

Package ‘paws.machine.learning’

July 23, 2025

Title 'Amazon Web Services' Machine Learning Services

Version 0.9.0

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>,
<https://paws-r.r-universe.dev/paws.machine.learning>

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

Imports paws.common (>= 0.8.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.3.2

Collate 'augmentedairruntime_service.R'
'augmentedairruntime_interfaces.R'
'augmentedairruntime_operations.R' 'bedrock_service.R'
'bedrock_interfaces.R' 'bedrock_operations.R'
'bedrockagent_service.R' 'bedrockagent_interfaces.R'
'bedrockagent_operations.R' 'bedrockagentruntime_service.R'
'bedrockagentruntime_interfaces.R'
'bedrockagentruntime_operations.R'
'bedrockdataautomation_service.R'
'bedrockdataautomation_interfaces.R'
'bedrockdataautomation_operations.R'
'bedrockdataautomationruntime_service.R'
'bedrockdataautomationruntime_interfaces.R'
'bedrockdataautomationruntime_operations.R'
'bedrockruntime_service.R' 'bedrockruntime_interfaces.R'
'bedrockruntime_operations.R' 'comprehend_service.R'
'comprehend_interfaces.R' 'comprehend_operations.R'
'comprehendmedical_service.R' 'comprehendmedical_interfaces.R'

'comprehendmedical_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'
 'sagemakergeospatialcapabilities_service.R'
 'sagemakergeospatialcapabilities_interfaces.R'
 'sagemakergeospatialcapabilities_operations.R'
 'sagemakermetrics_service.R' 'sagemakermetrics_interfaces.R'
 'sagemakermetrics_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 2025-03-15 06:50:02 UTC

Contents

augmentedairruntime	4
bedrock	6
bedrockagent	10
bedrockagentruntime	14
bedrockdataautomation	16
bedrockdataautomationruntime	19
bedrockruntime	21
comprehend	23
comprehendmedical	27
forecastqueryservice	30
forecastservice	32
frauddetector	36
lexmodelbuildingservice	40
lexmodelsv2	43
lexruntimev2	48
lexruntimev2	50
lookoutequipment	52
lookoutmetrics	56
machinelearning	58
panorama	61
personalize	64
personalizeevents	68
personalizeruntime	70
polly	73
rekognition	75
sagemaker	81
sagemakeredgemanager	91
sagemakerfeaturestoreruntime	93
sagemakergeospatialcapabilities	96
sagemakermetrics	98
sagemakerruntime	101
textract	103
transcribeservice	106
translate	109

voiceid 112

Index 115

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. <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 • 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)

```

bedrock

Amazon Bedrock

Description

Describes the API operations for creating, managing, fine-tuning, and evaluating Amazon Bedrock models.

Usage

```
bedrock(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. <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 • 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 <- bedrock(
  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_delete_evaluation_job	Deletes a batch of evaluation jobs
create_evaluation_job	Creates an evaluation job
create_guardrail	Creates a guardrail to block topics and to implement safeguards for your g
create_guardrail_version	Creates a version of the guardrail
create_inference_profile	Creates an application inference profile to track metrics and costs when inv
create_marketplace_model_endpoint	Creates an endpoint for a model from Amazon Bedrock Marketplace
create_model_copy_job	Copies a model to another region so that it can be used there
create_model_customization_job	Creates a fine-tuning job to customize a base model
create_model_import_job	Creates a model import job to import model that you have customized in o
create_model_invocation_job	Creates a batch inference job to invoke a model on multiple prompts
create_provisioned_model_throughput	Creates dedicated throughput for a base or custom model with the model u
delete_custom_model	Deletes a custom model that you created earlier
delete_guardrail	Deletes a guardrail

<code>delete_imported_model</code>	Deletes a custom model that you imported earlier
<code>delete_inference_profile</code>	Deletes an application inference profile
<code>delete_marketplace_model_endpoint</code>	Deletes an endpoint for a model from Amazon Bedrock Marketplace
<code>delete_model_invocation_logging_configuration</code>	Delete the invocation logging
<code>delete_provisioned_model_throughput</code>	Deletes a Provisioned Throughput
<code>deregister_marketplace_model_endpoint</code>	Deregisters an endpoint for a model from Amazon Bedrock Marketplace
<code>get_custom_model</code>	Get the properties associated with a Amazon Bedrock custom model that y
<code>get_evaluation_job</code>	Gets information about an evaluation job, such as the status of the job
<code>get_foundation_model</code>	Get details about a Amazon Bedrock foundation model
<code>get_guardrail</code>	Gets details about a guardrail
<code>get_imported_model</code>	Gets properties associated with a customized model you imported
<code>get_inference_profile</code>	Gets information about an inference profile
<code>get_marketplace_model_endpoint</code>	Retrieves details about a specific endpoint for a model from Amazon Bedro
<code>get_model_copy_job</code>	Retrieves information about a model copy job
<code>get_model_customization_job</code>	Retrieves the properties associated with a model-customization job, includ
<code>get_model_import_job</code>	Retrieves the properties associated with import model job, including the sta
<code>get_model_invocation_job</code>	Gets details about a batch inference job
<code>get_model_invocation_logging_configuration</code>	Get the current configuration values for model invocation logging
<code>get_prompt_router</code>	Retrieves details about a prompt router
<code>get_provisioned_model_throughput</code>	Returns details for a Provisioned Throughput
<code>list_custom_models</code>	Returns a list of the custom models that you have created with the CreateM
<code>list_evaluation_jobs</code>	Lists all existing evaluation jobs
<code>list_foundation_models</code>	Lists Amazon Bedrock foundation models that you can use
<code>list_guardrails</code>	Lists details about all the guardrails in an account
<code>list_imported_models</code>	Returns a list of models you've imported
<code>list_inference_profiles</code>	Returns a list of inference profiles that you can use
<code>list_marketplace_model_endpoints</code>	Lists the endpoints for models from Amazon Bedrock Marketplace in your
<code>list_model_copy_jobs</code>	Returns a list of model copy jobs that you have submitted
<code>list_model_customization_jobs</code>	Returns a list of model customization jobs that you have submitted
<code>list_model_import_jobs</code>	Returns a list of import jobs you've submitted
<code>list_model_invocation_jobs</code>	Lists all batch inference jobs in the account
<code>list_prompt_routers</code>	Retrieves a list of prompt routers
<code>list_provisioned_model_throughputs</code>	Lists the Provisioned Throughputs in the account
<code>list_tags_for_resource</code>	List the tags associated with the specified resource
<code>put_model_invocation_logging_configuration</code>	Set the configuration values for model invocation logging
<code>register_marketplace_model_endpoint</code>	Registers an existing Amazon SageMaker endpoint with Amazon Bedrock
<code>stop_evaluation_job</code>	Stops an evaluation job that is current being created or running
<code>stop_model_customization_job</code>	Stops an active model customization job
<code>stop_model_invocation_job</code>	Stops a batch inference job
<code>tag_resource</code>	Associate tags with a resource
<code>untag_resource</code>	Remove one or more tags from a resource
<code>update_guardrail</code>	Updates a guardrail with the values you specify
<code>update_marketplace_model_endpoint</code>	Updates the configuration of an existing endpoint for a model from Amazo
<code>update_provisioned_model_throughput</code>	Updates the name or associated model for a Provisioned Throughput

Examples

```
## Not run:
svc <- bedrock()
svc$batch_delete_evaluation_job(
  Foo = 123
)

## End(Not run)
```

bedrockagent

Agents for Amazon Bedrock

Description

Describes the API operations for creating and managing Amazon Bedrock agents.

Usage

```
bedrockagent(
  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 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 • 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 <- bedrockagent(
  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>associate_agent_collaborator</code>	Makes an agent a collaborator for another agent
<code>associate_agent_knowledge_base</code>	Associates a knowledge base with an agent
<code>create_agent</code>	Creates an agent that orchestrates interactions between foundation models, data sources, and actions
<code>create_agent_action_group</code>	Creates an action group for an agent
<code>create_agent_alias</code>	Creates an alias of an agent that can be used to deploy the agent
<code>create_data_source</code>	Connects a knowledge base to a data source
<code>create_flow</code>	Creates a prompt flow that you can use to send an input through various steps to yield a response
<code>create_flow_alias</code>	Creates an alias of a flow for deployment
<code>create_flow_version</code>	Creates a version of the flow that you can deploy
<code>create_knowledge_base</code>	Creates a knowledge base
<code>create_prompt</code>	Creates a prompt in your prompt library that you can add to a flow
<code>create_prompt_version</code>	Creates a static snapshot of your prompt that can be deployed to production
<code>delete_agent</code>	Deletes an agent
<code>delete_agent_action_group</code>	Deletes an action group in an agent
<code>delete_agent_alias</code>	Deletes an alias of an agent
<code>delete_agent_version</code>	Deletes a version of an agent
<code>delete_data_source</code>	Deletes a data source from a knowledge base
<code>delete_flow</code>	Deletes a flow
<code>delete_flow_alias</code>	Deletes an alias of a flow
<code>delete_flow_version</code>	Deletes a version of a flow
<code>delete_knowledge_base</code>	Deletes a knowledge base
<code>delete_knowledge_base_documents</code>	Deletes documents from a data source and syncs the changes to the knowledge base that is connected to the data source
<code>delete_prompt</code>	Deletes a prompt or a version of it, depending on whether you include the <code>promptVersion</code> parameter
<code>disassociate_agent_collaborator</code>	Disassociates an agent collaborator
<code>disassociate_agent_knowledge_base</code>	Disassociates a knowledge base from an agent
<code>get_agent</code>	Gets information about an agent
<code>get_agent_action_group</code>	Gets information about an action group for an agent
<code>get_agent_alias</code>	Gets information about an alias of an agent
<code>get_agent_collaborator</code>	Retrieves information about an agent's collaborator
<code>get_agent_knowledge_base</code>	Gets information about a knowledge base associated with an agent
<code>get_agent_version</code>	Gets details about a version of an agent
<code>get_data_source</code>	Gets information about a data source
<code>get_flow</code>	Retrieves information about a flow
<code>get_flow_alias</code>	Retrieves information about a flow
<code>get_flow_version</code>	Retrieves information about a version of a flow
<code>get_ingestion_job</code>	Gets information about a data ingestion job
<code>get_knowledge_base</code>	Gets information about a knowledge base
<code>get_knowledge_base_documents</code>	Retrieves specific documents from a data source that is connected to a knowledge base
<code>get_prompt</code>	Retrieves information about the working draft (DRAFT version) of a prompt or a version of a prompt
<code>ingest_knowledge_base_documents</code>	Ingests documents directly into the knowledge base that is connected to the data source

<code>list_agent_action_groups</code>	Lists the action groups for an agent and information about each one
<code>list_agent_aliases</code>	Lists the aliases of an agent and information about each one
<code>list_agent_collaborators</code>	Retrieve a list of an agent's collaborators
<code>list_agent_knowledge_bases</code>	Lists knowledge bases associated with an agent and information about each one
<code>list_agents</code>	Lists the agents belonging to an account and information about each agent
<code>list_agent_versions</code>	Lists the versions of an agent and information about each version
<code>list_data_sources</code>	Lists the data sources in a knowledge base and information about each one
<code>list_flow_aliases</code>	Returns a list of aliases for a flow
<code>list_flows</code>	Returns a list of flows and information about each flow
<code>list_flow_versions</code>	Returns a list of information about each flow
<code>list_ingestion_jobs</code>	Lists the data ingestion jobs for a data source
<code>list_knowledge_base_documents</code>	Retrieves all the documents contained in a data source that is connected to a knowledge base
<code>list_knowledge_bases</code>	Lists the knowledge bases in an account
<code>list_prompts</code>	Returns either information about the working draft (DRAFT version) of each prompt in your prompt library
<code>list_tags_for_resource</code>	List all the tags for the resource you specify
<code>prepare_agent</code>	Creates a DRAFT version of the agent that can be used for internal testing
<code>prepare_flow</code>	Prepares the DRAFT version of a flow so that it can be invoked
<code>start_ingestion_job</code>	Begins a data ingestion job
<code>stop_ingestion_job</code>	Stops a currently running data ingestion job
<code>tag_resource</code>	Associate tags with a resource
<code>untag_resource</code>	Remove tags from a resource
<code>update_agent</code>	Updates the configuration of an agent
<code>update_agent_action_group</code>	Updates the configuration for an action group for an agent
<code>update_agent_alias</code>	Updates configurations for an alias of an agent
<code>update_agent_collaborator</code>	Updates an agent's collaborator
<code>update_agent_knowledge_base</code>	Updates the configuration for a knowledge base that has been associated with an agent
<code>update_data_source</code>	Updates the configurations for a data source connector
<code>update_flow</code>	Modifies a flow
<code>update_flow_alias</code>	Modifies the alias of a flow
<code>update_knowledge_base</code>	Updates the configuration of a knowledge base with the fields that you specify
<code>update_prompt</code>	Modifies a prompt in your prompt library
<code>validate_flow_definition</code>	Validates the definition of a flow

Examples

```
## Not run:
svc <- bedrockagent()
svc$associate_agent_collaborator(
  Foo = 123
)

## End(Not run)
```

bedrockagentruntime *Agents for Amazon Bedrock Runtime*

Description

Contains APIs related to model invocation and querying of knowledge bases.

Usage

```
bedrockagentruntime(
    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 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 <- bedrockagentruntime(
  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_agent_memory	Deletes memory from the specified memory identifier
generate_query	Generates an SQL query from a natural language query
get_agent_memory	Gets the sessions stored in the memory of the agent

<code>invoke_agent</code>	Sends a prompt for the agent to process and respond to
<code>invoke_flow</code>	Invokes an alias of a flow to run the inputs that you specify and return the output of each node
<code>invoke_inline_agent</code>	Invokes an inline Amazon Bedrock agent using the configurations you provide with the request
<code>optimize_prompt</code>	Optimizes a prompt for the task that you specify
<code>rerank</code>	Reranks the relevance of sources based on queries
<code>retrieve</code>	Queries a knowledge base and retrieves information from it
<code>retrieve_and_generate</code>	Queries a knowledge base and generates responses based on the retrieved results and using the model
<code>retrieve_and_generate_stream</code>	Queries a knowledge base and generates responses based on the retrieved results, with output streaming

