

Package ‘paws.machine.learning’

October 8, 2026

Title 'Amazon Web Services' Machine Learning Services

Version 0.11.0

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>,
<https://www.paws-r-sdk.com>

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

Depends R (>= 3.5.0)

Imports paws.common (>= 0.9.0)

Suggests testthat

Encoding UTF-8

Config/roxygen2/version 8.1.0

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' 'bedrockagentcore_service.R'
'bedrockagentcore_interfaces.R' 'bedrockagentcore_operations.R'
'bedrockagentcorecontrol_service.R'
'bedrockagentcorecontrol_interfaces.R'
'bedrockagentcorecontrol_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' 'machinelearning_service.R'
 'machinelearning_interfaces.R' 'machinelearning_operations.R'
 'personalize_service.R' 'personalize_interfaces.R'
 'personalize_operations.R' 'personalizeevents_service.R'
 'personalizeevents_interfaces.R'
 'personalizeevents_operations.R' 'personalizeruntime_service.R'
 'personalizeruntime_interfaces.R'
 'personalizeruntime_operations.R' 'polly_service.R'
 'polly_interfaces.R' 'polly_operations.R'
 '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'

'extract_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 2026-10-08 08:40:08 UTC

Contents

augmentedairuntime	4
bedrock	7
bedrockagent	11
bedrockagentcore	15
bedrockagentcorecontrol	19
bedrockagentruntime	25
bedrockdataautomation	28
bedrockdataautomationruntime	31
bedrockruntime	33
comprehend	36
comprehendmedical	40
forecastqueryservice	43
forecastservice	46
frauddetector	49
lexmodelbuildingservice	53
lexmodelsv2	57
lexruntimev2	62
lexruntimev2	64
lookoutequipment	67
machinelearning	70
personalize	73
personalizeevents	77
personalizeruntime	80
polly	82
rekognition	85
sagemaker	91
sagemakeredgemanager	102
sagemakerfeaturestoreruntime	104
sagemakergeospatialcapabilities	107

sagemakermetrics	110
sagemakerruntime	112
textextract	114
transcribeservice	117
translate	121
voiceid	123

Index	127
--------------	------------

augmentedairuntime	<i>Amazon Augmented AI Runtime</i>
--------------------	------------------------------------

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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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	<i>Amazon Bedrock</i>
---------	-----------------------

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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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>batch_delete_advanced_prompt_optimization_job</code>	Deletes one or more advanced prompt optimization jobs
<code>batch_delete_evaluation_job</code>	Deletes a batch of evaluation jobs
<code>cancel_automated_reasoning_policy_build_workflow</code>	Cancels a running Automated Reasoning policy build workflow
<code>create_advanced_prompt_optimization_job</code>	Creates an advanced prompt optimization job
<code>create_automated_reasoning_policy</code>	Creates an Automated Reasoning policy for Amazon Bedrock
<code>create_automated_reasoning_policy_test_case</code>	Creates a test for an Automated Reasoning policy
<code>create_automated_reasoning_policy_version</code>	Creates a new version of an existing Automated Reasoning policy
<code>create_custom_model</code>	Creates a new custom model in Amazon Bedrock
<code>create_custom_model_deployment</code>	Deploys a custom model for on-demand inference in Amazon Bedrock
<code>create_evaluation_job</code>	Creates an evaluation job
<code>create_foundation_model_agreement</code>	Request a model access agreement for the specified model
<code>create_guardrail</code>	Creates a guardrail to block topics and to implement safeguards
<code>create_guardrail_version</code>	Creates a version of the guardrail
<code>create_inference_profile</code>	Creates an application inference profile to track metrics and costs
<code>create_marketplace_model_endpoint</code>	Creates an endpoint for a model from Amazon Bedrock Marketplace
<code>create_model_copy_job</code>	Copies a model to another region so that it can be used there
<code>create_model_customization_job</code>	Creates a fine-tuning job to customize a base model
<code>create_model_import_job</code>	Creates a model import job to import model that you have created elsewhere
<code>create_model_invocation_job</code>	Creates a batch inference job to invoke a model on multiple prompts
<code>create_prompt_router</code>	Creates a prompt router that manages the routing of requests
<code>create_provisioned_model_throughput</code>	Creates dedicated throughput for a base or custom model with Amazon Bedrock
<code>delete_automated_reasoning_policy</code>	Deletes an Automated Reasoning policy or policy version
<code>delete_automated_reasoning_policy_build_workflow</code>	Deletes an Automated Reasoning policy build workflow and any associated resources
<code>delete_automated_reasoning_policy_test_case</code>	Deletes an Automated Reasoning policy test
<code>delete_custom_model</code>	Deletes a custom model that you created earlier
<code>delete_custom_model_deployment</code>	Deletes a custom model deployment
<code>delete_enforced_guardrail_configuration</code>	Deletes the account-level enforced guardrail configuration
<code>delete_foundation_model_agreement</code>	Delete the model access agreement for the specified model
<code>delete_guardrail</code>	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_prompt_router</code>	Deletes a specified prompt router
<code>delete_provisioned_model_throughput</code>	Deletes a Provisioned Throughput
<code>delete_resource_policy</code>	Deletes a previously created Bedrock resource policy
<code>deregister_marketplace_model_endpoint</code>	Deregisters an endpoint for a model from Amazon Bedrock Marketplace
<code>export_automated_reasoning_policy_version</code>	Exports the policy definition for an Automated Reasoning policy
<code>get_account_data_retention</code>	Returns the account-wide data retention mode for Amazon Bedrock
<code>get_advanced_prompt_optimization_job</code>	Gets information about an advanced prompt optimization job
<code>get_automated_reasoning_policy</code>	Retrieves details about an Automated Reasoning policy or policy version
<code>get_automated_reasoning_policy_annotations</code>	Retrieves the current annotations for an Automated Reasoning policy
<code>get_automated_reasoning_policy_build_workflow</code>	Retrieves detailed information about an Automated Reasoning policy build workflow
<code>get_automated_reasoning_policy_build_workflow_result_assets</code>	Retrieves the resulting assets from a completed Automated Reasoning policy build workflow
<code>get_automated_reasoning_policy_next_scenario</code>	Retrieves the next test scenario for validating an Automated Reasoning policy
<code>get_automated_reasoning_policy_test_case</code>	Retrieves details about a specific Automated Reasoning policy test case

<code>get_automated_reasoning_policy_test_result</code>	Retrieves the test result for a specific Automated Reasoning
<code>get_custom_model</code>	Get the properties associated with a Amazon Bedrock custom
<code>get_custom_model_deployment</code>	Retrieves information about a custom model deployment, in
<code>get_evaluation_job</code>	Gets information about an evaluation job, such as the status
<code>get_foundation_model</code>	Get details about a Amazon Bedrock foundation model
<code>get_foundation_model_availability</code>	Get information about the Foundation model availability
<code>get_guardrail</code>	Gets details about a guardrail
<code>get_imported_model</code>	Gets properties associated with a customized model you im
<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
<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-customiza
<code>get_model_import_job</code>	Retrieves the properties associated with import model job, i
<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 l
<code>get_prompt_router</code>	Retrieves details about a prompt router
<code>get_provisioned_model_throughput</code>	Returns details for a Provisioned Throughput
<code>get_resource_policy</code>	Gets the resource policy document for a Bedrock resource
<code>get_use_case_for_model_access</code>	Get usecase for model access
<code>list_advanced_prompt_optimization_jobs</code>	Lists the advanced prompt optimization jobs in your account
<code>list_automated_reasoning_policies</code>	Lists all Automated Reasoning policies in your account, wi
<code>list_automated_reasoning_policy_build_workflows</code>	Lists all build workflows for an Automated Reasoning polic
<code>list_automated_reasoning_policy_test_cases</code>	Lists tests for an Automated Reasoning policy
<code>list_automated_reasoning_policy_test_results</code>	Lists test results for an Automated Reasoning policy, showi
<code>list_custom_model_deployments</code>	Lists custom model deployments in your account
<code>list_custom_models</code>	Returns a list of the custom models that you have created w
<code>list_enforced_guardrails_configuration</code>	Lists the account-level enforced guardrail configurations
<code>list_evaluation_jobs</code>	Lists all existing evaluation jobs
<code>list_foundation_model_agreement_offers</code>	Get the offers associated with the specified model
<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 Marl
<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 su
<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_account_data_retention</code>	Sets the account-wide data retention mode for Amazon Be
<code>put_enforced_guardrail_configuration</code>	Sets the account-level enforced guardrail configuration
<code>put_model_invocation_logging_configuration</code>	Set the configuration values for model invocation logging
<code>put_resource_policy</code>	Adds a resource policy for a Bedrock resource
<code>put_use_case_for_model_access</code>	Put usecase for model access
<code>register_marketplace_model_endpoint</code>	Registers an existing Amazon SageMaker endpoint with Ar
<code>start_automated_reasoning_policy_build_workflow</code>	Starts a new build workflow for an Automated Reasoning p

start_automated_reasoning_policy_test_workflow	Initiates a test workflow to validate Automated Reasoning p
stop_advanced_prompt_optimization_job	Stops an advanced prompt optimization job that is in progr
stop_evaluation_job	Stops an evaluation job that is current being created or runn
stop_model_customization_job	Stops an active model customization job
stop_model_invocation_job	Stops a batch inference job
tag_resource	Associate tags with a resource
untag_resource	Remove one or more tags from a resource
update_automated_reasoning_policy	Updates an existing Automated Reasoning policy with new
update_automated_reasoning_policy_annotations	Updates the annotations for an Automated Reasoning polic
update_automated_reasoning_policy_test_case	Updates an existing Automated Reasoning policy test
update_custom_model_deployment	Updates a custom model deployment with a new custom m
update_guardrail	Updates a guardrail with the values you specify
update_marketplace_model_endpoint	Updates the configuration of an existing endpoint for a mod
update_provisioned_model_throughput	Updates the name or associated model for a Provisioned TH

Examples

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

## End(Not run)
```

bedrockagent	<i>Agents for Amazon Bedrock</i>
--------------	----------------------------------

Description

Amazon Bedrock Agents (now Amazon Bedrock Agents Classic) is no longer open to new customers. For capabilities similar to Bedrock Agents Classic, explore Amazon Bedrock AgentCore. Existing customers can continue to use the service as normal. For more information, see [Amazon Bedrock Agents Classic availability change](#).

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

Usage

```
bedrockagent(
  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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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_agent_collaborator	Makes an agent a collaborator for another agent
associate_agent_knowledge_base	Associates a knowledge base with an agent
create_agent	Amazon Bedrock Agents (now Amazon Bedrock Agents Classic) is no longer open to
create_agent_action_group	Creates an action group for an agent
create_agent_alias	Creates an alias of an agent that can be used to deploy the agent
create_data_source	Connects a knowledge base to a data source
create_flow	Creates a prompt flow that you can use to send an input through various steps to yield a
create_flow_alias	Creates an alias of a flow for deployment
create_flow_version	Creates a version of the flow that you can deploy
create_knowledge_base	Creates a knowledge base
create_prompt	Creates a prompt in your prompt library that you can add to a flow
create_prompt_version	Creates a static snapshot of your prompt that can be deployed to production

<code>create_vpc_configuration</code>	Creates a VPC configuration that lets a knowledge base connect to a resource in your p
<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 th
<code>delete_prompt</code>	Deletes a prompt or a version of it, depending on whether you include the promptVersi
<code>delete_resource_policy</code>	Removes the resource policy associated with a knowledge base
<code>delete_vpc_configuration</code>	Deletes a VPC configuration
<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 versi
<code>get_resource_policy</code>	Retrieves the resource policy associated with a knowledge base
<code>get_vpc_configuration</code>	Returns the details and current status of a single VPC configuration
<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 knowledg
<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
<code>list_tags_for_resource</code>	List all the tags for the resource you specify

<code>list_vpc_configurations</code>	Returns a paginated list of the VPC configurations for a knowledge base
<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>put_resource_policy</code>	Associates a resource policy with a knowledge base
<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)
```

bedrockagentcore

Amazon Bedrock AgentCore

Description

Welcome to the Amazon Bedrock AgentCore Data Plane API reference. Data Plane actions process and handle data or workloads within Amazon Web Services services.

Usage

```
bedrockagentcore(
  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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- bedrockagentcore(
  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",
    use_dual_stack = "logical"
  ),
  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_memory_records](#)
[batch_delete_memory_records](#)
[batch_update_memory_records](#)
[complete_resource_token_auth](#)
[create_ab_test](#)
[create_event](#)
[create_payment_instrument](#)
[create_payment_session](#)
[delete_ab_test](#)
[delete_batch_evaluation](#)
[delete_capacity_provider_session](#)
[delete_event](#)

Creates multiple memory records in a single batch operation for the specified mem
 Deletes multiple memory records in a single batch operation from the specified me
 Updates multiple memory records with custom content in a single batch operation
 Confirms the user authentication session for obtaining OAuth2
 Creates an A/B test for comparing agent configurations
 Creates an event in an AgentCore Memory resource
 Create a new payment instrument for a connector
 Create a new payment session
 Deletes an A/B test and its associated gateway rules
 Deletes a batch evaluation and its associated results
 Deletes a session associated with a capacity provider in Amazon Bedrock AgentC
 Deletes an event from an AgentCore Memory resource

<code>delete_memory_record</code>	Deletes a memory record from an AgentCore Memory resource
<code>delete_payment_instrument</code>	Deletes a payment instrument
<code>delete_payment_session</code>	Deletes a payment session
<code>delete_recommendation</code>	Deletes a recommendation and its associated results
<code>evaluate</code>	Performs on-demand evaluation of agent traces using a specified evaluator
<code>get_ab_test</code>	Retrieves detailed information about an A/B test, including its configuration, status, and results
<code>get_agent_card</code>	Retrieves the A2A agent card associated with an AgentCore Runtime agent
<code>get_batch_evaluation</code>	Retrieves detailed information about a batch evaluation, including its status, configuration, and results
<code>get_browser_session</code>	Retrieves detailed information about a specific browser session in Amazon Bedrock AgentCore
<code>get_code_interpreter_session</code>	Retrieves detailed information about a specific code interpreter session in Amazon Bedrock AgentCore
<code>get_event</code>	Retrieves information about a specific event in an AgentCore Memory resource
<code>get_memory_record</code>	Retrieves a specific memory record from an AgentCore Memory resource
<code>get_payment_instrument</code>	Get a payment instrument by ID
<code>get_payment_instrument_balance</code>	Get the balance of a payment instrument
<code>get_payment_session</code>	Get a payment session
<code>get_recommendation</code>	Retrieves detailed information about a recommendation, including its configuration, status, and results
<code>get_resource_api_key</code>	Retrieves the API key associated with an API key credential provider
<code>get_resource_oauth_2_token</code>	Returns the OAuth 2
<code>get_resource_payment_token</code>	Generates authentication tokens for payment providers that use vendor-specific authentication
<code>get_workload_access_token</code>	Obtains a workload access token for agentic workloads not acting on behalf of a user
<code>get_workload_access_token_for_jwt</code>	Obtains a workload access token for agentic workloads acting on behalf of a user, using a JWT
<code>get_workload_access_token_for_user_id</code>	Obtains a workload access token for agentic workloads acting on behalf of a user, using a user ID
<code>ingest_data</code>	Submits content directly for ingestion to generate long-term memory records in an AgentCore Memory resource
<code>invoke_agent_runtime</code>	Sends a request to an agent or tool hosted in an Amazon Bedrock AgentCore Runtime
<code>invoke_agent_runtime_command</code>	Executes a command in a runtime session container and streams the output back to the caller
<code>invoke_browser</code>	Invokes an operating system-level action on a browser session in Amazon Bedrock AgentCore
<code>invoke_code_interpreter</code>	Executes code within an active code interpreter session in Amazon Bedrock AgentCore
<code>invoke_harness</code>	Operation to invoke a Harness
<code>list_ab_tests</code>	Lists all A/B tests in the account
<code>list_actors</code>	Lists all actors in an AgentCore Memory resource
<code>list_batch_evaluations</code>	Lists all batch evaluations in the account, providing summary information about each
<code>list_browser_sessions</code>	Retrieves a list of browser sessions in Amazon Bedrock AgentCore that match the specified criteria
<code>list_code_interpreter_sessions</code>	Retrieves a list of code interpreter sessions in Amazon Bedrock AgentCore that match the specified criteria
<code>list_events</code>	Lists events in an AgentCore Memory resource based on specified criteria
<code>list_memory_extraction_jobs</code>	Lists all long-term memory extraction jobs that are eligible to be started with optional filters
<code>list_memory_records</code>	Lists memory records in an AgentCore Memory resource based on specified criteria
<code>list_payment_instruments</code>	List payment instruments for a manager
<code>list_payment_sessions</code>	List payment sessions
<code>list_recommendations</code>	Lists all recommendations in the account, with optional filtering by status
<code>list_sessions</code>	Lists sessions in an AgentCore Memory resource based on specified criteria
<code>process_payment</code>	Processes a payment using a payment instrument within a payment session
<code>retrieve_memory_records</code>	Searches for and retrieves memory records from an AgentCore Memory resource based on specified criteria
<code>save_browser_session_profile</code>	Saves the current state of a browser session as a reusable profile in Amazon Bedrock AgentCore
<code>search_registry_records</code>	Searches for registry records using semantic, lexical, or hybrid queries
<code>start_batch_evaluation</code>	Starts a batch evaluation job that evaluates agent performance across multiple sessions
<code>start_browser_session</code>	Creates and initializes a browser session in Amazon Bedrock AgentCore
<code>start_code_interpreter_session</code>	Creates and initializes a code interpreter session in Amazon Bedrock AgentCore
<code>start_memory_extraction_job</code>	Starts a memory extraction job that processes events that failed extraction previously

start_recommendation	Starts a recommendation job that analyzes agent traces and generates optimization
stop_batch_evaluation	Stops a running batch evaluation
stop_browser_session	Terminates an active browser session in Amazon Bedrock AgentCore
stop_code_interpreter_session	Terminates an active code interpreter session in Amazon Bedrock AgentCore
stop_runtime_session	Stops a session that is running in an running AgentCore Runtime agent
update_ab_test	Updates an A/B test's configuration, including variants, traffic allocation, evaluation
update_browser_stream	Updates a browser stream

Examples

```
## Not run:
svc <- bedrockagentcore()
svc$batch_create_memory_records(
  Foo = 123
)

## End(Not run)
```

bedrockagentcorecontrol

Amazon Bedrock AgentCore Control

Description

Welcome to the Amazon Bedrock AgentCore Control plane API reference. Control plane actions configure, create, modify, and monitor Amazon Web Services resources.

