection in a stored video
<code>start_stream_processor</code>	Starts processing a stream processor
<code>start_text_detection</code>	Starts asynchronous detection of text in a stored video
<code>stop_project_version</code>	Stops a running model
<code>stop_stream_processor</code>	Stops a running stream processor that was created by CreateStreamProcessor
<code>tag_resource</code>	Adds one or more key-value tags to an Amazon Rekognition collection, stream processor,
<code>untag_resource</code>	Removes one or more tags from an Amazon Rekognition collection, stream processor, or
<code>update_dataset_entries</code>	Adds or updates one or more entries (images) in a dataset
<code>update_stream_processor</code>	Allows you to update a stream processor

Examples

```
## 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-6cf79a3d806e",
    "851cb847-dccc-4fea-9309-9f4805967855",
    "35ebbb41-7f67-4263-908d-dd0ecba05ab9"
  ),
  UserId = "DemoUser",
  UserMatchThreshold = 70L
)

## End(Not run)
```

sagemaker

*Amazon SageMaker Service***Description**

Provides APIs for creating and managing SageMaker resources.

Other Resources:

- [SageMaker Developer Guide](#)
- [Amazon Augmented AI Runtime API Reference](#)

Usage

```
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 <https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.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 <- 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

<code>add_association</code>	Creates an association between the source and the destination
<code>add_tags</code>	Adds or overwrites one or more tags for the specified SageMaker resource
<code>associate_trial_component</code>	Associates a trial component with a trial
<code>batch_describe_model_package</code>	This action batch describes a list of versioned model packages
<code>create_action</code>	Creates an action
<code>create_algorithm</code>	Create a machine learning algorithm that you can use in SageMaker and Amazon SageMaker
<code>create_app</code>	Creates a running app for the specified UserProfile
<code>create_app_image_config</code>	Creates a configuration for running a SageMaker image as a KernelGateway
<code>create_artifact</code>	Creates an artifact
<code>create_auto_ml_job</code>	Creates an Autopilot job also referred to as Autopilot experiment or AutoML
<code>create_auto_ml_job_v2</code>	Creates an Autopilot job also referred to as Autopilot experiment or AutoML
<code>create_code_repository</code>	Creates a Git repository as a resource in your SageMaker account
<code>create_compilation_job</code>	Starts a model compilation job
<code>create_context</code>	Creates a context
<code>create_data_quality_job_definition</code>	Creates a definition for a job that monitors data quality and drift
<code>create_device_fleet</code>	Creates a device fleet
<code>create_domain</code>	Creates a Domain used by Amazon SageMaker Studio
<code>create_edge_deployment_plan</code>	Creates an edge deployment plan, consisting of multiple stages
<code>create_edge_deployment_stage</code>	Creates a new stage in an existing edge deployment plan
<code>create_edge_packaging_job</code>	Starts a SageMaker Edge Manager model packaging job
<code>create_endpoint</code>	Creates an endpoint using the endpoint configuration specified in the request
<code>create_endpoint_config</code>	Creates an endpoint configuration that SageMaker hosting services use
<code>create_experiment</code>	Creates a SageMaker experiment
<code>create_feature_group</code>	Create a new FeatureGroup
<code>create_flow_definition</code>	Creates a flow definition
<code>create_hub</code>	Create a hub
<code>create_human_task_ui</code>	Defines the settings you will use for the human review workflow user interface
<code>create_hyper_parameter_tuning_job</code>	Starts a hyperparameter tuning job
<code>create_image</code>	Creates a custom SageMaker image
<code>create_image_version</code>	Creates a version of the SageMaker image specified by ImageName
<code>create_inference_experiment</code>	Creates an inference experiment using the configurations specified in the request
<code>create_inference_recommendations_job</code>	Starts a recommendation job
<code>create_labeling_job</code>	Creates a job that uses workers to label the data objects in your input data
<code>create_model</code>	Creates a model in SageMaker
<code>create_model_bias_job_definition</code>	Creates the definition for a model bias job
<code>create_model_card</code>	Creates an Amazon SageMaker Model Card
<code>create_model_card_export_job</code>	Creates an Amazon SageMaker Model Card export job
<code>create_model_explainability_job_definition</code>	Creates the definition for a model explainability job
<code>create_model_package</code>	Creates a model package that you can use to create SageMaker models
<code>create_model_package_group</code>	Creates a model group
<code>create_model_quality_job_definition</code>	Creates a definition for a job that monitors model quality and drift
<code>create_monitoring_schedule</code>	Creates a schedule that regularly starts Amazon SageMaker Processing
<code>create_notebook_instance</code>	Creates an SageMaker notebook instance