Examples

```
## Not run:
svc <- bedrockagentruntime()
svc$delete_agent_memory(
  Foo = 123
)

## End(Not run)
```

bedrockdataautomation	<i>Data Automation for Amazon Bedrock</i>
-----------------------	---

Description

Amazon Bedrock Data Automation BuildTime

Usage

```
bedrockdataautomation(
  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. <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 • 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 <- bedrockdataautomation(
  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_blueprint	Creates an Amazon Bedrock Data Automation Blueprint
create_blueprint_version	Creates a new version of an existing Amazon Bedrock Data Automation Blueprint
create_data_automation_project	Creates an Amazon Bedrock Data Automation Project
delete_blueprint	Deletes an existing Amazon Bedrock Data Automation Blueprint
delete_data_automation_project	Deletes an existing Amazon Bedrock Data Automation Project
get_blueprint	Gets an existing Amazon Bedrock Data Automation Blueprint
get_data_automation_project	Gets an existing Amazon Bedrock Data Automation Project
list_blueprints	Lists all existing Amazon Bedrock Data Automation Blueprints
list_data_automation_projects	Lists all existing Amazon Bedrock Data Automation Projects
update_blueprint	Updates an existing Amazon Bedrock Data Automation Blueprint
update_data_automation_project	Updates an existing Amazon Bedrock Data Automation Project

Examples

```

## Not run:
svc <- bedrockdataautomation()
svc$create_blueprint(
  Foo = 123
)

## End(Not run)

```

bedrockdataautomationruntime

Runtime for Amazon Bedrock Data Automation

Description

Amazon Bedrock Data Automation Runtime

Usage

```
bedrockdataautomationruntime(
    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 <- bedrockdataautomationruntime(
  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>get_data_automation_status</code>	API used to get data automation status
<code>invoke_data_automation_async</code>	Async API: Invoke data automation

Examples

```
## Not run:
svc <- bedrockdataautomationruntime()
svc$get_data_automation_status(
  Foo = 123
)

## End(Not run)
```

bedrockruntime	<i>Amazon Bedrock Runtime</i>
----------------	-------------------------------

Description

Describes the API operations for running inference using Amazon Bedrock models.

Usage

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

Arguments

<code>config</code>	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. <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 • 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 <- bedrockruntime(
  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

apply_guardrail	The action to apply a guardrail
converse	Sends messages to the specified Amazon Bedrock model
converse_stream	Sends messages to the specified Amazon Bedrock model and returns the response in a
get_async_invoke	Retrieve information about an asynchronous invocation
invoke_model	Invokes the specified Amazon Bedrock model to run inference using the prompt and i
invoke_model_with_response_stream	Invoke the specified Amazon Bedrock model to run inference using the prompt and in
list_async_invokes	Lists asynchronous invocations
start_async_invoke	Starts an asynchronous invocation

Examples

```
## Not run:
svc <- bedrockruntime()
svc$apply_guardrail(
  Foo = 123
)

## End(Not run)
```

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

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. • 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 • 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 <- 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 classification request to analyze a single document in real-time

[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

<code>delete_document_classifier</code>	Deletes a previously created document classifier
<code>delete_endpoint</code>	Deletes a model-specific endpoint for a previously-trained custom model
<code>delete_entity_recognizer</code>	Deletes an entity recognizer
<code>delete_flywheel</code>	Deletes a flywheel
<code>delete_resource_policy</code>	Deletes a resource-based policy that is attached to a custom model
<code>describe_dataset</code>	Returns information about the dataset that you specify
<code>describe_document_classification_job</code>	Gets the properties associated with a document classification job
<code>describe_document_classifier</code>	Gets the properties associated with a document classifier
<code>describe_dominant_language_detection_job</code>	Gets the properties associated with a dominant language detection job
<code>describe_endpoint</code>	Gets the properties associated with a specific endpoint
<code>describe_entities_detection_job</code>	Gets the properties associated with an entities detection job
<code>describe_entity_recognizer</code>	Provides details about an entity recognizer including status, S3 buckets contain
<code>describe_events_detection_job</code>	Gets the status and details of an events detection job
<code>describe_flywheel</code>	Provides configuration information about the flywheel
<code>describe_flywheel_iteration</code>	Retrieve the configuration properties of a flywheel iteration
<code>describe_key_phrases_detection_job</code>	Gets the properties associated with a key phrases detection job
<code>describe_pii_entities_detection_job</code>	Gets the properties associated with a PII entities detection job
<code>describe_resource_policy</code>	Gets the details of a resource-based policy that is attached to a custom model, i
<code>describe_sentiment_detection_job</code>	Gets the properties associated with a sentiment detection job
<code>describe_targeted_sentiment_detection_job</code>	Gets the properties associated with a targeted sentiment detection job
<code>describe_topics_detection_job</code>	Gets the properties associated with a topic detection job
<code>detect_dominant_language</code>	Determines the dominant language of the input text
<code>detect_entities</code>	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, N
<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>detect_toxic_content</code>	Performs toxicity analysis on the list of text strings that you provide as input
<code>import_model</code>	Creates a new custom model that replicates a source custom model that you im
<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 documentation classification jobs that you have submitted
<code>list_document_classifiers</code>	Gets a list of the document classifiers that you have created
<code>list_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
<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 using a custom classification
<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. • 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 • 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 <- 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 norm
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

<code>list_phi_detection_jobs</code>	Gets a list of protected health information (PHI) detection jobs you have submitted
<code>list_rx_norm_inference_jobs</code>	Gets a list of InferRxNorm jobs that you have submitted
<code>list_snomedct_inference_jobs</code>	Gets a list of InferSNOMEDCT jobs a user has submitted
<code>start_entities_detection_v2_job</code>	Starts an asynchronous medical entity detection job for a collection of documents
<code>start_icd10cm_inference_job</code>	Starts an asynchronous job to detect medical conditions and link them to the ICD-10-CM
<code>start_phi_detection_job</code>	Starts an asynchronous job to detect protected health information (PHI)
<code>start_rx_norm_inference_job</code>	Starts an asynchronous job to detect medication entities and link them to the RxNorm on
<code>start_snomedct_inference_job</code>	Starts an asynchronous job to detect medical concepts and link them to the SNOMED-C
<code>stop_entities_detection_v2_job</code>	Stops a medical entities detection job in progress
<code>stop_icd10cm_inference_job</code>	Stops an InferICD10CM inference job in progress
<code>stop_phi_detection_job</code>	Stops a protected health information (PHI) detection job in progress
<code>stop_rx_norm_inference_job</code>	Stops an InferRxNorm inference job in progress
<code>stop_snomedct_inference_job</code>	Stops an InferSNOMEDCT inference job in progress

Examples

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

## End(Not run)
```

forecastqueryservice *Amazon Forecast Query Service*

Description

Provides APIs for creating and managing Amazon Forecast resources.

Usage

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

Arguments

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

- **credentials:**

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

Value

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

Service syntax

```
svc <- 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)

```

forecastservice

Amazon Forecast Service

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

```

Operations

create_auto_predictor	Creates an Amazon Forecast predictor
create_dataset	Creates an Amazon Forecast dataset
create_dataset_group	Creates a dataset group, which holds a collection of related datasets
create_dataset_import_job	Imports your training data to an Amazon Forecast dataset
create_explainability	Explainability is only available for Forecasts and Predictors generated from an AutoPredictor
create_explainability_export	Exports an Explainability resource created by the CreateExplainability operation
create_forecast	Creates a forecast for each item in the TARGET_TIME_SERIES dataset that was used to train the predictor
create_forecast_export_job	Exports a forecast created by the CreateForecast operation to your Amazon Simple Storage Service bucket
create_monitor	Creates a predictor monitor resource for an existing auto predictor
create_predictor	This operation creates a legacy predictor that does not include all the predictor functions available in the Amazon Forecast console
create_predictor_backtest_export_job	Exports backtest forecasts and accuracy metrics generated by the CreateAutoPredictor operation
create_what_if_analysis	What-if analysis is a scenario modeling technique where you make a hypothetical change to the input data and see how it affects the forecast
create_what_if_forecast	A what-if forecast is a forecast that is created from a modified version of the baseline forecast

create_what_if_forecast_export	Exports a forecast created by the CreateWhatIfForecast operation to your Amazon Forecast resource
delete_dataset	Deletes an Amazon Forecast dataset that was created using the CreateDataset operation
delete_dataset_group	Deletes a dataset group created using the CreateDatasetGroup operation
delete_dataset_import_job	Deletes a dataset import job created using the CreateDatasetImportJob operation
delete_explainability	Deletes an Explainability resource
delete_explainability_export	Deletes an Explainability export
delete_forecast	Deletes a forecast created using the CreateForecast operation
delete_forecast_export_job	Deletes a forecast export job created using the CreateForecastExportJob operation
delete_monitor	Deletes a monitor resource
delete_predictor	Deletes a predictor created using the DescribePredictor or CreatePredictor operation
delete_predictor_backtest_export_job	Deletes a predictor backtest export job
delete_resource_tree	Deletes an entire resource tree
delete_what_if_analysis	Deletes a what-if analysis created using the CreateWhatIfAnalysis operation
delete_what_if_forecast	Deletes a what-if forecast created using the CreateWhatIfForecast operation
delete_what_if_forecast_export	Deletes a what-if forecast export created using the CreateWhatIfForecastExport operation
describe_auto_predictor	Describes a predictor created using the CreateAutoPredictor operation
describe_dataset	Describes an Amazon Forecast dataset created using the CreateDataset operation
describe_dataset_group	Describes a dataset group created using the CreateDatasetGroup operation
describe_dataset_import_job	Describes a dataset import job created using the CreateDatasetImportJob operation
describe_explainability	Describes an Explainability resource created using the CreateExplainability operation
describe_explainability_export	Describes an Explainability export created using the CreateExplainabilityExport operation
describe_forecast	Describes a forecast created using the CreateForecast operation
describe_forecast_export_job	Describes a forecast export job created using the CreateForecastExportJob operation
describe_monitor	Describes a monitor resource
describe_predictor	This operation is only valid for legacy predictors created with CreatePredictor
describe_predictor_backtest_export_job	Describes a predictor backtest export job created using the CreatePredictorBacktestExportJob operation
describe_what_if_analysis	Describes the what-if analysis created using the CreateWhatIfAnalysis operation
describe_what_if_forecast	Describes the what-if forecast created using the CreateWhatIfForecast operation
describe_what_if_forecast_export	Describes the what-if forecast export created using the CreateWhatIfForecastExport operation
get_accuracy_metrics	Provides metrics on the accuracy of the models that were trained by the CreatePredictor operation
list_dataset_groups	Returns a list of dataset groups created using the CreateDatasetGroup operation
list_dataset_import_jobs	Returns a list of dataset import jobs created using the CreateDatasetImportJob operation
list_datasets	Returns a list of datasets created using the CreateDataset operation
list_explainabilities	Returns a list of Explainability resources created using the CreateExplainability operation
list_explainability_exports	Returns a list of Explainability exports created using the CreateExplainabilityExport operation
list_forecast_export_jobs	Returns a list of forecast export jobs created using the CreateForecastExportJob operation
list_forecasts	Returns a list of forecasts created using the CreateForecast operation
list_monitor_evaluations	Returns a list of the monitoring evaluation results and predictor events collected by the CreateMonitor operation
list_monitors	Returns a list of monitors created with the CreateMonitor operation and CreateAutoPredictor
list_predictor_backtest_export_jobs	Returns a list of predictor backtest export jobs created using the CreatePredictorBacktestExportJob operation
list_predictors	Returns a list of predictors created using the CreateAutoPredictor or CreatePredictor operation
list_tags_for_resource	Lists the tags for an Amazon Forecast resource
list_what_if_analyses	Returns a list of what-if analyses created using the CreateWhatIfAnalysis operation
list_what_if_forecast_exports	Returns a list of what-if forecast exports created using the CreateWhatIfForecastExport operation
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](#)
[update_dataset_group](#)

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

Value

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

Service syntax

```
svc <- frauddetector(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
```

```

        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    anonymous = "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
<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

update_model_version	Updates a model version
update_model_version_status	Updates the status of a model version
update_rule_metadata	Updates a rule's metadata
update_rule_version	Updates a rule version resulting in a new rule version
update_variable	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

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

Value

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

Service syntax

```
svc <- lexmodelsv2(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
```

```

        secret_access_key = "string",
        session_token = "string"
    ),
    profile = "string",
    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_replica	Action to create a replication of the source bot in the secondary region
create_bot_version	Creates an immutable version of the bot
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

