Package: paws.machine.learning (via r-universe)

October 28, 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

Roxygen list(markdown = TRUE, roclets = c(``rd", ``namespace", ``collate"))

Config/roxygen2/version 8.1.0

Collate 'augmentedairuntime_service.R' 'augmentedairuntime_interfaces.R' 'augmentedairuntime_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'
'textract_operations.R' 'transcribeservice_service.R'
'transcribeservice_interfaces.R'
'transcribeservice_operations.R' 'translate_service.R'
'translate_interfaces.R' 'translate_operations.R'
'voiceid_service.R' 'voiceid_interfaces.R'
'voiceid_operations.R'

Config/pak/sysreqs libxml2-dev libssl-dev

Repository https://paws-r.r-universe.dev

Date/Publication 2026-10-28 19:14:01 UTC

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

RemoteRef HEAD

RemoteSha 1341bca0616aa9ee814720f471634b968e45e133

RemoteSubdir cran/paws.machine.learning

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
lexruntimeservice . . . 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
textract 114
transcribeservice 117
translate 121
voiceid 123

Index **127**

`augmentedairuntime` *Amazon Augmented AI Runtime*

Description

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

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

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

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

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

Usage

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

Arguments

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

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

Description

Describes the API operations for creating, managing, fine-turning, 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. `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-e
  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 <- 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

batch_delete_advanced_prompt_optimization_job	Deletes one or more advanced prompt optimization jobs
batch_delete_evaluation_job	Deletes a batch of evaluation jobs
cancel_automated_reasoning_policy_build_workflow	Cancels a running Automated Reasoning policy build work
create_advanced_prompt_optimization_job	Creates an advanced prompt optimization job
create_automated_reasoning_policy	Creates an Automated Reasoning policy for Amazon Bedro
create_automated_reasoning_policy_test_case	Creates a test for an Automated Reasoning policy
create_automated_reasoning_policy_version	Creates a new version of an existing Automated Reasoning
create_custom_model	Creates a new custom model in Amazon Bedrock
create_custom_model_deployment	Deploys a custom model for on-demand inference in Amaz
create_evaluation_job	Creates an evaluation job
create_foundation_model_agreement	Request a model access agreement for the specified model
create_guardrail	Creates a guardrail to block topics and to implement safegu
create_guardrail_version	Creates a version of the guardrail
create_inference_profile	Creates an application inference profile to track metrics and
create_marketplace_model_endpoint	Creates an endpoint for a model from Amazon Bedrock Ma
create_model_copy_job	Copies a model to another region so that it can be used ther
create_model_customization_job	Creates a fine-tuning job to customize a base model
create_model_import_job	Creates a model import job to import model that you have c
create_model_invocation_job	Creates a batch inference job to invoke a model on multiple
create_prompt_router	Creates a prompt router that manages the routing of request
create_provisioned_model_throughput	Creates dedicated throughput for a base or custom model w
delete_automated_reasoning_policy	Deletes an Automated Reasoning policy or policy version
delete_automated_reasoning_policy_build_workflow	Deletes an Automated Reasoning policy build workflow an
delete_automated_reasoning_policy_test_case	Deletes an Automated Reasoning policy test
delete_custom_model	Deletes a custom model that you created earlier
delete_custom_model_deployment	Deletes a custom model deployment
delete_enforced_guardrail_configuration	Deletes the account-level enforced guardrail configuration
delete_foundation_model_agreement	Delete the model access agreement for the specified model
delete_guardrail	Deletes a guardrail
delete_imported_model	Deletes a custom model that you imported earlier
delete_inference_profile	Deletes an application inference profile
delete_marketplace_model_endpoint	Deletes an endpoint for a model from Amazon Bedrock Ma
delete_model_invocation_logging_configuration	Delete the invocation logging
delete_prompt_router	Deletes a specified prompt router
delete_provisioned_model_throughput	Deletes a Provisioned Throughput
delete_resource_policy	Deletes a previously created Bedrock resource policy
deregister_marketplace_model_endpoint	Deregisters an endpoint for a model from Amazon Bedrock
export_automated_reasoning_policy_version	Exports the policy definition for an Automated Reasoning p
get_account_data_retention	Returns the account-wide data retention mode for Amazon
get_advanced_prompt_optimization_job	Gets information about an advanced prompt optimization jo
get_automated_reasoning_policy	Retrieves details about an Automated Reasoning policy or p
get_automated_reasoning_policy_annotations	Retrieves the current annotations for an Automated Reason
get_automated_reasoning_policy_build_workflow	Retrieves detailed information about an Automated Reason
get_automated_reasoning_policy_build_workflow_result_assets	Retrieves the resulting assets from a completed Automated
get_automated_reasoning_policy_next_scenario	Retrieves the next test scenario for validating an Automated
get_automated_reasoning_policy_test_case	Retrieves details about a specific Automated Reasoning pol

get_automated_reasoning_policy_test_result	Retrieves the test result for a specific Automated Reasoning
get_custom_model	Get the properties associated with a Amazon Bedrock custo
get_custom_model_deployment	Retrieves information about a custom model deployment, in
get_evaluation_job	Gets information about an evaluation job, such as the status
get_foundation_model	Get details about a Amazon Bedrock foundation model
get_foundation_model_availability	Get information about the Foundation model availability
get_guardrail	Gets details about a guardrail
get_imported_model	Gets properties associated with a customized model you im
get_inference_profile	Gets information about an inference profile
get_marketplace_model_endpoint	Retrieves details about a specific endpoint for a model from
get_model_copy_job	Retrieves information about a model copy job
get_model_customization_job	Retrieves the properties associated with a model-customiza
get_model_import_job	Retrieves the properties associated with import model job, i
get_model_invocation_job	Gets details about a batch inference job
get_model_invocation_logging_configuration	Get the current configuration values for model invocation lo
get_prompt_router	Retrieves details about a prompt router
get_provisioned_model_throughput	Returns details for a Provisioned Throughput
get_resource_policy	Gets the resource policy document for a Bedrock resource
get_use_case_for_model_access	Get usecase for model access
list_advanced_prompt_optimization_jobs	Lists the advanced prompt optimization jobs in your accour
list_automated_reasoning_policies	Lists all Automated Reasoning policies in your account, wi
list_automated_reasoning_policy_build_workflows	Lists all build workflows for an Automated Reasoning polic
list_automated_reasoning_policy_test_cases	Lists tests for an Automated Reasoning policy
list_automated_reasoning_policy_test_results	Lists test results for an Automated Reasoning policy, showi
list_custom_model_deployments	Lists custom model deployments in your account
list_custom_models	Returns a list of the custom models that you have created w
list_enforced_guardrails_configuration	Lists the account-level enforced guardrail configurations
list_evaluation_jobs	Lists all existing evaluation jobs
list_foundation_model_agreement_offers	Get the offers associated with the specified model
list_foundation_models	Lists Amazon Bedrock foundation models that you can use
list_guardrails	Lists details about all the guardrails in an account
list_imported_models	Returns a list of models you've imported
list_inference_profiles	Returns a list of inference profiles that you can use
list_marketplace_model_endpoints	Lists the endpoints for models from Amazon Bedrock Marl
list_model_copy_jobs	Returns a list of model copy jobs that you have submitted
list_model_customization_jobs	Returns a list of model customization jobs that you have su
list_model_import_jobs	Returns a list of import jobs you've submitted
list_model_invocation_jobs	Lists all batch inference jobs in the account
list_prompt_routers	Retrieves a list of prompt routers
list_provisioned_model_throughputs	Lists the Provisioned Throughputs in the account
list_tags_for_resource	List the tags associated with the specified resource
put_account_data_retention	Sets the account-wide data retention mode for Amazon Bed
put_enforced_guardrail_configuration	Sets the account-level enforced guardrail configuration
put_model_invocation_logging_configuration	Set the configuration values for model invocation logging
put_resource_policy	Adds a resource policy for a Bedrock resource
put_use_case_for_model_access	Put usecase for model access
register_marketplace_model_endpoint	Registers an existing Amazon SageMaker endpoint with Ar
start_automated_reasoning_policy_build_workflow	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 progre
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 moc
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` *Agents for Amazon Bedrock*

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. `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-e
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 <- 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