Usage

```
bedrockagentcorecontrol(
  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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- bedrockagentcorecontrol(
  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",
    use_dual_stack = "logical"
  ),
  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_dataset_examples	Adds examples to the dataset's DRAFT
batch_put_gateway_rate_limits	Atomically creates or updates multiple rate limits for a gateway
create_agent_runtime	Creates an Amazon Bedrock AgentCore Runtime
create_agent_runtime_endpoint	Creates an AgentCore Runtime endpoint
create_api_key_credential_provider	Creates a new API key credential provider
create_browser	Creates a custom browser
create_browser_profile	Creates a browser profile in Amazon Bedrock AgentCore
create_capacity_provider	Creates a capacity provider
create_code_interpreter	Creates a custom code interpreter
create_configuration_bundle	Creates a new configuration bundle resource
create_consent_portal	Creates a new consent portal
create_dataset	Creates a new dataset resource asynchronously
create_dataset_version	Publishes the current DRAFT as a new numbered version
create_evaluator	Creates a custom evaluator for agent quality assessment
create_gateway	Creates a gateway for Amazon Bedrock Agent
create_gateway_rate_limit	Creates a rate limit for a gateway
create_gateway_rule	Creates a rule for a gateway
create_gateway_target	Creates a target for a gateway
create_harness	Operation to create a harness
create_harness_endpoint	Operation to create a harness endpoint
create_memory	Creates a new Amazon Bedrock AgentCore Memory resource
create_oauth2_credential_provider	Creates a new OAuth2 credential provider
create_online_evaluation_config	Creates an online evaluation configuration for continuous monitoring of
create_payment_connector	Creates a new payment connector for a payment manager

<code>create_payment_credential_provider</code>	Creates a new payment credential provider for storing authentication credentials
<code>create_payment_manager</code>	Creates a new payment manager in your Amazon Web Services account
<code>create_policy</code>	Creates a policy within the AgentCore Policy system
<code>create_policy_engine</code>	Creates a new policy engine within the AgentCore Policy system
<code>create_registry</code>	Creates a new registry in your Amazon Web Services account
<code>create_registry_record</code>	Creates a new registry record within the specified registry
<code>create_workload_identity</code>	Creates a new workload identity
<code>delete_agent_runtime</code>	Deletes an Amazon Bedrock AgentCore Runtime, or a single version of an AgentCore Runtime
<code>delete_agent_runtime_endpoint</code>	Deletes an AgentCore Runtime endpoint
<code>delete_api_key_credential_provider</code>	Deletes an API key credential provider
<code>delete_browser</code>	Deletes a custom browser
<code>delete_browser_profile</code>	Deletes a browser profile
<code>delete_capacity_provider</code>	Deletes a capacity provider
<code>delete_code_interpreter</code>	Deletes a custom code interpreter
<code>delete_configuration_bundle</code>	Deletes a configuration bundle and all of its versions
<code>delete_consent_portal</code>	Deletes a consent portal
<code>delete_dataset</code>	Deletes a dataset version or an entire dataset asynchronously
<code>delete_dataset_examples</code>	Deletes specific examples by ID from DRAFT
<code>delete_evaluator</code>	Deletes a custom evaluator
<code>delete_gateway</code>	Deletes a gateway
<code>delete_gateway_rate_limit</code>	Deletes a gateway rate limit
<code>delete_gateway_rule</code>	Deletes a gateway rule
<code>delete_gateway_target</code>	Deletes a gateway target
<code>delete_harness</code>	Operation to delete a Harness
<code>delete_harness_endpoint</code>	Operation to delete a harness endpoint
<code>delete_memory</code>	Deletes an Amazon Bedrock AgentCore Memory resource
<code>delete_oauth_2_credential_provider</code>	Deletes an OAuth2 credential provider
<code>delete_online_evaluation_config</code>	Deletes an online evaluation configuration and stops any ongoing evaluation
<code>delete_payment_connector</code>	Deletes a payment connector
<code>delete_payment_credential_provider</code>	Deletes a payment credential provider and its associated stored credentials
<code>delete_payment_manager</code>	Deletes a payment manager
<code>delete_policy</code>	Deletes an existing policy from the AgentCore Policy system
<code>delete_policy_engine</code>	Deletes an existing policy engine from the AgentCore Policy system
<code>delete_registry</code>	Deletes a registry
<code>delete_registry_record</code>	Deletes a registry record
<code>delete_resource_policy</code>	Deletes the resource-based policy for a specified resource
<code>delete_workload_identity</code>	Deletes a workload identity
<code>get_agent_runtime</code>	Gets an Amazon Bedrock AgentCore Runtime
<code>get_agent_runtime_endpoint</code>	Gets information about an Amazon Secure AgentEndpoint
<code>get_api_key_credential_provider</code>	Retrieves information about an API key credential provider
<code>get_browser</code>	Gets information about a custom browser
<code>get_browser_profile</code>	Gets information about a browser profile
<code>get_capacity_provider</code>	Retrieves information about a capacity provider, including its status, permissions, and configuration
<code>get_code_interpreter</code>	Gets information about a custom code interpreter
<code>get_configuration_bundle</code>	Gets the latest version of a configuration bundle
<code>get_configuration_bundle_version</code>	Gets a specific version of a configuration bundle by its version identifier
<code>get_consent_portal</code>	Retrieves information about a consent portal
<code>get_dataset</code>	Retrieves dataset metadata

get_evaluator	Retrieves detailed information about an evaluator, including its configuration
get_gateway	Retrieves information about a specific Gateway
get_gateway_rate_limit	Retrieves information about a gateway rate limit
get_gateway_rule	Retrieves detailed information about a specific gateway rule
get_gateway_target	Retrieves information about a specific gateway target
get_harness	Operation to get a single harness
get_harness_endpoint	Operation to get a single harness endpoint
get_memory	Retrieve an existing Amazon Bedrock AgentCore Memory resource
get_oauth_2_credential_provider	Retrieves information about an OAuth2 credential provider
get_online_evaluation_config	Retrieves detailed information about an online evaluation configuration
get_payment_connector	Retrieves information about a specific payment connector
get_payment_credential_provider	Retrieves information about a specific payment credential provider
get_payment_manager	Retrieves information about a specific payment manager
get_policy	Retrieves detailed information about a specific policy within the AgentCore
get_policy_engine	Retrieves detailed information about a specific policy engine within the AgentCore
get_policy_engine_summary	Retrieves a metadata-only summary of a specific policy engine without decryption
get_policy_generation	Retrieves information about a policy generation request within the AgentCore
get_policy_generation_summary	Retrieves a metadata-only summary of a specific policy generation request without decryption
get_policy_summary	Retrieves a metadata-only summary of a specific policy without decryption
get_registry	Retrieves information about a specific registry
get_registry_record	Retrieves information about a specific registry record
get_resource_policy	Retrieves the resource-based policy for a specified resource
get_token_vault	Retrieves information about a token vault
get_workload_identity	Retrieves information about a workload identity
list_agent_runtime_endpoints	Lists all endpoints for a specific Amazon Secure Agent
list_agent_runtimes	Lists all Amazon Secure Agents in your account
list_agent_runtime_versions	Lists all versions of a specific Amazon Secure Agent
list_agent_runtime_versions_by_capacity_provider	Lists the agent runtime versions that are associated with a capacity provider
list_api_key_credential_providers	Lists all API key credential providers in your account
list_browser_profiles	Lists all browser profiles in your account
list_browsers	Lists all custom browsers in your account
list_capacity_providers	Lists the capacity providers in your account and returns summary information
list_code_interpreters	Lists all custom code interpreters in your account
list_configuration_bundles	Lists all configuration bundles in the account
list_configuration_bundle_versions	Lists all versions of a configuration bundle, with optional filtering by browser
list_consent_portals	Lists all of the consent portals in your account
list_dataset_examples	Returns paginated examples from the dataset
list_datasets	Lists all datasets in the caller's account, paginated
list_dataset_versions	Lists all published versions of a dataset, sorted by version number descending
list_evaluators	Lists all available evaluators, including both builtin evaluators provided by Amazon Bedrock and custom evaluators
list_gateway_rate_limits	Lists all rate limits for a gateway
list_gateway_rules	Lists all rules for a gateway
list_gateways	Lists all gateways in the account
list_gateway_targets	Lists all targets for a specific gateway
list_harness_endpoints	Operation to list the endpoints of a harness
list_harnesses	Operation to list harnesses
list_harness_versions	Operation to list the versions of a Harness
list_memories	Lists the available Amazon Bedrock AgentCore Memory resources in the account

<code>list_oauth_2_credential_providers</code>	Lists all OAuth2 credential providers in your account
<code>list_online_evaluation_configs</code>	Lists all online evaluation configurations in the account, providing summaries
<code>list_payment_connectors</code>	Lists all payment connectors for a specified payment manager
<code>list_payment_credential_providers</code>	Lists all payment credential providers in the account
<code>list_payment_managers</code>	Lists all payment managers in the account
<code>list_policies</code>	Retrieves a list of policies within the AgentCore Policy engine
<code>list_policy_engines</code>	Retrieves a list of policy engines within the AgentCore Policy system
<code>list_policy_engine_summaries</code>	Retrieves a paginated list of metadata-only policy engine summaries with filters
<code>list_policy_generation_assets</code>	Retrieves a list of generated policy assets from a policy generation request
<code>list_policy_generations</code>	Retrieves a list of policy generation requests within the AgentCore Policy system
<code>list_policy_generation_summaries</code>	Retrieves a paginated list of metadata-only policy generation summaries with filters
<code>list_policy_summaries</code>	Retrieves a paginated list of metadata-only policy summaries within a policy engine
<code>list_registries</code>	Lists all registries in the account
<code>list_registry_records</code>	Lists registry records within a registry
<code>list_tags_for_resource</code>	Lists the tags associated with the specified resource
<code>list_workload_identities</code>	Lists all workload identities in your account
<code>put_resource_policy</code>	Creates or updates a resource-based policy for a resource with the specified name
<code>rotate_payment_connector_credentials</code>	Replaces the service-managed credentials of a payment connector with new ones
<code>set_token_vault_cmk</code>	Sets the customer master key (CMK) for a token vault
<code>start_policy_generation</code>	Initiates the AI-powered generation of Dogwood policies from natural language
<code>submit_registry_record_for_approval</code>	Submits a registry record for approval
<code>synchronize_gateway_targets</code>	Synchronizes the gateway targets by fetching the latest tool definitions from the tool provider
<code>tag_resource</code>	Associates the specified tags to a resource with the specified resourceArn
<code>untag_resource</code>	Removes the specified tags from the specified resource
<code>update_agent_runtime</code>	Updates an existing Amazon Secure Agent
<code>update_agent_runtime_endpoint</code>	Updates an existing Amazon Bedrock AgentCore Runtime endpoint
<code>update_api_key_credential_provider</code>	Updates an existing API key credential provider
<code>update_capacity_provider</code>	Updates a capacity provider
<code>update_configuration_bundle</code>	Updates a configuration bundle by creating a new version with the specified name
<code>update_consent_portal</code>	Updates an existing consent portal
<code>update_dataset</code>	Updates a dataset's metadata
<code>update_dataset_examples</code>	Updates multiple existing examples in-place on DRAFT
<code>update_evaluator</code>	Updates a custom evaluator's configuration, description, or evaluation logic
<code>update_gateway</code>	Updates an existing gateway
<code>update_gateway_rate_limit</code>	Updates the entries of a gateway rate limit
<code>update_gateway_rule</code>	Updates a gateway rule's priority, conditions, actions, or description
<code>update_gateway_target</code>	Updates an existing gateway target
<code>update_harness</code>	Operation to update a harness
<code>update_harness_endpoint</code>	Operation to update a harness endpoint
<code>update_memory</code>	Update an Amazon Bedrock AgentCore Memory resource memory
<code>update_oauth_2_credential_provider</code>	Updates an existing OAuth2 credential provider
<code>update_online_evaluation_config</code>	Updates an online evaluation configuration's settings, including rules, data sources, and metrics
<code>update_payment_connector</code>	Updates an existing payment connector
<code>update_payment_credential_provider</code>	Updates an existing payment credential provider with new authentication details
<code>update_payment_manager</code>	Updates an existing payment manager
<code>update_policy</code>	Updates an existing policy within the AgentCore Policy system
<code>update_policy_engine</code>	Updates an existing policy engine within the AgentCore Policy system
<code>update_registry</code>	Updates an existing registry

update_registry_record	Updates an existing registry record
update_registry_record_status	Updates the status of a registry record
update_workload_identity	Updates an existing workload identity

Examples

```
## Not run:
svc <- bedrockagentcorecontrol()
svc$add_dataset_examples(
  Foo = 123
)

## End(Not run)
```

bedrockagentruntime *Agents for Amazon Bedrock Runtime*

Description

Amazon Bedrock Agents (now Amazon Bedrock Agents Classic) is no longer open to new customers. For capabilities similar to Bedrock Agents Classic, explore Amazon Bedrock AgentCore. Existing customers can continue to use the service as normal. For more information, see [Amazon Bedrock Agents Classic availability change](#).

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

Usage

```
bedrockagentruntime(
  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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- 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",
        use_dual_stack = "logical"
    ),
    credentials = list(
        creds = list(
            access_key_id = "string",
            secret_access_key = "string",
            session_token = "string"
        ),
        profile = "string",
        anonymous = "logical"
    ),
    endpoint = "string",
    region = "string"
)

```

Operations

agentic_retrieve_stream	Retrieves information from one or more knowledge bases using an agentic approach
check_ingested_document_acl	Checks whether a user has access to a specific document by verifying against the ingested access control list (ACL)
create_invocation	Creates a new invocation within a session
create_session	Creates a session to temporarily store conversations for generative AI (GenAI) applications
delete_agent_memory	Deletes memory from the specified memory identifier
delete_session	Deletes a session that you ended
end_session	Ends the session
generate_query	Generates an SQL query from a natural language query
get_agent_memory	Gets the sessions stored in the memory of the agent
get_document_content	Retrieves the content of an ingested document from a knowledge base
get_execution_flow_snapshot	Retrieves the flow definition snapshot used for a flow execution
get_flow_execution	Retrieves details about a specific flow execution, including its status, start and end times, and the output
get_ingested_document_acl	Retrieves the ingested access control list (ACL) for a specific document in a knowledge base
get_invocation_step	Retrieves the details of a specific invocation step within an invocation in a session
get_session	Retrieves details about a specific session
invoke_agent	Amazon Bedrock Agents (now Amazon Bedrock Agents Classic) is no longer open to new customers. Use invoke_flow instead.
invoke_flow	Invokes an alias of a flow to run the inputs that you specify and return the output of each node
invoke_inline_agent	Invokes an inline Amazon Bedrock agent using the configurations you provide with the request
list_flow_execution_events	Lists events that occurred during a flow execution
list_flow_executions	Lists all executions of a flow
list_invocations	Lists all invocations associated with a specific session
list_invocation_steps	Lists all invocation steps associated with a session and optionally, an invocation within the session
list_sessions	Lists all sessions in your Amazon Web Services account
list_tags_for_resource	List all the tags for the resource you specify
optimize_prompt	Optimizes a prompt for the task that you specify
put_invocation_step	Add an invocation step to an invocation in a session

rerank	Reranks the relevance of sources based on queries
retrieve	Queries a knowledge base and retrieves information from it
retrieve_and_generate	Queries a knowledge base and generates responses based on the retrieved results and using th
retrieve_and_generate_stream	Queries a knowledge base and generates responses based on the retrieved results, with output
start_flow_execution	Starts an execution of an Amazon Bedrock flow
stop_flow_execution	Stops an Amazon Bedrock flow's execution
tag_resource	Associate tags with a resource
untag_resource	Remove tags from a resource
update_session	Updates the metadata or encryption settings of a session

Examples

```
## Not run:
svc <- bedrockagentruntime()
svc$agentic_retrieve_stream(
  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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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>copy_blueprint_stage</code>	Copies a Blueprint from one stage to another
<code>create_blueprint</code>	Creates an Amazon Bedrock Data Automation Blueprint
<code>create_blueprint_version</code>	Creates a new version of an existing Amazon Bedrock Data Automation Blueprint
<code>create_data_automation_library</code>	Creates an Amazon Bedrock Data Automation Library
<code>create_data_automation_project</code>	Creates an Amazon Bedrock Data Automation Project
<code>delete_blueprint</code>	Deletes an existing Amazon Bedrock Data Automation Blueprint
<code>delete_data_automation_library</code>	Deletes an existing Amazon Bedrock Data Automation Library
<code>delete_data_automation_project</code>	Deletes an existing Amazon Bedrock Data Automation Project
<code>get_blueprint</code>	Gets an existing Amazon Bedrock Data Automation Blueprint
<code>get_blueprint_optimization_status</code>	API used to get blueprint optimization status
<code>get_data_automation_library</code>	Gets an existing Amazon Bedrock Data Automation Library
<code>get_data_automation_library_entity</code>	Gets an existing entity based on entity type from the library
<code>get_data_automation_library_ingestion_job</code>	API used to get status of data automation library ingestion job
<code>get_data_automation_project</code>	Gets an existing Amazon Bedrock Data Automation Project
<code>invoke_blueprint_optimization_async</code>	Invoke an async job to perform Blueprint Optimization
<code>invoke_data_automation_library_ingestion_job</code>	Async API: Invoke data automation library ingestion job
<code>list_blueprints</code>	Lists all existing Amazon Bedrock Data Automation Blueprints
<code>list_data_automation_libraries</code>	Lists all existing Amazon Bedrock Data Automation Libraries
<code>list_data_automation_library_entities</code>	Lists all stored entities in the library
<code>list_data_automation_library_ingestion_jobs</code>	Lists all data automation library ingestion jobs
<code>list_data_automation_projects</code>	Lists all existing Amazon Bedrock Data Automation Projects
<code>list_tags_for_resource</code>	List tags for an Amazon Bedrock Data Automation resource
<code>tag_resource</code>	Tag an Amazon Bedrock Data Automation resource
<code>untag_resource</code>	Untag an Amazon Bedrock Data Automation resource

update_blueprint	Updates an existing Amazon Bedrock Data Automation Blueprint
update_data_automation_library	Updates an existing Amazon Bedrock Data Automation Library
update_data_automation_project	Updates an existing Amazon Bedrock Data Automation Project