<code>delete_bot_replica</code>	The action to delete the replicated bot in the secondary region
<code>delete_bot_version</code>	Deletes a specific version of a bot
<code>delete_custom_vocabulary</code>	Removes a custom vocabulary from the specified locale in the specified bot
<code>delete_export</code>	Removes a previous export and the associated files stored in an S3 bucket
<code>delete_import</code>	Removes a previous import and the associated file stored in an S3 bucket
<code>delete_intent</code>	Removes the specified intent
<code>delete_resource_policy</code>	Removes an existing policy from a bot or bot alias
<code>delete_resource_policy_statement</code>	Deletes a policy statement from a resource policy
<code>delete_slot</code>	Deletes the specified slot from an intent
<code>delete_slot_type</code>	Deletes a slot type from a bot locale
<code>delete_test_set</code>	The action to delete the selected test set
<code>delete utterances</code>	Deletes stored utterances
<code>describe_bot</code>	Provides metadata information about a bot
<code>describe_bot_alias</code>	Get information about a specific bot alias
<code>describe_bot_locale</code>	Describes the settings that a bot has for a specific locale
<code>describe_bot_recommendation</code>	Provides metadata information about a bot recommendation
<code>describe_bot_replica</code>	Monitors the bot replication status through the UI console
<code>describe_bot_resource_generation</code>	Returns information about a request to generate a bot through natural language des
<code>describe_bot_version</code>	Provides metadata about a version of a bot
<code>describe_custom_vocabulary_metadata</code>	Provides metadata information about a custom vocabulary
<code>describe_export</code>	Gets information about a specific export
<code>describe_import</code>	Gets information about a specific import
<code>describe_intent</code>	Returns metadata about an intent
<code>describe_resource_policy</code>	Gets the resource policy and policy revision for a bot or bot alias
<code>describe_slot</code>	Gets metadata information about a slot
<code>describe_slot_type</code>	Gets metadata information about a slot type
<code>describe_test_execution</code>	Gets metadata information about the test execution
<code>describe_test_set</code>	Gets metadata information about the test set
<code>describe_test_set_discrepancy_report</code>	Gets metadata information about the test set discrepancy report
<code>describe_test_set_generation</code>	Gets metadata information about the test set generation
<code>generate_bot_element</code>	Generates sample utterances for an intent
<code>get_test_execution_artifacts_url</code>	The pre-signed Amazon S3 URL to download the test execution result artifacts
<code>list_aggregated_utterances</code>	Provides a list of utterances that users have sent to the bot
<code>list_bot_aliases</code>	Gets a list of aliases for the specified bot
<code>list_bot_alias_replicas</code>	The action to list the replicated bots created from the source bot alias
<code>list_bot_locales</code>	Gets a list of locales for the specified bot
<code>list_bot_recommendations</code>	Get a list of bot recommendations that meet the specified criteria
<code>list_bot_replicas</code>	The action to list the replicated bots
<code>list_bot_resource_generations</code>	Lists the generation requests made for a bot locale
<code>list_bots</code>	Gets a list of available bots
<code>list_bot_version_replicas</code>	Contains information about all the versions replication statuses applicable for Globa
<code>list_bot_versions</code>	Gets information about all of the versions of a bot
<code>list_built_in_intents</code>	Gets a list of built-in intents provided by Amazon Lex that you can use in your bot
<code>list_built_in_slot_types</code>	Gets a list of built-in slot types that meet the specified criteria
<code>list_custom_vocabulary_items</code>	Paginated list of custom vocabulary items for a given bot locale's custom vocabular
<code>list_exports</code>	Lists the exports for a bot, bot locale, or custom vocabulary
<code>list_imports</code>	Lists the imports for a bot, bot locale, or custom vocabulary
<code>list_intent_metrics</code>	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 to improve 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 ListUtteranceAnalyticsData action
list_utterance_metrics	To use this API operation, your IAM role must have permissions to perform the ListUtteranceMetrics action
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 process
start_bot_resource_generation	Starts a request for the descriptive bot builder to generate a bot locale configuration
start_import	Starts importing a bot, bot locale, or custom vocabulary from a zip archive that you upload
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.

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

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

Operations

<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 <- lexruntimev2()
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

<code>config</code>	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. <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 • 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 <- 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

<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
<code>put_session</code>	Creates a new session or modifies an existing session with an Amazon Lex V2 bot
<code>recognize_text</code>	Sends user input to Amazon Lex V2
<code>recognize_utterance</code>	Sends user input to Amazon Lex V2
<code>start_conversation</code>	Starts an HTTP/2 bidirectional event stream that enables you to send audio, text, or DTMF input in real

Examples

```

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

## End(Not run)

```

Description

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

Usage

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

Arguments

config	Optional configuration of credentials, endpoint, and/or region. • credentials: – creds: * access_key_id: AWS access key ID * secret_access_key: AWS secret access key * session_token: AWS temporary session token – profile: The name of a profile to use. If not given, then the default profile is used. – anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <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 • 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 <- 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 a machine learning model for data inference
create_retraining_scheduler	Creates a retraining scheduler on the specified model
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 a machine learning model currently available for Amazon Lookout for Equipment
delete_resource_policy	Deletes the resource policy attached to the resource
delete_retraining_scheduler	Deletes a retraining scheduler from a model

<code>describe_data_ingestion_job</code>	Provides information on a specific data ingestion job such as creation time, dataset ARN, and
<code>describe_dataset</code>	Provides a JSON description of the data in each time series dataset, including names, column
<code>describe_inference_scheduler</code>	Specifies information about the inference scheduler being used, including name, model, statu
<code>describe_label</code>	Returns the name of the label
<code>describe_label_group</code>	Returns information about the label group
<code>describe_model</code>	Provides a JSON containing the overall information about a specific machine learning model,
<code>describe_model_version</code>	Retrieves information about a specific machine learning model version
<code>describe_resource_policy</code>	Provides the details of a resource policy attached to a resource
<code>describe_retraining_scheduler</code>	Provides a description of the retraining scheduler, including information such as the model na
<code>import_dataset</code>	Imports a dataset
<code>import_model_version</code>	Imports a model that has been trained successfully
<code>list_data_ingestion_jobs</code>	Provides a list of all data ingestion jobs, including dataset name and ARN, S3 location of the
<code>list_datasets</code>	Lists all datasets currently available in your account, filtering on the dataset name
<code>list_inference_events</code>	Lists all inference events that have been found for the specified inference scheduler
<code>list_inference_executions</code>	Lists all inference executions that have been performed by the specified inference scheduler
<code>list_inference_schedulers</code>	Retrieves a list of all inference schedulers currently available for your account
<code>list_label_groups</code>	Returns a list of the label groups
<code>list_labels</code>	Provides a list of labels
<code>list_models</code>	Generates a list of all models in the account, including model name and ARN, dataset, and sta
<code>list_model_versions</code>	Generates a list of all model versions for a given model, including the model version, model v
<code>list_retraining_schedulers</code>	Lists all retraining schedulers in your account, filtering by model name prefix and status
<code>list_sensor_statistics</code>	Lists statistics about the data collected for each of the sensors that have been successfully ing
<code>list_tags_for_resource</code>	Lists all the tags for a specified resource, including key and value
<code>put_resource_policy</code>	Creates a resource control policy for a given resource
<code>start_data_ingestion_job</code>	Starts a data ingestion job
<code>start_inference_scheduler</code>	Starts an inference scheduler
<code>start_retraining_scheduler</code>	Starts a retraining scheduler
<code>stop_inference_scheduler</code>	Stops an inference scheduler
<code>stop_retraining_scheduler</code>	Stops a retraining scheduler
<code>tag_resource</code>	Associates a given tag to a resource in your account
<code>untag_resource</code>	Removes a specific tag from a given resource
<code>update_active_model_version</code>	Sets the active model version for a given machine learning model
<code>update_inference_scheduler</code>	Updates an inference scheduler
<code>update_label_group</code>	Updates the label group
<code>update_model</code>	Updates a model in the account
<code>update_retraining_scheduler</code>	Updates a retraining scheduler

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.

- **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 <- 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

<code>activate_anomaly_detector</code>	Activates an anomaly detector
<code>back_test_anomaly_detector</code>	Runs a backtest for anomaly detection for the specified resource
<code>create_alert</code>	Creates an alert for an anomaly detector
<code>create_anomaly_detector</code>	Creates an anomaly detector
<code>create_metric_set</code>	Creates a dataset
<code>deactivate_anomaly_detector</code>	Deactivates an anomaly detector
<code>delete_alert</code>	Deletes an alert
<code>delete_anomaly_detector</code>	Deletes a detector
<code>describe_alert</code>	Describes an alert
<code>describe_anomaly_detection_executions</code>	Returns information about the status of the specified anomaly detection jobs
<code>describe_anomaly_detector</code>	Describes a detector
<code>describe_metric_set</code>	Describes a dataset
<code>detect_metric_set_config</code>	Detects an Amazon S3 dataset's file format, interval, and offset
<code>get_anomaly_group</code>	Returns details about a group of anomalous metrics
<code>get_data_quality_metrics</code>	Returns details about the requested data quality metrics
<code>get_feedback</code>	Get feedback for an anomaly group
<code>get_sample_data</code>	Returns a selection of sample records from an Amazon S3 datasource
<code>list_alerts</code>	Lists the alerts attached to a detector
<code>list_anomaly_detectors</code>	Lists the detectors in the current AWS Region
<code>list_anomaly_group_related_metrics</code>	Returns a list of measures that are potential causes or effects of an anomaly group
<code>list_anomaly_group_summaries</code>	Returns a list of anomaly groups
<code>list_anomaly_group_time_series</code>	Gets a list of anomalous metrics for a measure in an anomaly group
<code>list_metric_sets</code>	Lists the datasets in the current AWS Region
<code>list_tags_for_resource</code>	Gets a list of tags for a detector, dataset, or alert
<code>put_feedback</code>	Add feedback for an anomalous metric
<code>tag_resource</code>	Adds tags to a detector, dataset, or alert
<code>untag_resource</code>	Removes tags from a detector, dataset, or alert
<code>update_alert</code>	Make changes to an existing alert
<code>update_anomaly_detector</code>	Updates a detector
<code>update_metric_set</code>	Updates a dataset

Examples

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

## End(Not run)
```

Description

Definition of the public APIs exposed by Amazon Machine Learning

Usage

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

Arguments

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

<code>list_devices</code>	Returns a list of devices
<code>list_devices_jobs</code>	Returns a list of jobs
<code>list_node_from_template_jobs</code>	Returns a list of camera stream node jobs
<code>list_nodes</code>	Returns a list of nodes
<code>list_package_import_jobs</code>	Returns a list of package import jobs
<code>list_packages</code>	Returns a list of packages
<code>list_tags_for_resource</code>	Returns a list of tags for a resource
<code>provision_device</code>	Creates a device and returns a configuration archive
<code>register_package_version</code>	Registers a package version
<code>remove_application_instance</code>	Removes an application instance
<code>signal_application_instance_node_instances</code>	Signal camera nodes to stop or resume
<code>tag_resource</code>	Tags a resource
<code>untag_resource</code>	Removes tags from a resource
<code>update_device_metadata</code>	Updates a device's metadata

Examples

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

## End(Not run)
```

personalize

Amazon Personalize

Description

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

Usage

```
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. • 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 • 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 <- 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	Generates batch recommendations based on a list of items or users stored in Amazon S3 and an Amazon Personalize solution
create_batch_segment_job	Creates a batch segment job
create_campaign	You incur campaign costs while it is active
create_data_deletion_job	Creates a batch job that deletes all references to specific users from an Amazon Personalize solution
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 solution
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	By default, all new solutions use automatic training
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

<code>delete_metric_attribution</code>	Deletes a metric attribution
<code>delete_recommender</code>	Deactivates and removes a recommender
<code>delete_schema</code>	Deletes a schema
<code>delete_solution</code>	Deletes all versions of a solution and the Solution object itself
<code>describe_algorithm</code>	Describes the given algorithm
<code>describe_batch_inference_job</code>	Gets the properties of a batch inference job including name, Amazon Resource Name (ARN)
<code>describe_batch_segment_job</code>	Gets the properties of a batch segment job including name, Amazon Resource Name (ARN)
<code>describe_campaign</code>	Describes the given campaign, including its status
<code>describe_data_deletion_job</code>	Describes the data deletion job created by <code>CreateDataDeletionJob</code> , including the job status
<code>describe_dataset</code>	Describes the given dataset
<code>describe_dataset_export_job</code>	Describes the dataset export job created by <code>CreateDatasetExportJob</code> , including the export job status
<code>describe_dataset_group</code>	Describes the given dataset group
<code>describe_dataset_import_job</code>	Describes the dataset import job created by <code>CreateDatasetImportJob</code> , including the import job status
<code>describe_event_tracker</code>	Describes an event tracker
<code>describe_feature_transformation</code>	Describes the given feature transformation
<code>describe_filter</code>	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_data_deletion_jobs</code>	Returns a list of data deletion jobs for a dataset group ordered by creation time, with the most recent first
<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 in a 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_PROGRESS
<code>tag_resource</code>	Add a list of tags to a resource
<code>untag_resource</code>	Removes the specified tags that are attached to a resource
<code>update_campaign</code>	Updates a campaign to deploy a retrained solution version with an existing campaign, changing the solution version
<code>update_dataset</code>	Update a dataset to replace its schema with a new or existing one

update_metric_attribution	Updates a metric attribution
update_recommender	Updates the recommender to modify the recommender configuration
update_solution	Updates an Amazon Personalize solution to use a different automatic training 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 item interaction events](#).

Usage

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

Arguments

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

	• endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to <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 • 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 <- 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_action_interactions	Records action interaction event data
put_actions	Adds one or more actions to an Actions dataset
put_events	Records item 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_action_interactions(
  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. • 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 • 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 <- 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_action_recommendations	Returns a list of recommended actions in sorted in descending order by prediction score
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_action_recommendations(
  Foo = 123
)

```

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

polly

Amazon Polly

Description

Amazon Polly is a web service that makes it easy to synthesize speech from text.

The Amazon Polly service provides API operations for synthesizing high-quality speech from plain text and Speech Synthesis Markup Language (SSML), along with managing pronunciations lexicons that enable you to get the best results for your application domain.

Usage

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

Arguments

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

- **credentials:**

- **creds:**

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

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

- **anonymous:** Set anonymous credentials.

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

- **region:** The AWS Region used in instantiating the client.

- **close_connection:** Immediately close all HTTP connections.