create_vpc_configuration	Creates a VPC configuration that lets a knowledge base connect to a resource in your p
delete_agent	Deletes an agent
delete_agent_action_group	Deletes an action group in an agent
delete_agent_alias	Deletes an alias of an agent
delete_agent_version	Deletes a version of an agent
delete_data_source	Deletes a data source from a knowledge base
delete_flow	Deletes a flow
delete_flow_alias	Deletes an alias of a flow
delete_flow_version	Deletes a version of a flow
delete_knowledge_base	Deletes a knowledge base
delete_knowledge_base_documents	Deletes documents from a data source and syncs the changes to the knowledge base tha
delete_prompt	Deletes a prompt or a version of it, depending on whether you include the promptVersi
delete_resource_policy	Removes the resource policy associated with a knowledge base
delete_vpc_configuration	Deletes a VPC configuration
disassociate_agent_collaborator	Disassociates an agent collaborator
disassociate_agent_knowledge_base	Disassociates a knowledge base from an agent
get_agent	Gets information about an agent
get_agent_action_group	Gets information about an action group for an agent
get_agent_alias	Gets information about an alias of an agent
get_agent_collaborator	Retrieves information about an agent's collaborator
get_agent_knowledge_base	Gets information about a knowledge base associated with an agent
get_agent_version	Gets details about a version of an agent
get_data_source	Gets information about a data source
get_flow	Retrieves information about a flow
get_flow_alias	Retrieves information about a flow
get_flow_version	Retrieves information about a version of a flow
get_ingestion_job	Gets information about a data ingestion job
get_knowledge_base	Gets information about a knowledge base
get_knowledge_base_documents	Retrieves specific documents from a data source that is connected to a knowledge base
get_prompt	Retrieves information about the working draft (DRAFT version) of a prompt or a versi
get_resource_policy	Retrieves the resource policy associated with a knowledge base
get_vpc_configuration	Returns the details and current status of a single VPC configuration
ingest_knowledge_base_documents	Ingests documents directly into the knowledge base that is connected to the data source
list_agent_action_groups	Lists the action groups for an agent and information about each one
list_agent_aliases	Lists the aliases of an agent and information about each one
list_agent_collaborators	Retrieve a list of an agent's collaborators
list_agent_knowledge_bases	Lists knowledge bases associated with an agent and information about each one
list_agents	Lists the agents belonging to an account and information about each agent
list_agent_versions	Lists the versions of an agent and information about each version
list_data_sources	Lists the data sources in a knowledge base and information about each one
list_flow_aliases	Returns a list of aliases for a flow
list_flows	Returns a list of flows and information about each flow
list_flow_versions	Returns a list of information about each flow
list_ingestion_jobs	Lists the data ingestion jobs for a data source
list_knowledge_base_documents	Retrieves all the documents contained in a data source that is connected to a knowledge
list_knowledge_bases	Lists the knowledge bases in an account
list_prompts	Returns either information about the working draft (DRAFT version) of each prompt in
list_tags_for_resource	List all the tags for the resource you specify

list_vpc_configurations	Returns a paginated list of the VPC configurations for a knowledge base
prepare_agent	Creates a DRAFT version of the agent that can be used for internal testing
prepare_flow	Prepares the DRAFT version of a flow so that it can be invoked
put_resource_policy	Associates a resource policy with a knowledge base
start_ingestion_job	Begins a data ingestion job
stop_ingestion_job	Stops a currently running data ingestion job
tag_resource	Associate tags with a resource
untag_resource	Remove tags from a resource
update_agent	Updates the configuration of an agent
update_agent_action_group	Updates the configuration for an action group for an agent
update_agent_alias	Updates configurations for an alias of an agent
update_agent_collaborator	Updates an agent's collaborator
update_agent_knowledge_base	Updates the configuration for a knowledge base that has been associated with an agent
update_data_source	Updates the configurations for a data source connector
update_flow	Modifies a flow
update_flow_alias	Modifies the alias of a flow
update_knowledge_base	Updates the configuration of a knowledge base with the fields that you specify
update_prompt	Modifies a prompt in your prompt library
validate_flow_definition	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. `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-e 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 <- 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	Creates multiple memory records in a single batch operation for the specified mem
batch_delete_memory_records	Deletes multiple memory records in a single batch operation from the specified me
batch_update_memory_records	Updates multiple memory records with custom content in a single batch operation
complete_resource_token_auth	Confirms the user authentication session for obtaining OAuth2
create_ab_test	Creates an A/B test for comparing agent configurations
create_event	Creates an event in an AgentCore Memory resource
create_payment_instrument	Create a new payment instrument for a connector
create_payment_session	Create a new payment session
delete_ab_test	Deletes an A/B test and its associated gateway rules
delete_batch_evaluation	Deletes a batch evaluation and its associated results
delete_capacity_provider_session	Deletes a session associated with a capacity provider in Amazon Bedrock AgentC
delete_event	Deletes an event from an AgentCore Memory resource

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

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, evaluatic
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. `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-e 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 <- 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_oauth_2_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

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

get_evaluator	Retrieves detailed information about an evaluator, including its configur
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 AgentC
get_policy_engine	Retrieves detailed information about a specific policy engine within the
get_policy_engine_summary	Retrieves a metadata-only summary of a specific policy engine without
get_policy_generation	Retrieves information about a policy generation request within the Agen
get_policy_generation_summary	Retrieves a metadata-only summary of a specific policy generation requ
get_policy_summary	Retrieves a metadata-only summary of a specific policy without decrypt
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 prov
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 inform
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 br
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 desce
list_evaluators	Lists all available evaluators, including both builtin evaluators provided
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 th

list_oauth_2_credential_providers	Lists all OAuth2 credential providers in your account
list_online_evaluation_configs	Lists all online evaluation configurations in the account, providing summ
list_payment_connectors	Lists all payment connectors for a specified payment manager
list_payment_credential_providers	Lists all payment credential providers in the account
list_payment_managers	Lists all payment managers in the account
list_policies	Retrieves a list of policies within the AgentCore Policy engine
list_policy_engines	Retrieves a list of policy engines within the AgentCore Policy system
list_policy_engine_summaries	Retrieves a paginated list of metadata-only policy engine summaries wit
list_policy_generation_assets	Retrieves a list of generated policy assets from a policy generation reque
list_policy_generations	Retrieves a list of policy generation requests within the AgentCore Polic
list_policy_generation_summaries	Retrieves a paginated list of metadata-only policy generation summaries
list_policy_summaries	Retrieves a paginated list of metadata-only policy summaries within a po
list_registries	Lists all registries in the account
list_registry_records	Lists registry records within a registry
list_tags_for_resource	Lists the tags associated with the specified resource
list_workload_identities	Lists all workload identities in your account
put_resource_policy	Creates or updates a resource-based policy for a resource with the specif
rotate_payment_connector_credentials	Replaces the service-managed credentials of a payment connector with n
set_token_vault_cmk	Sets the customer master key (CMK) for a token vault
start_policy_generation	Initiates the AI-powered generation of Dogwood policies from natural la
submit_registry_record_for_approval	Submits a registry record for approval
synchronize_gateway_targets	Synchronizes the gateway targets by fetching the latest tool definitions f
tag_resource	Associates the specified tags to a resource with the specified resourceAr
untag_resource	Removes the specified tags from the specified resource
update_agent_runtime	Updates an existing Amazon Secure Agent
update_agent_runtime_endpoint	Updates an existing Amazon Bedrock AgentCore Runtime endpoint
update_api_key_credential_provider	Updates an existing API key credential provider
update_capacity_provider	Updates a capacity provider
update_configuration_bundle	Updates a configuration bundle by creating a new version with the speci
update_consent_portal	Updates an existing consent portal
update_dataset	Updates a dataset's metadata
update_dataset_examples	Updates multiple existing examples in-place on DRAFT
update_evaluator	Updates a custom evaluator's configuration, description, or evaluation le
update_gateway	Updates an existing gateway
update_gateway_rate_limit	Updates the entries of a gateway rate limit
update_gateway_rule	Updates a gateway rule's priority, conditions, actions, or description
update_gateway_target	Updates an existing gateway target
update_harness	Operation to update a harness
update_harness_endpoint	Operation to update a harness endpoint
update_memory	Update an Amazon Bedrock AgentCore Memory resource memory
update_oauth_2_credential_provider	Updates an existing OAuth2 credential provider
update_online_evaluation_config	Updates an online evaluation configuration's settings, including rules, da
update_payment_connector	Updates an existing payment connector
update_payment_credential_provider	Updates an existing payment credential provider with new authenticatio
update_payment_manager	Updates an existing payment manager
update_policy	Updates an existing policy within the AgentCore Policy system
update_policy_engine	Updates an existing policy engine within the AgentCore Policy system
update_registry	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. `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-e 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 <- 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 ac
create_invocation	Creates a new invocation within a session
create_session	Creates a session to temporarily store conversations for generative AI (GenAI) applications b
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
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 c
invoke_flow	Invokes an alias of a flow to run the inputs that you specify and return the output of each nod
invoke_inline_agent	Invokes an inline Amazon Bedrock agent using the configurations you provide with the requ
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 se
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 outpu
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` *Data Automation for Amazon Bedrock*

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. `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-e 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 <- 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

copy_blueprint_stage	Copies a Blueprint from one stage to another
create_blueprint	Creates an Amazon Bedrock Data Automation Blueprint
create_blueprint_version	Creates a new version of an existing Amazon Bedrock Data Automation Blu
create_data_automation_library	Creates an Amazon Bedrock Data Automation Library
create_data_automation_project	Creates an Amazon Bedrock Data Automation Project
delete_blueprint	Deletes an existing Amazon Bedrock Data Automation Blueprint
delete_data_automation_library	Deletes an existing Amazon Bedrock Data Automation Library
delete_data_automation_project	Deletes an existing Amazon Bedrock Data Automation Project
get_blueprint	Gets an existing Amazon Bedrock Data Automation Blueprint
get_blueprint_optimization_status	API used to get blueprint optimization status
get_data_automation_library	Gets an existing Amazon Bedrock Data Automation Library
get_data_automation_library_entity	Gets an existing entity based on entity type from the library
get_data_automation_library_ingestion_job	API used to get status of data automation library ingestion job
get_data_automation_project	Gets an existing Amazon Bedrock Data Automation Project
invoke_blueprint_optimization_async	Invoke an async job to perform Blueprint Optimization
invoke_data_automation_library_ingestion_job	Async API: Invoke data automation library ingestion job
list_blueprints	Lists all existing Amazon Bedrock Data Automation Blueprints
list_data_automation_libraries	Lists all existing Amazon Bedrock Data Automation Libraries
list_data_automation_library_entities	Lists all stored entities in the library
list_data_automation_library_ingestion_jobs	Lists all data automation library ingestion jobs
list_data_automation_projects	Lists all existing Amazon Bedrock Data Automation Projects
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

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. `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-e 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 <- 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. `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-e
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 <- 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	The action to apply a guardrail
converse	Sends messages to the specified Amazon Bedrock model
converse_stream	Sends messages to the specified Amazon Bedrock model and returns the response
count_tokens	Returns the token count for a given inference request
get_async_invoke	Retrieve information about an asynchronous invocation
invoke_guardrail_checks	Evaluates messages against inline guardrail checks
invoke_model	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 bidirectio
invoke_model_with_response_stream	Invoke the specified Amazon Bedrock model to run inference using the prompt an
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` *Amazon Comprehend*

Description

Amazon Comprehend is an Amazon Web Services service for gaining insight into the content of documents. Use these actions to determine the topics contained in your documents, the topics they discuss, the predominant sentiment expressed in them, the predominant language used, and more.