Examples

```
## Not run:
svc <- bedrockdataautomation()
svc$copy_blueprint_stage(
  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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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_data_automation_status	API used to get data automation status
invoke_data_automation	Sync API: Invoke data automation
invoke_data_automation_async	Async API: Invoke data automation
list_tags_for_resource	List tags for an Amazon Bedrock Data Automation resource
tag_resource	Tag an Amazon Bedrock Data Automation resource
untag_resource	Untag an Amazon Bedrock Data Automation resource

Examples

```

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

## End(Not run)

```

bedrockruntime

Amazon Bedrock Runtime

Description

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

Usage

```
bedrockruntime(
  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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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](#)

[converse](#)

[converse_stream](#)

[count_tokens](#)

[get_async_invoke](#)

[invoke_guardrail_checks](#)

[invoke_model](#)

The action to apply a guardrail

Sends messages to the specified Amazon Bedrock model

Sends messages to the specified Amazon Bedrock model and returns the response

Returns the token count for a given inference request

Retrieve information about an asynchronous invocation

Evaluates messages against inline guardrail checks

Invokes the specified Amazon Bedrock model to run inference using the prompt a

invoke_model_with_bidirectional_stream	Invoke the specified Amazon Bedrock model to run inference using the bidirectional stream.
invoke_model_with_response_stream	Invoke the specified Amazon Bedrock model to run inference using the prompt and response stream.
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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
        use_dual_stack = "logical"
    ),
    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](#)
[batch_detect_entities](#)
[batch_detect_key_phrases](#)
[batch_detect_sentiment](#)
[batch_detect_syntax](#)
[batch_detect_targeted_sentiment](#)
[classify_document](#)
[contains_pii_entities](#)
[create_dataset](#)
[create_document_classifier](#)
[create_endpoint](#)
[create_entity_recognizer](#)
[create_flywheel](#)
[delete_document_classifier](#)
[delete_endpoint](#)
[delete_entity_recognizer](#)
[delete_flywheel](#)
[delete_resource_policy](#)
[describe_dataset](#)
[describe_document_classification_job](#)
[describe_document_classifier](#)
[describe_dominant_language_detection_job](#)
[describe_endpoint](#)
[describe_entities_detection_job](#)
[describe_entity_recognizer](#)
[describe_events_detection_job](#)

Determines the dominant language of the input text for a batch of documents
 Inspects the text of a batch of documents for named entities and returns information
 Detects the key noun phrases found in a batch of documents
 Inspects a batch of documents and returns an inference of the prevailing sentiment
 Inspects the text of a batch of documents for the syntax and part of speech of the
 Inspects a batch of documents and returns a sentiment analysis for each entity in
 Creates a classification request to analyze a single document in real-time
 Analyzes input text for the presence of personally identifiable information (PII)
 Creates a dataset to upload training or test data for a model associated with a flywheel
 Creates a new document classifier that you can use to categorize documents
 Creates a model-specific endpoint for synchronous inference for a previously trained
 Creates an entity recognizer using submitted files
 A flywheel is an Amazon Web Services resource that orchestrates the ongoing training
 Deletes a previously created document classifier
 Deletes a model-specific endpoint for a previously-trained custom model
 Deletes an entity recognizer
 Deletes a flywheel
 Deletes a resource-based policy that is attached to a custom model
 Returns information about the dataset that you specify
 Gets the properties associated with a document classification job
 Gets the properties associated with a document classifier
 Gets the properties associated with a dominant language detection job
 Gets the properties associated with a specific endpoint
 Gets the properties associated with an entities detection job
 Provides details about an entity recognizer including status, S3 buckets containing
 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 doc
<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 docu
<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	<i>AWS Comprehend Medical</i>
-------------------	-------------------------------

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-endpoints.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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

<code>list_icd10cm_inference_jobs</code>	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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",  
    use_dual_stack = "logical"  
  ),  
  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	<i>Amazon Forecast Service</i>
-----------------	--------------------------------

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. `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>

- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

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",
    use_dual_stack = "logical"
  ),
  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_auto_predictor</code>	Creates an Amazon Forecast predictor
<code>create_dataset</code>	Creates an Amazon Forecast dataset
<code>create_dataset_group</code>	Creates a dataset group, which holds a collection of related datasets
<code>create_dataset_import_job</code>	Imports your training data to an Amazon Forecast dataset
<code>create_explainability</code>	Explainability is only available for Forecasts and Predictors generated from an AutoPredictor
<code>create_explainability_export</code>	Exports an Explainability resource created by the CreateExplainability operation
<code>create_forecast</code>	Creates a forecast for each item in the TARGET_TIME_SERIES dataset that was used to train the predictor
<code>create_forecast_export_job</code>	Exports a forecast created by the CreateForecast operation to your Amazon Simple Storage Service (S3) bucket
<code>create_monitor</code>	Creates a predictor monitor resource for an existing auto predictor
<code>create_predictor</code>	Creates a predictor resource. This operation creates a legacy predictor that does not include all the predictor functionality
<code>create_predictor_backtest_export_job</code>	Exports backtest forecasts and accuracy metrics generated by the CreateAutoPredictor operation
<code>create_what_if_analysis</code>	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
<code>create_what_if_forecast</code>	A what-if forecast is a forecast that is created from a modified version of the baseline forecast
<code>create_what_if_forecast_export</code>	Exports a forecast created by the CreateWhatIfForecast operation to your Amazon S3 bucket
<code>delete_dataset</code>	Deletes an Amazon Forecast dataset that was created using the CreateDataset operation
<code>delete_dataset_group</code>	Deletes a dataset group created using the CreateDatasetGroup operation
<code>delete_dataset_import_job</code>	Deletes a dataset import job created using the CreateDatasetImportJob operation
<code>delete_explainability</code>	Deletes an Explainability resource
<code>delete_explainability_export</code>	Deletes an Explainability export
<code>delete_forecast</code>	Deletes a forecast created using the CreateForecast operation
<code>delete_forecast_export_job</code>	Deletes a forecast export job created using the CreateForecastExportJob operation
<code>delete_monitor</code>	Deletes a monitor resource
<code>delete_predictor</code>	Deletes a predictor created using the DescribePredictor or CreatePredictor operation
<code>delete_predictor_backtest_export_job</code>	Deletes a predictor backtest export job
<code>delete_resource_tree</code>	Deletes an entire resource tree
<code>delete_what_if_analysis</code>	Deletes a what-if analysis created using the CreateWhatIfAnalysis operation
<code>delete_what_if_forecast</code>	Deletes a what-if forecast created using the CreateWhatIfForecast operation
<code>delete_what_if_forecast_export</code>	Deletes a what-if forecast export created using the CreateWhatIfForecastExport operation
<code>describe_auto_predictor</code>	Describes a predictor created using the CreateAutoPredictor operation
<code>describe_dataset</code>	Describes an Amazon Forecast dataset created using the CreateDataset operation
<code>describe_dataset_group</code>	Describes a dataset group created using the CreateDatasetGroup operation
<code>describe_dataset_import_job</code>	Describes a dataset import job created using the CreateDatasetImportJob operation
<code>describe_explainability</code>	Describes an Explainability resource created using the CreateExplainability operation
<code>describe_explainability_export</code>	Describes an Explainability export created using the CreateExplainabilityExport operation
<code>describe_forecast</code>	Describes a forecast created using the CreateForecast operation
<code>describe_forecast_export_job</code>	Describes a forecast export job created using the CreateForecastExportJob operation
<code>describe_monitor</code>	Describes a monitor resource
<code>describe_predictor</code>	This operation is only valid for legacy predictors created with CreatePredictor
<code>describe_predictor_backtest_export_job</code>	Describes a predictor backtest export job created using the CreatePredictorBacktestExportJob operation
<code>describe_what_if_analysis</code>	Describes the what-if analysis created using the CreateWhatIfAnalysis operation
<code>describe_what_if_forecast</code>	Describes the what-if forecast created using the CreateWhatIfForecast operation
<code>describe_what_if_forecast_export</code>	Describes the what-if forecast export created using the CreateWhatIfForecastExport operation
<code>get_accuracy_metrics</code>	Provides metrics on the accuracy of the models that were trained by the CreatePredictor operation
<code>list_dataset_groups</code>	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 a monitor
list_monitors	Returns a list of monitors created with the CreateMonitor operation and CreateAutoPredictor operation
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	Deletes the specified tags from a resource
update_dataset_group	Replaces the datasets in a dataset group with the specified datasets

Examples

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

## End(Not run)
```

frauddetector

Amazon Fraud Detector

Description

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

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

The Amazon Fraud Detector Query API provides HTTPS requests that use the HTTP verb GET or POST and a Query parameter Action. AWS SDK provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS. These libraries provide basic functions

that automatically take care of tasks such as cryptographically signing your requests, retrying requests, and handling error responses, so that it is easier for you to get started. For more information about the AWS SDKs, go to [Tools to build on AWS](#) page, scroll down to the **SDK** section, and choose plus (+) sign to expand the section.

Usage

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

Arguments

- | | |
|-------------|--|
| config | <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-endpoints.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset. |
| 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 <- 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",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

batch_create_variable	Creates a batch of variables
batch_get_variable	Gets a batch of variables

cancel_batch_import_job	Cancels an in-progress batch import job
cancel_batch_prediction_job	Cancels the specified batch prediction job
create_batch_import_job	Creates a batch import job
create_batch_prediction_job	Creates a batch prediction job
create_detector_version	Creates a detector version
create_list	Creates a list
create_model	Creates a model using the specified model type
create_model_version	Creates a version of the model using the specified model type and model id
create_rule	Creates a rule for use with the specified detector
create_variable	Creates a variable
delete_batch_import_job	Deletes the specified batch import job ID record
delete_batch_prediction_job	Deletes a batch prediction job
delete_detector	Deletes the detector
delete_detector_version	Deletes the detector version
delete_entity_type	Deletes an entity type
delete_event	Deletes the specified event
delete_events_by_event_type	Deletes all events of a particular event type
delete_event_type	Deletes an event type
delete_external_model	Removes a SageMaker model from Amazon Fraud Detector
delete_label	Deletes a label
delete_list	Deletes the list, provided it is not used in a rule
delete_model	Deletes a model
delete_model_version	Deletes a model version
delete_outcome	Deletes an outcome
delete_rule	Deletes the rule
delete_variable	Deletes a variable
describe_detector	Gets all versions for a specified detector
describe_model_versions	Gets all of the model versions for the specified model type or for the specified model id
get_batch_import_jobs	Gets all batch import jobs or a specific job of the specified ID
get_batch_prediction_jobs	Gets all batch prediction jobs or a specific job if you specify a job ID
get_delete_events_by_event_type_status	Retrieves the status of a DeleteEventsByEventType action
get_detectors	Gets all detectors or a single detector if a detectorId is specified
get_detector_version	Gets a particular detector version
get_entity_types	Gets all entity types or a specific entity type if a name is specified
get_event	Retrieves details of events stored with Amazon Fraud Detector
get_event_prediction	Evaluates an event against a detector version
get_event_prediction_metadata	Gets details of the past fraud predictions for the specified event ID, event type, detector version, and rule
get_event_types	Gets all event types or a specific event type if name is provided
get_external_models	Gets the details for one or more Amazon SageMaker models that have been imported into Amazon Fraud Detector
get_kms_encryption_key	Gets the encryption key if a KMS key has been specified to be used to encrypt content
get_labels	Gets all labels or a specific label if name is provided
get_list_elements	Gets all the elements in the specified list
get_lists_metadata	Gets the metadata of either all the lists under the account or the specified list
get_models	Gets one or more models
get_model_version	Gets the details of the specified model version
get_outcomes	Gets one or more outcomes
get_rules	Get all rules for a detector (paginated) if ruleId and ruleVersion are not specified
get_variables	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 t
<code>tag_resource</code>	Assigns tags to a resource
<code>untag_resource</code>	Removes tags from a resource
<code>update_detector_version</code>	Updates a detector version
<code>update_detector_version_metadata</code>	Updates the detector version's description
<code>update_detector_version_status</code>	Updates the detector version's status
<code>update_event_label</code>	Updates the specified event with a new label
<code>update_list</code>	Updates a list
<code>update_model</code>	Updates model description
<code>update_model_version</code>	Updates a model version
<code>update_model_version_status</code>	Updates the status of a model version
<code>update_rule_metadata</code>	Updates a rule's metadata
<code>update_rule_version</code>	Updates a rule version resulting in a new rule version
<code>update_variable</code>	Updates a variable

Examples

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

## End(Not run)
```

lexmodelbuildingservice

Amazon Lex Model Building Service

Description

Amazon Lex Build-Time Actions

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

Usage

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

Arguments

config	Optional configuration of credentials, endpoint, and/or region. • credentials: – creds: * access_key_id: AWS access key ID * secret_access_key: AWS secret access key * session_token: AWS temporary session token – 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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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

<code>delete_intent</code>	Deletes all versions of the intent, including the \$LATEST version
<code>delete_intent_version</code>	Deletes a specific version of an intent
<code>delete_slot_type</code>	Deletes all versions of the slot type, including the \$LATEST version
<code>delete_slot_type_version</code>	Deletes a specific version of a slot type
<code>delete_utterances</code>	Deletes stored utterances
<code>get_bot</code>	Returns metadata information for a specific bot
<code>get_bot_alias</code>	Returns information about an Amazon Lex bot alias
<code>get_bot_aliases</code>	Returns a list of aliases for a specified Amazon Lex bot
<code>get_bot_channel_association</code>	Returns information about the association between an Amazon Lex bot and a messaging platform
<code>get_bot_channel_associations</code>	Returns a list of all of the channels associated with the specified bot
<code>get_bots</code>	Returns bot information as follows:
<code>get_bot_versions</code>	Gets information about all of the versions of a bot
<code>get_builtin_intent</code>	Returns information about a built-in intent
<code>get_builtin_intents</code>	Gets a list of built-in intents that meet the specified criteria
<code>get_builtin_slot_types</code>	Gets a list of built-in slot types that meet the specified criteria
<code>get_export</code>	Exports the contents of a Amazon Lex resource in a specified format
<code>get_import</code>	Gets information about an import job started with the StartImport operation
<code>get_intent</code>	Returns information about an intent
<code>get_intents</code>	Returns intent information as follows:
<code>get_intent_versions</code>	Gets information about all of the versions of an intent
<code>get_migration</code>	Provides details about an ongoing or complete migration from an Amazon Lex V1 bot to an Amazon Lex V2 bot
<code>get_migrations</code>	Gets a list of migrations between Amazon Lex V1 and Amazon Lex V2
<code>get_slot_type</code>	Returns information about a specific version of a slot type
<code>get_slot_types</code>	Returns slot type information as follows:
<code>get_slot_type_versions</code>	Gets information about all versions of a slot type
<code>get_utterances_view</code>	Use the GetUtterancesView operation to get information about the utterances that your user has said
<code>list_tags_for_resource</code>	Gets a list of tags associated with the specified resource
<code>put_bot</code>	Creates an Amazon Lex conversational bot or replaces an existing bot
<code>put_bot_alias</code>	Creates an alias for the specified version of the bot or replaces an alias for the specified bot
<code>put_intent</code>	Creates an intent or replaces an existing intent
<code>put_slot_type</code>	Creates a custom slot type or replaces an existing custom slot type
<code>start_import</code>	Starts a job to import a resource to Amazon Lex
<code>start_migration</code>	Starts migrating a bot from Amazon Lex V1 to Amazon Lex V2
<code>tag_resource</code>	Adds the specified tags to the specified resource
<code>untag_resource</code>	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)
```

Description

This document provides detailed information about the Amazon Lex V2 API actions and their parameters.

For information about the IAM access control permissions you need to use this API, see [Identity-based policies for Amazon Lex V2](#).

Amazon Lex V2 Model Building V2 operations let you build and manage bots.

If you use a custom HTTP client to call Amazon Lex Model Building V2 operations, you must set the "Content-Type" HTTP header to "application/x-amz-json-1.1". Otherwise, you receive an HTTP 404 - UnknownOperationException in the response.

Amazon Lex Model Building V2 operations return the responses with the "application/x-amz-json-1.1" content type.

You can use [Amazon Web Services SDKs](#) to access Amazon Lex V2 APIs using your favorite programming language. The SDKs automatically perform useful tasks for you, such as:

- Cryptographically sign your service requests
- Retry requests
- Handle error responses

The following resources provide additional information about the Amazon Lex V2 Model Building API.

- *Amazon Web Services General Reference*
 - [Amazon Lex V2 Endpoints for each region](#).
- *Command Line Interface*
 - [Amazon Lex Model Building V2 CLI commands](#).

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-endpoints.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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

<code>create_resource_policy_statement</code>	Adds a new resource policy statement to a bot or bot alias
<code>create_slot</code>	Creates a slot in an intent
<code>create_slot_type</code>	Creates a custom slot type
<code>create_test_set_discrepancy_report</code>	Create a report that describes the differences between the bot and the test set
<code>create_upload_url</code>	Gets a pre-signed S3 write URL that you use to upload the zip archive when importing
<code>delete_bot</code>	Deletes all versions of a bot, including the Draft version
<code>delete_bot_alias</code>	Deletes the specified bot alias
<code>delete_bot_analyzer_recommendation</code>	Permanently deletes the recommendations and analysis results for a specific bot and
<code>delete_bot_locale</code>	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_analyzer_recommendation</code>	Retrieves the analysis results and recommendations for bot optimization
<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_analyzer_history</code>	Retrieves a list of historical bot analysis executions for a specific bot
<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

list_bot_replicas	The action to list the replicated bots
list_bot_resource_generations	Lists the generation requests made for a bot locale
list_bots	Gets a list of available bots
list_bot_version_replicas	Contains information about all the versions replication statuses applicable for Glob
list_bot_versions	Gets information about all of the versions of a bot
list_built_in_intents	Gets a list of built-in intents provided by Amazon Lex that you can use in your bot
list_built_in_slot_types	Gets a list of built-in slot types that meet the specified criteria
list_custom_vocabulary_items	Paginated list of custom vocabulary items for a given bot locale's custom vocabula
list_exports	Lists the exports for a bot, bot locale, or custom vocabulary
list_imports	Lists the imports for a bot, bot locale, or custom vocabulary
list_intent_metrics	Retrieves summary metrics for the intents in your bot
list_intent_paths	Retrieves summary statistics for a path of intents that users take over sessions with
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 c
list_session_analytics_data	Retrieves a list of metadata for individual user sessions with your bot
list_session_metrics	Retrieves summary metrics for the user sessions with your bot
list_slots	Gets a list of slots that match the specified criteria
list_slot_types	Gets a list of slot types that match the specified criteria
list_tags_for_resource	Gets a list of tags associated with a resource
list_test_execution_result_items	Gets a list of test execution result items
list_test_executions	The list of test set executions
list_test_set_records	The list of test set records
list_test_sets	The list of the test sets
list_utterance_analytics_data	To use this API operation, your IAM role must have permissions to perform the Lis
list_utterance_metrics	To use this API operation, your IAM role must have permissions to perform the Lis
search_associated_transcripts	Search for associated transcripts that meet the specified criteria
start_bot_analyzer	Initiates an asynchronous analysis of your bot configuration using AI-powered anal
start_bot_recommendation	Use this to provide your transcript data, and to start the bot recommendation proces
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
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_analyzer	Cancels an ongoing bot analysis execution
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	<i>Amazon Lex Runtime Service</i>
--------------------	-----------------------------------

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. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- lexruntime-service(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
```

```

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