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

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

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

rekognition

Amazon Rekognition

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
- get_media_analysis_job
- index_faces
- list_collections
- ListMediaAnalysisJob
- list_faces
- list_users
- recognize_celebrities
- search_faces
- search_faces_by_image
- search_users
- search_users_by_image
- start_media_analysis_job

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

Value

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

Service syntax

```
svc <- rekognition(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
```

```

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

```

Operations

associate_faces	Associates one or more faces with an existing UserID
compare_faces	Compares a face in the source input image with each of the 100 largest faces detected in the target image
copy_project_version	This operation applies only to Amazon Rekognition Custom Labels
create_collection	Creates a collection in an AWS Region
create_dataset	This operation applies only to Amazon Rekognition Custom Labels
create_face_liveness_session	This API operation initiates a Face Liveness session
create_project	Creates a new Amazon Rekognition project
create_project_version	Creates a new version of Amazon Rekognition project (like a Custom Labels model or a classifier)
create_stream_processor	Creates an Amazon Rekognition stream processor that you can use to detect and recognize faces in real time
create_user	Creates a new User within a collection specified by CollectionId
delete_collection	Deletes the specified collection
delete_dataset	This operation applies only to Amazon Rekognition Custom Labels
delete_faces	Deletes faces from a collection
delete_project	Deletes a Amazon Rekognition project
delete_project_policy	This operation applies only to Amazon Rekognition Custom Labels
delete_project_version	Deletes a Rekognition project model or project version, like a Amazon Rekognition Custom Labels model or classifier
delete_stream_processor	Deletes the stream processor identified by Name
delete_user	Deletes the specified UserID within the collection
describe_collection	Describes the specified collection
describe_dataset	This operation applies only to Amazon Rekognition Custom Labels

<code>describe_projects</code>	Gets information about your Rekognition projects
<code>describe_project_versions</code>	Lists and describes the versions of an Amazon Rekognition project
<code>describe_stream_processor</code>	Provides information about a stream processor created by <code>CreateStreamProcessor</code>
<code>detect_custom_labels</code>	This operation applies only to Amazon Rekognition Custom Labels
<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 <code>FaceIds</code> and the User
<code>distribute_dataset_entries</code>	This operation applies only to Amazon Rekognition Custom Labels
<code>get_celebrity_info</code>	Gets the name and additional information about a celebrity based on their Amazon Rekognition ID
<code>get_celebrity_recognition</code>	Gets the celebrity recognition results for a Amazon Rekognition Video analysis started by <code>StartCelebrityRecognition</code>
<code>get_content_moderation</code>	Gets the inappropriate, unwanted, or offensive content analysis results for a Amazon Rekognition Video analysis started by <code>StartContentModeration</code>
<code>get_face_detection</code>	Gets face detection results for a Amazon Rekognition Video analysis started by <code>StartFaceDetection</code>
<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 <code>StartFaceSearch</code>
<code>get_label_detection</code>	Gets the label detection results of a Amazon Rekognition Video analysis started by <code>StartLabelDetection</code>
<code>get_media_analysis_job</code>	Retrieves the results for a given media analysis job
<code>get_person_tracking</code>	Gets the path tracking results of a Amazon Rekognition Video analysis started by <code>StartPersonTracking</code>
<code>get_segment_detection</code>	Gets the segment detection results of a Amazon Rekognition Video analysis started by <code>StartSegmentDetection</code>
<code>get_text_detection</code>	Gets the text detection results of a Amazon Rekognition Video analysis started by <code>StartTextDetection</code>
<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>	This operation applies only to Amazon Rekognition Custom Labels
<code>list_dataset_labels</code>	This operation applies only to Amazon Rekognition Custom Labels
<code>list_faces</code>	Returns metadata for faces in the specified collection
<code>list_media_analysis_jobs</code>	Returns a list of media analysis jobs
<code>list_project_policies</code>	This operation applies only to Amazon Rekognition Custom Labels
<code>list_stream_processors</code>	Gets a list of stream processors that you have created with <code>CreateStreamProcessor</code>
<code>list_tags_for_resource</code>	Returns a list of tags in an Amazon Rekognition collection, stream processor, or Custom Label
<code>list_users</code>	Returns metadata of the User such as <code>UserID</code> in the specified collection
<code>put_project_policy</code>	This operation applies only to Amazon Rekognition Custom Labels
<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 specified collection for matching faces
<code>search_users</code>	Searches for <code>UserID</code> s within a collection based on a <code>FaceId</code> or <code>UserId</code>
<code>search_users_by_image</code>	Searches for <code>UserID</code> s 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 video
<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 detected in the input image
<code>start_label_detection</code>	Starts asynchronous detection of labels in a stored video
<code>start_media_analysis_job</code>	Initiates a new media analysis job
<code>start_person_tracking</code>	Starts the asynchronous tracking of a person's path in a stored video
<code>start_project_version</code>	This operation applies only to Amazon Rekognition Custom Labels
<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

start_text_detection	Starts asynchronous detection of text in a stored video
stop_project_version	This operation applies only to Amazon Rekognition Custom Labels
stop_stream_processor	Stops a running stream processor that was created by CreateStreamProcessor
tag_resource	Adds one or more key-value tags to an Amazon Rekognition collection, stream processor,
untag_resource	Removes one or more tags from an Amazon Rekognition collection, stream processor, or
update_dataset_entries	This operation applies only to Amazon Rekognition Custom Labels
update_stream_processor	Allows you to update a stream processor

Examples

```
## Not run:
svc <- rekognition()
# This operation compares the largest face detected in the source image
# with each face detected in the target image.
svc$compare_faces(
  SimilarityThreshold = 90L,
  SourceImage = list(
    S3Object = list(
      Bucket = "mybucket",
      Name = "mysourceimage"
    )
  ),
  TargetImage = list(
    S3Object = list(
      Bucket = "mybucket",
      Name = "mytargetimage"
    )
  )
)

## 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. <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 • 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

add_association	Creates an association between the source and the destination
add_tags	Adds or overwrites one or more tags for the specified SageMaker resource
associate_trial_component	Associates a trial component with a trial
batch_delete_cluster_nodes	Deletes specific nodes within a SageMaker HyperPod cluster
batch_describe_model_package	This action batch describes a list of versioned model packages
create_action	Creates an action
create_algorithm	Create a machine learning algorithm that you can use in SageMaker and Amazon SageMaker
create_app	Creates a running app for the specified UserProfile
create_app_image_config	Creates a configuration for running a SageMaker AI image as a KernelSpec
create_artifact	Creates an artifact
create_auto_ml_job	Creates an Autopilot job also referred to as Autopilot experiment or AutoML job
create_auto_ml_job_v2	Creates an Autopilot job also referred to as Autopilot experiment or AutoML job
create_cluster	Creates a SageMaker HyperPod cluster

<code>create_cluster_scheduler_config</code>	Create cluster policy configuration
<code>create_code_repository</code>	Creates a Git repository as a resource in your SageMaker AI account
<code>create_compilation_job</code>	Starts a model compilation job
<code>create_compute_quota</code>	Create compute allocation definition
<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
<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_hub_content_reference</code>	Create a hub content reference in order to add a model in the JumpStart console
<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 AI image
<code>create_image_version</code>	Creates a version of the SageMaker AI image specified by ImageName
<code>create_inference_component</code>	Creates an inference component, which is a SageMaker AI hosting object
<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_mlflow_tracking_server</code>	Creates an MLflow Tracking Server using a general purpose Amazon SageMaker instance
<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 AI Processing Jobs
<code>create_notebook_instance</code>	Creates an SageMaker AI notebook instance
<code>create_notebook_instance_lifecycle_config</code>	Creates a lifecycle configuration that you can associate with a notebook instance
<code>create_optimization_job</code>	Creates a job that optimizes a model for inference performance
<code>create_partner_app</code>	Creates an Amazon SageMaker Partner AI App
<code>create_partner_app_presigned_url</code>	Creates a presigned URL to access an Amazon SageMaker Partner AI App
<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_mlflow_tracking_server_url</code>	Returns a presigned URL that you can use to connect to the MLflow UI
<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 training jobs
<code>create_space</code>	Creates a private space or a space used for real time collaboration in a domain

<code>create_studio_lifecycle_config</code>	Creates a new Amazon SageMaker AI Studio Lifecycle Configuration
<code>create_training_job</code>	Starts a model training job
<code>create_training_plan</code>	Creates a new training plan in SageMaker to reserve compute capacity
<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_cluster</code>	Delete a SageMaker HyperPod cluster
<code>delete_cluster_scheduler_config</code>	Deletes the cluster policy of the cluster
<code>delete_code_repository</code>	Deletes the specified Git repository from your account
<code>delete_compilation_job</code>	Deletes the specified compilation job
<code>delete_compute_quota</code>	Deletes the compute allocation from the cluster
<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 plan
<code>delete_edge_deployment_stage</code>	Delete a stage in an edge deployment plan if (and only if) the stage is in the plan
<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 OnlineStore
<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_hub_content_reference</code>	Delete a hub content reference in order to remove a model from a private model registry
<code>delete_human_task_ui</code>	Use this operation to delete a human task user interface (worker task template)
<code>delete_hyper_parameter_tuning_job</code>	Deletes a hyperparameter tuning job
<code>delete_image</code>	Deletes a SageMaker AI image and all versions of the image
<code>delete_image_version</code>	Deletes a version of a SageMaker AI image
<code>delete_inference_component</code>	Deletes an inference component
<code>delete_inference_experiment</code>	Deletes an inference experiment
<code>delete_mlflow_tracking_server</code>	Deletes an MLflow Tracking Server
<code>delete_model</code>	Deletes a model
<code>delete_model_bias_job_definition</code>	Deletes an Amazon SageMaker AI 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 AI 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 specified model quality monitoring job definition
<code>delete_monitoring_schedule</code>	Deletes a monitoring schedule
<code>delete_notebook_instance</code>	Deletes an SageMaker AI notebook instance
<code>delete_notebook_instance_lifecycle_config</code>	Deletes a notebook instance lifecycle configuration
<code>delete_optimization_job</code>	Deletes an optimization job
<code>delete_partner_app</code>	Deletes a SageMaker Partner AI App
<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 Amazon SageMaker AI 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 <code>CreateAutoMLJob</code>
<code>describe_auto_ml_job_v2</code>	Returns information about an AutoML job created by calling <code>CreateAutoMLJobV2</code>
<code>describe_cluster</code>	Retrieves information of a SageMaker HyperPod cluster
<code>describe_cluster_node</code>	Retrieves information of a node (also called a instance interchangeably)
<code>describe_cluster_scheduler_config</code>	Description of the cluster policy
<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_compute_quota</code>	Description of the compute allocation definition
<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 <code>CreateEndpointConfig</code> operation
<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>	Describes 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 interface)
<code>describe_hyper_parameter_tuning_job</code>	Returns a description of a hyperparameter tuning job, depending on the <code>HyperParameterTuningJobName</code> parameter
<code>describe_image</code>	Describes a SageMaker AI image

<code>describe_image_version</code>	Describes a version of a SageMaker AI image
<code>describe_inference_component</code>	Returns information about an inference component
<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_mlflow_tracking_server</code>	Returns information about an MLflow Tracking Server
<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 model
<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_optimization_job</code>	Provides the properties of the specified optimization job
<code>describe_partner_app</code>	Gets information about a SageMaker Partner AI App
<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 Amazon SageMaker AI 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_training_plan</code>	Retrieves detailed information about a specific training plan
<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 trial's component's properties
<code>describe_user_profile</code>	Describes a user profile
<code>describe_workforce</code>	Lists private workforce information, including workforce name, Amazon SageMaker AI Studio, and SageMaker AI Studio
<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 recommendation
<code>get_search_suggestions</code>	An auto-complete API for the search functionality in the SageMaker console
<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_cluster_nodes</code>	Retrieves the list of instances (also called nodes interchangeably) in a S
<code>list_clusters</code>	Retrieves the list of SageMaker HyperPod clusters
<code>list_cluster_scheduler_configs</code>	List the cluster policy configurations
<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_compute_quotas</code>	List the resource allocation definitions
<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
<code>list_experiments</code>	Lists all the experiments in your account
<code>list_feature_groups</code>	List FeatureGroups based on given filter and order
<code>list_flow_definitions</code>	Returns information about the flow definitions in your account
<code>list_hub_contents</code>	List the contents of a hub
<code>list_hub_content_versions</code>	List hub content versions
<code>list_hubs</code>	List all existing hubs
<code>list_human_task_uis</code>	Returns information about the human task user interfaces in your accoun
<code>list_hyper_parameter_tuning_jobs</code>	Gets a list of HyperParameterTuningJobSummary objects that describe
<code>list_images</code>	Lists the images in your account and their properties
<code>list_image_versions</code>	Lists the versions of a specified image and their properties
<code>list_inference_components</code>	Lists the inference components in your account and their properties
<code>list_inference_experiments</code>	Returns the list of all inference experiments
<code>list_inference_recommendations_jobs</code>	Lists recommendation jobs that satisfy various filters
<code>list_inference_recommendations_job_steps</code>	Returns a list of the subtasks for an Inference Recommender job
<code>list_labeling_jobs</code>	Gets a list of labeling jobs
<code>list_labeling_jobs_for_workteam</code>	Gets a list of labeling jobs assigned to a specified work team
<code>list_lineage_groups</code>	A list of lineage groups shared with your Amazon Web Services account
<code>list_mlflow_tracking_servers</code>	Lists all MLflow Tracking Servers
<code>list_model_bias_job_definitions</code>	Lists model bias jobs definitions that satisfy various filters
<code>list_model_card_export_jobs</code>	List the export jobs for the Amazon SageMaker Model Card
<code>list_model_cards</code>	List existing model cards
<code>list_model_card_versions</code>	List existing versions of an Amazon SageMaker Model Card
<code>list_model_explainability_job_definitions</code>	Lists model explainability job definitions that satisfy various filters
<code>list_model_metadata</code>	Lists the domain, framework, task, and model name of standard machin
<code>list_model_package_groups</code>	Gets a list of the model groups in your Amazon Web Services account
<code>list_model_packages</code>	Lists the model packages that have been created

<code>list_model_quality_job_definitions</code>	Gets a list of model quality monitoring job definitions in your account
<code>list_models</code>	Lists models created with the CreateModel API
<code>list_monitoring_alert_history</code>	Gets a list of past alerts in a model monitoring schedule
<code>list_monitoring_alerts</code>	Gets the alerts for a single monitoring schedule
<code>list_monitoring_executions</code>	Returns list of all monitoring job executions
<code>list_monitoring_schedules</code>	Returns list of all monitoring schedules
<code>list_notebook_instance_lifecycle_configs</code>	Lists notebook instance lifestyle configurations created with the CreateNotebookInstanceLifecycleConfig API
<code>list_notebook_instances</code>	Returns a list of the SageMaker AI notebook instances in the requester's account
<code>list_optimization_jobs</code>	Lists the optimization jobs in your account and their properties
<code>list_partner_apps</code>	Lists all of the SageMaker Partner AI Apps in an account
<code>list_pipeline_executions</code>	Gets a list of the pipeline executions
<code>list_pipeline_execution_steps</code>	Gets a list of PipelineExecutionStep objects
<code>list_pipeline_parameters_for_execution</code>	Gets a list of parameters for a pipeline execution
<code>list_pipelines</code>	Gets a list of pipelines
<code>list_processing_jobs</code>	Lists processing jobs that satisfy various filters
<code>list_projects</code>	Gets a list of the projects in an Amazon Web Services account
<code>list_resource_catalogs</code>	Lists Amazon SageMaker Catalogs based on given filters and orders
<code>list_spaces</code>	Lists spaces
<code>list_stage_devices</code>	Lists devices allocated to the stage, containing detailed device information
<code>list_studio_lifecycle_configs</code>	Lists the Amazon SageMaker AI Studio Lifecycle Configurations in your account
<code>list_subscribed_workteams</code>	Gets a list of the work teams that you are subscribed to in the Amazon VPC
<code>list_tags</code>	Returns the tags for the specified SageMaker resource
<code>list_training_jobs</code>	Lists training jobs
<code>list_training_jobs_for_hyper_parameter_tuning_job</code>	Gets a list of TrainingJobSummary objects that describe the training jobs
<code>list_training_plans</code>	Retrieves a list of training plans for the current account
<code>list_transform_jobs</code>	Lists transform jobs
<code>list_trial_components</code>	Lists the trial components in your account
<code>list_trials</code>	Lists the trials in your account
<code>list_user_profiles</code>	Lists user profiles
<code>list_workforces</code>	Use this operation to list all private and vendor workforces in an Amazon VPC
<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 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>search_training_plan_offerings</code>	Searches for available training plan offerings based on specified criteria
<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 the step is complete
<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_mlflow_tracking_server</code>	Programmatically start an MLflow Tracking Server
<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_mlflow_tracking_server</code>	Programmatically stop an MLflow Tracking Server
<code>stop_monitoring_schedule</code>	Stops a previously started monitoring schedule
<code>stop_notebook_instance</code>	Terminates the ML compute instance
<code>stop_optimization_job</code>	Ends a running inference optimization job
<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_cluster</code>	Updates a SageMaker HyperPod cluster
<code>update_cluster_scheduler_config</code>	Update the cluster policy configuration
<code>update_cluster_software</code>	Updates the platform software of a SageMaker HyperPod cluster for ses
<code>update_code_repository</code>	Updates the specified Git repository with the specified values
<code>update_compute_quota</code>	Update the compute allocation definition
<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 EndpointConfig specified in the request to a new fleet of in
<code>update_endpoint_weights_and_capacities</code>	Updates variant weight of one or more variants associated with an existi
<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 onli
<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 AI image
<code>update_image_version</code>	Updates the properties of a SageMaker AI image version
<code>update_inference_component</code>	Updates an inference component
<code>update_inference_component_runtime_config</code>	Runtime settings for a model that is deployed with an inference compon
<code>update_inference_experiment</code>	Updates an inference experiment that you created
<code>update_mlflow_tracking_server</code>	Updates properties of an existing MLflow Tracking Server
<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_partner_app</code>	Updates all of the SageMaker Partner AI Apps in an account
<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 templat

<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.

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

Value

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

Service syntax