<code>create_notebook_instance_lifecycle_config</code>	Creates a lifecycle configuration that you can associate with a notebook instance
<code>create_pipeline</code>	Creates a pipeline using a JSON pipeline definition
<code>create_presigned_domain_url</code>	Creates a URL for a specified UserProfile in a Domain
<code>create_presigned_notebook_instance_url</code>	Returns a URL that you can use to connect to the Jupyter server from a notebook instance
<code>create_processing_job</code>	Creates a processing job

<code>create_project</code>	Creates a machine learning (ML) project that can contain one or more t
<code>create_space</code>	Creates a space used for real time collaboration in a Domain
<code>create_studio_lifecycle_config</code>	Creates a new Studio Lifecycle Configuration
<code>create_training_job</code>	Starts a model training job
<code>create_transform_job</code>	Starts a transform job
<code>create_trial</code>	Creates an SageMaker trial
<code>create_trial_component</code>	Creates a trial component, which is a stage of a machine learning trial
<code>create_user_profile</code>	Creates a user profile
<code>create_workforce</code>	Use this operation to create a workforce
<code>create_workteam</code>	Creates a new work team for labeling your data
<code>delete_action</code>	Deletes an action
<code>delete_algorithm</code>	Removes the specified algorithm from your account
<code>delete_app</code>	Used to stop and delete an app
<code>delete_app_image_config</code>	Deletes an AppImageConfig
<code>delete_artifact</code>	Deletes an artifact
<code>delete_association</code>	Deletes an association
<code>delete_code_repository</code>	Deletes the specified Git repository from your account
<code>delete_context</code>	Deletes an context
<code>delete_data_quality_job_definition</code>	Deletes a data quality monitoring job definition
<code>delete_device_fleet</code>	Deletes a fleet
<code>delete_domain</code>	Used to delete a domain
<code>delete_edge_deployment_plan</code>	Deletes an edge deployment plan if (and only if) all the stages in the pla
<code>delete_edge_deployment_stage</code>	Delete a stage in an edge deployment plan if (and only if) the stage is in
<code>delete_endpoint</code>	Deletes an endpoint
<code>delete_endpoint_config</code>	Deletes an endpoint configuration
<code>delete_experiment</code>	Deletes an SageMaker experiment
<code>delete_feature_group</code>	Delete the FeatureGroup and any data that was written to the OnlineSto
<code>delete_flow_definition</code>	Deletes the specified flow definition
<code>delete_hub</code>	Delete a hub
<code>delete_hub_content</code>	Delete the contents of a hub
<code>delete_human_task_ui</code>	Use this operation to delete a human task user interface (worker task ter
<code>delete_image</code>	Deletes a SageMaker image and all versions of the image
<code>delete_image_version</code>	Deletes a version of a SageMaker image
<code>delete_inference_experiment</code>	Deletes an inference experiment
<code>delete_model</code>	Deletes a model
<code>delete_model_bias_job_definition</code>	Deletes an Amazon SageMaker model bias job definition
<code>delete_model_card</code>	Deletes an Amazon SageMaker Model Card
<code>delete_model_explainability_job_definition</code>	Deletes an Amazon SageMaker model explainability job definition
<code>delete_model_package</code>	Deletes a model package
<code>delete_model_package_group</code>	Deletes the specified model group
<code>delete_model_package_group_policy</code>	Deletes a model group resource policy
<code>delete_model_quality_job_definition</code>	Deletes the secified model quality monitoring job definition
<code>delete_monitoring_schedule</code>	Deletes a monitoring schedule
<code>delete_notebook_instance</code>	Deletes an SageMaker notebook instance
<code>delete_notebook_instance_lifecycle_config</code>	Deletes a notebook instance lifecycle configuration
<code>delete_pipeline</code>	Deletes a pipeline if there are no running instances of the pipeline
<code>delete_project</code>	Delete the specified project
<code>delete_space</code>	Used to delete a space

<code>delete_studio_lifecycle_config</code>	Deletes the Studio Lifecycle Configuration
<code>delete_tags</code>	Deletes the specified tags from an SageMaker resource
<code>delete_trial</code>	Deletes the specified trial
<code>delete_trial_component</code>	Deletes the specified trial component
<code>delete_user_profile</code>	Deletes a user profile
<code>delete_workforce</code>	Use this operation to delete a workforce
<code>delete_workteam</code>	Deletes an existing work team
<code>deregister_devices</code>	Deregisters the specified devices
<code>describe_action</code>	Describes an action
<code>describe_algorithm</code>	Returns a description of the specified algorithm that is in your account
<code>describe_app</code>	Describes the app
<code>describe_app_image_config</code>	Describes an AppImageConfig
<code>describe_artifact</code>	Describes an artifact
<code>describe_auto_ml_job</code>	Returns information about an AutoML job created by calling CreateAutoMLJob
<code>describe_auto_ml_job_v2</code>	Returns information about an AutoML job created by calling CreateAutoMLJobV2
<code>describe_code_repository</code>	Gets details about the specified Git repository