Usage

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

Arguments

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

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

  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **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-e 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 <- 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	Determines the dominant language of the input text for a batch of documents
batch_detect_entities	Inspects the text of a batch of documents for named entities and returns informa
batch_detect_key_phrases	Detects the key noun phrases found in a batch of documents
batch_detect_sentiment	Inspects a batch of documents and returns an inference of the prevailing sentim
batch_detect_syntax	Inspects the text of a batch of documents for the syntax and part of speech of th
batch_detect_targeted_sentiment	Inspects a batch of documents and returns a sentiment analysis for each entity i
classify_document	Creates a classification request to analyze a single document in real-time
contains_pii_entities	Analyzes input text for the presence of personally identifiable information (PII)
create_dataset	Creates a dataset to upload training or test data for a model associated with a fl
create_document_classifier	Creates a new document classifier that you can use to categorize documents
create_endpoint	Creates a model-specific endpoint for synchronous inference for a previously tr
create_entity_recognizer	Creates an entity recognizer using submitted files
create_flywheel	A flywheel is an Amazon Web Services resource that orchestrates the ongoing t
delete_document_classifier	Deletes a previously created document classifier
delete_endpoint	Deletes a model-specific endpoint for a previously-trained custom model
delete_entity_recognizer	Deletes an entity recognizer
delete_flywheel	Deletes a flywheel
delete_resource_policy	Deletes a resource-based policy that is attached to a custom model
describe_dataset	Returns information about the dataset that you specify
describe_document_classification_job	Gets the properties associated with a document classification job
describe_document_classifier	Gets the properties associated with a document classifier
describe_dominant_language_detection_job	Gets the properties associated with a dominant language detection job
describe_endpoint	Gets the properties associated with a specific endpoint
describe_entities_detection_job	Gets the properties associated with an entities detection job
describe_entity_recognizer	Provides details about an entity recognizer including status, S3 buckets contain
describe_events_detection_job	Gets the status and details of an events detection job

describe_flywheel	Provides configuration information about the flywheel
describe_flywheel_iteration	Retrieve the configuration properties of a flywheel iteration
describe_key_phrases_detection_job	Gets the properties associated with a key phrases detection job
describe_pii_entities_detection_job	Gets the properties associated with a PII entities detection job
describe_resource_policy	Gets the details of a resource-based policy that is attached to a custom model, i
describe_sentiment_detection_job	Gets the properties associated with a sentiment detection job
describe_targeted_sentiment_detection_job	Gets the properties associated with a targeted sentiment detection job
describe_topics_detection_job	Gets the properties associated with a topic detection job
detect_dominant_language	Determines the dominant language of the input text
detect_entities	Detects named entities in input text when you use the pre-trained model
detect_key_phrases	Detects the key noun phrases found in the text
detect_pii_entities	Inspects the input text for entities that contain personally identifiable informatio
detect_sentiment	Inspects text and returns an inference of the prevailing sentiment (POSITIVE, N
detect_syntax	Inspects text for syntax and the part of speech of words in the document
detect_targeted_sentiment	Inspects the input text and returns a sentiment analysis for each entity identifiec
detect_toxic_content	Performs toxicity analysis on the list of text strings that you provide as input
import_model	Creates a new custom model that replicates a source custom model that you imp
list_datasets	List the datasets that you have configured in this Region
list_document_classification_jobs	Gets a list of the documentation classification jobs that you have submitted
list_document_classifiers	Gets a list of the document classifiers that you have created
list_document_classifier_summaries	Gets a list of summaries of the document classifiers that you have created
list_dominant_language_detection_jobs	Gets a list of the dominant language detection jobs that you have submitted
list_endpoints	Gets a list of all existing endpoints that you've created
list_entities_detection_jobs	Gets a list of the entity detection jobs that you have submitted
list_entity_recognizers	Gets a list of the properties of all entity recognizers that you created, including
list_entity_recognizer_summaries	Gets a list of summaries for the entity recognizers that you have created
list_events_detection_jobs	Gets a list of the events detection jobs that you have submitted
list_flywheel_iteration_history	Information about the history of a flywheel iteration
list_flywheels	Gets a list of the flywheels that you have created
list_key_phrases_detection_jobs	Get a list of key phrase detection jobs that you have submitted
list_pii_entities_detection_jobs	Gets a list of the PII entity detection jobs that you have submitted
list_sentiment_detection_jobs	Gets a list of sentiment detection jobs that you have submitted
list_tags_for_resource	Lists all tags associated with a given Amazon Comprehend resource
list_targeted_sentiment_detection_jobs	Gets a list of targeted sentiment detection jobs that you have submitted
list_topics_detection_jobs	Gets a list of the topic detection jobs that you have submitted
put_resource_policy	Attaches a resource-based policy to a custom model
start_document_classification_job	Starts an asynchronous document classification job using a custom classificatio
start_dominant_language_detection_job	Starts an asynchronous dominant language detection job for a collection of doc
start_entities_detection_job	Starts an asynchronous entity detection job for a collection of documents
start_events_detection_job	Starts an asynchronous event detection job for a collection of documents
start_flywheel_iteration	Start the flywheel iteration
start_key_phrases_detection_job	Starts an asynchronous key phrase detection job for a collection of documents
start_pii_entities_detection_job	Starts an asynchronous PII entity detection job for a collection of documents
start_sentiment_detection_job	Starts an asynchronous sentiment detection job for a collection of documents
start_targeted_sentiment_detection_job	Starts an asynchronous targeted sentiment detection job for a collection of docu
start_topics_detection_job	Starts an asynchronous topic detection job
stop_dominant_language_detection_job	Stops a dominant language detection job in progress
stop_entities_detection_job	Stops an entities detection job in progress

stop_events_detection_job	Stops an events detection job in progress
stop_key_phrases_detection_job	Stops a key phrases detection job in progress
stop_pii_entities_detection_job	Stops a PII entities detection job in progress
stop_sentiment_detection_job	Stops a sentiment detection job in progress
stop_targeted_sentiment_detection_job	Stops a targeted sentiment detection job in progress
stop_training_document_classifier	Stops a document classifier training job while in progress
stop_training_entity_recognizer	Stops an entity recognizer training job while in progress
tag_resource	Associates a specific tag with an Amazon Comprehend resource
untag_resource	Removes a specific tag associated with an Amazon Comprehend resource
update_endpoint	Updates information about the specified endpoint
update_flywheel	Update the configuration information for an existing flywheel

Examples

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

## End(Not run)
```

`comprehendmedical` *AWS Comprehend Medical*

Description

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

Usage

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

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `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-e
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 <- 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 e
infer_icd10cm	InferICD10CM detects medical conditions as entities listed in a patient record and links
infer_rx_norm	InferRxNorm detects medications as entities listed in a patient record and links to the no
infer_snomedct	InferSNOMEDCT detects possible medical concepts as entities and links them to codes
list_entities_detection_v2_jobs	Gets a list of medical entity detection jobs that you have submitted

list_icd10cm_inference_jobs	Gets a list of InferICD10CM jobs that you have submitted
list_phi_detection_jobs	Gets a list of protected health information (PHI) detection jobs you have submitted
list_rx_norm_inference_jobs	Gets a list of InferRxNorm jobs that you have submitted
list_snomedct_inference_jobs	Gets a list of InferSNOMEDCT jobs a user has submitted
start_entities_detection_v2_job	Starts an asynchronous medical entity detection job for a collection of documents
start_icd10cm_inference_job	Starts an asynchronous job to detect medical conditions and link them to the ICD-10-CM
start_phi_detection_job	Starts an asynchronous job to detect protected health information (PHI)
start_rx_norm_inference_job	Starts an asynchronous job to detect medication entities and link them to the RxNorm on
start_snomedct_inference_job	Starts an asynchronous job to detect medical concepts and link them to the SNOMED-C
stop_entities_detection_v2_job	Stops a medical entities detection job in progress
stop_icd10cm_inference_job	Stops an InferICD10CM inference job in progress
stop_phi_detection_job	Stops a protected health information (PHI) detection job in progress
stop_rx_norm_inference_job	Stops an InferRxNorm inference job in progress
stop_snomedct_inference_job	Stops an InferSNOMEDCT inference job in progress

Examples

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

## End(Not run)
```

---

forecastqueryservice *Amazon Forecast Query Service*

Description

Provides APIs for creating and managing Amazon Forecast resources.

Usage

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

Arguments

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

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

Description

Provides APIs for creating and managing Amazon Forecast resources.