```
svc <- sagemakeredgemanager(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        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. <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 • 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 <- 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)
```

sagemakergeospatialcapabilities

Amazon SageMaker geospatial capabilities

Description

Provides APIs for creating and managing SageMaker geospatial resources.

Usage

```
sagemakergeospatialcapabilities(
  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. <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 • 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 <- sagemakergeospatialcapabilities(
  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_earth_observation_job</code>	Use this operation to delete an Earth Observation job
<code>delete_vector_enrichment_job</code>	Use this operation to delete a Vector Enrichment job
<code>export_earth_observation_job</code>	Use this operation to export results of an Earth Observation job and optionally source images
<code>export_vector_enrichment_job</code>	Use this operation to copy results of a Vector Enrichment job to an Amazon S3 location
<code>get_earth_observation_job</code>	Get the details for a previously initiated Earth Observation job
<code>get_raster_data_collection</code>	Use this operation to get details of a specific raster data collection
<code>get_tile</code>	Gets a web mercator tile for the given Earth Observation job
<code>get_vector_enrichment_job</code>	Retrieves details of a Vector Enrichment Job for a given job Amazon Resource Name (ARN)
<code>list_earth_observation_jobs</code>	Use this operation to get a list of the Earth Observation jobs associated with the calling Amazon Resource Name (ARN)
<code>list_raster_data_collections</code>	Use this operation to get raster data collections
<code>list_tags_for_resource</code>	Lists the tags attached to the resource
<code>list_vector_enrichment_jobs</code>	Retrieves a list of vector enrichment jobs
<code>search_raster_data_collection</code>	Allows you run image query on a specific raster data collection to get a list of the satellite images
<code>start_earth_observation_job</code>	Use this operation to create an Earth observation job
<code>start_vector_enrichment_job</code>	Creates a Vector Enrichment job for the supplied job type
<code>stop_earth_observation_job</code>	Use this operation to stop an existing earth observation job
<code>stop_vector_enrichment_job</code>	Stops the Vector Enrichment job for a given job ARN
<code>tag_resource</code>	The resource you want to tag
<code>untag_resource</code>	The resource you want to untag

Examples

```

## Not run:
svc <- sagemakergeospatialcapabilities()
svc$delete_earth_observation_job(
  Foo = 123
)

## End(Not run)

```

Description

Contains all data plane API operations and data types for Amazon SageMaker Metrics. Use these APIs to put and retrieve (get) features related to your training run.

- [batch_put_metrics](#)

Usage

```
sagemakermetrics(
    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. <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 • 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 <- sagemakermetrics(
  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_metrics	Used to retrieve training metrics from SageMaker
batch_put_metrics	Used to ingest training metrics into SageMaker

Examples

```
## Not run:
```

```

svc <- sagemakermetrics()
svc$batch_get_metrics(
  Foo = 123
)

## End(Not run)

```

sagemakerruntime

Amazon SageMaker Runtime

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.

- **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 <- 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	<i>Amazon Textract</i>
----------	------------------------

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

Value

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

Service syntax

```
svc <- 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 relations
analyze_id	Analyzes identity documents for relevant information
create_adapter	Creates an adapter, which can be fine-tuned for enhanced performance on user provided documents
create_adapter_version	Creates a new version of an adapter
delete_adapter	Deletes an Amazon Textract adapter
delete_adapter_version	Deletes an Amazon Textract adapter version
detect_document_text	Detects text in the input document
get_adapter	Gets configuration information for an adapter specified by an AdapterId, returning information about the adapter
get_adapter_version	Gets configuration information for the specified adapter version, including: AdapterId, AdapterVersion, and AdapterName
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 for data like contact information, items, and totals
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
list_adapters	Lists all adapters that match the specified filtration criteria
list_adapter_versions	List all version of an adapter that meet the specified filtration criteria
list_tags_for_resource	Lists all tags for an Amazon Textract resource
start_document_analysis	Starts the asynchronous analysis of an input document for relationships between detected items
start_document_text_detection	Starts the asynchronous detection of text in a document
start_expense_analysis	Starts the asynchronous analysis of invoices or receipts for data like contact information, items, and totals
start_lending_analysis	Starts the classification and analysis of an input document
tag_resource	Adds one or more tags to the specified resource
untag_resource	Removes any tags with the specified keys from the specified resource
update_adapter	Update the configuration for an adapter

Examples

```

## Not run:
svc <- textract()
svc$analyze_document(

```

```

    Foo = 123
)

## End(Not run)

```

transcribesservice	<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

```

transcribesservice(
  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. <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 • 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 <- transcribesservice(
  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>create_call_analytics_category</code>	Creates a new Call Analytics category
<code>create_language_model</code>	Creates a new custom language model
<code>create_medical_vocabulary</code>	Creates a new custom medical vocabulary
<code>create_vocabulary</code>	Creates a new custom vocabulary
<code>create_vocabulary_filter</code>	Creates a new custom vocabulary filter
<code>delete_call_analytics_category</code>	Deletes a Call Analytics category
<code>delete_call_analytics_job</code>	Deletes a Call Analytics job
<code>delete_language_model</code>	Deletes a custom language model
<code>delete_medical_scribe_job</code>	Deletes a Medical Scribe job
<code>delete_medical_transcription_job</code>	Deletes a medical transcription job
<code>delete_medical_vocabulary</code>	Deletes a custom medical vocabulary
<code>delete_transcription_job</code>	Deletes a transcription job
<code>delete_vocabulary</code>	Deletes a custom vocabulary
<code>delete_vocabulary_filter</code>	Deletes a custom vocabulary filter
<code>describe_language_model</code>	Provides information about the specified custom language model
<code>get_call_analytics_category</code>	Provides information about the specified Call Analytics category
<code>get_call_analytics_job</code>	Provides information about the specified Call Analytics job
<code>get_medical_scribe_job</code>	Provides information about the specified Medical Scribe job
<code>get_medical_transcription_job</code>	Provides information about the specified medical transcription job
<code>get_medical_vocabulary</code>	Provides information about the specified custom medical vocabulary
<code>get_transcription_job</code>	Provides information about the specified transcription job
<code>get_vocabulary</code>	Provides information about the specified custom vocabulary
<code>get_vocabulary_filter</code>	Provides information about the specified custom vocabulary filter
<code>list_call_analytics_categories</code>	Provides a list of Call Analytics categories, including all rules that make up each category
<code>list_call_analytics_jobs</code>	Provides a list of Call Analytics jobs that match the specified criteria
<code>list_language_models</code>	Provides a list of custom language models that match the specified criteria
<code>list_medical_scribe_jobs</code>	Provides a list of Medical Scribe jobs that match the specified criteria
<code>list_medical_transcription_jobs</code>	Provides a list of medical transcription jobs that match the specified criteria
<code>list_medical_vocabularies</code>	Provides a list of custom medical vocabularies that match the specified criteria
<code>list_tags_for_resource</code>	Lists all tags associated with the specified transcription job, vocabulary, model, or resource
<code>list_transcription_jobs</code>	Provides a list of transcription jobs that match the specified criteria
<code>list_vocabularies</code>	Provides a list of custom vocabularies that match the specified criteria
<code>list_vocabulary_filters</code>	Provides a list of custom vocabulary filters that match the specified criteria
<code>start_call_analytics_job</code>	Transcribes the audio from a customer service call and applies any additional Request Parameters
<code>start_medical_scribe_job</code>	Transcribes patient-clinician conversations and generates clinical notes
<code>start_medical_transcription_job</code>	Transcribes the audio from a medical dictation or conversation and applies any additional

start_transcription_job	Transcribes the audio from a media file and applies any additional Request Parameters you
tag_resource	Adds one or more custom tags, each in the form of a key:value pair, to the specified resour
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
 - * **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-endpoints.html
credentials	Optional credentials shorthand for the config parameter • creds: – access_key_id: AWS access key ID – secret_access_key: AWS secret access key – session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
endpoint	Optional shorthand for complete URL to use for the constructed client.
region	Optional shorthand for AWS Region used in instantiating the client.

Value

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

Service syntax

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