<code>describe_compilation_job</code>	Returns information about a model compilation job
<code>describe_context</code>	Describes a context
<code>describe_data_quality_job_definition</code>	Gets the details of a data quality monitoring job definition
<code>describe_device</code>	Describes the device
<code>describe_device_fleet</code>	A description of the fleet the device belongs to
<code>describe_domain</code>	The description of the domain
<code>describe_edge_deployment_plan</code>	Describes an edge deployment plan with deployment status per stage
<code>describe_edge_packaging_job</code>	A description of edge packaging jobs
<code>describe_endpoint</code>	Returns the description of an endpoint
<code>describe_endpoint_config</code>	Returns the description of an endpoint configuration created using the CreateEndpointConfig API
<code>describe_experiment</code>	Provides a list of an experiment's properties
<code>describe_feature_group</code>	Use this operation to describe a FeatureGroup
<code>describe_feature_metadata</code>	Shows the metadata for a feature within a feature group
<code>describe_flow_definition</code>	Returns information about the specified flow definition
<code>describe_hub</code>	Describe a hub
<code>describe_hub_content</code>	Describe the content of a hub
<code>describe_human_task_ui</code>	Returns information about the requested human task user interface (worker)
<code>describe_hyper_parameter_tuning_job</code>	Returns a description of a hyperparameter tuning job, depending on the job type
<code>describe_image</code>	Describes a SageMaker image
<code>describe_image_version</code>	Describes a version of a SageMaker image
<code>describe_inference_experiment</code>	Returns details about an inference experiment
<code>describe_inference_recommendations_job</code>	Provides the results of the Inference Recommender job
<code>describe_labeling_job</code>	Gets information about a labeling job
<code>describe_lineage_group</code>	Provides a list of properties for the requested lineage group
<code>describe_model</code>	Describes a model that you created using the CreateModel API
<code>describe_model_bias_job_definition</code>	Returns a description of a model bias job definition
<code>describe_model_card</code>	Describes the content, creation time, and security configuration of an Amazon SageMaker Model Card
<code>describe_model_card_export_job</code>	Describes an Amazon SageMaker Model Card export job
<code>describe_model_explainability_job_definition</code>	Returns a description of a model explainability job definition
<code>describe_model_package</code>	Returns a description of the specified model package, which is used to create a SageMaker endpoint
<code>describe_model_package_group</code>	Gets a description for the specified model group
<code>describe_model_quality_job_definition</code>	Returns a description of a model quality job definition

<code>describe_monitoring_schedule</code>	Describes the schedule for a monitoring job
<code>describe_notebook_instance</code>	Returns information about a notebook instance
<code>describe_notebook_instance_lifecycle_config</code>	Returns a description of a notebook instance lifecycle configuration
<code>describe_pipeline</code>	Describes the details of a pipeline
<code>describe_pipeline_definition_for_execution</code>	Describes the details of an execution's pipeline definition
<code>describe_pipeline_execution</code>	Describes the details of a pipeline execution
<code>describe_processing_job</code>	Returns a description of a processing job
<code>describe_project</code>	Describes the details of a project
<code>describe_space</code>	Describes the space
<code>describe_studio_lifecycle_config</code>	Describes the Studio Lifecycle Configuration
<code>describe_subscribed_workteam</code>	Gets information about a work team provided by a vendor
<code>describe_training_job</code>	Returns information about a training job
<code>describe_transform_job</code>	Returns information about a transform job
<code>describe_trial</code>	Provides a list of a trial's properties
<code>describe_trial_component</code>	Provides a list of a trials component's properties
<code>describe_user_profile</code>	Describes a user profile
<code>describe_workforce</code>	Lists private workforce information, including workforce name, Amazon
<code>describe_workteam</code>	Gets information about a specific work team
<code>disable_sagemaker_servicecatalog_portfolio</code>	Disables using Service Catalog in SageMaker
<code>disassociate_trial_component</code>	Disassociates a trial component from a trial
<code>enable_sagemaker_servicecatalog_portfolio</code>	Enables using Service Catalog in SageMaker
<code>get_device_fleet_report</code>	Describes a fleet
<code>get_lineage_group_policy</code>	The resource policy for the lineage group
<code>get_model_package_group_policy</code>	Gets a resource policy that manages access for a model group
<code>get_sagemaker_servicecatalog_portfolio_status</code>	Gets the status of Service Catalog in SageMaker
<code>get_scaling_configuration_recommendation</code>	Starts an Amazon SageMaker Inference Recommender autoscaling reco
<code>get_search_suggestions</code>	An auto-complete API for the search functionality in the SageMaker co
<code>import_hub_content</code>	Import hub content