Usage

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

Arguments

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

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

create_auto_predictor	Creates an Amazon Forecast predictor
create_dataset	Creates an Amazon Forecast dataset
create_dataset_group	Creates a dataset group, which holds a collection of related datasets
create_dataset_import_job	Imports your training data to an Amazon Forecast dataset
create_explainability	Explainability is only available for Forecasts and Predictors generated from an Aut
create_explainability_export	Exports an Explainability resource created by the CreateExplainability operation
create_forecast	Creates a forecast for each item in the TARGET_TIME_SERIES dataset that was u
create_forecast_export_job	Exports a forecast created by the CreateForecast operation to your Amazon Simple
create_monitor	Creates a predictor monitor resource for an existing auto predictor
create_predictor	This operation creates a legacy predictor that does not include all the predictor func
create_predictor_backtest_export_job	Exports backtest forecasts and accuracy metrics generated by the CreateAutoPredic
create_what_if_analysis	What-if analysis is a scenario modeling technique where you make a hypothetical c
create_what_if_forecast	A what-if forecast is a forecast that is created from a modified version of the baseli
create_what_if_forecast_export	Exports a forecast created by the CreateWhatIfForecast operation to your Amazon
delete_dataset	Deletes an Amazon Forecast dataset that was created using the CreateDataset opera
delete_dataset_group	Deletes a dataset group created using the CreateDatasetGroup operation
delete_dataset_import_job	Deletes a dataset import job created using the CreateDatasetImportJob operation
delete_explainability	Deletes an Explainability resource
delete_explainability_export	Deletes an Explainability export
delete_forecast	Deletes a forecast created using the CreateForecast operation
delete_forecast_export_job	Deletes a forecast export job created using the CreateForecastExportJob operation
delete_monitor	Deletes a monitor resource
delete_predictor	Deletes a predictor created using the DescribePredictor or CreatePredictor operatio
delete_predictor_backtest_export_job	Deletes a predictor backtest export job
delete_resource_tree	Deletes an entire resource tree
delete_what_if_analysis	Deletes a what-if analysis created using the CreateWhatIfAnalysis operation
delete_what_if_forecast	Deletes a what-if forecast created using the CreateWhatIfForecast operation
delete_what_if_forecast_export	Deletes a what-if forecast export created using the CreateWhatIfForecastExport ope
describe_auto_predictor	Describes a predictor created using the CreateAutoPredictor operation
describe_dataset	Describes an Amazon Forecast dataset created using the CreateDataset operation
describe_dataset_group	Describes a dataset group created using the CreateDatasetGroup operation
describe_dataset_import_job	Describes a dataset import job created using the CreateDatasetImportJob operation
describe_explainability	Describes an Explainability resource created using the CreateExplainability operati
describe_explainability_export	Describes an Explainability export created using the CreateExplainabilityExport op
describe_forecast	Describes a forecast created using the CreateForecast operation
describe_forecast_export_job	Describes a forecast export job created using the CreateForecastExportJob operatio
describe_monitor	Describes a monitor resource
describe_predictor	This operation is only valid for legacy predictors created with CreatePredictor
describe_predictor_backtest_export_job	Describes a predictor backtest export job created using the CreatePredictorBacktest
describe_what_if_analysis	Describes the what-if analysis created using the CreateWhatIfAnalysis operation
describe_what_if_forecast	Describes the what-if forecast created using the CreateWhatIfForecast operation
describe_what_if_forecast_export	Describes the what-if forecast export created using the CreateWhatIfForecastExpor
get_accuracy_metrics	Provides metrics on the accuracy of the models that were trained by the CreatePred
list_dataset_groups	Returns a list of dataset groups created using the CreateDatasetGroup operation

list_dataset_import_jobs	Returns a list of dataset import jobs created using the CreateDatasetImportJob oper
list_datasets	Returns a list of datasets created using the CreateDataset operation
list_explainabilities	Returns a list of Explainability resources created using the CreateExplainability ope
list_explainability_exports	Returns a list of Explainability exports created using the CreateExplainabilityExpor
list_forecast_export_jobs	Returns a list of forecast export jobs created using the CreateForecastExportJob ope
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
list_monitors	Returns a list of monitors created with the CreateMonitor operation and CreateAuto
list_predictor_backtest_export_jobs	Returns a list of predictor backtest export jobs created using the CreatePredictorBac
list_predictors	Returns a list of predictors created using the CreateAutoPredictor or CreatePredicto
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 CreateWhatIfForecastEx
list_what_if_forecasts	Returns a list of what-if forecasts created using the CreateWhatIfForecast operation
resume_resource	Resumes a stopped monitor resource
stop_resource	Stops a resource
tag_resource	Associates the specified tags to a resource with the specified resourceArn
untag_resource	Deletes the specified tags from a resource
update_dataset_group	Replaces the datasets in a dataset group with the specified datasets

Examples

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

## End(Not run)
```

`frauddetector` *Amazon Fraud Detector*

Description

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

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

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

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

Usage

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

Arguments

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

- **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-e
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 <- 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 mod
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, dete
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 import
get_kms_encryption_key	Gets the encryption key if a KMS key has been specified to be used to encrypt cont
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

list_event_predictions	Gets a list of past predictions
list_tags_for_resource	Lists all tags associated with the resource
put_detector	Creates or updates a detector
put_entity_type	Creates or updates an entity type
put_event_type	Creates or updates an event type
put_external_model	Creates or updates an Amazon SageMaker model endpoint
put_kms_encryption_key	Specifies the KMS key to be used to encrypt content in Amazon Fraud Detector
put_label	Creates or updates label
put_outcome	Creates or updates an outcome
send_event	Stores events in Amazon Fraud Detector without generating fraud predictions for t
tag_resource	Assigns tags to a resource
untag_resource	Removes tags from a resource
update_detector_version	Updates a detector version
update_detector_version_metadata	Updates the detector version's description
update_detector_version_status	Updates the detector version's status
update_event_label	Updates the specified event with a new label
update_list	Updates a list
update_model	Updates model description
update_model_version	Updates a model version
update_model_version_status	Updates the status of a model version
update_rule_metadata	Updates a rule's metadata
update_rule_version	Updates a rule version resulting in a new rule version
update_variable	Updates a variable

Examples

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

## End(Not run)
```

`lexmodelbuildingservice`

Amazon Lex Model Building Service

Description

Amazon Lex Build-Time Actions

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

Usage

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

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `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-e
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 <- 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 typ
delete_bot	Deletes all versions of the bot, including the $LATEST version
delete_bot_alias	Deletes an alias for the specified bot
delete_bot_channel_association	Deletes the association between an Amazon Lex bot and a messaging platform
delete_bot_version	Deletes a specific version of a bot

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

Examples

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

## End(Not run)
```

`lexmodelsv2` *Amazon Lex Model Building V2*

Description

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. `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-e
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 <- 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 vocabula
batch_delete_custom_vocabulary_item	Delete a batch of custom vocabulary items for a given bot locale's custom vocabula
batch_update_custom_vocabulary_item	Update a batch of custom vocabulary items for a given bot locale's custom vocabul
build_bot_locale	Builds a bot, its intents, and its slot types into a specific locale
create_bot	Creates an Amazon Lex conversational bot
create_bot_alias	Creates an alias for the specified version of a bot
create_bot_locale	Creates a locale in the bot
create_bot_replica	Action to create a replication of the source bot in the secondary region
create_bot_version	Creates an immutable version of the bot
create_export	Creates a zip archive containing the contents of a bot or a bot locale
create_intent	Creates an intent
create_resource_policy	Creates a new resource policy with the specified policy statements

create_resource_policy_statement	Adds a new resource policy statement to a bot or bot alias
create_slot	Creates a slot in an intent
create_slot_type	Creates a custom slot type
create_test_set_discrepancy_report	Create a report that describes the differences between the bot and the test set
create_upload_url	Gets a pre-signed S3 write URL that you use to upload the zip archive when impor
delete_bot	Deletes all versions of a bot, including the Draft version
delete_bot_alias	Deletes the specified bot alias
delete_bot_analyzer_recommendation	Permanently deletes the recommendations and analysis results for a specific bot an
delete_bot_locale	Removes a locale from a bot
delete_bot_replica	The action to delete the replicated bot in the secondary region
delete_bot_version	Deletes a specific version of a bot
delete_custom_vocabulary	Removes a custom vocabulary from the specified locale in the specified bot
delete_export	Removes a previous export and the associated files stored in an S3 bucket
delete_import	Removes a previous import and the associated file stored in an S3 bucket
delete_intent	Removes the specified intent
delete_resource_policy	Removes an existing policy from a bot or bot alias
delete_resource_policy_statement	Deletes a policy statement from a resource policy
delete_slot	Deletes the specified slot from an intent
delete_slot_type	Deletes a slot type from a bot locale
delete_test_set	The action to delete the selected test set
delete_utterances	Deletes stored utterances
describe_bot	Provides metadata information about a bot
describe_bot_alias	Get information about a specific bot alias
describe_bot_analyzer_recommendation	Retrieves the analysis results and recommendations for bot optimization
describe_bot_locale	Describes the settings that a bot has for a specific locale
describe_bot_recommendation	Provides metadata information about a bot recommendation
describe_bot_replica	Monitors the bot replication status through the UI console
describe_bot_resource_generation	Returns information about a request to generate a bot through natural language des
describe_bot_version	Provides metadata about a version of a bot
describe_custom_vocabulary_metadata	Provides metadata information about a custom vocabulary
describe_export	Gets information about a specific export
describe_import	Gets information about a specific import
describe_intent	Returns metadata about an intent
describe_resource_policy	Gets the resource policy and policy revision for a bot or bot alias
describe_slot	Gets metadata information about a slot
describe_slot_type	Gets metadata information about a slot type
describe_test_execution	Gets metadata information about the test execution
describe_test_set	Gets metadata information about the test set
describe_test_set_discrepancy_report	Gets metadata information about the test set discrepancy report
describe_test_set_generation	Gets metadata information about the test set generation
generate_bot_element	Generates sample utterances for an intent
get_test_execution_artifacts_url	The pre-signed Amazon S3 URL to download the test execution result artifacts
list_aggregated_utterances	Provides a list of utterances that users have sent to the bot
list_bot_aliases	Gets a list of aliases for the specified bot
list_bot_alias_replicas	The action to list the replicated bots created from the source bot alias
list_bot_analyzer_history	Retrieves a list of historical bot analysis executions for a specific bot
list_bot_locales	Gets a list of locales for the specified bot
list_bot_recommendations	Get a list of bot recommendations that meet the specified criteria

list_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)
```