```

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

```

Operations

create_parallel_data	Creates a parallel data resource in Amazon Translate by importing an input file from Amazon
delete_parallel_data	Deletes a parallel data resource in Amazon Translate
delete_terminology	A synchronous action that deletes a custom terminology
describe_text_translation_job	Gets the properties associated with an asynchronous batch translation job including name, ID,
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. <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 • 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](#)
[create_domain](#)
[create_watchlist](#)
[delete_domain](#)
[delete_fraudster](#)
[delete_speaker](#)
[delete_watchlist](#)
[describe_domain](#)

Associates the fraudsters with the watchlist specified in the same domain
 Creates a domain that contains all Amazon Connect Voice ID data, such as speakers, fraudsters, and watchlists
 Creates a watchlist that fraudsters can be a part of
 Deletes the specified domain from Voice ID
 Deletes the specified fraudster from Voice ID
 Deletes the specified speaker from Voice ID
 Deletes the specified watchlist from Voice ID
 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 Amazon Connect session
<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, 58
add_association, 83
add_tags, 60, 83
analyze_document, 105
analyze_expense, 105
analyze_id, 105
apply_guardrail, 23
associate_agent_collaborator, 12
associate_agent_knowledge_base, 12
associate_faces, 75, 79
associate_fraudster, 113
associate_trial_component, 83
augmentedairuntime, 4

back_test_anomaly_detector, 58
batch_create_custom_vocabulary_item, 45
batch_create_variable, 38
batch_delete_cluster_nodes, 83
batch_delete_custom_vocabulary_item, 45
batch_delete_evaluation_job, 8
batch_describe_model_package, 83
batch_detect_dominant_language, 25
batch_detect_entities, 25
batch_detect_key_phrases, 25
batch_detect_sentiment, 25
batch_detect_syntax, 25
batch_detect_targeted_sentiment, 25
batch_get_metrics, 100
batch_get_record, 95
batch_get_variable, 38
batch_put_metrics, 99, 100
batch_update_custom_vocabulary_item, 45
bedrock, 6
bedrockagent, 10
bedrockagentruntime, 14
bedrockdataautomation, 16
bedrockdataautomationruntime, 19
bedrockruntime, 21
build_bot_locale, 45

cancel_batch_import_job, 38
cancel_batch_prediction_job, 38
classify_document, 25
compare_faces, 75, 79
comprehend, 23
comprehendmedical, 27
contains_pii_entities, 25
converse, 23
converse_stream, 23
copy_project_version, 76, 79
create_action, 83
create_adapter, 105
create_adapter_version, 105
create_agent, 12
create_agent_action_group, 12
create_agent_alias, 12
create_alert, 58
create_algorithm, 83
create_anomaly_detector, 58
create_app, 83
create_app_image_config, 83
create_application_instance, 63
create_artifact, 83
create_auto_ml_job, 83
create_auto_ml_job_v2, 83
create_auto_predictor, 34
create_batch_import_job, 38
create_batch_inference_job, 66
create_batch_prediction, 60
create_batch_prediction_job, 38
create_batch_segment_job, 66
create_blueprint, 18
create_blueprint_version, 18
create_bot, 45
create_bot_alias, 45
create_bot_locale, 45
create_bot_replica, 45

create_bot_version, [42, 45](#)
create_call_analytics_category, [108](#)
create_campaign, [66](#)
create_cluster, [83](#)
create_cluster_scheduler_config, [84](#)
create_code_repository, [84](#)
create_collection, [75, 79](#)
create_compilation_job, [84](#)
create_compute_quota, [84](#)
create_context, [84](#)
create_data_automation_project, [18](#)
create_data_deletion_job, [66](#)
create_data_quality_job_definition, [84](#)
create_data_source, [12](#)
create_data_source_from_rds, [60](#)
create_data_source_from_redshift, [60](#)
create_data_source_from_s3, [60](#)
create_dataset, [25, 34, 54, 66, 76, 79](#)
create_dataset_export_job, [66](#)
create_dataset_group, [34, 66](#)
create_dataset_import_job, [34, 66](#)
create_detector_version, [38](#)
create_device_fleet, [84](#)
create_document_classifier, [25](#)
create_domain, [84, 113](#)
create_edge_deployment_plan, [84](#)
create_edge_deployment_stage, [84](#)
create_edge_packaging_job, [84](#)
create_endpoint, [25, 84](#)
create_endpoint_config, [84](#)
create_entity_recognizer, [25](#)
create_evaluation, [60](#)
create_evaluation_job, [8](#)
create_event_tracker, [66](#)
create_experiment, [84](#)
create_explainability, [34](#)
create_explainability_export, [34](#)
create_export, [45](#)
create_face_liveness_session, [79](#)
create_feature_group, [84](#)
create_filter, [66](#)
create_flow, [12](#)
create_flow_alias, [12](#)
create_flow_definition, [84](#)
create_flow_version, [12](#)
create_flywheel, [25](#)
create_forecast, [34](#)
create_forecast_export_job, [34](#)
create_guardrail, [8](#)
create_guardrail_version, [8](#)
create_hub, [84](#)
create_hub_content_reference, [84](#)
create_human_task_ui, [84](#)
create_hyper_parameter_tuning_job, [84](#)
create_image, [84](#)
create_image_version, [84](#)
create_inference_component, [84](#)
create_inference_experiment, [84](#)
create_inference_profile, [8](#)
create_inference_recommendations_job, [84](#)
create_inference_scheduler, [54](#)
create_intent, [45](#)
create_intent_version, [42](#)
create_job_for_devices, [63](#)
create_knowledge_base, [12](#)
create_label, [54](#)
create_label_group, [54](#)
create_labeling_job, [84](#)
create_language_model, [108](#)
create_list, [38](#)
create_marketplace_model_endpoint, [8](#)
create_medical_vocabulary, [108](#)
create_metric_attribution, [66](#)
create_metric_set, [58](#)
create_ml_model, [60](#)
create_mlflow_tracking_server, [84](#)
create_model, [38, 54, 84](#)
create_model_bias_job_definition, [84](#)
create_model_card, [84](#)
create_model_card_export_job, [84](#)
create_model_copy_job, [8](#)
create_model_customization_job, [8](#)
create_model_explainability_job_definition, [84](#)
create_model_import_job, [8](#)
create_model_invocation_job, [8](#)
create_model_package, [84](#)
create_model_package_group, [84](#)
create_model_quality_job_definition, [84](#)
create_model_version, [38](#)
create_monitor, [34](#)
create_monitoring_schedule, [84](#)
create_node_from_template_job, [63](#)
create_notebook_instance, [84](#)

- create_notebook_instance_lifecycle_config, 84
- create_optimization_job, 84
- create_package, 63
- create_package_import_job, 63
- create_parallel_data, 111
- create_partner_app, 84
- create_partner_app_presigned_url, 84
- create_pipeline, 84
- create_predictor, 34
- create_predictor_backtest_export_job, 34
- create_presigned_domain_url, 84
- create_presigned_mlflow_tracking_server_url, 84
- create_presigned_notebook_instance_url, 84
- create_processing_job, 84
- create_project, 76, 79, 84
- create_project_version, 76, 79
- create_prompt, 12
- create_prompt_version, 12
- create_provisioned_model_throughput, 8
- create_realtime_endpoint, 60
- create_recommender, 66
- create_resource_policy, 45
- create_resource_policy_statement, 45
- create_retraining_scheduler, 54
- create_rule, 38
- create_schema, 66
- create_slot, 45
- create_slot_type, 45
- create_slot_type_version, 42
- create_solution, 66
- create_solution_version, 66
- create_space, 84
- create_stream_processor, 77, 79
- create_studio_lifecycle_config, 85
- create_test_set_discrepancy_report, 45
- create_training_job, 85
- create_training_plan, 85
- create_transform_job, 85
- create_trial, 85
- create_trial_component, 85
- create_upload_url, 45
- create_user, 75, 79
- create_user_profile, 85
- create_variable, 38
- create_vocabulary, 108
- create_vocabulary_filter, 108
- create_watchlist, 113
- create_what_if_analysis, 34
- create_what_if_forecast, 34
- create_what_if_forecast_export, 35
- create_workforce, 85
- create_workteam, 85
- deactivate_anomaly_detector, 58
- delete_action, 85
- delete_adapter, 105
- delete_adapter_version, 105
- delete_agent, 12
- delete_agent_action_group, 12
- delete_agent_alias, 12
- delete_agent_memory, 15
- delete_agent_version, 12
- delete_alert, 58
- delete_algorithm, 85
- delete_anomaly_detector, 58
- delete_app, 85
- delete_app_image_config, 85
- delete_artifact, 85
- delete_association, 85
- delete_batch_import_job, 38
- delete_batch_prediction, 61
- delete_batch_prediction_job, 38
- delete_blueprint, 18
- delete_bot, 42, 45
- delete_bot_alias, 42, 45
- delete_bot_channel_association, 42
- delete_bot_locale, 45
- delete_bot_replica, 46
- delete_bot_version, 42, 46
- delete_call_analytics_category, 108
- delete_call_analytics_job, 108
- delete_campaign, 66
- delete_cluster, 85
- delete_cluster_scheduler_config, 85
- delete_code_repository, 85
- delete_collection, 75, 79
- delete_compilation_job, 85
- delete_compute_quota, 85
- delete_context, 85
- delete_custom_model, 8
- delete_custom_vocabulary, 46
- delete_data_automation_project, 18
- delete_data_quality_job_definition, 85

- delete_data_source, [12](#), [61](#)
- delete_dataset, [35](#), [54](#), [66](#), [76](#), [79](#)
- delete_dataset_group, [35](#), [66](#)
- delete_dataset_import_job, [35](#)
- delete_detector, [38](#)
- delete_detector_version, [38](#)
- delete_device, [63](#)
- delete_device_fleet, [85](#)
- delete_document_classifier, [26](#)
- delete_domain, [85](#), [113](#)
- delete_earth_observation_job, [98](#)
- delete_edge_deployment_plan, [85](#)
- delete_edge_deployment_stage, [85](#)
- delete_endpoint, [26](#), [85](#)
- delete_endpoint_config, [85](#)
- delete_entity_recognizer, [26](#)
- delete_entity_type, [38](#)
- delete_evaluation, [61](#)
- delete_event, [38](#)
- delete_event_tracker, [66](#)
- delete_event_type, [38](#)
- delete_events_by_event_type, [38](#)
- delete_experiment, [85](#)
- delete_explainability, [35](#)
- delete_explainability_export, [35](#)
- delete_export, [46](#)
- delete_external_model, [39](#)
- delete_faces, [75](#), [79](#)
- delete_feature_group, [85](#)
- delete_filter, [66](#)
- delete_flow, [12](#)
- delete_flow_alias, [12](#)
- delete_flow_definition, [85](#)
- delete_flow_version, [12](#)
- delete_flywheel, [26](#)
- delete_forecast, [35](#)
- delete_forecast_export_job, [35](#)
- delete_fraudster, [113](#)
- delete_guardrail, [8](#)
- delete_hub, [85](#)
- delete_hub_content, [85](#)
- delete_hub_content_reference, [85](#)
- delete_human_loop, [6](#)
- delete_human_task_ui, [85](#)
- delete_hyper_parameter_tuning_job, [85](#)
- delete_image, [85](#)
- delete_image_version, [85](#)
- delete_import, [46](#)
- delete_imported_model, [9](#)
- delete_inference_component, [85](#)
- delete_inference_experiment, [85](#)
- delete_inference_profile, [9](#)
- delete_inference_scheduler, [54](#)
- delete_intent, [42](#), [46](#)
- delete_intent_version, [42](#)
- delete_knowledge_base, [12](#)
- delete_knowledge_base_documents, [12](#)
- delete_label, [39](#), [54](#)
- delete_label_group, [54](#)
- delete_language_model, [108](#)
- delete_lexicon, [75](#)
- delete_list, [39](#)
- delete_marketplace_model_endpoint, [9](#)
- delete_medical_scribe_job, [108](#)
- delete_medical_transcription_job, [108](#)
- delete_medical_vocabulary, [108](#)
- delete_metric_attribution, [67](#)
- delete_ml_model, [61](#)
- delete_mlflow_tracking_server, [85](#)
- delete_model, [39](#), [54](#), [85](#)
- delete_model_bias_job_definition, [85](#)
- delete_model_card, [85](#)
- delete_model_explainability_job_definition, [85](#)
- delete_model_invocation_logging_configuration, [9](#)
- delete_model_package, [85](#)
- delete_model_package_group, [85](#)
- delete_model_package_group_policy, [85](#)
- delete_model_quality_job_definition, [86](#)
- delete_model_version, [39](#)
- delete_monitor, [35](#)
- delete_monitoring_schedule, [86](#)
- delete_notebook_instance, [86](#)
- delete_notebook_instance_lifecycle_config, [86](#)
- delete_optimization_job, [86](#)
- delete_outcome, [39](#)
- delete_package, [63](#)
- delete_parallel_data, [111](#)
- delete_partner_app, [86](#)
- delete_pipeline, [86](#)
- delete_predictor, [35](#)
- delete_predictor_backtest_export_job, [35](#)

- delete_project, [76, 79, 86](#)
- delete_project_policy, [76, 79](#)
- delete_project_version, [76, 79](#)
- delete_prompt, [12](#)
- delete_provisioned_model_throughput, [9](#)
- delete_realtime_endpoint, [61](#)
- delete_recommender, [67](#)
- delete_record, [95](#)
- delete_resource_policy, [26, 46, 54](#)
- delete_resource_policy_statement, [46](#)
- delete_resource_tree, [35](#)
- delete_retraining_scheduler, [54](#)
- delete_rule, [39](#)
- delete_schema, [67](#)
- delete_session, [50, 52](#)
- delete_slot, [46](#)
- delete_slot_type, [42, 46](#)
- delete_slot_type_version, [42](#)
- delete_solution, [67](#)
- delete_space, [86](#)
- delete_speaker, [113](#)
- delete_stream_processor, [77, 79](#)
- delete_studio_lifecycle_config, [86](#)
- delete_tags, [61, 86](#)
- delete_terminology, [111](#)
- delete_test_set, [46](#)
- delete_transcription_job, [108](#)
- delete_trial, [86](#)
- delete_trial_component, [86](#)
- delete_user, [75, 79](#)
- delete_user_profile, [86](#)
- delete_utterances, [42, 46](#)
- delete_variable, [39](#)
- delete_vector_enrichment_job, [98](#)
- delete_vocabulary, [108](#)
- delete_vocabulary_filter, [108](#)
- delete_watchlist, [113](#)
- delete_what_if_analysis, [35](#)
- delete_what_if_forecast, [35](#)
- delete_what_if_forecast_export, [35](#)
- delete_workforce, [86](#)
- delete_workteam, [86](#)
- deregister_devices, [86](#)
- deregister_marketplace_model_endpoint, [9](#)
- deregister_package_version, [63](#)
- describe_action, [86](#)
- describe_alert, [58](#)
- describe_algorithm, [67, 86](#)
- describe_anomaly_detection_executions, [58](#)