```

Operations

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

Examples

```

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

## End(Not run)

```

Description

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

Usage

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

Arguments

config	Optional configuration of credentials, endpoint, and/or region. • credentials: – creds: * access_key_id: AWS access key ID * secret_access_key: AWS secret access key * session_token: AWS temporary session token – profile: The name of a profile to use. If not given, then the default profile is used. – anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

delete_session	Removes session information for a specified bot, alias, and user ID
get_session	Returns session information for a specified bot, alias, and user
put_session	Creates a new session or modifies an existing session with an Amazon Lex V2 bot
recognize_text	Sends user input to Amazon Lex V2
recognize_utterance	Sends user input to Amazon Lex V2
start_conversation	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)
```

lookoutequipment

Amazon Lookout for Equipment

Description

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

Usage

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

Arguments

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

	• s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
```

```

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
describe_data_ingestion_job	Provides information on a specific data ingestion job such as creation time, dataset ARN, and status
describe_dataset	Provides a JSON description of the data in each time series dataset, including names, columns, and data types
describe_inference_scheduler	Specifies information about the inference scheduler being used, including name, model, status, and schedule
describe_label	Returns the name of the label
describe_label_group	Returns information about the label group
describe_model	Provides a JSON containing the overall information about a specific machine learning model, including name, ARN, and status
describe_model_version	Retrieves information about a specific machine learning model version
describe_resource_policy	Provides the details of a resource policy attached to a resource
describe_retraining_scheduler	Provides a description of the retraining scheduler, including information such as the model name, dataset ARN, and schedule
import_dataset	Imports a dataset
import_model_version	Imports a model that has been trained successfully
list_data_ingestion_jobs	Provides a list of all data ingestion jobs, including dataset name and ARN, S3 location of the data, and status
list_datasets	Lists all datasets currently available in your account, filtering on the dataset name
list_inference_events	Lists all inference events that have been found for the specified inference scheduler
list_inference_executions	Lists all inference executions that have been performed by the specified inference scheduler
list_inference_schedulers	Retrieves a list of all inference schedulers currently available for your account
list_label_groups	Returns a list of the label groups
list_labels	Provides a list of labels
list_models	Generates a list of all models in the account, including model name and ARN, dataset, and status
list_model_versions	Generates a list of all model versions for a given model, including the model version, model name, and status

list_retraining_schedulers	Lists all retraining schedulers in your account, filtering by model name prefix and status
list_sensor_statistics	Lists statistics about the data collected for each of the sensors that have been successfully ing
list_tags_for_resource	Lists all the tags for a specified resource, including key and value
put_resource_policy	Creates a resource control policy for a given resource
start_data_ingestion_job	Starts a data ingestion job
start_inference_scheduler	Starts an inference scheduler
start_retraining_scheduler	Starts a retraining scheduler
stop_inference_scheduler	Stops an inference scheduler
stop_retraining_scheduler	Stops a retraining scheduler
tag_resource	Associates a given tag to a resource in your account
untag_resource	Removes a specific tag from a given resource
update_active_model_version	Sets the active model version for a given machine learning model
update_inference_scheduler	Updates an inference scheduler
update_label_group	Updates the label group
update_model	Updates a model in the account
update_retraining_scheduler	Updates a retraining scheduler

Examples

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

## End(Not run)
```

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

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-endpoints.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)

```

Operations

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

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

Examples

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

## End(Not run)
```

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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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
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
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
create_event_tracker	Creates an event tracker that you use when adding event data to a specified dataset group u
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

<code>create_schema</code>	Creates an Amazon Personalize schema from the specified schema string
<code>create_solution</code>	By default, all new solutions use automatic training
<code>create_solution_version</code>	Trains or retrains an active solution in a Custom dataset group
<code>delete_campaign</code>	Removes a campaign by deleting the solution deployment
<code>delete_dataset</code>	Deletes a dataset
<code>delete_dataset_group</code>	Deletes a dataset group
<code>delete_event_tracker</code>	Deletes the event tracker
<code>delete_filter</code>	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
<code>update_metric_attribution</code>	Updates a metric attribution
<code>update_recommender</code>	Updates the recommender to modify the recommender configuration
<code>update_solution</code>	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	<i>Amazon Personalize Events</i>
-------------------	----------------------------------

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 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>
- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

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",
use_dual_stack = "logical"
),
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	<i>Amazon Personalize Runtime</i>
--------------------	-----------------------------------

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. `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>
- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

`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",
    use_dual_stack = "logical"
  ),
  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>
- **use_dual_stack**: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

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",
    use_dual_stack = "logical"
  ),
  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 Service
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_stream	Synthesizes UTF-8 input, plain text, or SSML over a bidirectional streaming connection
start_speech_synthesis_task	Allows the creation of an asynchronous synthesis task, by starting a new SpeechSynthesisTa
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 Bulk Image Analysis](#), [Amazon Rekognition Custom Labels](#), [Amazon Rekognition Stored Video](#), [Amazon Rekognition Face Liveness](#), [Amazon Rekognition Streaming Video](#). It provides descriptions of actions, data types, common parameters, and common errors.

Amazon Rekognition Image

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

Amazon Rekognition Bulk Image Analysis

- [get_media_analysis_job](#)
- [ListMediaAnalysisJob](#)

- `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 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 Face Liveness

- `create_face_liveness_session`
- `get_face_liveness_session_results`
- `StartFaceLivenessSession`

Amazon Rekognition 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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
```

```

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 custom classifier)
create_stream_processor	Service availability notice: Streaming Video and Bulk Image Analysis is no longer available
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
delete_stream_processor	Service availability notice: Streaming Video and Bulk Image Analysis is no longer available
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
describe_projects	Gets information about your Rekognition projects
describe_project_versions	Lists and describes the versions of an Amazon Rekognition project
describe_stream_processor	Service availability notice: Streaming Video and Bulk Image Analysis is no longer available
detect_custom_labels	This operation applies only to Amazon Rekognition Custom Labels
detect_faces	Detects faces within an image that is provided as input
detect_labels	Detects instances of real-world entities within an image (JPEG or PNG) provided as input
detect_moderation_labels	Detects unsafe content in a specified JPEG or PNG format image
detect_protective_equipment	Detects Personal Protective Equipment (PPE) worn by people detected in an image
detect_text	Detects text in the input image and converts it into machine-readable text
disassociate_faces	Removes the association between a Face supplied in an array of FaceIds and the User
distribute_dataset_entries	This operation applies only to Amazon Rekognition Custom Labels
get_celebrity_info	Gets the name and additional information about a celebrity based on their Amazon Rekognition Video analysis
get_celebrity_recognition	Gets the celebrity recognition results for a Amazon Rekognition Video analysis started by the DetectFaces operation

<code>get_content_moderation</code>	Gets the inappropriate, unwanted, or offensive content analysis results for a Amazon Rekognition Video analysis started by StartContentModeration
<code>get_face_detection</code>	Gets face detection results for a Amazon Rekognition Video analysis started by StartFaceDetection
<code>get_face_liveness_session_results</code>	Retrieves the results of a specific Face Liveness session
<code>get_face_search</code>	Gets the face search results for Amazon Rekognition Video face search started by StartFaceSearch
<code>get_label_detection</code>	Gets the label detection results of a Amazon Rekognition Video analysis started by StartLabelDetection
<code>get_media_analysis_job</code>	Service availability notice: Streaming Video and Bulk Image Analysis is no longer available. End of support notice: On October 31, 2025, AWS will discontinue support for Amazon Rekognition Media Analysis jobs.
<code>get_person_tracking</code>	End of support notice: On October 31, 2025, AWS will discontinue support for Amazon Rekognition Person Tracking.
<code>get_segment_detection</code>	Gets the segment detection results of a Amazon Rekognition Video analysis started by StartSegmentDetection
<code>get_text_detection</code>	Gets the text detection results of a Amazon Rekognition Video analysis started by StartTextDetection
<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>	Service availability notice: Streaming Video and Bulk Image Analysis is no longer available. End of support notice: On October 31, 2025, AWS will discontinue support for Amazon Rekognition Media Analysis jobs.
<code>list_project_policies</code>	This operation applies only to Amazon Rekognition Custom Labels
<code>list_stream_processors</code>	Service availability notice: Streaming Video and Bulk Image Analysis is no longer available. End of support notice: On October 31, 2025, AWS will discontinue support for Amazon Rekognition Stream Processors.
<code>list_tags_for_resource</code>	Returns a list of tags in an Amazon Rekognition collection, stream processor, or Custom Label Project
<code>list_users</code>	Returns metadata of the User such as UserID 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 UserIDs within a collection based on a FaceId or UserId
<code>search_users_by_image</code>	Searches for UserIDs using a supplied image
<code>start_celebrity_recognition</code>	Starts asynchronous recognition of celebrities in a stored video
<code>start_content_moderation</code>	Starts asynchronous detection of inappropriate, unwanted, or offensive content in a stored 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>	Service availability notice: Streaming Video and Bulk Image Analysis is no longer available. End of support notice: On October 31, 2025, AWS will discontinue support for Amazon Rekognition Media Analysis jobs.
<code>start_person_tracking</code>	End of support notice: On October 31, 2025, AWS will discontinue support for Amazon Rekognition Person Tracking.
<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>	Service availability notice: Streaming Video and Bulk Image Analysis is no longer available. End of support notice: On October 31, 2025, AWS will discontinue support for Amazon Rekognition Stream Processors.
<code>start_text_detection</code>	Starts asynchronous detection of text in a stored video
<code>stop_project_version</code>	This operation applies only to Amazon Rekognition Custom Labels
<code>stop_stream_processor</code>	Service availability notice: Streaming Video and Bulk Image Analysis is no longer available. End of support notice: On October 31, 2025, AWS will discontinue support for Amazon Rekognition Stream Processors.
<code>tag_resource</code>	Adds one or more key-value tags to an Amazon Rekognition collection, stream processor, or Custom Label Project
<code>untag_resource</code>	Removes one or more tags from an Amazon Rekognition collection, stream processor, or Custom Label Project
<code>update_dataset_entries</code>	This operation applies only to Amazon Rekognition Custom Labels
<code>update_stream_processor</code>	Service availability notice: Streaming Video and Bulk Image Analysis is no longer available. End of support notice: On October 31, 2025, AWS will discontinue support for Amazon Rekognition Stream Processors.

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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
),
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

[attach_cluster_node_network_interface](#)

Attaches an elastic network interface (ENI) to a node in a HyperPod cluster

[attach_cluster_node_volume](#)

Attaches your Amazon Elastic Block Store (Amazon EBS) volume to a node

[batch_add_cluster_nodes](#)

Adds nodes to a HyperPod cluster by incrementing the target count for a node type

[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

[batch_reboot_cluster_nodes](#)

Reboots specific nodes within a SageMaker HyperPod cluster using a system reboot

[batch_replace_cluster_nodes](#)

Replaces specific nodes within a SageMaker HyperPod cluster with new nodes

[create_action](#)

Creates an action

[create_ai_benchmark_job](#)

Creates a benchmark job that runs performance benchmarks against inference

[create_ai_recommendation_job](#)

Creates a recommendation job that generates intelligent optimization recommendations

[create_ai_workload_config](#)

Creates a reusable AI workload configuration that defines datasets, data access, and model

[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 an Amazon SageMaker HyperPod cluster

[create_cluster_scheduler_config](#)

Create cluster policy configuration

[create_code_repository](#)

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_presigned_urls</code>	Creates presigned URLs for accessing hub content artifacts
<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_hyperparameter_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 and ImageVersion
<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_job</code>	Creates a model customization job in Amazon SageMaker
<code>create_labeling_job</code>	Creates a job that uses workers to label the data objects in your input dataset
<code>create_mlflow_app</code>	Creates an MLflow Tracking Server using a general purpose Amazon SageMaker AI instance
<code>create_mlflow_tracking_server</code>	Creates an MLflow Tracking Server using a general purpose Amazon SageMaker AI 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_app_url</code>	Returns a presigned URL that you can use to connect to the MLflow Tracking Server
<code>create_presigned_mlflow_tracking_server_url</code>	Returns a presigned URL that you can use to connect to the MLflow Tracking Server
<code>create_presigned_notebook_instance_url</code>	Returns a URL that you can use to connect to the Jupyter server from a SageMaker AI 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 trials
<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_ai_benchmark_job</code>	Deletes the specified AI benchmark job
<code>delete_ai_recommendation_job</code>	Deletes the specified AI recommendation job
<code>delete_ai_workload_config</code>	Deletes the specified AI workload configuration
<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 are deleted
<code>delete_edge_deployment_stage</code>	Delete a stage in an edge deployment plan if (and only if) the stage is in a pending state
<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 plan)
<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_job</code>	Deletes a job
<code>delete_mlflow_app</code>	Deletes an MLflow App
<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_processing_job</code>	Deletes a processing job
<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_training_job</code>	Deletes a training job
<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_ai_benchmark_job</code>	Returns details of an AI benchmark job, including its status, configuration, and results
<code>describe_ai_recommendation_job</code>	Returns details of an AI recommendation job, including its status, model, and results
<code>describe_ai_workload_config</code>	Returns details of an AI workload configuration, including the dataset configuration
<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_event</code>	Retrieves detailed information about a specific event for a given 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 C
<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 (wor
<code>describe_hyperparameter_tuning_job</code>	Returns a description of a hyperparameter tuning job, depending on the
<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_job</code>	Returns detailed information about a job, including its current status, se
<code>describe_job_schema_version</code>	Returns the JSON schema for a specified job category and schema versi
<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_app</code>	Returns information about an MLflow App
<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 A
<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 c
<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_reserved_capacity</code>	Retrieves details about a reserved capacity
<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_training_plan_extension_history</code>	Retrieves the extension history for a specified 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 trials component's properties
<code>describe_user_profile</code>	Describes a user profile
<code>describe_workforce</code>	Lists private workforce information, including workforce name, Amazon SageMaker
<code>describe_workteam</code>	Gets information about a specific work team
<code>detach_cluster_node_volume</code>	Detaches your Amazon Elastic Block Store (Amazon EBS) volume from a SageMaker
<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>extend_training_plan</code>	Extends an existing training plan by purchasing an extension offering
<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_ai_benchmark_jobs</code>	Returns a list of AI benchmark jobs in your account
<code>list_ai_recommendation_jobs</code>	Returns a list of AI recommendation jobs in your account
<code>list_ai_workload_configs</code>	Returns a list of AI workload configurations in your account
<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_events</code>	Retrieves a list of event summaries for a specified HyperPod cluster
<code>list_cluster_nodes</code>	Retrieves the list of instances (also called nodes interchangeably) in a SageMaker
<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 account
<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_jobs</code>	Lists jobs in a specified category
<code>list_job_schema_versions</code>	Lists available configuration schema versions for a specified job category
<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_apps</code>	Lists all MLflow Apps
<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 machine learning
<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 lifecycle configurations created with the CreateNotebookInstanceLifecycleConfig
<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_pipeline_versions</code>	Gets a list of all versions of the pipeline
<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 V
<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 job
<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_ultra_servers_by_reserved_capacity</code>	Lists all UltraServers that are part of a specified reserved capacity
<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
<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
<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 w
<code>send_pipeline_execution_step_success</code>	Notifies the pipeline that the execution of a callback step succeeded and
<code>start_cluster_health_check</code>	Start deep health checks for a SageMaker HyperPod cluster
<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>start_session</code>	Initiates a remote connection session between a local integrated develop
<code>stop_ai_benchmark_job</code>	Stops a running AI benchmark job
<code>stop_ai_recommendation_job</code>	Stops a running AI recommendation job
<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_hyper_parameter_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_job</code>	Stops a running 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