`lexruntimeservice` *Amazon Lex Runtime Service*

Description

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

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

Arguments

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

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

  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e 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 <- lexruntimeservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string",
      anonymous = "logical"
    ),
    endpoint = "string",
```

```
    region = "string",
    close_connection = "logical",
    timeout = "numeric",
    s3_force_path_style = "logical",
    sts_regional_endpoint = "string",
    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 <- lexruntimeservice()
svc$delete_session(
  Foo = 123
)

## End(Not run)
```

---

lexruntimev2 *Amazon Lex Runtime V2*

Description

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

Usage

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

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e 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 <- 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. `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-e 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 <- 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
describe_dataset	Provides a JSON description of the data in each time series dataset, including names, column
describe_inference_scheduler	Specifies information about the inference scheduler being used, including name, model, statu
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,
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 na
import_dataset	Imports a dataset
import_model_version	Imports a model that has been trained successfully
list_data_ingestion_jobs	Provides a list of all data ingestion jobs, including dataset name and ARN, S3 location of the
list_datasets	Lists all datasets currently available in your account, filtering on the dataset name
list_inference_events	Lists all inference events that have been found for the specified inference scheduler
list_inference_executions	Lists all inference executions that have been performed by the specified inference scheduler
list_inference_schedulers	Retrieves a list of all inference schedulers currently available for your account
list_label_groups	Returns a list of the label groups
list_labels	Provides a list of labels
list_models	Generates a list of all models in the account, including model name and ARN, dataset, and sta
list_model_versions	Generates a list of all model versions for a given model, including the model version, model v

list_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` *Amazon Machine Learning*

Description

Definition of the public APIs exposed by Amazon Machine Learning

Usage

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

Arguments

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

- **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-e
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 <- 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 informatio
get_data_source	Returns a DataSource that includes metadata and data file information, as well as the curr
get_evaluation	Returns an Evaluation that includes metadata as well as the current status of the Evaluatio
get_ml_model	Returns an MLModel that includes detailed metadata, data source information, and the cu
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. `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-e 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 <- 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 an
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 us
create_filter	Creates a recommendation filter
create_metric_attribution	Creates a metric attribution
create_recommender	Creates a recommender with the recipe (a Domain dataset group use case) you specify

create_schema	Creates an Amazon Personalize schema from the specified schema string
create_solution	By default, all new solutions use automatic training
create_solution_version	Trains or retrains an active solution in a Custom dataset group
delete_campaign	Removes a campaign by deleting the solution deployment
delete_dataset	Deletes a dataset
delete_dataset_group	Deletes a dataset group
delete_event_tracker	Deletes the event tracker
delete_filter	Deletes a filter
delete_metric_attribution	Deletes a metric attribution
delete_recommender	Deactivates and removes a recommender
delete_schema	Deletes a schema
delete_solution	Deletes all versions of a solution and the Solution object itself
describe_algorithm	Describes the given algorithm
describe_batch_inference_job	Gets the properties of a batch inference job including name, Amazon Resource Name (ARI
describe_batch_segment_job	Gets the properties of a batch segment job including name, Amazon Resource Name (ARN
describe_campaign	Describes the given campaign, including its status
describe_data_deletion_job	Describes the data deletion job created by CreateDataDeletionJob, including the job status
describe_dataset	Describes the given dataset
describe_dataset_export_job	Describes the dataset export job created by CreateDatasetExportJob, including the export j
describe_dataset_group	Describes the given dataset group
describe_dataset_import_job	Describes the dataset import job created by CreateDatasetImportJob, including the import j
describe_event_tracker	Describes an event tracker
describe_feature_transformation	Describes the given feature transformation
describe_filter	Describes a filter's properties
describe_metric_attribution	Describes a metric attribution
describe_recipe	Describes a recipe
describe_recommender	Describes the given recommender, including its status
describe_schema	Describes a schema
describe_solution	Describes a solution
describe_solution_version	Describes a specific version of a solution
get_solution_metrics	Gets the metrics for the specified solution version
list_batch_inference_jobs	Gets a list of the batch inference jobs that have been performed off of a solution version
list_batch_segment_jobs	Gets a list of the batch segment jobs that have been performed off of a solution version that
list_campaigns	Returns a list of campaigns that use the given solution
list_data_deletion_jobs	Returns a list of data deletion jobs for a dataset group ordered by creation time, with the m
list_dataset_export_jobs	Returns a list of dataset export jobs that use the given dataset
list_dataset_groups	Returns a list of dataset groups
list_dataset_import_jobs	Returns a list of dataset import jobs that use the given dataset
list_datasets	Returns the list of datasets contained in the given dataset group
list_event_trackers	Returns the list of event trackers associated with the account
list_filters	Lists all filters that belong to a given dataset group
list_metric_attribution_metrics	Lists the metrics for the metric attribution
list_metric_attributions	Lists metric attributions
list_recipes	Returns a list of available recipes
list_recommenders	Returns a list of recommenders in a given Domain dataset group
list_schemas	Returns the list of schemas associated with the account
list_solutions	Returns a list of solutions in a given dataset group
list_solution_versions	Returns a list of solution versions for the given solution

list_tags_for_resource	Get a list of tags attached to a resource
start_recommender	Starts a recommender that is INACTIVE
stop_recommender	Stops a recommender that is ACTIVE
stop_solution_version_creation	Stops creating a solution version that is in a state of CREATE_PENDING or CREATE IN_
tag_resource	Add a list of tags to a resource
untag_resource	Removes the specified tags that are attached to a resource
update_campaign	Updates a campaign to deploy a retrained solution version with an existing campaign, chan
update_dataset	Update a dataset to replace its schema with a new or existing one
update_metric_attribution	Updates a metric attribution
update_recommender	Updates the recommender to modify the recommender configuration
update_solution	Updates an Amazon Personalize solution to use a different automatic training configuration

Examples

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

## End(Not run)
```

`personalizeevents` *Amazon Personalize Events*

Description

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

Usage

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

Arguments

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

- **credentials**:

    - **creds**:
      * **access_key_id**: AWS access key ID
      * **secret_access_key**: AWS secret access key
      * **session_token**: AWS temporary session token
    - **profile**: The name of a profile to use. If not given, then the default profile is used.
    - **anonymous**: Set anonymous credentials.
  - **endpoint**: The complete URL to use for the constructed client.
  - **region**: The AWS Region used in instantiating the client.
  - **close_connection**: Immediately close all HTTP connections.
  - **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
  - **s3_force_path_style**: Set this to `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-e 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` *Amazon Personalize Runtime*

Description

Amazon Personalize Runtime

Usage

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

Arguments

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

- **credentials**:
  - **creds**:
    - **access_key_id**: AWS access key ID
    - **secret_access_key**: AWS secret access key
    - **session_token**: AWS temporary session token
  - **profile**: The name of a profile to use. If not given, then the default profile is used.
  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `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-e 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-e 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 SpeechSynthesisTas
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. `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-e 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 <- 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 t
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 c
create_stream_processor	Service availability notice: Streaming Video and Bulk Image Analysis is no longer availa
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 Custo
delete_stream_processor	Service availability notice: Streaming Video and Bulk Image Analysis is no longer availa
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 availa
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 Rekog
get_celebrity_recognition	Gets the celebrity recognition results for a Amazon Rekognition Video analysis started by

get_content_moderation	Gets the inappropriate, unwanted, or offensive content analysis results for a Amazon Reko
get_face_detection	Gets face detection results for a Amazon Rekognition Video analysis started by StartFace
get_face_liveness_session_results	Retrieves the results of a specific Face Liveness session
get_face_search	Gets the face search results for Amazon Rekognition Video face search started by StartFa
get_label_detection	Gets the label detection results of a Amazon Rekognition Video analysis started by StartL
get_media_analysis_job	Service availability notice: Streaming Video and Bulk Image Analysis is no longer availa
get_person_tracking	End of support notice: On October 31, 2025, AWS will discontinue support for Amazon I
get_segment_detection	Gets the segment detection results of a Amazon Rekognition Video analysis started by Sta
get_text_detection	Gets the text detection results of a Amazon Rekognition Video analysis started by StartTe
index_faces	Detects faces in the input image and adds them to the specified collection
list_collections	Returns list of collection IDs in your account
list_dataset_entries	This operation applies only to Amazon Rekognition Custom Labels
list_dataset_labels	This operation applies only to Amazon Rekognition Custom Labels
list_faces	Returns metadata for faces in the specified collection
list_media_analysis_jobs	Service availability notice: Streaming Video and Bulk Image Analysis is no longer availa
list_project_policies	This operation applies only to Amazon Rekognition Custom Labels
list_stream_processors	Service availability notice: Streaming Video and Bulk Image Analysis is no longer availa
list_tags_for_resource	Returns a list of tags in an Amazon Rekognition collection, stream processor, or Custom I
list_users	Returns metadata of the User such as UserID in the specified collection
put_project_policy	This operation applies only to Amazon Rekognition Custom Labels
recognize_celebrities	Returns an array of celebrities recognized in the input image
search_faces	For a given input face ID, searches for matching faces in the collection the face belongs to
search_faces_by_image	For a given input image, first detects the largest face in the image, and then searches the s
search_users	Searches for UserIDs within a collection based on a FaceId or UserId
search_users_by_image	Searches for UserIDs using a supplied image
start_celebrity_recognition	Starts asynchronous recognition of celebrities in a stored video
start_content_moderation	Starts asynchronous detection of inappropriate, unwanted, or offensive content in a stored
start_face_detection	Starts asynchronous detection of faces in a stored video
start_face_search	Starts the asynchronous search for faces in a collection that match the faces of persons de
start_label_detection	Starts asynchronous detection of labels in a stored video
start_media_analysis_job	Service availability notice: Streaming Video and Bulk Image Analysis is no longer availa
start_person_tracking	End of support notice: On October 31, 2025, AWS will discontinue support for Amazon I
start_project_version	This operation applies only to Amazon Rekognition Custom Labels
start_segment_detection	Starts asynchronous detection of segment detection in a stored video
start_stream_processor	Service availability notice: Streaming Video and Bulk Image Analysis is no longer availa
start_text_detection	Starts asynchronous detection of text in a stored video
stop_project_version	This operation applies only to Amazon Rekognition Custom Labels
stop_stream_processor	Service availability notice: Streaming Video and Bulk Image Analysis is no longer availa
tag_resource	Adds one or more key-value tags to an Amazon Rekognition collection, stream processor,
untag_resource	Removes one or more tags from an Amazon Rekognition collection, stream processor, or
update_dataset_entries	This operation applies only to Amazon Rekognition Custom Labels
update_stream_processor	Service availability notice: Streaming Video and Bulk Image Analysis is no longer availa