<code>list_actions</code>	Lists the actions in your account and their properties
<code>list_algorithms</code>	Lists the machine learning algorithms that have been created
<code>list_aliases</code>	Lists the aliases of a specified image or image version
<code>list_app_image_configs</code>	Lists the AppImageConfigs in your account and their properties
<code>list_apps</code>	Lists apps
<code>list_artifacts</code>	Lists the artifacts in your account and their properties
<code>list_associations</code>	Lists the associations in your account and their properties
<code>list_auto_ml_jobs</code>	Request a list of jobs
<code>list_candidates_for_auto_ml_job</code>	List the candidates created for the job
<code>list_code_repositories</code>	Gets a list of the Git repositories in your account
<code>list_compilation_jobs</code>	Lists model compilation jobs that satisfy various filters
<code>list_contexts</code>	Lists the contexts in your account and their properties
<code>list_data_quality_job_definitions</code>	Lists the data quality job definitions in your account
<code>list_device_fleets</code>	Returns a list of devices in the fleet
<code>list_devices</code>	A list of devices
<code>list_domains</code>	Lists the domains
<code>list_edge_deployment_plans</code>	Lists all edge deployment plans
<code>list_edge_packaging_jobs</code>	Returns a list of edge packaging jobs
<code>list_endpoint_configs</code>	Lists endpoint configurations
<code>list_endpoints</code>	Lists endpoints

list_experiments	Lists all the experiments in your account
list_feature_groups	List FeatureGroups based on given filter and order
list_flow_definitions	Returns information about the flow definitions in your account
list_hub_contents	List the contents of a hub
list_hub_content_versions	List hub content versions
list_hubs	List all existing hubs
list_human_task_uis	Returns information about the human task user interfaces in your account
list_hyper_parameter_tuning_jobs	Gets a list of HyperParameterTuningJobSummary objects that describe
list_images	Lists the images in your account and their properties
list_image_versions	Lists the versions of a specified image and their properties
list_inference_experiments	Returns the list of all inference experiments
list_inference_recommendations_jobs	Lists recommendation jobs that satisfy various filters
list_inference_recommendations_job_steps	Returns a list of the subtasks for an Inference Recommender job
list_labeling_jobs	Gets a list of labeling jobs
list_labeling_jobs_for_workteam	Gets a list of labeling jobs assigned to a specified work team
list_lineage_groups	A list of lineage groups shared with your Amazon Web Services account
list_model_bias_job_definitions	Lists model bias jobs definitions that satisfy various filters
list_model_card_export_jobs	List the export jobs for the Amazon SageMaker Model Card
list_model_cards	List existing model cards
list_model_card_versions	List existing versions of an Amazon SageMaker Model Card
list_model_explainability_job_definitions	Lists model explainability job definitions that satisfy various filters
list_model_metadata	Lists the domain, framework, task, and model name of standard machine learning
list_model_package_groups	Gets a list of the model groups in your Amazon Web Services account
list_model_packages	Lists the model packages that have been created
list_model_quality_job_definitions	Gets a list of model quality monitoring job definitions in your account
list_models	Lists models created with the CreateModel API
list_monitoring_alert_history	Gets a list of past alerts in a model monitoring schedule
list_monitoring_alerts	Gets the alerts for a single monitoring schedule
list_monitoring_executions	Returns list of all monitoring job executions
list_monitoring_schedules	Returns list of all monitoring schedules
list_notebook_instance_lifecycle_configs	Lists notebook instance lifecycle configurations created with the CreateNotebookInstanceLifecycleConfig
list_notebook_instances	Returns a list of the SageMaker notebook instances in the requester's account
list_pipeline_executions	Gets a list of the pipeline executions
list_pipeline_execution_steps	Gets a list of PipeLineExecutionStep objects
list_pipeline_parameters_for_execution	Gets a list of parameters for a pipeline execution
list_pipelines	Gets a list of pipelines
list_processing_jobs	Lists processing jobs that satisfy various filters
list_projects	Gets a list of the projects in an Amazon Web Services account
list_resource_catalogs	Lists Amazon SageMaker Catalogs based on given filters and orders
list_spaces	Lists spaces
list_stage_devices	Lists devices allocated to the stage, containing detailed device information
list_studio_lifecycle_configs	Lists the Studio Lifecycle Configurations in your Amazon Web Services account
list_subscribed_workteams	Gets a list of the work teams that you are subscribed to in the Amazon Web Services account
list_tags	Returns the tags for the specified SageMaker resource
list_training_jobs	Lists training jobs
list_training_jobs_for_hyper_parameter_tuning_job	Gets a list of TrainingJobSummary objects that describe the training jobs
list_transform_jobs	Lists transform jobs
list_trial_components	Lists the trial components in your account