update_action	Updates an action
update_app_image_config	Updates the properties of an AppImageConfig
update_artifact	Updates an artifact
update_cluster	Updates a SageMaker HyperPod cluster
update_cluster_scheduler_config	Update the cluster policy configuration
update_cluster_software	Updates the platform software of a SageMaker HyperPod cluster for se
update_code_repository	Updates the specified Git repository with the specified values
update_compute_quota	Update the compute allocation definition
update_context	Updates a context
update_device_fleet	Updates a fleet of devices
update_devices	Updates one or more devices in a fleet
update_domain	Updates the default settings for new user profiles in the domain
update_endpoint	Deploys the EndpointConfig specified in the request to a new fleet of in
update_endpoint_weights_and_capacities	Updates variant weight of one or more variants associated with an existi
update_experiment	Adds, updates, or removes the description of an experiment
update_feature_group	Updates the feature group by either adding features or updating the onli
update_feature_metadata	Updates the description and parameters of the feature group
update_hub	Update a hub
update_hub_content	Updates SageMaker hub content (either a Model or Notebook resource)
update_hub_content_reference	Updates the contents of a SageMaker hub for a ModelReference resourc
update_image	Updates the properties of a SageMaker AI image
update_image_version	Updates the properties of a SageMaker AI image version
update_inference_component	Updates an inference component
update_inference_component_runtime_config	Runtime settings for a model that is deployed with an inference compon
update_inference_experiment	Updates an inference experiment that you created
update_mlflow_app	Updates an MLflow App
update_mlflow_tracking_server	Updates properties of an existing MLflow Tracking Server
update_model_card	Update an Amazon SageMaker Model Card
update_model_package	Updates a versioned model
update_monitoring_alert	Update the parameters of a model monitor alert
update_monitoring_schedule	Updates a previously created schedule
update_notebook_instance	Updates a notebook instance
update_notebook_instance_lifecycle_config	Updates a notebook instance lifecycle configuration created with the Cr
update_partner_app	Updates all of the SageMaker Partner AI Apps in an account
update_pipeline	Updates a pipeline
update_pipeline_execution	Updates a pipeline execution
update_pipeline_version	Updates a pipeline version
update_project	Updates a machine learning (ML) project that is created from a templat
update_space	Updates the settings of a space
update_training_job	Update a model training job to request a new Debugger profiling config
update_trial	Updates the display name of a trial
update_trial_component	Updates one or more properties of a trial component
update_user_profile	Updates a user profile
update_workforce	Use this operation to update your workforce
update_workteam	Updates an existing work team with new member definitions or descrip

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-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the AWS_USE_DUALSTACK_ENDPOINT environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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
batch_write_record	Writes a batch of Records to one or more FeatureGroups
delete_record	Deletes a Record from a FeatureGroup in the OnlineStore
get_record	Use for OnlineStore serving from a FeatureStore
list_records	Lists the RecordIdentifier values of all records stored in a FeatureGroup's OnlineStore
put_record	The PutRecord API is used to ingest a list of Records into your feature group
update_record	Updates one or more feature values for an existing record in the specified 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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
```

```

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

Examples

```

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

## End(Not run)

```

sagemakermetrics

*Amazon SageMaker Metrics Service***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. `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>

- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

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",
    use_dual_stack = "logical"
  ),
  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>batch_get_metrics</code>	Used to retrieve training metrics from SageMaker
<code>batch_put_metrics</code>	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 AI runtime API.

Usage

```
sagemakerruntime(
  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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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 AI hosting ser
invoke_endpoint_async	After you deploy a model into production using Amazon SageMaker AI hosting ser
invoke_endpoint_with_response_stream	Invokes a model at the specified endpoint to return the inference response as a stre

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-endpoint.html • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  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 doc
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 informat
get_adapter_version	Gets configuration information for the specified adapter version, including: AdapterId, Adap
get_document_analysis	Gets the results for an Amazon Textract asynchronous operation that analyzes text in a docu
get_document_text_detection	Gets the results for an Amazon Textract asynchronous operation that detects text in a docum

get_expense_analysis	Gets the results for an Amazon Textract asynchronous operation that analyzes invoices and
get_lending_analysis	Gets the results for an Amazon Textract asynchronous operation that analyzes text in a lending
get_lending_analysis_summary	Gets summarized results for the StartLendingAnalysis operation, which analyzes text in a lending
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, item
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)
```

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

Description

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

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

Usage

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

Arguments

config	Optional configuration of credentials, endpoint, and/or region. • credentials: – creds: * access_key_id: AWS access key ID * secret_access_key: AWS secret access key * session_token: AWS temporary session token – 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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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 <- transcribeservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)
```

Operations

create_call_analytics_category	Creates a new Call Analytics category
create_language_model	Creates a new custom language model
create_medical_vocabulary	Creates a new custom medical vocabulary
create_vocabulary	Creates a new custom vocabulary
create_vocabulary_filter	Creates a new custom vocabulary filter
delete_call_analytics_category	Deletes a Call Analytics category
delete_call_analytics_job	Deletes a Call Analytics job

<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 Request Parameters
<code>start_transcription_job</code>	Transcribes the audio from a media file and applies any additional Request Parameters
<code>tag_resource</code>	Adds one or more custom tags, each in the form of a key:value pair, to the specified resource
<code>untag_resource</code>	Removes the specified tags from the specified Amazon Transcribe resource
<code>update_call_analytics_category</code>	Updates the specified Call Analytics category with new rules
<code>update_language_model</code>	Updates the encryption configuration for an existing custom language model
<code>update_medical_vocabulary</code>	Updates an existing custom medical vocabulary with new values
<code>update_vocabulary</code>	Updates an existing custom vocabulary with new values
<code>update_vocabulary_filter</code>	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. `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>

- **use_dual_stack:** Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the `AWS_USE_DUALSTACK_ENDPOINT` environment variable when unset.

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",
    use_dual_stack = "logical"
  ),
  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 giv
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

Amazon Voice ID

Description

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

Usage

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

Arguments

config	Optional configuration of credentials, endpoint, and/or region. • credentials: – creds: * access_key_id: AWS access key ID * secret_access_key: AWS secret access key * session_token: AWS temporary session token – profile: The name of a profile to use. If not given, then the default profile is used. – anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e. <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 • use_dual_stack: Set this to true to use the dualstack (IPv4 and IPv6) endpoint for a service, where available, falling back to the regular endpoint when it isn't. Defaults to the <code>AWS_USE_DUALSTACK_ENDPOINT</code> environment variable when unset.
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",
    use_dual_stack = "logical"
  ),
  credentials = list(
    creds = list(
      access_key_id = "string",
      secret_access_key = "string",
      session_token = "string"
    ),
    profile = "string",
    anonymous = "logical"
  ),
  endpoint = "string",
  region = "string"
)

```

Operations

associate_fraudster	Associates the fraudsters with the watchlist specified in the same domain
create_domain	Creates a domain that contains all Amazon Connect Voice ID data, such as speakers, fra
create_watchlist	Creates a watchlist that fraudsters can be a part of
delete_domain	Deletes the specified domain from Voice ID
delete_fraudster	Deletes the specified fraudster from Voice ID
delete_speaker	Deletes the specified speaker from Voice ID
delete_watchlist	Deletes the specified watchlist from Voice ID
describe_domain	Describes the specified domain
describe_fraudster	Describes the specified fraudster
describe_fraudster_registration_job	Describes the specified fraudster registration job
describe_speaker	Describes the specified speaker
describe_speaker_enrollment_job	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

add_association, [93](#)
add_dataset_examples, [21](#)
add_tags, [72](#), [93](#)
agentic_retrieve_stream, [27](#)
analyze_document, [116](#)
analyze_expense, [116](#)
analyze_id, [116](#)
apply_guardrail, [35](#)
associate_agent_collaborator, [13](#)
associate_agent_knowledge_base, [13](#)
associate_faces, [85](#), [89](#)
associate_fraudster, [125](#)
associate_trial_component, [93](#)
attach_cluster_node_network_interface, [93](#)
attach_cluster_node_volume, [93](#)
augmentedairuntime, [4](#)

batch_add_cluster_nodes, [93](#)
batch_create_custom_vocabulary_item, [59](#)
batch_create_memory_records, [17](#)
batch_create_variable, [51](#)
batch_delete_advanced_prompt_optimization_job, [9](#)
batch_delete_cluster_nodes, [93](#)
batch_delete_custom_vocabulary_item, [59](#)
batch_delete_evaluation_job, [9](#)
batch_delete_memory_records, [17](#)
batch_describe_model_package, [93](#)
batch_detect_dominant_language, [38](#)
batch_detect_entities, [38](#)
batch_detect_key_phrases, [38](#)
batch_detect_sentiment, [38](#)
batch_detect_syntax, [38](#)
batch_detect_targeted_sentiment, [38](#)
batch_get_metrics, [112](#)
batch_get_record, [106](#)
batch_get_variable, [51](#)

batch_put_gateway_rate_limits, [21](#)
batch_put_metrics, [110](#), [112](#)
batch_reboot_cluster_nodes, [93](#)
batch_replace_cluster_nodes, [93](#)
batch_update_custom_vocabulary_item, [59](#)
batch_update_memory_records, [17](#)
batch_write_record, [106](#)
bedrock, [7](#)
bedrockagent, [11](#)
bedrockagentcore, [15](#)
bedrockagentcorecontrol, [19](#)
bedrockagentruntime, [25](#)
bedrockdataautomation, [28](#)
bedrockdataautomationruntime, [31](#)
bedrockruntime, [33](#)
build_bot_locale, [59](#)

cancel_automated_reasoning_policy_build_workflow, [9](#)
cancel_batch_import_job, [52](#)
cancel_batch_prediction_job, [52](#)
check_ingested_document_acl, [27](#)
classify_document, [38](#)
compare_faces, [85](#), [89](#)
complete_resource_token_auth, [17](#)
comprehend, [36](#)
comprehendmedical, [40](#)
contains_pii_entities, [38](#)
converse, [35](#)
converse_stream, [35](#)
copy_blueprint_stage, [30](#)
copy_project_version, [86](#), [89](#)
count_tokens, [35](#)
create_ab_test, [17](#)
create_action, [93](#)
create_adapter, [116](#)
create_adapter_version, [116](#)
create_advanced_prompt_optimization_job, [9](#)

create_agent, [13](#)
 create_agent_action_group, [13](#)
 create_agent_alias, [13](#)
 create_agent_runtime, [21](#)
 create_agent_runtime_endpoint, [21](#)
 create_ai_benchmark_job, [93](#)
 create_ai_recommendation_job, [93](#)
 create_ai_workload_config, [93](#)
 create_algorithm, [93](#)
 create_api_key_credential_provider, [21](#)
 create_app, [93](#)
 create_app_image_config, [93](#)
 create_artifact, [93](#)
 create_auto_ml_job, [93](#)
 create_auto_ml_job_v2, [93](#)
 create_auto_predictor, [48](#)
 create_automated_reasoning_policy, [9](#)
 create_automated_reasoning_policy_test_case, [9](#)
 create_automated_reasoning_policy_version, [9](#)
 create_batch_import_job, [52](#)
 create_batch_inference_job, [75](#)
 create_batch_prediction, [72](#)
 create_batch_prediction_job, [52](#)
 create_batch_segment_job, [75](#)
 create_blueprint, [30](#)
 create_blueprint_version, [30](#)
 create_bot, [59](#)
 create_bot_alias, [59](#)
 create_bot_locale, [59](#)
 create_bot_replica, [59](#)
 create_bot_version, [55](#), [59](#)
 create_browser, [21](#)
 create_browser_profile, [21](#)
 create_call_analytics_category, [119](#)
 create_campaign, [75](#)
 create_capacity_provider, [21](#)
 create_cluster, [93](#)
 create_cluster_scheduler_config, [93](#)
 create_code_interpreter, [21](#)
 create_code_repository, [93](#)
 create_collection, [85](#), [89](#)
 create_compilation_job, [94](#)
 create_compute_quota, [94](#)
 create_configuration_bundle, [21](#)
 create_consent_portal, [21](#)
 create_context, [94](#)
 create_custom_model, [9](#)
 create_custom_model_deployment, [9](#)
 create_data_automation_library, [30](#)
 create_data_automation_project, [30](#)
 create_data_deletion_job, [75](#)
 create_data_quality_job_definition, [94](#)
 create_data_source, [13](#)
 create_data_source_from_rds, [72](#)
 create_data_source_from_redshift, [72](#)
 create_data_source_from_s3, [72](#)
 create_dataset, [21](#), [38](#), [48](#), [69](#), [75](#), [86](#), [89](#)
 create_dataset_export_job, [75](#)
 create_dataset_group, [48](#), [75](#)
 create_dataset_import_job, [48](#), [75](#)
 create_dataset_version, [21](#)
 create_detector_version, [52](#)
 create_device_fleet, [94](#)
 create_document_classifier, [38](#)
 create_domain, [94](#), [125](#)
 create_edge_deployment_plan, [94](#)
 create_edge_deployment_stage, [94](#)
 create_edge_packaging_job, [94](#)
 create_endpoint, [38](#), [94](#)
 create_endpoint_config, [94](#)
 create_entity_recognizer, [38](#)
 create_evaluation, [72](#)
 create_evaluation_job, [9](#)
 create_evaluator, [21](#)
 create_event, [17](#)
 create_event_tracker, [75](#)
 create_experiment, [94](#)
 create_explainability, [48](#)
 create_explainability_export, [48](#)
 create_export, [59](#)
 create_face_liveness_session, [87](#), [89](#)
 create_feature_group, [94](#)
 create_filter, [75](#)
 create_flow, [13](#)
 create_flow_alias, [13](#)
 create_flow_definition, [94](#)
 create_flow_version, [13](#)
 create_flywheel, [38](#)
 create_forecast, [48](#)
 create_forecast_export_job, [48](#)
 create_foundation_model_agreement, [9](#)
 create_gateway, [21](#)
 create_gateway_rate_limit, [21](#)
 create_gateway_rule, [21](#)

create_gateway_target, [21](#)
create_guardrail, [9](#)
create_guardrail_version, [9](#)
create_harness, [21](#)
create_harness_endpoint, [21](#)
create_hub, [94](#)
create_hub_content_presigned_urls, [94](#)
create_hub_content_reference, [94](#)
create_human_task_ui, [94](#)
create_hyper_parameter_tuning_job, [94](#)
create_image, [94](#)
create_image_version, [94](#)
create_inference_component, [94](#)
create_inference_experiment, [94](#)
create_inference_profile, [9](#)
create_inference_recommendations_job,
[94](#)
create_inference_scheduler, [69](#)
create_intent, [59](#)
create_intent_version, [55](#)
create_invocation, [27](#)
create_job, [94](#)
create_knowledge_base, [13](#)
create_label, [69](#)
create_label_group, [69](#)
create_labeling_job, [94](#)
create_language_model, [119](#)
create_list, [52](#)
create_marketplace_model_endpoint, [9](#)
create_medical_vocabulary, [119](#)
create_memory, [21](#)
create_metric_attribution, [75](#)
create_ml_model, [72](#)
create_mlflow_app, [94](#)
create_mlflow_tracking_server, [94](#)
create_model, [52](#), [69](#), [94](#)
create_model_bias_job_definition, [94](#)
create_model_card, [94](#)
create_model_card_export_job, [94](#)
create_model_copy_job, [9](#)
create_model_customization_job, [9](#)
create_model_explainability_job_definition,
[94](#)
create_model_import_job, [9](#)
create_model_invocation_job, [9](#)
create_model_package, [94](#)
create_model_package_group, [94](#)
create_model_quality_job_definition,
[94](#)
create_model_version, [52](#)
create_monitor, [48](#)
create_monitoring_schedule, [94](#)
create_notebook_instance, [94](#)
create_notebook_instance_lifecycle_config,
[94](#)
create_oauth_2_credential_provider, [21](#)
create_online_evaluation_config, [21](#)
create_optimization_job, [94](#)
create_parallel_data, [123](#)
create_partner_app, [94](#)
create_partner_app_presigned_url, [94](#)
create_payment_connector, [21](#)
create_payment_credential_provider, [22](#)
create_payment_instrument, [17](#)
create_payment_manager, [22](#)
create_payment_session, [17](#)
create_pipeline, [94](#)
create_policy, [22](#)
create_policy_engine, [22](#)
create_predictor, [48](#)
create_predictor_backtest_export_job,
[48](#)
create_presigned_domain_url, [94](#)
create_presigned_mlflow_app_url, [94](#)
create_presigned_mlflow_tracking_server_url,
[94](#)
create_presigned_notebook_instance_url,
[94](#)
create_processing_job, [94](#)
create_project, [86](#), [89](#), [95](#)
create_project_version, [86](#), [89](#)
create_prompt, [13](#)
create_prompt_router, [9](#)
create_prompt_version, [13](#)
create_provisioned_model_throughput, [9](#)
create_realtime_endpoint, [72](#)
create_recommender, [75](#)
create_registry, [22](#)
create_registry_record, [22](#)
create_resource_policy, [59](#)
create_resource_policy_statement, [60](#)
create_retraining_scheduler, [69](#)
create_rule, [52](#)
create_schema, [76](#)
create_session, [27](#)
create_slot, [60](#)

- create_slot_type, [60](#)
- create_slot_type_version, [55](#)
- create_solution, [76](#)
- create_solution_version, [76](#)
- create_space, [95](#)
- create_stream_processor, [87](#), [89](#)
- create_studio_lifecycle_config, [95](#)
- create_test_set_discrepancy_report, [60](#)
- create_training_job, [95](#)
- create_training_plan, [95](#)
- create_transform_job, [95](#)
- create_trial, [95](#)
- create_trial_component, [95](#)
- create_upload_url, [60](#)
- create_user, [85](#), [89](#)
- create_user_profile, [95](#)
- create_variable, [52](#)
- create_vocabulary, [119](#)
- create_vocabulary_filter, [119](#)
- create_vpc_configuration, [14](#)
- create_watchlist, [125](#)
- create_what_if_analysis, [48](#)
- create_what_if_forecast, [48](#)
- create_what_if_forecast_export, [48](#)
- create_workforce, [95](#)
- create_workload_identity, [22](#)
- create_workteam, [95](#)