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. `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-e 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 <- 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 resou
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 clu
attach_cluster_node_volume	Attaches your Amazon Elastic Block Store (Amazon EBS) volume to a
batch_add_cluster_nodes	Adds nodes to a HyperPod cluster by incrementing the target count for
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 s
batch_replace_cluster_nodes	Replaces specific nodes within a SageMaker HyperPod cluster with nev
create_action	Creates an action
create_ai_benchmark_job	Creates a benchmark job that runs performance benchmarks against inf
create_ai_recommendation_job	Creates a recommendation job that generates intelligent optimization re
create_ai_workload_config	Creates a reusable AI workload configuration that defines datasets, data
create_algorithm	Create a machine learning algorithm that you can use in SageMaker and
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 KernelG
create_artifact	Creates an artifact
create_auto_ml_job	Creates an Autopilot job also referred to as Autopilot experiment or Au
create_auto_ml_job_v2	Creates an Autopilot job also referred to as Autopilot experiment or Au
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

create_compilation_job	Starts a model compilation job
create_compute_quota	Create compute allocation definition
create_context	Creates a context
create_data_quality_job_definition	Creates a definition for a job that monitors data quality and drift
create_device_fleet	Creates a device fleet
create_domain	Creates a Domain
create_edge_deployment_plan	Creates an edge deployment plan, consisting of multiple stages
create_edge_deployment_stage	Creates a new stage in an existing edge deployment plan
create_edge_packaging_job	Starts a SageMaker Edge Manager model packaging job
create_endpoint	Creates an endpoint using the endpoint configuration specified in the re
create_endpoint_config	Creates an endpoint configuration that SageMaker hosting services uses
create_experiment	Creates a SageMaker experiment
create_feature_group	Create a new FeatureGroup
create_flow_definition	Creates a flow definition
create_hub	Create a hub
create_hub_content_presigned_urls	Creates presigned URLs for accessing hub content artifacts
create_hub_content_reference	Create a hub content reference in order to add a model in the JumpStart
create_human_task_ui	Defines the settings you will use for the human review workflow user in
create_hyper_parameter_tuning_job	Starts a hyperparameter tuning job
create_image	Creates a custom SageMaker AI image
create_image_version	Creates a version of the SageMaker AI image specified by ImageName
create_inference_component	Creates an inference component, which is a SageMaker AI hosting obje
create_inference_experiment	Creates an inference experiment using the configurations specified in th
create_inference_recommendations_job	Starts a recommendation job
create_job	Creates a model customization job in Amazon SageMaker
create_labeling_job	Creates a job that uses workers to label the data objects in your input da
create_mlflow_app	Creates an MLflow Tracking Server using a general purpose Amazon S
create_mlflow_tracking_server	Creates an MLflow Tracking Server using a general purpose Amazon S
create_model	Creates a model in SageMaker
create_model_bias_job_definition	Creates the definition for a model bias job
create_model_card	Creates an Amazon SageMaker Model Card
create_model_card_export_job	Creates an Amazon SageMaker Model Card export job
create_model_explainability_job_definition	Creates the definition for a model explainability job
create_model_package	Creates a model package that you can use to create SageMaker models
create_model_package_group	Creates a model group
create_model_quality_job_definition	Creates a definition for a job that monitors model quality and drift
create_monitoring_schedule	Creates a schedule that regularly starts Amazon SageMaker AI Processi
create_notebook_instance	Creates an SageMaker AI notebook instance
create_notebook_instance_lifecycle_config	Creates a lifecycle configuration that you can associate with a notebook
create_optimization_job	Creates a job that optimizes a model for inference performance
create_partner_app	Creates an Amazon SageMaker Partner AI App
create_partner_app_presigned_url	Creates a presigned URL to access an Amazon SageMaker Partner AI A
create_pipeline	Creates a pipeline using a JSON pipeline definition
create_presigned_domain_url	Creates a URL for a specified UserProfile in a Domain
create_presigned_mlflow_app_url	Returns a presigned URL that you can use to connect to the MLflow UI
create_presigned_mlflow_tracking_server_url	Returns a presigned URL that you can use to connect to the MLflow UI
create_presigned_notebook_instance_url	Returns a URL that you can use to connect to the Jupyter server from a
create_processing_job	Creates a processing job

create_project	Creates a machine learning (ML) project that can contain one or more t
create_space	Creates a private space or a space used for real time collaboration in a d
create_studio_lifecycle_config	Creates a new Amazon SageMaker AI Studio Lifecycle Configuration
create_training_job	Starts a model training job
create_training_plan	Creates a new training plan in SageMaker to reserve compute capacity
create_transform_job	Starts a transform job
create_trial	Creates an SageMaker trial
create_trial_component	Creates a trial component, which is a stage of a machine learning trial
create_user_profile	Creates a user profile
create_workforce	Use this operation to create a workforce
create_workteam	Creates a new work team for labeling your data
delete_action	Deletes an action
delete_ai_benchmark_job	Deletes the specified AI benchmark job
delete_ai_recommendation_job	Deletes the specified AI recommendation job
delete_ai_workload_config	Deletes the specified AI workload configuration
delete_algorithm	Removes the specified algorithm from your account
delete_app	Used to stop and delete an app
delete_app_image_config	Deletes an AppImageConfig
delete_artifact	Deletes an artifact
delete_association	Deletes an association
delete_cluster	Delete a SageMaker HyperPod cluster
delete_cluster_scheduler_config	Deletes the cluster policy of the cluster
delete_code_repository	Deletes the specified Git repository from your account
delete_compilation_job	Deletes the specified compilation job
delete_compute_quota	Deletes the compute allocation from the cluster
delete_context	Deletes an context
delete_data_quality_job_definition	Deletes a data quality monitoring job definition
delete_device_fleet	Deletes a fleet
delete_domain	Used to delete a domain
delete_edge_deployment_plan	Deletes an edge deployment plan if (and only if) all the stages in the pla
delete_edge_deployment_stage	Delete a stage in an edge deployment plan if (and only if) the stage is in
delete_endpoint	Deletes an endpoint
delete_endpoint_config	Deletes an endpoint configuration
delete_experiment	Deletes an SageMaker experiment
delete_feature_group	Delete the FeatureGroup and any data that was written to the OnlineSto
delete_flow_definition	Deletes the specified flow definition
delete_hub	Delete a hub
delete_hub_content	Delete the contents of a hub
delete_hub_content_reference	Delete a hub content reference in order to remove a model from a privat
delete_human_task_ui	Use this operation to delete a human task user interface (worker task te
delete_hyper_parameter_tuning_job	Deletes a hyperparameter tuning job
delete_image	Deletes a SageMaker AI image and all versions of the image
delete_image_version	Deletes a version of a SageMaker AI image
delete_inference_component	Deletes an inference component
delete_inference_experiment	Deletes an inference experiment
delete_job	Deletes a job
delete_mlflow_app	Deletes an MLflow App
delete_mlflow_tracking_server	Deletes an MLflow Tracking Server