<code>list_trials</code>	Lists the trials in your account
<code>list_user_profiles</code>	Lists user profiles
<code>list_workforces</code>	Use this operation to list all private and vendor workforces in an Amazon SageMaker region
<code>list_workteams</code>	Gets a list of private work teams that you have defined in a region
<code>put_model_package_group_policy</code>	Adds a resource policy to control access to a model group
<code>query_lineage</code>	Use this action to inspect your lineage and discover relationships between SageMaker resources
<code>register_devices</code>	Register devices
<code>render_ui_template</code>	Renders the UI template so that you can preview the worker's experience
<code>retry_pipeline_execution</code>	Retry the execution of the pipeline
<code>search</code>	Finds SageMaker resources that match a search query
<code>send_pipeline_execution_step_failure</code>	Notifies the pipeline that the execution of a callback step failed, along with the error message
<code>send_pipeline_execution_step_success</code>	Notifies the pipeline that the execution of a callback step succeeded and provides the output
<code>start_edge_deployment_stage</code>	Starts a stage in an edge deployment plan
<code>start_inference_experiment</code>	Starts an inference experiment
<code>start_monitoring_schedule</code>	Starts a previously stopped monitoring schedule
<code>start_notebook_instance</code>	Launches an ML compute instance with the latest version of the libraries
<code>start_pipeline_execution</code>	Starts a pipeline execution
<code>stop_auto_ml_job</code>	A method for forcing a running job to shut down
<code>stop_compilation_job</code>	Stops a model compilation job
<code>stop_edge_deployment_stage</code>	Stops a stage in an edge deployment plan
<code>stop_edge_packaging_job</code>	Request to stop an edge packaging job
<code>stop_hyperparameter_tuning_job</code>	Stops a running hyperparameter tuning job and all running training jobs
<code>stop_inference_experiment</code>	Stops an inference experiment
<code>stop_inference_recommendations_job</code>	Stops an Inference Recommender job
<code>stop_labeling_job</code>	Stops a running labeling job
<code>stop_monitoring_schedule</code>	Stops a previously started monitoring schedule
<code>stop_notebook_instance</code>	Terminates the ML compute instance
<code>stop_pipeline_execution</code>	Stops a pipeline execution
<code>stop_processing_job</code>	Stops a processing job
<code>stop_training_job</code>	Stops a training job
<code>stop_transform_job</code>	Stops a batch transform job
<code>update_action</code>	Updates an action
<code>update_app_image_config</code>	Updates the properties of an AppImageConfig
<code>update_artifact</code>	Updates an artifact
<code>update_code_repository</code>	Updates the specified Git repository with the specified values
<code>update_context</code>	Updates a context
<code>update_device_fleet</code>	Updates a fleet of devices
<code>update_devices</code>	Updates one or more devices in a fleet
<code>update_domain</code>	Updates the default settings for new user profiles in the domain
<code>update_endpoint</code>	Deploys the new EndpointConfig specified in the request, switches to use the new EndpointConfig
<code>update_endpoint_weights_and_capacities</code>	Updates variant weight of one or more variants associated with an existing endpoint
<code>update_experiment</code>	Adds, updates, or removes the description of an experiment
<code>update_feature_group</code>	Updates the feature group by either adding features or updating the online features
<code>update_feature_metadata</code>	Updates the description and parameters of the feature group
<code>update_hub</code>	Update a hub
<code>update_image</code>	Updates the properties of a SageMaker image
<code>update_image_version</code>	Updates the properties of a SageMaker image version
<code>update_inference_experiment</code>	Updates an inference experiment that you created

<code>update_model_card</code>	Update an Amazon SageMaker Model Card
<code>update_model_package</code>	Updates a versioned model
<code>update_monitoring_alert</code>	Update the parameters of a model monitor alert
<code>update_monitoring_schedule</code>	Updates a previously created schedule
<code>update_notebook_instance</code>	Updates a notebook instance
<code>update_notebook_instance_lifecycle_config</code>	Updates a notebook instance lifecycle configuration created with the CR
<code>update_pipeline</code>	Updates a pipeline
<code>update_pipeline_execution</code>	Updates a pipeline execution
<code>update_project</code>	Updates a machine learning (ML) project that is created from a template
<code>update_space</code>	Updates the settings of a space
<code>update_training_job</code>	Update a model training job to request a new Debugger profiling configuration
<code>update_trial</code>	Updates the display name of a trial
<code>update_trial_component</code>	Updates one or more properties of a trial component
<code>update_user_profile</code>	Updates a user profile
<code>update_workforce</code>	Use this operation to update your workforce
<code>update_workteam</code>	Updates an existing work team with new member definitions or descriptions

Examples