- delete_ab_test, [17](#)
- delete_action, [95](#)
- delete_adapter, [116](#)
- delete_adapter_version, [116](#)
- delete_agent, [14](#)
- delete_agent_action_group, [14](#)
- delete_agent_alias, [14](#)
- delete_agent_memory, [27](#)
- delete_agent_runtime, [22](#)
- delete_agent_runtime_endpoint, [22](#)
- delete_agent_version, [14](#)
- delete_ai_benchmark_job, [95](#)
- delete_ai_recommendation_job, [95](#)
- delete_ai_workload_config, [95](#)
- delete_algorithm, [95](#)
- delete_api_key_credential_provider, [22](#)
- delete_app, [95](#)
- delete_app_image_config, [95](#)
- delete_artifact, [95](#)
- delete_association, [95](#)
- delete_automated_reasoning_policy, [9](#)
- delete_automated_reasoning_policy_build_workflow, [9](#)
- delete_automated_reasoning_policy_test_case, [9](#)
- delete_batch_evaluation, [17](#)
- delete_batch_import_job, [52](#)
- delete_batch_prediction, [72](#)
- delete_batch_prediction_job, [52](#)
- delete_blueprint, [30](#)
- delete_bot, [55](#), [60](#)
- delete_bot_alias, [55](#), [60](#)
- delete_bot_analyzer_recommendation, [60](#)
- delete_bot_channel_association, [55](#)
- delete_bot_locale, [60](#)
- delete_bot_replica, [60](#)
- delete_bot_version, [55](#), [60](#)
- delete_browser, [22](#)
- delete_browser_profile, [22](#)
- delete_call_analytics_category, [119](#)
- delete_call_analytics_job, [119](#)
- delete_campaign, [76](#)
- delete_capacity_provider, [22](#)
- delete_capacity_provider_session, [17](#)
- delete_cluster, [95](#)
- delete_cluster_scheduler_config, [95](#)
- delete_code_interpreter, [22](#)
- delete_code_repository, [95](#)
- delete_collection, [85](#), [89](#)
- delete_compilation_job, [95](#)
- delete_compute_quota, [95](#)
- delete_configuration_bundle, [22](#)
- delete_consent_portal, [22](#)
- delete_context, [95](#)
- delete_custom_model, [9](#)
- delete_custom_model_deployment, [9](#)
- delete_custom_vocabulary, [60](#)
- delete_data_automation_library, [30](#)
- delete_data_automation_project, [30](#)
- delete_data_quality_job_definition, [95](#)
- delete_data_source, [14](#), [72](#)
- delete_dataset, [22](#), [48](#), [69](#), [76](#), [86](#), [89](#)
- delete_dataset_examples, [22](#)
- delete_dataset_group, [48](#), [76](#)
- delete_dataset_import_job, [48](#)
- delete_detector, [52](#)
- delete_detector_version, [52](#)
- delete_device_fleet, [95](#)
- delete_document_classifier, [38](#)

- delete_domain, [95](#), [125](#)
- delete_earth_observation_job, [109](#)
- delete_edge_deployment_plan, [95](#)
- delete_edge_deployment_stage, [95](#)
- delete_endpoint, [38](#), [95](#)
- delete_endpoint_config, [95](#)
- delete_enforced_guardrail_configuration, [9](#)
- delete_entity_recognizer, [38](#)
- delete_entity_type, [52](#)
- delete_evaluation, [72](#)
- delete_evaluator, [22](#)
- delete_event, [17](#), [52](#)
- delete_event_tracker, [76](#)
- delete_event_type, [52](#)
- delete_events_by_event_type, [52](#)
- delete_experiment, [95](#)
- delete_explainability, [48](#)
- delete_explainability_export, [48](#)
- delete_export, [60](#)
- delete_external_model, [52](#)
- delete_faces, [85](#), [89](#)
- delete_feature_group, [95](#)
- delete_filter, [76](#)
- delete_flow, [14](#)
- delete_flow_alias, [14](#)
- delete_flow_definition, [95](#)
- delete_flow_version, [14](#)
- delete_flywheel, [38](#)
- delete_forecast, [48](#)
- delete_forecast_export_job, [48](#)
- delete_foundation_model_agreement, [9](#)
- delete_fraudster, [125](#)
- delete_gateway, [22](#)
- delete_gateway_rate_limit, [22](#)
- delete_gateway_rule, [22](#)
- delete_gateway_target, [22](#)
- delete_guardrail, [9](#)
- delete_harness, [22](#)
- delete_harness_endpoint, [22](#)
- delete_hub, [95](#)
- delete_hub_content, [95](#)
- delete_hub_content_reference, [95](#)
- delete_human_loop, [6](#)
- delete_human_task_ui, [95](#)
- delete_hyper_parameter_tuning_job, [95](#)
- delete_image, [95](#)
- delete_image_version, [95](#)
- delete_import, [60](#)
- delete_imported_model, [9](#)
- delete_inference_component, [95](#)
- delete_inference_experiment, [95](#)
- delete_inference_profile, [9](#)
- delete_inference_scheduler, [69](#)
- delete_intent, [56](#), [60](#)
- delete_intent_version, [56](#)
- delete_job, [95](#)
- delete_knowledge_base, [14](#)
- delete_knowledge_base_documents, [14](#)
- delete_label, [52](#), [69](#)
- delete_label_group, [69](#)
- delete_language_model, [120](#)
- delete_lexicon, [84](#)
- delete_list, [52](#)
- delete_marketplace_model_endpoint, [9](#)
- delete_medical_scribe_job, [120](#)
- delete_medical_transcription_job, [120](#)
- delete_medical_vocabulary, [120](#)
- delete_memory, [22](#)
- delete_memory_record, [18](#)
- delete_metric_attribution, [76](#)
- delete_ml_model, [72](#)
- delete_mlflow_app, [95](#)
- delete_mlflow_tracking_server, [95](#)
- delete_model, [52](#), [69](#), [96](#)
- delete_model_bias_job_definition, [96](#)
- delete_model_card, [96](#)
- delete_model_explainability_job_definition, [96](#)
- delete_model_invocation_logging_configuration, [9](#)
- delete_model_package, [96](#)
- delete_model_package_group, [96](#)
- delete_model_package_group_policy, [96](#)
- delete_model_quality_job_definition, [96](#)
- delete_model_version, [52](#)
- delete_monitor, [48](#)
- delete_monitoring_schedule, [96](#)
- delete_notebook_instance, [96](#)
- delete_notebook_instance_lifecycle_config, [96](#)
- delete_oauth_2_credential_provider, [22](#)
- delete_online_evaluation_config, [22](#)
- delete_optimization_job, [96](#)
- delete_outcome, [52](#)

- [delete_parallel_data, 123](#)
- [delete_partner_app, 96](#)
- [delete_payment_connector, 22](#)
- [delete_payment_credential_provider, 22](#)
- [delete_payment_instrument, 18](#)
- [delete_payment_manager, 22](#)
- [delete_payment_session, 18](#)
- [delete_pipeline, 96](#)
- [delete_policy, 22](#)
- [delete_policy_engine, 22](#)
- [delete_predictor, 48](#)
- [delete_predictor_backtest_export_job, 48](#)
- [delete_processing_job, 96](#)
- [delete_project, 86, 89, 96](#)
- [delete_project_policy, 86, 89](#)
- [delete_project_version, 86, 89](#)
- [delete_prompt, 14](#)
- [delete_prompt_router, 9](#)
- [delete_provisioned_model_throughput, 9](#)
- [delete_realtime_endpoint, 73](#)
- [delete_recommendation, 18](#)
- [delete_recommender, 76](#)
- [delete_record, 106](#)
- [delete_registry, 22](#)
- [delete_registry_record, 22](#)
- [delete_resource_policy, 9, 14, 22, 38, 60, 69](#)
- [delete_resource_policy_statement, 60](#)
- [delete_resource_tree, 48](#)
- [delete_retraining_scheduler, 69](#)
- [delete_rule, 52](#)
- [delete_schema, 76](#)
- [delete_session, 27, 64, 66](#)
- [delete_slot, 60](#)
- [delete_slot_type, 56, 60](#)
- [delete_slot_type_version, 56](#)
- [delete_solution, 76](#)
- [delete_space, 96](#)
- [delete_speaker, 125](#)
- [delete_stream_processor, 87, 89](#)
- [delete_studio_lifecycle_config, 96](#)
- [delete_tags, 73, 96](#)
- [delete_terminology, 123](#)
- [delete_test_set, 60](#)
- [delete_training_job, 96](#)
- [delete_transcription_job, 120](#)
- [delete_trial, 96](#)
- [delete_trial_component, 96](#)
- [delete_user, 85, 89](#)
- [delete_user_profile, 96](#)
- [delete_utterances, 56, 60](#)
- [delete_variable, 52](#)
- [delete_vector_enrichment_job, 109](#)
- [delete_vocabulary, 120](#)
- [delete_vocabulary_filter, 120](#)
- [delete_vpc_configuration, 14](#)
- [delete_watchlist, 125](#)
- [delete_what_if_analysis, 48](#)
- [delete_what_if_forecast, 48](#)
- [delete_what_if_forecast_export, 48](#)
- [delete_workforce, 96](#)
- [delete_workload_identity, 22](#)
- [delete_workteam, 96](#)
- [deregister_devices, 96](#)
- [deregister_marketplace_model_endpoint, 9](#)
- [describe_action, 96](#)
- [describe_ai_benchmark_job, 96](#)
- [describe_ai_recommendation_job, 96](#)
- [describe_ai_workload_config, 96](#)
- [describe_algorithm, 76, 96](#)
- [describe_app, 96](#)
- [describe_app_image_config, 96](#)
- [describe_artifact, 96](#)
- [describe_auto_ml_job, 96](#)
- [describe_auto_ml_job_v2, 96](#)
- [describe_auto_predictor, 48](#)
- [describe_batch_inference_job, 76](#)
- [describe_batch_predictions, 73](#)
- [describe_batch_segment_job, 76](#)
- [describe_bot, 60](#)
- [describe_bot_alias, 60](#)
- [describe_bot_analyzer_recommendation, 60](#)
- [describe_bot_locale, 60](#)
- [describe_bot_recommendation, 60](#)
- [describe_bot_replica, 60](#)
- [describe_bot_resource_generation, 60](#)
- [describe_bot_version, 60](#)
- [describe_campaign, 76](#)
- [describe_cluster, 96](#)
- [describe_cluster_event, 96](#)
- [describe_cluster_node, 96](#)
- [describe_cluster_scheduler_config, 96](#)
- [describe_code_repository, 96](#)

describe_collection, [85](#), [89](#)
describe_compilation_job, [96](#)
describe_compute_quota, [96](#)
describe_context, [96](#)
describe_custom_vocabulary_metadata,
 [60](#)
describe_data_deletion_job, [76](#)
describe_data_ingestion_job, [69](#)
describe_data_quality_job_definition,
 [96](#)
describe_data_sources, [73](#)
describe_dataset, [38](#), [48](#), [69](#), [76](#), [86](#), [89](#)
describe_dataset_export_job, [76](#)
describe_dataset_group, [48](#), [76](#)
describe_dataset_import_job, [48](#), [76](#)
describe_detector, [52](#)
describe_device, [96](#)
describe_device_fleet, [96](#)
describe_document_classification_job,
 [38](#)
describe_document_classifier, [38](#)
describe_domain, [96](#), [125](#)
describe_dominant_language_detection_job,
 [38](#)
describe_edge_deployment_plan, [97](#)
describe_edge_packaging_job, [97](#)
describe_endpoint, [38](#), [97](#)
describe_endpoint_config, [97](#)
describe_entities_detection_job, [38](#)
describe_entities_detection_v2_job, [42](#)
describe_entity_recognizer, [38](#)
describe_evaluations, [73](#)
describe_event_tracker, [76](#)
describe_events_detection_job, [38](#)
describe_experiment, [97](#)
describe_explainability, [48](#)
describe_explainability_export, [48](#)
describe_export, [60](#)
describe_feature_group, [97](#)
describe_feature_metadata, [97](#)
describe_feature_transformation, [76](#)
describe_filter, [76](#)
describe_flow_definition, [97](#)
describe_flywheel, [39](#)
describe_flywheel_iteration, [39](#)
describe_forecast, [48](#)
describe_forecast_export_job, [48](#)
describe_fraudster, [125](#)
describe_fraudster_registration_job,
 [125](#)
describe_hub, [97](#)
describe_hub_content, [97](#)
describe_human_loop, [6](#)
describe_human_task_ui, [97](#)
describe_hyper_parameter_tuning_job,
 [97](#)
describe_icd10cm_inference_job, [42](#)
describe_image, [97](#)
describe_image_version, [97](#)
describe_import, [60](#)
describe_inference_component, [97](#)
describe_inference_experiment, [97](#)
describe_inference_recommendations_job,
 [97](#)
describe_inference_scheduler, [69](#)
describe_intent, [60](#)
describe_job, [97](#)
describe_job_schema_version, [97](#)
describe_key_phrases_detection_job, [39](#)
describe_label, [69](#)
describe_label_group, [69](#)
describe_labeling_job, [97](#)
describe_language_model, [120](#)
describe_lineage_group, [97](#)
describe_metric_attribution, [76](#)
describe_ml_models, [73](#)
describe_mlflow_app, [97](#)
describe_mlflow_tracking_server, [97](#)
describe_model, [69](#), [97](#)
describe_model_bias_job_definition, [97](#)
describe_model_card, [97](#)
describe_model_card_export_job, [97](#)
describe_model_explainability_job_definition,
 [97](#)
describe_model_package, [97](#)
describe_model_package_group, [97](#)
describe_model_quality_job_definition,
 [97](#)
describe_model_version, [69](#)
describe_model_versions, [52](#)
describe_monitor, [48](#)
describe_monitoring_schedule, [97](#)
describe_notebook_instance, [97](#)
describe_notebook_instance_lifecycle_config,
 [97](#)
describe_optimization_job, [97](#)

- describe_partner_app, [97](#)
- describe_phi_detection_job, [42](#)
- describe_pii_entities_detection_job, [39](#)
- describe_pipeline, [97](#)
- describe_pipeline_definition_for_execution, [97](#)
- describe_pipeline_execution, [97](#)
- describe_predictor, [48](#)
- describe_predictor_backtest_export_job, [48](#)
- describe_processing_job, [97](#)
- describe_project, [97](#)
- describe_project_versions, [86](#), [89](#)
- describe_projects, [86](#), [89](#)
- describe_recipe, [76](#)
- describe_recommender, [76](#)
- describe_reserved_capacity, [97](#)
- describe_resource_policy, [39](#), [60](#), [69](#)
- describe_retraining_scheduler, [69](#)
- describe_rx_norm_inference_job, [42](#)
- describe_schema, [76](#)
- describe_sentiment_detection_job, [39](#)
- describe_slot, [60](#)
- describe_slot_type, [60](#)
- describe_snomedct_inference_job, [42](#)
- describe_solution, [76](#)
- describe_solution_version, [76](#)
- describe_space, [97](#)
- describe_speaker, [125](#)
- describe_speaker_enrollment_job, [125](#)
- describe_stream_processor, [87](#), [89](#)
- describe_studio_lifecycle_config, [97](#)
- describe_subscribed_workteam, [97](#)
- describe_tags, [73](#)
- describe_targeted_sentiment_detection_job, [39](#)
- describe_test_execution, [60](#)
- describe_test_set, [60](#)
- describe_test_set_discrepancy_report, [60](#)
- describe_test_set_generation, [60](#)
- describe_text_translation_job, [123](#)
- describe_topics_detection_job, [39](#)
- describe_training_job, [97](#)
- describe_training_plan, [97](#)
- describe_training_plan_extension_history, [97](#)
- describe_transform_job, [98](#)
- describe_trial, [98](#)
- describe_trial_component, [98](#)
- describe_user_profile, [98](#)
- describe_voices, [84](#)
- describe_watchlist, [126](#)
- describe_what_if_analysis, [48](#)
- describe_what_if_forecast, [48](#)
- describe_what_if_forecast_export, [48](#)
- describe_workforce, [98](#)
- describe_workteam, [98](#)
- detach_cluster_node_volume, [98](#)
- detect_custom_labels, [86](#), [89](#)
- detect_document_text, [116](#)
- detect_dominant_language, [39](#)
- detect_entities, [39](#), [42](#)
- detect_entities_v2, [42](#)
- detect_faces, [85](#), [89](#)
- detect_key_phrases, [39](#)
- detect_labels, [85](#), [89](#)
- detect_moderation_labels, [85](#), [89](#)
- detect_phi, [42](#)
- detect_pii_entities, [39](#)
- detect_protective_equipment, [85](#), [89](#)
- detect_sentiment, [39](#)
- detect_syntax, [39](#)
- detect_targeted_sentiment, [39](#)
- detect_text, [85](#), [89](#)
- detect_toxic_content, [39](#)
- disable_sagemaker_servicecatalog_portfolio, [98](#)
- disassociate_agent_collaborator, [14](#)
- disassociate_agent_knowledge_base, [14](#)
- disassociate_faces, [85](#), [89](#)
- disassociate_fraudster, [126](#)
- disassociate_trial_component, [98](#)
- distribute_dataset_entries, [86](#), [89](#)
- enable_sagemaker_servicecatalog_portfolio, [98](#)
- end_session, [27](#)
- evaluate, [18](#)
- evaluate_session, [126](#)
- export_automated_reasoning_policy_version, [9](#)
- export_earth_observation_job, [109](#)
- export_vector_enrichment_job, [109](#)
- extend_training_plan, [98](#)