- describe_anomaly_detector, [58](#)
- describe_app, [86](#)
- describe_app_image_config, [86](#)
- describe_application_instance, [63](#)
- describe_application_instance_details, [63](#)
- describe_artifact, [86](#)
- describe_auto_ml_job, [86](#)
- describe_auto_ml_job_v2, [86](#)
- describe_auto_predictor, [35](#)
- describe_batch_inference_job, [67](#)
- describe_batch_predictions, [61](#)
- describe_batch_segment_job, [67](#)
- describe_bot, [46](#)
- describe_bot_alias, [46](#)
- describe_bot_locale, [46](#)
- describe_bot_recommendation, [46](#)
- describe_bot_replica, [46](#)
- describe_bot_resource_generation, [46](#)
- describe_bot_version, [46](#)
- describe_campaign, [67](#)
- describe_cluster, [86](#)
- describe_cluster_node, [86](#)
- describe_cluster_scheduler_config, [86](#)
- describe_code_repository, [86](#)
- describe_collection, [75, 79](#)
- describe_compilation_job, [86](#)
- describe_compute_quota, [86](#)
- describe_context, [86](#)
- describe_custom_vocabulary_metadata, [46](#)
- describe_data_deletion_job, [67](#)
- describe_data_ingestion_job, [55](#)
- describe_data_quality_job_definition, [86](#)
- describe_data_sources, [61](#)
- describe_dataset, [26, 35, 55, 67, 76, 79](#)
- describe_dataset_export_job, [67](#)
- describe_dataset_group, [35, 67](#)
- describe_dataset_import_job, [35, 67](#)
- describe_detector, [39](#)
- describe_device, [63, 86](#)
- describe_device_fleet, [86](#)
- describe_device_job, [63](#)
- describe_document_classification_job,

- 26
- describe_document_classifier, 26
- describe_domain, 86, 113
- describe_dominant_language_detection_job, 26
- describe_edge_deployment_plan, 86
- describe_edge_packaging_job, 86
- describe_endpoint, 26, 86
- describe_endpoint_config, 86
- describe_entities_detection_job, 26
- describe_entities_detection_v2_job, 29
- describe_entity_recognizer, 26
- describe_evaluations, 61
- describe_event_tracker, 67
- describe_events_detection_job, 26
- describe_experiment, 86
- describe_explainability, 35
- describe_explainability_export, 35
- describe_export, 46
- describe_feature_group, 86
- describe_feature_metadata, 86
- describe_feature_transformation, 67
- describe_filter, 67
- describe_flow_definition, 86
- describe_flywheel, 26
- describe_flywheel_iteration, 26
- describe_forecast, 35
- describe_forecast_export_job, 35
- describe_fraudster, 114
- describe_fraudster_registration_job, 114
- describe_hub, 86
- describe_hub_content, 86
- describe_human_loop, 6
- describe_human_task_ui, 86
- describe_hyper_parameter_tuning_job, 86
- describe_icdl0cm_inference_job, 29
- describe_image, 86
- describe_image_version, 87
- describe_import, 46
- describe_inference_component, 87
- describe_inference_experiment, 87
- describe_inference_recommendations_job, 87
- describe_inference_scheduler, 55
- describe_intent, 46
- describe_key_phrases_detection_job, 26
- describe_label, 55
- describe_label_group, 55
- describe_labeling_job, 87
- describe_language_model, 108
- describe_lineage_group, 87
- describe_metric_attribution, 67
- describe_metric_set, 58
- describe_ml_models, 61
- describe_mlflow_tracking_server, 87
- describe_model, 55, 87
- describe_model_bias_job_definition, 87
- describe_model_card, 87
- describe_model_card_export_job, 87
- describe_model_explainability_job_definition, 87
- describe_model_package, 87
- describe_model_package_group, 87
- describe_model_quality_job_definition, 87
- describe_model_version, 55
- describe_model_versions, 39
- describe_monitor, 35
- describe_monitoring_schedule, 87
- describe_node, 63
- describe_node_from_template_job, 63
- describe_notebook_instance, 87
- describe_notebook_instance_lifecycle_config, 87
- describe_optimization_job, 87
- describe_package, 63
- describe_package_import_job, 63
- describe_package_version, 63
- describe_partner_app, 87
- describe_phi_detection_job, 29
- describe_pii_entities_detection_job, 26
- describe_pipeline, 87
- describe_pipeline_definition_for_execution, 87
- describe_pipeline_execution, 87
- describe_predictor, 35
- describe_predictor_backtest_export_job, 35
- describe_processing_job, 87
- describe_project, 87
- describe_project_versions, 76, 80
- describe_projects, 76, 80
- describe_recipe, 67

- describe_recommender, 67
- describe_resource_policy, 26, 46, 55
- describe_retraining_scheduler, 55
- describe_rx_norm_inference_job, 29
- describe_schema, 67
- describe_sentiment_detection_job, 26
- describe_slot, 46
- describe_slot_type, 46
- describe_snomedct_inference_job, 29
- describe_solution, 67
- describe_solution_version, 67
- describe_space, 87
- describe_speaker, 114
- describe_speaker_enrollment_job, 114
- describe_stream_processor, 77, 80
- describe_studio_lifecycle_config, 87
- describe_subscribed_workteam, 87
- describe_tags, 61
- describe_targeted_sentiment_detection_job, 26
- describe_test_execution, 46
- describe_test_set, 46
- describe_test_set_discrepancy_report, 46
- describe_test_set_generation, 46
- describe_text_translation_job, 111
- describe_topics_detection_job, 26
- describe_training_job, 87
- describe_training_plan, 87
- describe_transform_job, 87
- describe_trial, 87
- describe_trial_component, 87
- describe_user_profile, 87
- describe_voices, 75
- describe_watchlist, 114
- describe_what_if_analysis, 35
- describe_what_if_forecast, 35
- describe_what_if_forecast_export, 35
- describe_workforce, 87
- describe_workteam, 87
- detect_custom_labels, 76, 80
- detect_document_text, 105
- detect_dominant_language, 26
- detect_entities, 26, 29
- detect_entities_v2, 29
- detect_faces, 75, 80
- detect_key_phrases, 26
- detect_labels, 75, 80
- detect_metric_set_config, 58
- detect_moderation_labels, 76, 80
- detect_phi, 29
- detect_pii_entities, 26
- detect_protective_equipment, 76, 80
- detect_sentiment, 26
- detect_syntax, 26
- detect_targeted_sentiment, 26
- detect_text, 76, 80
- detect_toxic_content, 26
- disable_sagemaker_servicecatalog_portfolio, 87
- disassociate_agent_collaborator, 12
- disassociate_agent_knowledge_base, 12
- disassociate_faces, 76, 80
- disassociate_fraudster, 114
- disassociate_trial_component, 87
- distribute_dataset_entries, 76, 80
- enable_sagemaker_servicecatalog_portfolio, 87
- evaluate_session, 114
- export_earth_observation_job, 98
- export_vector_enrichment_job, 98
- forecastqueryservice, 30
- forecastservice, 32
- frauddetector, 36
- generate_bot_element, 46
- generate_query, 15
- get_accuracy_metrics, 35
- get_action_recommendations, 72
- get_adapter, 105
- get_adapter_version, 105
- get_agent, 12
- get_agent_action_group, 12
- get_agent_alias, 12
- get_agent_collaborator, 12
- get_agent_knowledge_base, 12
- get_agent_memory, 15
- get_agent_version, 12
- get_anomaly_group, 58
- get_async_invoke, 23
- get_batch_import_jobs, 39
- get_batch_prediction, 61
- get_batch_prediction_jobs, 39
- get_blueprint, 18
- get_bot, 42

get_bot_alias, 42
get_bot_aliases, 42
get_bot_channel_association, 42
get_bot_channel_associations, 42
get_bot_versions, 42
get_bots, 42
get_builtin_intent, 42
get_builtin_intents, 42
get_builtin_slot_types, 42
get_call_analytics_category, 108
get_call_analytics_job, 108
get_celebrity_info, 76, 80
get_celebrity_recognition, 77, 80
get_content_moderation, 77, 80
get_custom_model, 9
get_data_automation_project, 18
get_data_automation_status, 21
get_data_quality_metrics, 58
get_data_source, 12, 61
get_delete_events_by_event_type_status, 39
get_deployments, 93
get_detector_version, 39
get_detectors, 39
get_device_fleet_report, 87
get_device_registration, 93
get_document_analysis, 105
get_document_text_detection, 105
get_earth_observation_job, 98
get_entity_types, 39
get_evaluation, 61
get_evaluation_job, 9
get_event, 39
get_event_prediction, 39
get_event_prediction_metadata, 39
get_event_types, 39
get_expense_analysis, 105
get_export, 42
get_external_models, 39
get_face_detection, 77, 80
get_face_liveness_session_results, 80
get_face_search, 77, 80
get_feedback, 58
get_flow, 12
get_flow_alias, 12
get_flow_version, 12
get_foundation_model, 9
get_guardrail, 9
get_import, 42
get_imported_model, 9
get_inference_profile, 9
get_ingestion_job, 12
get_intent, 42
get_intent_versions, 42
get_intents, 42
get_kms_encryption_key, 39
get_knowledge_base, 12
get_knowledge_base_documents, 12
get_label_detection, 77, 80
get_labels, 39
get_lending_analysis, 105
get_lending_analysis_summary, 105
get_lexicon, 75
get_lineage_group_policy, 87
get_list_elements, 39
get_lists_metadata, 39
get_marketplace_model_endpoint, 9
get_media_analysis_job, 76, 80
get_medical_scribe_job, 108
get_medical_transcription_job, 108
get_medical_vocabulary, 108
get_migration, 42
get_migrations, 42
get_ml_model, 61
get_model_copy_job, 9
get_model_customization_job, 9
get_model_import_job, 9
get_model_invocation_job, 9
get_model_invocation_logging_configuration, 9
get_model_package_group_policy, 87
get_model_version, 39
get_models, 39
get_outcomes, 39
get_parallel_data, 111
get_person_tracking, 77, 80
get_personalized_ranking, 72
get_prompt, 12
get_prompt_router, 9
get_provisioned_model_throughput, 9
get_raster_data_collection, 98
get_recommendations, 72
get_record, 95
get_rules, 39
get_sagemaker_servicecatalog_portfolio_status, 87

get_sample_data, 58
get_scaling_configuration_recommendation, 87
get_search_suggestions, 87
get_segment_detection, 77, 80
get_session, 50, 52
get_slot_type, 43
get_slot_type_versions, 43
get_slot_types, 43
get_solution_metrics, 67
get_speech_synthesis_task, 75
get_terminology, 111
get_test_execution_artifacts_url, 46
get_text_detection, 77, 80
get_tile, 98
get_transcription_job, 108
get_utterances_view, 43
get_variables, 39
get_vector_enrichment_job, 98
get_vocabulary, 108
get_vocabulary_filter, 108

import_dataset, 55
import_hub_content, 87
import_model, 26
import_model_version, 55
import_terminology, 111
index_faces, 76, 80
infer_icd10cm, 29
infer_rx_norm, 29
infer_snomedct, 29
ingest_knowledge_base_documents, 12
invoke_agent, 16
invoke_data_automation_async, 21
invoke_endpoint, 103
invoke_endpoint_async, 103
invoke_endpoint_with_response_stream, 103
invoke_flow, 16
invoke_inline_agent, 16
invoke_model, 23
invoke_model_with_response_stream, 23

lexmodelbuildingservice, 40
lexmodelsv2, 43
lexruntimev2, 48
lexruntimev2, 50
list_actions, 87
list_adapter_versions, 105
list_adapters, 105
list_agent_action_groups, 13
list_agent_aliases, 13
list_agent_collaborators, 13
list_agent_knowledge_bases, 13
list_agent_versions, 13
list_agents, 13
list_aggregated_utterances, 46
list_alerts, 58
list_algorithms, 87
list_aliases, 88
list_anomaly_detectors, 58
list_anomaly_group_related_metrics, 58
list_anomaly_group_summaries, 58
list_anomaly_group_time_series, 58
list_app_image_configs, 88
list_application_instance_dependencies, 63
list_application_instance_node_instances, 63
list_application_instances, 63
list_apps, 88
list_artifacts, 88
list_associations, 88
list_async_invokes, 23
list_auto_ml_jobs, 88
list_batch_inference_jobs, 67
list_batch_segment_jobs, 67
list_blueprints, 18
list_bot_alias_replicas, 46
list_bot_aliases, 46
list_bot_locales, 46
list_bot_recommendations, 46
list_bot_replicas, 46
list_bot_resource_generations, 46
list_bot_version_replicas, 46
list_bot_versions, 46
list_bots, 46
list_built_in_intents, 46
list_built_in_slot_types, 46
list_call_analytics_categories, 108
list_call_analytics_jobs, 108
list_campaigns, 67
list_candidates_for_auto_ml_job, 88
list_cluster_nodes, 88
list_cluster_scheduler_configs, 88
list_clusters, 88
list_code_repositories, 88

list_collections, [76, 80](#)
list_compilation_jobs, [88](#)
list_compute_quotas, [88](#)
list_contexts, [88](#)
list_custom_models, [9](#)
list_custom_vocabulary_items, [46](#)
list_data_automation_projects, [18](#)
list_data_deletion_jobs, [67](#)
list_data_ingestion_jobs, [55](#)
list_data_quality_job_definitions, [88](#)
list_data_sources, [13](#)
list_dataset_entries, [76, 80](#)
list_dataset_export_jobs, [67](#)
list_dataset_groups, [35, 67](#)
list_dataset_import_jobs, [35, 67](#)
list_dataset_labels, [76, 80](#)
list_datasets, [26, 35, 55, 67](#)
list_device_fleets, [88](#)
list_devices, [64, 88](#)
list_devices_jobs, [64](#)
list_document_classification_jobs, [26](#)
list_document_classifier_summaries, [26](#)
list_document_classifiers, [26](#)
list_domains, [88, 114](#)
list_dominant_language_detection_jobs, [26](#)
list_earth_observation_jobs, [98](#)
list_edge_deployment_plans, [88](#)
list_edge_packaging_jobs, [88](#)
list_endpoint_configs, [88](#)
list_endpoints, [26, 88](#)
list_entities_detection_jobs, [26](#)
list_entities_detection_v2_jobs, [29](#)
list_entity_recognizer_summaries, [26](#)
list_entity_recognizers, [26](#)
list_evaluation_jobs, [9](#)
list_event_predictions, [39](#)
list_event_trackers, [67](#)
list_events_detection_jobs, [26](#)
list_experiments, [88](#)
list_explainabilities, [35](#)
list_explainability_exports, [35](#)
list_exports, [46](#)
list_faces, [76, 80](#)
list_feature_groups, [88](#)
list_filters, [67](#)
list_flow_aliases, [13](#)
list_flow_definitions, [88](#)
list_flow_versions, [13](#)
list_flows, [13](#)
list_flywheel_iteration_history, [26](#)
list_flywheels, [26](#)
list_forecast_export_jobs, [35](#)
list_forecasts, [35](#)
list_foundation_models, [9](#)
list_fraudster_registration_jobs, [114](#)
list_fraudsters, [114](#)
list_guardrails, [9](#)
list_hub_content_versions, [88](#)
list_hub_contents, [88](#)
list_hubs, [88](#)
list_human_loops, [6](#)
list_human_task_uis, [88](#)
list_hyper_parameter_tuning_jobs, [88](#)
list_icd10cm_inference_jobs, [29](#)
list_image_versions, [88](#)
list_images, [88](#)
list_imported_models, [9](#)
list_imports, [46](#)