```
## 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

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

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds: <ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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",
```

```

        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

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

```

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

## End(Not run)

```

 sagemakerfeaturestoreruntime

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`.

	<ul style="list-style-type: none"> • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	<p>Optional credentials shorthand for the config parameter</p> <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- sagemakerfeaturestoreruntime(
  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

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

```

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

## End(Not run)

```

sagemakerruntime	<i>Amazon SageMaker Runtime</i>
------------------	---------------------------------

Description

The Amazon SageMaker runtime API.

Usage

```

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

```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • credentials: <ul style="list-style-type: none"> – creds:
--------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- sagemakerruntime(
  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

[invoke_endpoint](#)

After you deploy a model into production using Amazon SageMaker hosting service,

[invoke_endpoint_async](#)

After you deploy a model into production using Amazon SageMaker hosting service,

[invoke_endpoint_with_response_stream](#)

Invokes a model at the specified endpoint to return the inference response as a stream.

Examples

```

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

## End(Not run)

```

textract

Amazon Textract

Description

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

Usage

```
textract(config = list(), 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-endpoint.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 <- 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
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 the results for an Amazon Textract asynchronous operation that 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, item numbers, and amounts
start_lending_analysis	Starts the classification and analysis of an input document

Examples

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

## End(Not run)
```

transcribeservice	<i>Amazon Transcribe Service</i>
-------------------	----------------------------------

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

```
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

	<ul style="list-style-type: none"> – 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. <ul style="list-style-type: none"> • 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 <code>true</code> to force the request to use path-style addressing, i.e. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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 <- 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	Provides a list of custom vocabulary filters that match the specified criteria
start_call_analytics_job	Transcribes the audio from a customer service call and applies any additional Request Parameters
start_medical_transcription_job	Transcribes the audio from a medical dictation or conversation and applies any additional Request Parameters
start_transcription_job	Transcribes the audio from a media file and applies any additional Request Parameters
tag_resource	Adds one or more custom tags, each in the form of a key:value pair, to the specified resource
untag_resource	Removes the specified tags from the specified Amazon Transcribe resource
update_call_analytics_category	Updates the specified Call Analytics category with new rules
update_medical_vocabulary	Updates an existing custom medical vocabulary with new values
update_vocabulary	Updates an existing custom vocabulary with new values
update_vocabulary_filter	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

```
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

	<ul style="list-style-type: none"> * 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. <code>http://s3.amazonaws.com/BUCKET/KEY</code>. • sts_regional_endpoint: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-endpoint.html
credentials	Optional credentials shorthand for the config parameter <ul style="list-style-type: none"> • creds: <ul style="list-style-type: none"> – 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,
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 given
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
)

```