forecastqueryservice, 43
 forecastservice, 46
 frauddetector, 49

 generate_bot_element, 60
 generate_query, 27
 get_ab_test, 18
 get_account_data_retention, 9
 get_accuracy_metrics, 48
 get_action_recommendations, 82
 get_adapter, 116
 get_adapter_version, 116
 get_advanced_prompt_optimization_job, 9
 get_agent, 14
 get_agent_action_group, 14
 get_agent_alias, 14
 get_agent_card, 18
 get_agent_collaborator, 14
 get_agent_knowledge_base, 14
 get_agent_memory, 27
 get_agent_runtime, 22
 get_agent_runtime_endpoint, 22
 get_agent_version, 14
 get_api_key_credential_provider, 22
 get_async_invoke, 35
 get_automated_reasoning_policy, 9
 get_automated_reasoning_policy_annotations, 9
 get_automated_reasoning_policy_build_workflow, 9
 get_automated_reasoning_policy_build_workflow_result_assets, 9
 get_automated_reasoning_policy_next_scenario, 9
 get_automated_reasoning_policy_test_case, 9
 get_automated_reasoning_policy_test_result, 10
 get_batch_evaluation, 18
 get_batch_import_jobs, 52
 get_batch_prediction, 73
 get_batch_prediction_jobs, 52
 get_blueprint, 30
 get_blueprint_optimization_status, 30
 get_bot, 56
 get_bot_alias, 56
 get_bot_aliases, 56
 get_bot_channel_association, 56
 get_bot_channel_associations, 56
 get_bot_versions, 56
 get_bots, 56
 get_browser, 22
 get_browser_profile, 22
 get_browser_session, 18
 get_builtin_intent, 56
 get_builtin_intents, 56
 get_builtin_slot_types, 56
 get_call_analytics_category, 120
 get_call_analytics_job, 120
 get_capacity_provider, 22
 get_celebrity_info, 85, 89
 get_celebrity_recognition, 86, 89
 get_code_interpreter, 22
 get_code_interpreter_session, 18
 get_configuration_bundle, 22
 get_configuration_bundle_version, 22
 get_consent_portal, 22
 get_content_moderation, 86, 90
 get_custom_model, 10
 get_custom_model_deployment, 10
 get_data_automation_library, 30
 get_data_automation_library_entity, 30
 get_data_automation_library_ingestion_job, 30
 get_data_automation_project, 30
 get_data_automation_status, 33
 get_data_source, 14, 73
 get_dataset, 22
 get_delete_events_by_event_type_status, 52
 get_deployments, 104
 get_detector_version, 52
 get_detectors, 52
 get_device_fleet_report, 98
 get_device_registration, 104
 get_document_analysis, 116
 get_document_content, 27
 get_document_text_detection, 116
 get_earth_observation_job, 109
 get_entity_types, 52
 get_evaluation, 73
 get_evaluation_job, 10
 get_evaluator, 23
 get_event, 18, 52
 get_event_prediction, 52
 get_event_prediction_metadata, 52

- [get_event_types](#), [52](#)
- [get_execution_flow_snapshot](#), [27](#)
- [get_expense_analysis](#), [117](#)
- [get_export](#), [56](#)
- [get_external_models](#), [52](#)
- [get_face_detection](#), [86](#), [90](#)
- [get_face_liveness_session_results](#), [87](#), [90](#)
- [get_face_search](#), [86](#), [90](#)
- [get_flow](#), [14](#)
- [get_flow_alias](#), [14](#)
- [get_flow_execution](#), [27](#)
- [get_flow_version](#), [14](#)
- [get_foundation_model](#), [10](#)
- [get_foundation_model_availability](#), [10](#)
- [get_gateway](#), [23](#)
- [get_gateway_rate_limit](#), [23](#)
- [get_gateway_rule](#), [23](#)
- [get_gateway_target](#), [23](#)
- [get_guardrail](#), [10](#)
- [get_harness](#), [23](#)
- [get_harness_endpoint](#), [23](#)
- [get_import](#), [56](#)
- [get_imported_model](#), [10](#)
- [get_inference_profile](#), [10](#)
- [get_ingested_document_acl](#), [27](#)
- [get_ingestion_job](#), [14](#)
- [get_intent](#), [56](#)
- [get_intent_versions](#), [56](#)
- [get_intents](#), [56](#)
- [get_invocation_step](#), [27](#)
- [get_kms_encryption_key](#), [52](#)
- [get_knowledge_base](#), [14](#)
- [get_knowledge_base_documents](#), [14](#)
- [get_label_detection](#), [86](#), [90](#)
- [get_labels](#), [52](#)
- [get_lending_analysis](#), [117](#)
- [get_lending_analysis_summary](#), [117](#)
- [get_lexicon](#), [84](#)
- [get_lineage_group_policy](#), [98](#)
- [get_list_elements](#), [52](#)
- [get_lists_metadata](#), [52](#)
- [get_marketplace_model_endpoint](#), [10](#)
- [get_media_analysis_job](#), [85](#), [90](#)
- [get_medical_scribe_job](#), [120](#)
- [get_medical_transcription_job](#), [120](#)
- [get_medical_vocabulary](#), [120](#)
- [get_memory](#), [23](#)
- [get_memory_record](#), [18](#)
- [get_migration](#), [56](#)
- [get_migrations](#), [56](#)
- [get_ml_model](#), [73](#)
- [get_model_copy_job](#), [10](#)
- [get_model_customization_job](#), [10](#)
- [get_model_import_job](#), [10](#)
- [get_model_invocation_job](#), [10](#)
- [get_model_invocation_logging_configuration](#), [10](#)
- [get_model_package_group_policy](#), [98](#)
- [get_model_version](#), [52](#)
- [get_models](#), [52](#)
- [get_oauth_2_credential_provider](#), [23](#)
- [get_online_evaluation_config](#), [23](#)
- [get_outcomes](#), [52](#)
- [get_parallel_data](#), [123](#)
- [get_payment_connector](#), [23](#)
- [get_payment_credential_provider](#), [23](#)
- [get_payment_instrument](#), [18](#)
- [get_payment_instrument_balance](#), [18](#)
- [get_payment_manager](#), [23](#)
- [get_payment_session](#), [18](#)
- [get_person_tracking](#), [86](#), [90](#)
- [get_personalized_ranking](#), [82](#)
- [get_policy](#), [23](#)
- [get_policy_engine](#), [23](#)
- [get_policy_engine_summary](#), [23](#)
- [get_policy_generation](#), [23](#)
- [get_policy_generation_summary](#), [23](#)
- [get_policy_summary](#), [23](#)
- [get_prompt](#), [14](#)
- [get_prompt_router](#), [10](#)
- [get_provisioned_model_throughput](#), [10](#)
- [get_raster_data_collection](#), [109](#)
- [get_recommendation](#), [18](#)
- [get_recommendations](#), [82](#)
- [get_record](#), [106](#)
- [get_registry](#), [23](#)
- [get_registry_record](#), [23](#)
- [get_resource_api_key](#), [18](#)
- [get_resource_oauth_2_token](#), [18](#)
- [get_resource_payment_token](#), [18](#)
- [get_resource_policy](#), [10](#), [14](#), [23](#)
- [get_rules](#), [52](#)
- [get_sagemaker_servicecatalog_portfolio_status](#), [98](#)
- [get_scaling_configuration_recommendation](#), [10](#)

- 98
- get_search_suggestions, 98
- get_segment_detection, 86, 90
- get_session, 27, 64, 66
- get_slot_type, 56
- get_slot_type_versions, 56
- get_slot_types, 56
- get_solution_metrics, 76
- get_speech_synthesis_task, 84
- get_terminology, 123
- get_test_execution_artifacts_url, 60
- get_text_detection, 86, 90
- get_tile, 109
- get_token_vault, 23
- get_transcription_job, 120
- get_use_case_for_model_access, 10
- get_utterances_view, 56
- get_variables, 52
- get_vector_enrichment_job, 109
- get_vocabulary, 120
- get_vocabulary_filter, 120
- get_vpc_configuration, 14
- get_workload_access_token, 18
- get_workload_access_token_for_jwt, 18
- get_workload_access_token_for_user_id, 18
- get_workload_identity, 23
- import_dataset, 69
- import_hub_content, 98
- import_model, 39
- import_model_version, 69
- import_terminology, 123
- index_faces, 85, 90
- infer_icd10cm, 42
- infer_rx_norm, 42
- infer_snomedct, 42
- ingest_data, 18
- ingest_knowledge_base_documents, 14
- invoke_agent, 27
- invoke_agent_runtime, 18
- invoke_agent_runtime_command, 18
- invoke_blueprint_optimization_async, 30
- invoke_browser, 18
- invoke_code_interpreter, 18
- invoke_data_automation, 33
- invoke_data_automation_async, 33
- invoke_data_automation_library_ingestion_job, 30
- invoke_endpoint, 114
- invoke_endpoint_async, 114
- invoke_endpoint_with_response_stream, 114
- invoke_flow, 27
- invoke_guardrail_checks, 35
- invoke_harness, 18
- invoke_inline_agent, 27
- invoke_model, 35
- invoke_model_with_bidirectional_stream, 36
- invoke_model_with_response_stream, 36
- lexmodelbuildingservice, 53
- lexmodelsv2, 57
- lexruntime, 62
- lexruntimev2, 64
- list_ab_tests, 18
- list_actions, 98
- list_actors, 18
- list_adapter_versions, 117
- list_adapters, 117
- list_advanced_prompt_optimization_jobs, 10
- list_agent_action_groups, 14
- list_agent_aliases, 14
- list_agent_collaborators, 14
- list_agent_knowledge_bases, 14
- list_agent_runtime_endpoints, 23
- list_agent_runtime_versions, 23
- list_agent_runtime_versions_by_capacity_provider, 23
- list_agent_runtimes, 23
- list_agent_versions, 14
- list_agents, 14
- list_aggregated_utterances, 60
- list_ai_benchmark_jobs, 98
- list_ai_recommendation_jobs, 98
- list_ai_workload_configs, 98
- list_algorithms, 98
- list_aliases, 98
- list_api_key_credential_providers, 23
- list_app_image_configs, 98
- list_apps, 98
- list_artifacts, 98
- list_associations, 98
- list_async_invokes, 36

list_auto_ml_jobs, [98](#)
 list_automated_reasoning_policies, [10](#)
 list_automated_reasoning_policy_build_workflows, [10](#)
 list_automated_reasoning_policy_test_cases, [10](#)
 list_automated_reasoning_policy_test_results, [10](#)
 list_batch_evaluations, [18](#)
 list_batch_inference_jobs, [76](#)
 list_batch_segment_jobs, [76](#)
 list_blueprints, [30](#)
 list_bot_alias_replicas, [60](#)
 list_bot_aliases, [60](#)
 list_bot_analyzer_history, [60](#)
 list_bot_locales, [60](#)
 list_bot_recommendations, [60](#)
 list_bot_replicas, [61](#)
 list_bot_resource_generations, [61](#)
 list_bot_version_replicas, [61](#)
 list_bot_versions, [61](#)
 list_bots, [61](#)
 list_browser_profiles, [23](#)
 list_browser_sessions, [18](#)
 list_browsers, [23](#)
 list_built_in_intents, [61](#)
 list_built_in_slot_types, [61](#)
 list_call_analytics_categories, [120](#)
 list_call_analytics_jobs, [120](#)
 list_campaigns, [76](#)
 list_candidates_for_auto_ml_job, [98](#)
 list_capacity_providers, [23](#)
 list_cluster_events, [98](#)
 list_cluster_nodes, [98](#)
 list_cluster_scheduler_configs, [98](#)
 list_clusters, [98](#)
 list_code_interpreter_sessions, [18](#)
 list_code_interpreters, [23](#)
 list_code_repositories, [98](#)
 list_collections, [85](#), [90](#)
 list_compilation_jobs, [98](#)
 list_compute_quotas, [98](#)
 list_configuration_bundle_versions, [23](#)
 list_configuration_bundles, [23](#)
 list_consent_portals, [23](#)
 list_contexts, [98](#)
 list_custom_model_deployments, [10](#)
 list_custom_models, [10](#)
 list_custom_vocabulary_items, [61](#)
 list_data_automation_libraries, [30](#)
 list_data_automation_library_entities, [30](#)
 list_data_automation_library_ingestion_jobs, [30](#)
 list_data_automation_projects, [30](#)
 list_data_deletion_jobs, [76](#)
 list_data_ingestion_jobs, [69](#)
 list_data_quality_job_definitions, [98](#)
 list_data_sources, [14](#)
 list_dataset_entries, [86](#), [90](#)
 list_dataset_examples, [23](#)
 list_dataset_export_jobs, [76](#)
 list_dataset_groups, [48](#), [76](#)
 list_dataset_import_jobs, [49](#), [76](#)
 list_dataset_labels, [86](#), [90](#)
 list_dataset_versions, [23](#)
 list_datasets, [23](#), [39](#), [49](#), [69](#), [76](#)
 list_device_fleets, [98](#)
 list_devices, [98](#)
 list_document_classification_jobs, [39](#)
 list_document_classifier_summaries, [39](#)
 list_document_classifiers, [39](#)
 list_domains, [98](#), [126](#)
 list_dominant_language_detection_jobs, [39](#)
 list_earth_observation_jobs, [109](#)
 list_edge_deployment_plans, [98](#)
 list_edge_packaging_jobs, [98](#)
 list_endpoint_configs, [98](#)
 list_endpoints, [39](#), [98](#)
 list_enforced_guardrails_configuration, [10](#)
 list_entities_detection_jobs, [39](#)
 list_entities_detection_v2_jobs, [42](#)
 list_entity_recognizer_summaries, [39](#)
 list_entity_recognizers, [39](#)
 list_evaluation_jobs, [10](#)
 list_evaluators, [23](#)
 list_event_predictions, [53](#)
 list_event_trackers, [76](#)
 list_events, [18](#)
 list_events_detection_jobs, [39](#)
 list_experiments, [98](#)
 list_explainabilities, [49](#)
 list_explainability_exports, [49](#)
 list_exports, [61](#)

list_faces, [85, 90](#)
list_feature_groups, [98](#)
list_filters, [76](#)
list_flow_aliases, [14](#)
list_flow_definitions, [99](#)
list_flow_execution_events, [27](#)
list_flow_executions, [27](#)
list_flow_versions, [14](#)
list_flows, [14](#)
list_flywheel_iteration_history, [39](#)
list_flywheels, [39](#)
list_forecast_export_jobs, [49](#)
list_forecasts, [49](#)
list_foundation_model_agreement_offers, [10](#)
list_foundation_models, [10](#)
list_fraudster_registration_jobs, [126](#)
list_fraudsters, [126](#)
list_gateway_rate_limits, [23](#)
list_gateway_rules, [23](#)
list_gateway_targets, [23](#)
list_gateways, [23](#)
list_guardrails, [10](#)
list_harness_endpoints, [23](#)
list_harness_versions, [23](#)
list_harnesses, [23](#)
list_hub_content_versions, [99](#)
list_hub_contents, [99](#)
list_hubs, [99](#)
list_human_loops, [6](#)
list_human_task_uis, [99](#)
list_hyper_parameter_tuning_jobs, [99](#)
list_icd10cm_inference_jobs, [43](#)
list_image_versions, [99](#)
list_images, [99](#)
list_imported_models, [10](#)
list_imports, [61](#)
list_inference_components, [99](#)
list_inference_events, [69](#)
list_inference_executions, [69](#)
list_inference_experiments, [99](#)
list_inference_profiles, [10](#)
list_inference_recommendations_job_steps, [99](#)
list_inference_recommendations_jobs, [99](#)
list_inference_schedulers, [69](#)
list_ingestion_jobs, [14](#)
list_intent_metrics, [61](#)
list_intent_paths, [61](#)
list_intent_stage_metrics, [61](#)
list_intents, [61](#)
list_invocation_steps, [27](#)
list_invocations, [27](#)
list_job_schema_versions, [99](#)
list_jobs, [99](#)
list_key_phrases_detection_jobs, [39](#)
list_knowledge_base_documents, [14](#)
list_knowledge_bases, [14](#)
list_label_groups, [69](#)
list_labeling_jobs, [99](#)
list_labeling_jobs_for_workteam, [99](#)
list_labels, [69](#)
list_language_models, [120](#)
list_languages, [123](#)
list_lexicons, [84](#)
list_lineage_groups, [99](#)
list_marketplace_model_endpoints, [10](#)
list_media_analysis_jobs, [90](#)
list_medical_scribe_jobs, [120](#)
list_medical_transcription_jobs, [120](#)
list_medical_vocabularies, [120](#)
list_memories, [23](#)
list_memory_extraction_jobs, [18](#)
list_memory_records, [18](#)
list_metric_attribution_metrics, [76](#)
list_metric_attributions, [76](#)
list_mlflow_apps, [99](#)
list_mlflow_tracking_servers, [99](#)
list_model_bias_job_definitions, [99](#)
list_model_card_export_jobs, [99](#)
list_model_card_versions, [99](#)
list_model_cards, [99](#)
list_model_copy_jobs, [10](#)
list_model_customization_jobs, [10](#)
list_model_explainability_job_definitions, [99](#)
list_model_import_jobs, [10](#)
list_model_invocation_jobs, [10](#)
list_model_metadata, [99](#)
list_model_package_groups, [99](#)
list_model_packages, [99](#)
list_model_quality_job_definitions, [99](#)
list_model_versions, [69](#)
list_models, [69, 99](#)
list_monitor_evaluations, [49](#)