list_inference_components, [88](#)
list_inference_events, [55](#)
list_inference_executions, [55](#)
list_inference_experiments, [88](#)
list_inference_profiles, [9](#)
list_inference_recommendations_job_steps, [88](#)
list_inference_recommendations_jobs, [88](#)
list_inference_schedulers, [55](#)
list_ingestion_jobs, [13](#)
list_intent_metrics, [46](#)
list_intent_paths, [47](#)
list_intent_stage_metrics, [47](#)
list_intents, [47](#)
list_key_phrases_detection_jobs, [26](#)
list_knowledge_base_documents, [13](#)
list_knowledge_bases, [13](#)
list_label_groups, [55](#)
list_labeling_jobs, [88](#)
list_labeling_jobs_for_workteam, [88](#)
list_labels, [55](#)
list_language_models, [108](#)
list_languages, [111](#)
list_lexicons, [75](#)
list_lineage_groups, [88](#)
list_marketplace_model_endpoints, [9](#)

list_media_analysis_jobs, 80
list_medical_scribe_jobs, 108
list_medical_transcription_jobs, 108
list_medical_vocabularies, 108
list_metric_attribution_metrics, 67
list_metric_attributions, 67
list_metric_sets, 58
list_mlflow_tracking_servers, 88
list_model_bias_job_definitions, 88
list_model_card_export_jobs, 88
list_model_card_versions, 88
list_model_cards, 88
list_model_copy_jobs, 9
list_model_customization_jobs, 9
list_model_explainability_job_definitions, 88
list_model_import_jobs, 9
list_model_invocation_jobs, 9
list_model_metadata, 88
list_model_package_groups, 88
list_model_packages, 88
list_model_quality_job_definitions, 89
list_model_versions, 55
list_models, 55, 89
list_monitor_evaluations, 35
list_monitoring_alert_history, 89
list_monitoring_alerts, 89
list_monitoring_executions, 89
list_monitoring_schedules, 89
list_monitors, 35
list_node_from_template_jobs, 64
list_nodes, 64
list_notebook_instance_lifecycle_configs, 89
list_notebook_instances, 89
list_optimization_jobs, 89
list_package_import_jobs, 64
list_packages, 64
list_parallel_data, 111
list_partner_apps, 89
list_phi_detection_jobs, 30
list_pii_entities_detection_jobs, 26
list_pipeline_execution_steps, 89
list_pipeline_executions, 89
list_pipeline_parameters_for_execution, 89
list_pipelines, 89
list_predictor_backtest_export_jobs, 35
list_predictors, 35
list_processing_jobs, 89
list_project_policies, 76, 80
list_projects, 89
list_prompt_routers, 9
list_prompts, 13
list_provisioned_model_throughputs, 9
list_raster_data_collections, 98
list_recipes, 67
list_recommended_intents, 47
list_recommenders, 67
list_resource_catalogs, 89
list_retraining_schedulers, 55
list_rx_norm_inference_jobs, 30
list_schemas, 67
list_sensor_statistics, 55
list_sentiment_detection_jobs, 26
list_session_analytics_data, 47
list_session_metrics, 47
list_slot_types, 47
list_slots, 47
list_snomedct_inference_jobs, 30
list_solution_versions, 67
list_solutions, 67
list_spaces, 89
list_speaker_enrollment_jobs, 114
list_speakers, 114
list_speech_synthesis_tasks, 75
list_stage_devices, 89
list_stream_processors, 77, 80
list_studio_lifecycle_configs, 89
list_subscribed_workteams, 89
list_tags, 89
list_tags_for_resource, 9, 13, 26, 35, 39, 43, 47, 55, 58, 64, 67, 80, 98, 105, 108, 111, 114
list_targeted_sentiment_detection_jobs, 26
list_terminologies, 111
list_test_execution_result_items, 47
list_test_executions, 47
list_test_set_records, 47
list_test_sets, 47
list_text_translation_jobs, 111
list_topics_detection_jobs, 26
list_training_jobs, 89
list_training_jobs_for_hyper_parameter_tuning_job,

- 89
- list_training_plans, 89
- list_transcription_jobs, 108
- list_transform_jobs, 89
- list_trial_components, 89
- list_trials, 89
- list_user_profiles, 89
- list_users, 76, 80
- list_utterance_analytics_data, 47
- list_utterance_metrics, 47
- list_vector_enrichment_jobs, 98
- list_vocabularies, 108
- list_vocabulary_filters, 108
- list_watchlists, 114
- list_what_if_analyses, 35
- list_what_if_forecast_exports, 35
- list_what_if_forecasts, 35
- list_workforces, 89
- list_workteams, 89
- lookoutequipment, 52
- lookoutmetrics, 56
- machinelearning, 58
- opt_out_speaker, 114
- optimize_prompt, 16
- panorama, 61
- personalize, 64
- personalizeevents, 68
- personalizeruntime, 70
- polly, 73
- post_content, 50
- post_text, 50
- predict, 61
- prepare_agent, 13
- prepare_flow, 13
- provision_device, 64
- put_action_interactions, 70
- put_actions, 70
- put_bot, 43
- put_bot_alias, 43
- put_detector, 39
- put_entity_type, 39
- put_event_type, 39
- put_events, 70
- put_external_model, 39
- put_feedback, 58
- put_intent, 43
- put_items, 70
- put_kms_encryption_key, 39
- put_label, 39
- put_lexicon, 75
- put_model_invocation_logging_configuration, 9
- put_model_package_group_policy, 89
- put_outcome, 39
- put_project_policy, 76, 80
- put_record, 95
- put_resource_policy, 27, 55
- put_session, 50, 52
- put_slot_type, 43
- put_users, 70
- query_forecast, 32
- query_lineage, 89
- query_what_if_forecast, 32
- recognize_celebrities, 76, 80
- recognize_text, 52
- recognize_utterance, 52
- register_devices, 89
- register_marketplace_model_endpoint, 9
- register_package_version, 64
- rekognition, 75
- remove_application_instance, 64
- render_ui_template, 89
- rerank, 16
- resume_resource, 35
- retrieve, 16
- retrieve_and_generate, 16
- retrieve_and_generate_stream, 16
- retry_pipeline_execution, 89
- sagemaker, 81
- sagemakeredgemanager, 91
- sagemakerfeaturestoreruntime, 93
- sagemakergeospatialcapabilities, 96
- sagemakermetrics, 98
- sagemakerruntime, 101
- search, 89
- search_associated_transcripts, 47
- search_faces, 76, 80
- search_faces_by_image, 76, 80
- search_raster_data_collection, 98
- search_training_plan_offerings, 89
- search_users, 76, 80
- search_users_by_image, 76, 80

send_event, [39](#)
send_heartbeat, [93](#)
send_pipeline_execution_step_failure, [89](#)
send_pipeline_execution_step_success, [89](#)
signal_application_instance_node_instances, [64](#)
start_async_invoke, [23](#)
start_bot_recommendation, [47](#)
start_bot_resource_generation, [47](#)
start_call_analytics_job, [108](#)
start_celebrity_recognition, [77](#), [80](#)
start_content_moderation, [77](#), [80](#)
start_conversation, [52](#)
start_data_ingestion_job, [55](#)
start_document_analysis, [105](#)
start_document_classification_job, [27](#)
start_document_text_detection, [105](#)
start_dominant_language_detection_job, [27](#)
start_earth_observation_job, [98](#)
start_edge_deployment_stage, [89](#)
start_entities_detection_job, [27](#)
start_entities_detection_v2_job, [30](#)
start_events_detection_job, [27](#)
start_expense_analysis, [105](#)
start_face_detection, [77](#), [80](#)
start_face_search, [77](#), [80](#)
start_flywheel_iteration, [27](#)
start_fraudster_registration_job, [114](#)
start_human_loop, [4](#), [6](#)
start_icd10cm_inference_job, [30](#)
start_import, [43](#), [47](#)
start_inference_experiment, [89](#)
start_inference_scheduler, [55](#)
start_ingestion_job, [13](#)
start_key_phrases_detection_job, [27](#)
start_label_detection, [77](#), [80](#)
start_lending_analysis, [105](#)
start_media_analysis_job, [76](#), [80](#)
start_medical_scribe_job, [108](#)
start_medical_transcription_job, [108](#)
start_migration, [43](#)
start_mlflow_tracking_server, [89](#)
start_monitoring_schedule, [89](#)
start_notebook_instance, [89](#)
start_person_tracking, [77](#), [80](#)
start_phi_detection_job, [30](#)
start_pii_entities_detection_job, [27](#)
start_pipeline_execution, [89](#)
start_project_version, [76](#), [80](#)
start_recommender, [67](#)
start_retraining_scheduler, [55](#)
start_rx_norm_inference_job, [30](#)
start_segment_detection, [77](#), [80](#)
start_sentiment_detection_job, [27](#)
start_snomedct_inference_job, [30](#)
start_speaker_enrollment_job, [114](#)
start_speech_synthesis_task, [75](#)
start_stream_processor, [77](#), [80](#)
start_targeted_sentiment_detection_job, [27](#)
start_test_execution, [47](#)
start_test_set_generation, [47](#)
start_text_detection, [77](#), [81](#)
start_text_translation_job, [111](#)
start_topics_detection_job, [27](#)
start_transcription_job, [109](#)
start_vector_enrichment_job, [98](#)
stop_auto_ml_job, [89](#)
stop_bot_recommendation, [47](#)
stop_compilation_job, [89](#)
stop_dominant_language_detection_job, [27](#)
stop_earth_observation_job, [98](#)
stop_edge_deployment_stage, [90](#)
stop_edge_packaging_job, [90](#)
stop_entities_detection_job, [27](#)
stop_entities_detection_v2_job, [30](#)
stop_evaluation_job, [9](#)
stop_events_detection_job, [27](#)
stop_human_loop, [6](#)
stop_hyper_parameter_tuning_job, [90](#)
stop_icd10cm_inference_job, [30](#)
stop_inference_experiment, [90](#)
stop_inference_recommendations_job, [90](#)
stop_inference_scheduler, [55](#)
stop_ingestion_job, [13](#)
stop_key_phrases_detection_job, [27](#)
stop_labeling_job, [90](#)
stop_mlflow_tracking_server, [90](#)
stop_model_customization_job, [9](#)
stop_model_invocation_job, [9](#)
stop_monitoring_schedule, [90](#)
stop_notebook_instance, [90](#)

- stop_optimization_job, [90](#)
- stop_phi_detection_job, [30](#)
- stop_pii_entities_detection_job, [27](#)
- stop_pipeline_execution, [90](#)
- stop_processing_job, [90](#)
- stop_project_version, [76](#), [81](#)
- stop_recommender, [67](#)
- stop_resource, [35](#)
- stop_retraining_scheduler, [55](#)
- stop_rx_norm_inference_job, [30](#)
- stop_sentiment_detection_job, [27](#)
- stop_snomedct_inference_job, [30](#)
- stop_solution_version_creation, [67](#)
- stop_stream_processor, [77](#), [81](#)
- stop_targeted_sentiment_detection_job, [27](#)
- stop_text_translation_job, [111](#)
- stop_training_document_classifier, [27](#)
- stop_training_entity_recognizer, [27](#)
- stop_training_job, [90](#)
- stop_transform_job, [90](#)
- stop_vector_enrichment_job, [98](#)
- synthesize_speech, [75](#)
- tag_resource, [9](#), [13](#), [27](#), [35](#), [39](#), [43](#), [47](#), [55](#), [58](#), [64](#), [67](#), [81](#), [98](#), [105](#), [109](#), [111](#), [114](#)
- textract, [103](#)
- transcribeservice, [106](#)
- translate, [109](#)
- translate_document, [111](#)
- translate_text, [111](#)
- untag_resource, [9](#), [13](#), [27](#), [36](#), [39](#), [43](#), [47](#), [55](#), [58](#), [64](#), [67](#), [81](#), [98](#), [105](#), [109](#), [111](#), [114](#)
- update_action, [90](#)
- update_active_model_version, [55](#)
- update_adapter, [105](#)
- update_agent, [13](#)
- update_agent_action_group, [13](#)
- update_agent_alias, [13](#)
- update_agent_collaborator, [13](#)
- update_agent_knowledge_base, [13](#)
- update_alert, [58](#)
- update_anomaly_detector, [58](#)
- update_app_image_config, [90](#)
- update_artifact, [90](#)
- update_batch_prediction, [61](#)
- update_blueprint, [18](#)
- update_bot, [47](#)
- update_bot_alias, [47](#)
- update_bot_locale, [47](#)
- update_bot_recommendation, [47](#)
- update_call_analytics_category, [109](#)
- update_campaign, [67](#)
- update_cluster, [90](#)
- update_cluster_scheduler_config, [90](#)
- update_cluster_software, [90](#)
- update_code_repository, [90](#)
- update_compute_quota, [90](#)
- update_context, [90](#)
- update_data_automation_project, [18](#)
- update_data_source, [13](#), [61](#)
- update_dataset, [67](#)
- update_dataset_entries, [77](#), [81](#)
- update_dataset_group, [36](#)
- update_detector_version, [39](#)
- update_detector_version_metadata, [39](#)
- update_detector_version_status, [39](#)
- update_device_fleet, [90](#)
- update_device_metadata, [64](#)
- update_devices, [90](#)
- update_domain, [90](#), [114](#)
- update_endpoint, [27](#), [90](#)
- update_endpoint_weights_and_capacities, [90](#)
- update_evaluation, [61](#)
- update_event_label, [39](#)
- update_experiment, [90](#)
- update_export, [47](#)
- update_feature_group, [90](#)
- update_feature_metadata, [90](#)
- update_flow, [13](#)
- update_flow_alias, [13](#)
- update_flywheel, [27](#)
- update_guardrail, [9](#)
- update_hub, [90](#)
- update_image, [90](#)
- update_image_version, [90](#)
- update_inference_component, [90](#)
- update_inference_component_runtime_config, [90](#)
- update_inference_experiment, [90](#)
- update_inference_scheduler, [55](#)
- update_intent, [47](#)
- update_knowledge_base, [13](#)
- update_label_group, [55](#)
- update_list, [39](#)

- update_marketplace_model_endpoint, [9](#)
- update_medical_vocabulary, [109](#)
- update_metric_attribution, [68](#)
- update_metric_set, [58](#)
- update_ml_model, [61](#)
- update_mlflow_tracking_server, [90](#)
- update_model, [39](#), [55](#)
- update_model_card, [90](#)
- update_model_package, [90](#)
- update_model_version, [40](#)
- update_model_version_status, [40](#)
- update_monitoring_alert, [90](#)
- update_monitoring_schedule, [90](#)
- update_notebook_instance, [90](#)
- update_notebook_instance_lifecycle_config,
[90](#)
- update_parallel_data, [111](#)
- update_partner_app, [90](#)
- update_pipeline, [90](#)
- update_pipeline_execution, [90](#)
- update_project, [90](#)
- update_prompt, [13](#)
- update_provisioned_model_throughput, [9](#)
- update_recommender, [68](#)
- update_resource_policy, [47](#)
- update_retraining_scheduler, [55](#)
- update_rule_metadata, [40](#)
- update_rule_version, [40](#)
- update_slot, [47](#)
- update_slot_type, [47](#)
- update_solution, [68](#)
- update_space, [91](#)
- update_stream_processor, [77](#), [81](#)
- update_test_set, [47](#)
- update_training_job, [91](#)
- update_trial, [91](#)
- update_trial_component, [91](#)
- update_user_profile, [91](#)
- update_variable, [40](#)
- update_vocabulary, [109](#)
- update_vocabulary_filter, [109](#)
- update_watchlist, [114](#)
- update_workforce, [91](#)
- update_workteam, [91](#)

- validate_flow_definition, [13](#)
- voiceid, [112](#)