delete_model	Deletes a model
delete_model_bias_job_definition	Deletes an Amazon SageMaker AI model bias job definition
delete_model_card	Deletes an Amazon SageMaker Model Card
delete_model_explainability_job_definition	Deletes an Amazon SageMaker AI model explainability job definition
delete_model_package	Deletes a model package
delete_model_package_group	Deletes the specified model group
delete_model_package_group_policy	Deletes a model group resource policy
delete_model_quality_job_definition	Deletes the secified model quality monitoring job definition
delete_monitoring_schedule	Deletes a monitoring schedule
delete_notebook_instance	Deletes an SageMaker AI notebook instance
delete_notebook_instance_lifecycle_config	Deletes a notebook instance lifecycle configuration
delete_optimization_job	Deletes an optimization job
delete_partner_app	Deletes a SageMaker Partner AI App
delete_pipeline	Deletes a pipeline if there are no running instances of the pipeline
delete_processing_job	Deletes a processing job
delete_project	Delete the specified project
delete_space	Used to delete a space
delete_studio_lifecycle_config	Deletes the Amazon SageMaker AI Studio Lifecycle Configuration
delete_tags	Deletes the specified tags from an SageMaker resource
delete_training_job	Deletes a training job
delete_trial	Deletes the specified trial
delete_trial_component	Deletes the specified trial component
delete_user_profile	Deletes a user profile
delete_workforce	Use this operation to delete a workforce
delete_workteam	Deletes an existing work team
deregister_devices	Deregisters the specified devices
describe_action	Describes an action
describe_ai_benchmark_job	Returns details of an AI benchmark job, including its status, configurati
describe_ai_recommendation_job	Returns details of an AI recommendation job, including its status, mode
describe_ai_workload_config	Returns details of an AI workload configuration, including the dataset c
describe_algorithm	Returns a description of the specified algorithm that is in your account
describe_app	Describes the app
describe_app_image_config	Describes an AppImageConfig
describe_artifact	Describes an artifact
describe_auto_ml_job	Returns information about an AutoML job created by calling CreateAu
describe_auto_ml_job_v2	Returns information about an AutoML job created by calling CreateAu
describe_cluster	Retrieves information of a SageMaker HyperPod cluster
describe_cluster_event	Retrieves detailed information about a specific event for a given HyperP
describe_cluster_node	Retrieves information of a node (also called a instance interchangeably)
describe_cluster_scheduler_config	Description of the cluster policy
describe_code_repository	Gets details about the specified Git repository
describe_compilation_job	Returns information about a model compilation job
describe_compute_quota	Description of the compute allocation definition
describe_context	Describes a context
describe_data_quality_job_definition	Gets the details of a data quality monitoring job definition
describe_device	Describes the device
describe_device_fleet	A description of the fleet the device belongs to
describe_domain	The description of the domain

describe_edge_deployment_plan	Describes an edge deployment plan with deployment status per stage
describe_edge_packaging_job	A description of edge packaging jobs
describe_endpoint	Returns the description of an endpoint
describe_endpoint_config	Returns the description of an endpoint configuration created using the C
describe_experiment	Provides a list of an experiment's properties
describe_feature_group	Use this operation to describe a FeatureGroup
describe_feature_metadata	Shows the metadata for a feature within a feature group
describe_flow_definition	Returns information about the specified flow definition
describe_hub	Describes a hub
describe_hub_content	Describe the content of a hub
describe_human_task_ui	Returns information about the requested human task user interface (wor
describe_hyper_parameter_tuning_job	Returns a description of a hyperparameter tuning job, depending on the
describe_image	Describes a SageMaker AI image
describe_image_version	Describes a version of a SageMaker AI image
describe_inference_component	Returns information about an inference component
describe_inference_experiment	Returns details about an inference experiment
describe_inference_recommendations_job	Provides the results of the Inference Recommender job
describe_job	Returns detailed information about a job, including its current status, se
describe_job_schema_version	Returns the JSON schema for a specified job category and schema versi
describe_labeling_job	Gets information about a labeling job
describe_lineage_group	Provides a list of properties for the requested lineage group
describe_mlflow_app	Returns information about an MLflow App
describe_mlflow_tracking_server	Returns information about an MLflow Tracking Server
describe_model	Describes a model that you created using the CreateModel API
describe_model_bias_job_definition	Returns a description of a model bias job definition
describe_model_card	Describes the content, creation time, and security configuration of an A
describe_model_card_export_job	Describes an Amazon SageMaker Model Card export job
describe_model_explainability_job_definition	Returns a description of a model explainability job definition
describe_model_package	Returns a description of the specified model package, which is used to c
describe_model_package_group	Gets a description for the specified model group
describe_model_quality_job_definition	Returns a description of a model quality job definition
describe_monitoring_schedule	Describes the schedule for a monitoring job
describe_notebook_instance	Returns information about a notebook instance
describe_notebook_instance_lifecycle_config	Returns a description of a notebook instance lifecycle configuration
describe_optimization_job	Provides the properties of the specified optimization job
describe_partner_app	Gets information about a SageMaker Partner AI App
describe_pipeline	Describes the details of a pipeline
describe_pipeline_definition_for_execution	Describes the details of an execution's pipeline definition
describe_pipeline_execution	Describes the details of a pipeline execution
describe_processing_job	Returns a description of a processing job
describe_project	Describes the details of a project
describe_reserved_capacity	Retrieves details about a reserved capacity
describe_space	Describes the space
describe_studio_lifecycle_config	Describes the Amazon SageMaker AI Studio Lifecycle Configuration
describe_subscribed_workteam	Gets information about a work team provided by a vendor
describe_training_job	Returns information about a training job
describe_training_plan	Retrieves detailed information about a specific training plan
describe_training_plan_extension_history	Retrieves the extension history for a specified training plan

describe_transform_job	Returns information about a transform job
describe_trial	Provides a list of a trial's properties
describe_trial_component	Provides a list of a trials component's properties
describe_user_profile	Describes a user profile
describe_workforce	Lists private workforce information, including workforce name, Amazo
describe_workteam	Gets information about a specific work team
detach_cluster_node_volume	Detaches your Amazon Elastic Block Store (Amazon EBS) volume fro
disable_sagemaker_servicecatalog_portfolio	Disables using Service Catalog in SageMaker
disassociate_trial_component	Disassociates a trial component from a trial
enable_sagemaker_servicecatalog_portfolio	Enables using Service Catalog in SageMaker
extend_training_plan	Extends an existing training plan by purchasing an extension offering
get_device_fleet_report	Describes a fleet
get_lineage_group_policy	The resource policy for the lineage group
get_model_package_group_policy	Gets a resource policy that manages access for a model group
get_sagemaker_servicecatalog_portfolio_status	Gets the status of Service Catalog in SageMaker
get_scaling_configuration_recommendation	Starts an Amazon SageMaker Inference Recommender autoscaling rec
get_search_suggestions	An auto-complete API for the search functionality in the SageMaker co
import_hub_content	Import hub content
list_actions	Lists the actions in your account and their properties
list_ai_benchmark_jobs	Returns a list of AI benchmark jobs in your account
list_ai_recommendation_jobs	Returns a list of AI recommendation jobs in your account
list_ai_workload_configs	Returns a list of AI workload configurations in your account
list_algorithms	Lists the machine learning algorithms that have been created
list_aliases	Lists the aliases of a specified image or image version
list_app_image_configs	Lists the AppImageConfigs in your account and their properties
list_apps	Lists apps
list_artifacts	Lists the artifacts in your account and their properties
list_associations	Lists the associations in your account and their properties
list_auto_ml_jobs	Request a list of jobs
list_candidates_for_auto_ml_job	List the candidates created for the job
list_cluster_events	Retrieves a list of event summaries for a specified HyperPod cluster
list_cluster_nodes	Retrieves the list of instances (also called nodes interchangeably) in a S
list_clusters	Retrieves the list of SageMaker HyperPod clusters
list_cluster_scheduler_configs	List the cluster policy configurations
list_code_repositories	Gets a list of the Git repositories in your account
list_compilation_jobs	Lists model compilation jobs that satisfy various filters
list_compute_quotas	List the resource allocation definitions
list_contexts	Lists the contexts in your account and their properties
list_data_quality_job_definitions	Lists the data quality job definitions in your account
list_device_fleets	Returns a list of devices in the fleet
list_devices	A list of devices
list_domains	Lists the domains
list_edge_deployment_plans	Lists all edge deployment plans
list_edge_packaging_jobs	Returns a list of edge packaging jobs
list_endpoint_configs	Lists endpoint configurations
list_endpoints	Lists endpoints
list_experiments	Lists all the experiments in your account
list_feature_groups	List FeatureGroups based on given filter and order

list_flow_definitions	Returns information about the flow definitions in your account
list_hub_contents	List the contents of a hub
list_hub_content_versions	List hub content versions
list_hubs	List all existing hubs
list_human_task_uis	Returns information about the human task user interfaces in your accou
list_hyper_parameter_tuning_jobs	Gets a list of HyperParameterTuningJobSummary objects that describe
list_images	Lists the images in your account and their properties
list_image_versions	Lists the versions of a specified image and their properties
list_inference_components	Lists the inference components in your account and their properties
list_inference_experiments	Returns the list of all inference experiments
list_inference_recommendations_jobs	Lists recommendation jobs that satisfy various filters
list_inference_recommendations_job_steps	Returns a list of the subtasks for an Inference Recommender job
list_jobs	Lists jobs in a specified category
list_job_schema_versions	Lists available configuration schema versions for a specified job categor
list_labeling_jobs	Gets a list of labeling jobs
list_labeling_jobs_for_workteam	Gets a list of labeling jobs assigned to a specified work team
list_lineage_groups	A list of lineage groups shared with your Amazon Web Services accour
list_mlflow_apps	Lists all MLflow Apps
list_mlflow_tracking_servers	Lists all MLflow Tracking Servers
list_model_bias_job_definitions	Lists model bias jobs definitions that satisfy various filters
list_model_card_export_jobs	List the export jobs for the Amazon SageMaker Model Card
list_model_cards	List existing model cards
list_model_card_versions	List existing versions of an Amazon SageMaker Model Card
list_model_explainability_job_definitions	Lists model explainability job definitions that satisfy various filters
list_model_metadata	Lists the domain, framework, task, and model name of standard machin
list_model_package_groups	Gets a list of the model groups in your Amazon Web Services account
list_model_packages	Lists the model packages that have been created
list_model_quality_job_definitions	Gets a list of model quality monitoring job definitions in your account
list_models	Lists models created with the CreateModel API
list_monitoring_alert_history	Gets a list of past alerts in a model monitoring schedule
list_monitoring_alerts	Gets the alerts for a single monitoring schedule
list_monitoring_executions	Returns list of all monitoring job executions
list_monitoring_schedules	Returns list of all monitoring schedules
list_notebook_instance_lifecycle_configs	Lists notebook instance lifestyle configurations created with the Create
list_notebook_instances	Returns a list of the SageMaker AI notebook instances in the requester'
list_optimization_jobs	Lists the optimization jobs in your account and their properties
list_partner_apps	Lists all of the SageMaker Partner AI Apps in an account
list_pipeline_executions	Gets a list of the pipeline executions
list_pipeline_execution_steps	Gets a list of PipeLineExecutionStep objects
list_pipeline_parameters_for_execution	Gets a list of parameters for a pipeline execution
list_pipelines	Gets a list of pipelines
list_pipeline_versions	Gets a list of all versions of the pipeline
list_processing_jobs	Lists processing jobs that satisfy various filters
list_projects	Gets a list of the projects in an Amazon Web Services account
list_resource_catalogs	Lists Amazon SageMaker Catalogs based on given filters and orders
list_spaces	Lists spaces
list_stage_devices	Lists devices allocated to the stage, containing detailed device informat
list_studio_lifecycle_configs	Lists the Amazon SageMaker AI Studio Lifecycle Configurations in yo