```
)
## End(Not run)
```

voiceid	<i>Amazon Voice ID</i>
---------	------------------------

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-endpoint.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

<code>create_domain</code>	Creates a domain that contains all Amazon Connect Voice ID data, such as speakers, fra
<code>create_watchlist</code>	Creates a watchlist that fraudsters can be a part of
<code>delete_domain</code>	Deletes the specified domain from Voice ID
<code>delete_fraudster</code>	Deletes the specified fraudster from Voice ID
<code>delete_speaker</code>	Deletes the specified speaker from Voice ID
<code>delete_watchlist</code>	Deletes the specified watchlist from Voice ID
<code>describe_domain</code>	Describes the specified domain
<code>describe_fraudster</code>	Describes the specified fraudster
<code>describe_fraudster_registration_job</code>	Describes the specified fraudster registration job
<code>describe_speaker</code>	Describes the specified speaker
<code>describe_speaker_enrollment_job</code>	Describes the specified speaker enrollment job
<code>describe_watchlist</code>	Describes the specified watchlist
<code>disassociate_fraudster</code>	Disassociates the fraudsters from the watchlist specified
<code>evaluate_session</code>	Evaluates a specified session based on audio data accumulated during a streaming Ama
<code>list_domains</code>	Lists all the domains in the Amazon Web Services account
<code>list_fraudster_registration_jobs</code>	Lists all the fraudster registration jobs in the domain with the given JobStatus
<code>list_fraudsters</code>	Lists all fraudsters in a specified watchlist or domain
<code>list_speaker_enrollment_jobs</code>	Lists all the speaker enrollment jobs in the domain with the specified JobStatus
<code>list_speakers</code>	Lists all speakers in a specified domain
<code>list_tags_for_resource</code>	Lists all tags associated with a specified Voice ID resource
<code>list_watchlists</code>	Lists all watchlists in a specified domain
<code>opt_out_speaker</code>	Opts out a speaker from Voice ID
<code>start_fraudster_registration_job</code>	Starts a new batch fraudster registration job using provided details
<code>start_speaker_enrollment_job</code>	Starts a new batch speaker enrollment job using specified details
<code>tag_resource</code>	Tags a Voice ID resource with the provided list of tags
<code>untag_resource</code>	Removes specified tags from a specified Amazon Connect Voice ID resource
<code>update_domain</code>	Updates the specified domain
<code>update_watchlist</code>	Updates the specified 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](#)
analyze_expense, [84](#)
analyze_id, [84](#)
associate_faces, [60](#), [63](#)
associate_fraudster, [92](#)
associate_trial_component, [69](#)
augmentedairuntime, [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
- delete_context, 70
- delete_custom_vocabulary, 31
- delete_data_quality_job_definition, 70
- delete_data_source, 45
- delete_dataset, 20, 39, 51, 61, 64
- delete_dataset_group, 20, 51
- delete_dataset_import_job, 20
- delete_detector, 23
- delete_detector_version, 23
- delete_device, 48
- delete_device_fleet, 70
- delete_document_classifier, 8
- delete_domain, 70, 93
- delete_edge_deployment_plan, 70
- delete_edge_deployment_stage, 70
- delete_endpoint, 8, 70
- delete_endpoint_config, 70
- delete_entity_recognizer, 8
- delete_entity_type, 23
- delete_evaluation, 45
- delete_event, 23
- delete_event_tracker, 51
- delete_event_type, 23
- delete_events_by_event_type, 23
- delete_experiment, 70
- delete_explainability, 20
- delete_explainability_export, 20
- delete_export, 31
- delete_external_model, 24
- delete_faces, 60, 64
- delete_feature_group, 70
- delete_filter, 51
- delete_flow_definition, 70
- delete_flywheel, 8
- delete_forecast, 20
- delete_forecast_export_job, 20
- delete_fraudster, 93
- delete_hub, 70
- delete_hub_content, 70
- delete_human_loop, 5
- delete_human_task_ui, 70
- delete_image, 70
- delete_image_version, 70
- delete_import, 31
- delete_inference_experiment, 70
- delete_inference_scheduler, 39
- delete_intent, 27, 31
- delete_intent_version, 27
- delete_label, 24, 39
- delete_label_group, 39
- delete_language_model, 87
- delete_lexicon, 59
- delete_list, 24
- delete_medical_transcription_job, 87
- delete_medical_vocabulary, 87
- delete_metric_attribution, 51
- delete_ml_model, 45
- delete_model, 24, 39, 70
- delete_model_bias_job_definition, 70
- delete_model_card, 70
- delete_model_explainability_job_definition, 70
- delete_model_package, 70
- delete_model_package_group, 70
- delete_model_package_group_policy, 70
- delete_model_quality_job_definition, 70
- delete_model_version, 24
- delete_monitor, 20
- delete_monitoring_schedule, 70
- delete_notebook_instance, 70
- delete_notebook_instance_lifecycle_config,