- [list_monitoring_alert_history, 99](#)
- [list_monitoring_alerts, 99](#)
- [list_monitoring_executions, 99](#)
- [list_monitoring_schedules, 99](#)
- [list_monitors, 49](#)
- [list_notebook_instance_lifecycle_configs, 99](#)
- [list_notebook_instances, 99](#)
- [list_oauth2_credential_providers, 24](#)
- [list_online_evaluation_configs, 24](#)
- [list_optimization_jobs, 99](#)
- [list_parallel_data, 123](#)
- [list_partner_apps, 99](#)
- [list_payment_connectors, 24](#)
- [list_payment_credential_providers, 24](#)
- [list_payment_instruments, 18](#)
- [list_payment_managers, 24](#)
- [list_payment_sessions, 18](#)
- [list_phi_detection_jobs, 43](#)
- [list_pii_entities_detection_jobs, 39](#)
- [list_pipeline_execution_steps, 99](#)
- [list_pipeline_executions, 99](#)
- [list_pipeline_parameters_for_execution, 99](#)
- [list_pipeline_versions, 99](#)
- [list_pipelines, 99](#)
- [list_policies, 24](#)
- [list_policy_engine_summaries, 24](#)
- [list_policy_engines, 24](#)
- [list_policy_generation_assets, 24](#)
- [list_policy_generation_summaries, 24](#)
- [list_policy_generations, 24](#)
- [list_policy_summaries, 24](#)
- [list_predictor_backtest_export_jobs, 49](#)
- [list_predictors, 49](#)
- [list_processing_jobs, 99](#)
- [list_project_policies, 86, 90](#)
- [list_projects, 99](#)
- [list_prompt_routers, 10](#)
- [list_prompts, 14](#)
- [list_provisioned_model_throughputs, 10](#)
- [list_raster_data_collections, 109](#)
- [list_recipes, 76](#)
- [list_recommendations, 18](#)
- [list_recommended_intents, 61](#)
- [list_recommenders, 76](#)
- [list_records, 106](#)
- [list_registries, 24](#)
- [list_registry_records, 24](#)
- [list_resource_catalogs, 99](#)
- [list_retraining_schedulers, 70](#)
- [list_rx_norm_inference_jobs, 43](#)
- [list_schemas, 76](#)
- [list_sensor_statistics, 70](#)
- [list_sentiment_detection_jobs, 39](#)
- [list_session_analytics_data, 61](#)
- [list_session_metrics, 61](#)
- [list_sessions, 18, 27](#)
- [list_slot_types, 61](#)
- [list_slots, 61](#)
- [list_snomedct_inference_jobs, 43](#)
- [list_solution_versions, 76](#)
- [list_solutions, 76](#)
- [list_spaces, 99](#)
- [list_speaker_enrollment_jobs, 126](#)
- [list_speakers, 126](#)
- [list_speech_synthesis_tasks, 84](#)
- [list_stage_devices, 99](#)
- [list_stream_processors, 87, 90](#)
- [list_studio_lifecycle_configs, 99](#)
- [list_subscribed_workteams, 100](#)
- [list_tags, 100](#)
- [list_tags_for_resource, 10, 14, 24, 27, 30, 33, 39, 49, 53, 56, 61, 70, 77, 90, 109, 117, 120, 123, 126](#)
- [list_targeted_sentiment_detection_jobs, 39](#)
- [list_terminologies, 123](#)
- [list_test_execution_result_items, 61](#)
- [list_test_executions, 61](#)
- [list_test_set_records, 61](#)
- [list_test_sets, 61](#)
- [list_text_translation_jobs, 123](#)
- [list_topics_detection_jobs, 39](#)
- [list_training_jobs, 100](#)
- [list_training_jobs_for_hyper_parameter_tuning_job, 100](#)
- [list_training_plans, 100](#)
- [list_transcription_jobs, 120](#)
- [list_transform_jobs, 100](#)
- [list_trial_components, 100](#)
- [list_trials, 100](#)
- [list_ultra_servers_by_reserved_capacity, 100](#)
- [list_user_profiles, 100](#)

- list_users, [85, 90](#)
- list_utterance_analytics_data, [61](#)
- list_utterance_metrics, [61](#)
- list_vector_enrichment_jobs, [109](#)
- list_vocabularies, [120](#)
- list_vocabulary_filters, [120](#)
- list_vpc_configurations, [15](#)
- list_watchlists, [126](#)
- list_what_if_analyses, [49](#)
- list_what_if_forecast_exports, [49](#)
- list_what_if_forecasts, [49](#)
- list_workforces, [100](#)
- list_workload_identities, [24](#)
- list_workteams, [100](#)
- lookoutequipment, [67](#)

- machinelearning, [70](#)

- opt_out_speaker, [126](#)
- optimize_prompt, [27](#)

- personalize, [73](#)
- personalizeevents, [77](#)
- personalizeruntime, [80](#)
- polly, [82](#)
- post_content, [64](#)
- post_text, [64](#)
- predict, [73](#)
- prepare_agent, [15](#)
- prepare_flow, [15](#)
- process_payment, [18](#)
- put_account_data_retention, [10](#)
- put_action_interactions, [79](#)
- put_actions, [79](#)
- put_bot, [56](#)
- put_bot_alias, [56](#)
- put_detector, [53](#)
- put_enforced_guardrail_configuration, [10](#)
- put_entity_type, [53](#)
- put_event_type, [53](#)
- put_events, [79](#)
- put_external_model, [53](#)
- put_intent, [56](#)
- put_invocation_step, [27](#)
- put_items, [79](#)
- put_kms_encryption_key, [53](#)
- put_label, [53](#)
- put_lexicon, [84](#)
- put_model_invocation_logging_configuration, [10](#)
- put_model_package_group_policy, [100](#)
- put_outcome, [53](#)
- put_project_policy, [86, 90](#)
- put_record, [106](#)
- put_resource_policy, [10, 15, 24, 39, 70](#)
- put_session, [64, 66](#)
- put_slot_type, [56](#)
- put_use_case_for_model_access, [10](#)
- put_users, [79](#)

- query_forecast, [45](#)
- query_lineage, [100](#)
- query_what_if_forecast, [45](#)

- recognize_celebrities, [85, 90](#)
- recognize_text, [66](#)
- recognize_utterance, [66](#)
- register_devices, [100](#)
- register_marketplace_model_endpoint, [10](#)
- rekognition, [85](#)
- render_ui_template, [100](#)
- rerank, [28](#)
- resume_resource, [49](#)
- retrieve, [28](#)
- retrieve_and_generate, [28](#)
- retrieve_and_generate_stream, [28](#)
- retrieve_memory_records, [18](#)
- retry_pipeline_execution, [100](#)
- rotate_payment_connector_credentials, [24](#)

- sagemaker, [91](#)
- sagemakeredgemanager, [102](#)
- sagemakerfeaturestoreruntime, [104](#)
- sagemakergeospatialcapabilities, [107](#)
- sagemakermetrics, [110](#)
- sagemakerruntime, [112](#)
- save_browser_session_profile, [18](#)
- search, [100](#)
- search_associated_transcripts, [61](#)
- search_faces, [85, 90](#)
- search_faces_by_image, [85, 90](#)
- search_raster_data_collection, [109](#)
- search_registry_records, [18](#)
- search_training_plan_offerings, [100](#)
- search_users, [85, 90](#)

- search_users_by_image, [85, 90](#)
- send_event, [53](#)
- send_heartbeat, [104](#)
- send_pipeline_execution_step_failure, [100](#)
- send_pipeline_execution_step_success, [100](#)
- set_token_vault_cmek, [24](#)
- start_async_invoke, [36](#)
- start_automated_reasoning_policy_build_workflow, [10](#)
- start_automated_reasoning_policy_test_workflow, [11](#)
- start_batch_evaluation, [18](#)
- start_bot_analyzer, [61](#)
- start_bot_recommendation, [61](#)
- start_bot_resource_generation, [61](#)
- start_browser_session, [18](#)
- start_call_analytics_job, [120](#)
- start_celebrity_recognition, [86, 90](#)
- start_cluster_health_check, [100](#)
- start_code_interpreter_session, [18](#)
- start_content_moderation, [86, 90](#)
- start_conversation, [66](#)
- start_data_ingestion_job, [70](#)
- start_document_analysis, [117](#)
- start_document_classification_job, [39](#)
- start_document_text_detection, [117](#)
- start_dominant_language_detection_job, [39](#)
- start_earth_observation_job, [109](#)
- start_edge_deployment_stage, [100](#)
- start_entities_detection_job, [39](#)
- start_entities_detection_v2_job, [43](#)
- start_events_detection_job, [39](#)
- start_expense_analysis, [117](#)
- start_face_detection, [86, 90](#)
- start_face_search, [86, 90](#)
- start_flow_execution, [28](#)
- start_flywheel_iteration, [39](#)
- start_fraudster_registration_job, [126](#)
- start_human_loop, [4, 6](#)
- start_icd10cm_inference_job, [43](#)
- start_import, [56, 61](#)
- start_inference_experiment, [100](#)
- start_inference_scheduler, [70](#)
- start_ingestion_job, [15](#)
- start_key_phrases_detection_job, [39](#)
- start_label_detection, [86, 90](#)
- start_lending_analysis, [117](#)
- start_media_analysis_job, [86, 90](#)
- start_medical_scribe_job, [120](#)
- start_medical_transcription_job, [120](#)
- start_memory_extraction_job, [18](#)
- start_migration, [56](#)
- start_mlflow_tracking_server, [100](#)
- start_monitoring_schedule, [100](#)
- start_notebook_instance, [100](#)
- start_person_tracking, [87, 90](#)
- start_phi_detection_job, [43](#)
- start_pii_entities_detection_job, [39](#)
- start_pipeline_execution, [100](#)
- start_policy_generation, [24](#)
- start_project_version, [86, 90](#)
- start_recommendation, [19](#)
- start_recommender, [77](#)
- start_retraining_scheduler, [70](#)
- start_rx_norm_inference_job, [43](#)
- start_segment_detection, [87, 90](#)
- start_sentiment_detection_job, [39](#)
- start_session, [100](#)
- start_snomedct_inference_job, [43](#)
- start_speaker_enrollment_job, [126](#)
- start_speech_synthesis_stream, [84](#)
- start_speech_synthesis_task, [84](#)
- start_stream_processor, [87, 90](#)
- start_targeted_sentiment_detection_job, [39](#)
- start_test_execution, [61](#)
- start_test_set_generation, [61](#)
- start_text_detection, [87, 90](#)
- start_text_translation_job, [123](#)
- start_topics_detection_job, [39](#)
- start_transcription_job, [120](#)
- start_vector_enrichment_job, [109](#)
- stop_advanced_prompt_optimization_job, [11](#)
- stop_ai_benchmark_job, [100](#)
- stop_ai_recommendation_job, [100](#)
- stop_auto_ml_job, [100](#)
- stop_batch_evaluation, [19](#)
- stop_bot_analyzer, [61](#)
- stop_bot_recommendation, [61](#)
- stop_browser_session, [19](#)
- stop_code_interpreter_session, [19](#)
- stop_compilation_job, [100](#)

- stop_dominant_language_detection_job, 39
- stop_earth_observation_job, 109
- stop_edge_deployment_stage, 100
- stop_edge_packaging_job, 100
- stop_entities_detection_job, 39
- stop_entities_detection_v2_job, 43
- stop_evaluation_job, 11
- stop_events_detection_job, 40
- stop_flow_execution, 28
- stop_human_loop, 6
- stop_hyper_parameter_tuning_job, 100
- stop_icd10cm_inference_job, 43
- stop_inference_experiment, 100
- stop_inference_recommendations_job, 100
- stop_inference_scheduler, 70
- stop_ingestion_job, 15
- stop_job, 100
- stop_key_phrases_detection_job, 40
- stop_labeling_job, 100
- stop_mlflow_tracking_server, 100
- stop_model_customization_job, 11
- stop_model_invocation_job, 11
- stop_monitoring_schedule, 100
- stop_notebook_instance, 100
- stop_optimization_job, 100
- stop_phi_detection_job, 43
- stop_pii_entities_detection_job, 40
- stop_pipeline_execution, 100
- stop_processing_job, 100
- stop_project_version, 86, 90
- stop_recommender, 77
- stop_resource, 49
- stop_retraining_scheduler, 70
- stop_runtime_session, 19
- stop_rx_norm_inference_job, 43
- stop_sentiment_detection_job, 40
- stop_snomedct_inference_job, 43
- stop_solution_version_creation, 77
- stop_stream_processor, 87, 90
- stop_targeted_sentiment_detection_job, 40
- stop_text_translation_job, 123
- stop_training_document_classifier, 40
- stop_training_entity_recognizer, 40
- stop_training_job, 100
- stop_transform_job, 100
- stop_vector_enrichment_job, 109
- submit_registry_record_for_approval, 24
- synchronize_gateway_targets, 24
- synthesize_speech, 84
- tag_resource, 11, 15, 24, 28, 30, 33, 40, 49, 53, 56, 61, 70, 77, 90, 109, 117, 120, 123, 126
- textract, 114
- transcribeservice, 117
- translate, 121
- translate_document, 123
- translate_text, 123
- untag_resource, 11, 15, 24, 28, 30, 33, 40, 49, 53, 56, 61, 70, 77, 90, 109, 117, 120, 123, 126
- update_ab_test, 19
- update_action, 101
- update_active_model_version, 70
- update_adapter, 117
- update_agent, 15
- update_agent_action_group, 15
- update_agent_alias, 15
- update_agent_collaborator, 15
- update_agent_knowledge_base, 15
- update_agent_runtime, 24
- update_agent_runtime_endpoint, 24
- update_api_key_credential_provider, 24
- update_app_image_config, 101
- update_artifact, 101
- update_automated_reasoning_policy, 11
- update_automated_reasoning_policy_annotations, 11
- update_automated_reasoning_policy_test_case, 11
- update_batch_prediction, 73
- update_blueprint, 31
- update_bot, 61
- update_bot_alias, 61
- update_bot_locale, 61
- update_bot_recommendation, 61
- update_browser_stream, 19
- update_call_analytics_category, 120
- update_campaign, 77
- update_capacity_provider, 24
- update_cluster, 101
- update_cluster_scheduler_config, 101

- update_cluster_software, [101](#)
- update_code_repository, [101](#)
- update_compute_quota, [101](#)
- update_configuration_bundle, [24](#)
- update_consent_portal, [24](#)
- update_context, [101](#)
- update_custom_model_deployment, [11](#)
- update_data_automation_library, [31](#)
- update_data_automation_project, [31](#)
- update_data_source, [15](#), [73](#)
- update_dataset, [24](#), [77](#)
- update_dataset_entries, [86](#), [90](#)
- update_dataset_examples, [24](#)
- update_dataset_group, [49](#)
- update_detector_version, [53](#)
- update_detector_version_metadata, [53](#)
- update_detector_version_status, [53](#)
- update_device_fleet, [101](#)
- update_devices, [101](#)
- update_domain, [101](#), [126](#)
- update_endpoint, [40](#), [101](#)
- update_endpoint_weights_and_capacities, [101](#)
- update_evaluation, [73](#)
- update_evaluator, [24](#)
- update_event_label, [53](#)
- update_experiment, [101](#)
- update_export, [61](#)
- update_feature_group, [101](#)
- update_feature_metadata, [101](#)
- update_flow, [15](#)
- update_flow_alias, [15](#)
- update_flywheel, [40](#)
- update_gateway, [24](#)
- update_gateway_rate_limit, [24](#)
- update_gateway_rule, [24](#)
- update_gateway_target, [24](#)
- update_guardrail, [11](#)
- update_harness, [24](#)
- update_harness_endpoint, [24](#)
- update_hub, [101](#)
- update_hub_content, [101](#)
- update_hub_content_reference, [101](#)
- update_image, [101](#)
- update_image_version, [101](#)
- update_inference_component, [101](#)
- update_inference_component_runtime_config, [101](#)
- update_inference_experiment, [101](#)
- update_inference_scheduler, [70](#)
- update_intent, [61](#)
- update_knowledge_base, [15](#)
- update_label_group, [70](#)
- update_language_model, [120](#)
- update_list, [53](#)
- update_marketplace_model_endpoint, [11](#)
- update_medical_vocabulary, [120](#)
- update_memory, [24](#)
- update_metric_attribution, [77](#)
- update_ml_model, [73](#)
- update_mlflow_app, [101](#)
- update_mlflow_tracking_server, [101](#)
- update_model, [53](#), [70](#)
- update_model_card, [101](#)
- update_model_package, [101](#)
- update_model_version, [53](#)
- update_model_version_status, [53](#)
- update_monitoring_alert, [101](#)
- update_monitoring_schedule, [101](#)
- update_notebook_instance, [101](#)
- update_notebook_instance_lifecycle_config, [101](#)
- update_oauth_2_credential_provider, [24](#)
- update_online_evaluation_config, [24](#)
- update_parallel_data, [123](#)
- update_partner_app, [101](#)
- update_payment_connector, [24](#)
- update_payment_credential_provider, [24](#)
- update_payment_manager, [24](#)
- update_pipeline, [101](#)
- update_pipeline_execution, [101](#)
- update_pipeline_version, [101](#)
- update_policy, [24](#)
- update_policy_engine, [24](#)
- update_project, [101](#)
- update_prompt, [15](#)
- update_provisioned_model_throughput, [11](#)
- update_recommender, [77](#)
- update_record, [106](#)
- update_registry, [24](#)
- update_registry_record, [25](#)
- update_registry_record_status, [25](#)
- update_resource_policy, [61](#)
- update_retraining_scheduler, [70](#)
- update_rule_metadata, [53](#)

- update_rule_version, [53](#)
- update_session, [28](#)
- update_slot, [61](#)
- update_slot_type, [61](#)
- update_solution, [77](#)
- update_space, [101](#)
- update_stream_processor, [87](#), [90](#)
- update_test_set, [61](#)
- update_training_job, [101](#)
- update_trial, [101](#)
- update_trial_component, [101](#)
- update_user_profile, [101](#)
- update_variable, [53](#)
- update_vocabulary, [120](#)
- update_vocabulary_filter, [120](#)
- update_watchlist, [126](#)
- update_workforce, [101](#)
- update_workload_identity, [25](#)
- update_workteam, [101](#)

- validate_flow_definition, [15](#)
- voiceid, [123](#)