list_subscribed_workteams	Gets a list of the work teams that you are subscribed to in the Amazon V
list_tags	Returns the tags for the specified SageMaker resource
list_training_jobs	Lists training jobs
list_training_jobs_for_hyper_parameter_tuning_job	Gets a list of TrainingJobSummary objects that describe the training job
list_training_plans	Retrieves a list of training plans for the current account
list_transform_jobs	Lists transform jobs
list_trial_components	Lists the trial components in your account
list_trials	Lists the trials in your account
list_ultra_servers_by_reserved_capacity	Lists all UltraServers that are part of a specified reserved capacity
list_user_profiles	Lists user profiles
list_workforces	Use this operation to list all private and vendor workforces in an Amazo
list_workteams	Gets a list of private work teams that you have defined in a region
put_model_package_group_policy	Adds a resouce policy to control access to a model group
query_lineage	Use this action to inspect your lineage and discover relationships betwe
register_devices	Register devices
render_ui_template	Renders the UI template so that you can preview the worker's experienc
retry_pipeline_execution	Retry the execution of the pipeline
search	Finds SageMaker resources that match a search query
search_training_plan_offerings	Searches for available training plan offerings based on specified criteria
send_pipeline_execution_step_failure	Notifies the pipeline that the execution of a callback step failed, along w
send_pipeline_execution_step_success	Notifies the pipeline that the execution of a callback step succeeded and
start_cluster_health_check	Start deep health checks for a SageMaker HyperPod cluster
start_edge_deployment_stage	Starts a stage in an edge deployment plan
start_inference_experiment	Starts an inference experiment
start_mlflow_tracking_server	Programmatically start an MLflow Tracking Server
start_monitoring_schedule	Starts a previously stopped monitoring schedule
start_notebook_instance	Launches an ML compute instance with the latest version of the librarie
start_pipeline_execution	Starts a pipeline execution
start_session	Initiates a remote connection session between a local integrated develop
stop_ai_benchmark_job	Stops a running AI benchmark job
stop_ai_recommendation_job	Stops a running AI recommendation job
stop_auto_ml_job	A method for forcing a running job to shut down
stop_compilation_job	Stops a model compilation job
stop_edge_deployment_stage	Stops a stage in an edge deployment plan
stop_edge_packaging_job	Request to stop an edge packaging job
stop_hyper_parameter_tuning_job	Stops a running hyperparameter tuning job and all running training jobs
stop_inference_experiment	Stops an inference experiment
stop_inference_recommendations_job	Stops an Inference Recommender job
stop_job	Stops a running job
stop_labeling_job	Stops a running labeling job
stop_mlflow_tracking_server	Programmatically stop an MLflow Tracking Server
stop_monitoring_schedule	Stops a previously started monitoring schedule
stop_notebook_instance	Terminates the ML compute instance
stop_optimization_job	Ends a running inference optimization job
stop_pipeline_execution	Stops a pipeline execution
stop_processing_job	Stops a processing job
stop_training_job	Stops a training job
stop_transform_job	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 exist
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 resour
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 compor
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. `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-e
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. `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-e
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 <- 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. `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-e 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 <- 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 Ama
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 im
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-e 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

batch_get_metrics	Used to retrieve training metrics from SageMaker
batch_put_metrics	Used to ingest training metrics into SageMaker

Examples

```
## Not run:
svc <- sagemakermetrics()
svc$batch_get_metrics(
  Foo = 123
)

## End(Not run)
```

`sagemakerruntime` *Amazon SageMaker Runtime*

Description

The Amazon SageMaker AI runtime API.

Usage

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

Arguments

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

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

  - **anonymous**: Set anonymous credentials.
- **endpoint**: The complete URL to use for the constructed client.
- **region**: The AWS Region used in instantiating the client.
- **close_connection**: Immediately close all HTTP connections.
- **timeout**: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.
- **s3_force_path_style**: Set this to `true` to force the request to use path-style addressing, i.e. `http://s3.amazonaws.com/BUCKET/KEY`.
- **sts_regional_endpoint**: Set sts regional endpoint resolver to regional or legacy https://docs.aws.amazon.com/sdkref/latest/guide/feature-sts-regionalized-e
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 <- 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 strea

Examples

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

## End(Not run)
```

---

textract *Amazon Textract*

Description

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

Usage

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

Arguments

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

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

Description

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

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

Usage

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

Arguments

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

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

delete_language_model	Deletes a custom language model
delete_medical_scribe_job	Deletes a Medical Scribe job
delete_medical_transcription_job	Deletes a medical transcription job
delete_medical_vocabulary	Deletes a custom medical vocabulary
delete_transcription_job	Deletes a transcription job
delete_vocabulary	Deletes a custom vocabulary
delete_vocabulary_filter	Deletes a custom vocabulary filter
describe_language_model	Provides information about the specified custom language model
get_call_analytics_category	Provides information about the specified Call Analytics category
get_call_analytics_job	Provides information about the specified Call Analytics job
get_medical_scribe_job	Provides information about the specified Medical Scribe job
get_medical_transcription_job	Provides information about the specified medical transcription job
get_medical_vocabulary	Provides information about the specified custom medical vocabulary
get_transcription_job	Provides information about the specified transcription job
get_vocabulary	Provides information about the specified custom vocabulary
get_vocabulary_filter	Provides information about the specified custom vocabulary filter
list_call_analytics_categories	Provides a list of Call Analytics categories, including all rules that make up each category
list_call_analytics_jobs	Provides a list of Call Analytics jobs that match the specified criteria
list_language_models	Provides a list of custom language models that match the specified criteria
list_medical_scribe_jobs	Provides a list of Medical Scribe jobs that match the specified criteria
list_medical_transcription_jobs	Provides a list of medical transcription jobs that match the specified criteria
list_medical_vocabularies	Provides a list of custom medical vocabularies that match the specified criteria
list_tags_for_resource	Lists all tags associated with the specified transcription job, vocabulary, model, or resourc
list_transcription_jobs	Provides a list of transcription jobs that match the specified criteria
list_vocabularies	Provides a list of custom vocabularies that match the specified criteria
list_vocabulary_filters	Provides a list of custom vocabulary filters that match the specified criteria
start_call_analytics_job	Transcribes the audio from a customer service call and applies any additional Request Par
start_medical_scribe_job	Transcribes patient-clinician conversations and generates clinical notes
start_medical_transcription_job	Transcribes the audio from a medical dictation or conversation and applies any additional
start_transcription_job	Transcribes the audio from a media file and applies any additional Request Parameters you
tag_resource	Adds one or more custom tags, each in the form of a key:value pair, to the specified resou
untag_resource	Removes the specified tags from the specified Amazon Transcribe resource
update_call_analytics_category	Updates the specified Call Analytics category with new rules
update_language_model	Updates the encryption configuration for an existing custom language model
update_medical_vocabulary	Updates an existing custom medical vocabulary with new values
update_vocabulary	Updates an existing custom vocabulary with new values
update_vocabulary_filter	Updates an existing custom vocabulary filter with a new list of words

Examples

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

## End(Not run)
```

translate *Amazon Translate*

Description

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

Usage

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

Arguments

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

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

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. `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-e
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 <- 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

describe_watchlist	Describes the specified watchlist
disassociate_fraudster	Disassociates the fraudsters from the watchlist specified
evaluate_session	Evaluates a specified session based on audio data accumulated during a streaming Amaz
list_domains	Lists all the domains in the Amazon Web Services account
list_fraudster_registration_jobs	Lists all the fraudster registration jobs in the domain with the given JobStatus
list_fraudsters	Lists all fraudsters in a specified watchlist or domain
list_speaker_enrollment_jobs	Lists all the speaker enrollment jobs in the domain with the specified JobStatus
list_speakers	Lists all speakers in a specified domain
list_tags_for_resource	Lists all tags associated with a specified Voice ID resource
list_watchlists	Lists all watchlists in a specified domain
opt_out_speaker	Opts out a speaker from Voice ID
start_fraudster_registration_job	Starts a new batch fraudster registration job using provided details
start_speaker_enrollment_job	Starts a new batch speaker enrollment job using specified details
tag_resource	Tags a Voice ID resource with the provided list of tags
untag_resource	Removes specified tags from a specified Amazon Connect Voice ID resource
update_domain	Updates the specified domain
update_watchlist	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,

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
lexruntimeservice, 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_oauth_2_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_cmk, *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