- 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_edge_packaging_job, [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_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_personalized_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](#)
- lexruntime, [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](#)

- [list_auto_ml_jobs](#), [72](#)
- [list_batch_inference_jobs](#), [52](#)
- [list_batch_segment_jobs](#), [52](#)
- [list_bot_aliases](#), [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](#), [72](#)
- [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_uis](#), [73](#)
- [list_hyper_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](#)

- [list_metric_sets, 43](#)
- [list_model_bias_job_definitions, 73](#)
- [list_model_card_export_jobs, 73](#)
- [list_model_card_versions, 73](#)
- [list_model_cards, 73](#)
- [list_model_explainability_job_definitions, 73](#)
- [list_model_metadata, 73](#)
- [list_model_package_groups, 73](#)
- [list_model_packages, 73](#)
- [list_model_quality_job_definitions, 73](#)
- [list_model_versions, 40](#)
- [list_models, 40, 73](#)
- [list_monitor_evaluations, 20](#)
- [list_monitoring_alert_history, 73](#)
- [list_monitoring_alerts, 73](#)
- [list_monitoring_executions, 73](#)
- [list_monitoring_schedules, 73](#)
- [list_monitors, 20](#)
- [list_node_from_template_jobs, 48](#)
- [list_nodes, 48](#)
- [list_notebook_instance_lifecycle_configs, 73](#)
- [list_notebook_instances, 73](#)
- [list_package_import_jobs, 48](#)
- [list_packages, 48](#)
- [list_parallel_data, 90](#)
- [list_phi_detection_jobs, 12](#)
- [list_pii_entities_detection_jobs, 9](#)
- [list_pipeline_execution_steps, 73](#)
- [list_pipeline_executions, 73](#)
- [list_pipeline_parameters_for_execution, 73](#)
- [list_pipelines, 73](#)
- [list_predictor_backtest_export_jobs, 20](#)
- [list_predictors, 20](#)
- [list_processing_jobs, 73](#)
- [list_project_policies, 61, 64](#)
- [list_projects, 73](#)
- [list_recipes, 52](#)
- [list_recommended_intents, 31](#)
- [list_recommenders, 52](#)
- [list_resource_catalogs, 73](#)
- [list_rx_norm_inference_jobs, 12](#)
- [list_schemas, 52](#)
- [list_sensor_statistics, 40](#)
- [list_sentiment_detection_jobs, 9](#)
- [list_session_analytics_data, 31](#)
- [list_session_metrics, 31](#)
- [list_slot_types, 31](#)
- [list_slots, 31](#)
- [list_snomedct_inference_jobs, 12](#)
- [list_solution_versions, 52](#)
- [list_solutions, 52](#)
- [list_spaces, 73](#)
- [list_speaker_enrollment_jobs, 93](#)
- [list_speakers, 93](#)
- [list_speech_synthesis_tasks, 59](#)
- [list_stage_devices, 73](#)
- [list_stream_processors, 62, 64](#)
- [list_studio_lifecycle_configs, 73](#)
- [list_subscribed_workteams, 73](#)
- [list_tags, 73](#)
- [list_tags_for_resource, 9, 15, 20, 24, 28, 31, 40, 43, 48, 52, 64, 87, 90, 93](#)
- [list_targeted_sentiment_detection_jobs, 9](#)
- [list_terminologies, 90](#)
- [list_test_execution_result_items, 32](#)
- [list_test_executions, 32](#)
- [list_test_set_records, 32](#)
- [list_test_sets, 32](#)
- [list_text_translation_jobs, 90](#)
- [list_topics_detection_jobs, 9](#)
- [list_training_jobs, 73](#)
- [list_training_jobs_for_hyper_parameter_tuning_job, 73](#)
- [list_transcription_jobs, 87](#)
- [list_transform_jobs, 73](#)
- [list_trial_components, 73](#)
- [list_trials, 74](#)
- [list_user_profiles, 74](#)
- [list_users, 60, 65](#)
- [list_utterance_analytics_data, 32](#)
- [list_utterance_metrics, 32](#)
- [list_vocabularies, 87](#)
- [list_vocabulary_filters, 88](#)
- [list_watchlists, 93](#)
- [list_what_if_analyses, 20](#)
- [list_what_if_forecast_exports, 21](#)
- [list_what_if_forecasts, 21](#)
- [list_workforces, 74](#)
- [list_workteams, 74](#)
- [lookoutequipment, 37](#)
- [lookoutmetrics, 40](#)

- 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
- sagemakerfeaturestoruntime, 78
- sagemakerruntime, 80
- search, 74
- search_associated_transcripts, 32
- search_faces, 60, 65
- search_faces_by_image, 60, 65
- search_users, 60, 65
- search_users_by_image, 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_dominant_language_detection_job, 9
- start_edge_deployment_stage, 74
- start_entities_detection_job, 9
- start_entities_detection_v2_job, 12
- start_events_detection_job, 9
- start_expense_analysis, 84
- start_face_detection, 61, 65
- start_face_search, 61, 65
- start_flywheel_iteration, 9
- start_fraudster_registration_job, 93
- start_human_loop, 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_dominant_language_detection_job, [9](#)
- 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_hyper_parameter_tuning_job, [74](#)
- stop_icd10cm_inference_job, [12](#)
- stop_inference_experiment, [74](#)
- stop_inference_recommendations_job, [74](#)
- stop_inference_scheduler, [40](#)
- stop_key_phrases_detection_job, [9](#)
- 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](#)

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