

Package ‘paws.application.integration’

February 8, 2023

Title 'Amazon Web Services' Application Integration Services

Version 0.2.0

Description Interface to 'Amazon Web Services' application integration services, including 'Simple Queue Service' ('SQS') message queue, 'Simple Notification Service' ('SNS') publish/subscribe messaging, and more <<https://aws.amazon.com/>>.

License Apache License (>= 2.0)

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

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

Imports paws.common (>= 0.5.4)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.2.2

Collate 'eventbridge_service.R' 'eventbridge_interfaces.R'
'eventbridge_operations.R' 'locationsservice_service.R'
'locationsservice_interfaces.R' 'locationsservice_operations.R'
'mq_service.R' 'mq_interfaces.R' 'mq_operations.R'
'mwaaservice.R' 'mwaaservice_interfaces.R' 'mwaaservice_operations.R'
'schemas_service.R' 'schemas_interfaces.R'
'schemas_operations.R' 'sfn_service.R' 'sfn_interfaces.R'
'sfn_operations.R' 'sns_service.R' 'sns_interfaces.R'
'sns_operations.R' 'sqs_service.R' 'sqs_interfaces.R'
'sqs_operations.R' 'swf_service.R' 'swf_interfaces.R'
'swf_operations.R'

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2023-02-08 11:00:02 UTC

R topics documented:

eventbridge	2
locationsservice	5
mq	7
mwaa	9
schemas	12
sfn	14
sns	16
sqs	19
swf	21

Index	24
--------------	-----------

eventbridge	<i>Amazon EventBridge</i>
-------------	---------------------------

Description

Amazon EventBridge helps you to respond to state changes in your Amazon Web Services resources. When your resources change state, they automatically send events to an event stream. You can create rules that match selected events in the stream and route them to targets to take action. You can also use rules to take action on a predetermined schedule. For example, you can configure rules to:

- Automatically invoke an Lambda function to update DNS entries when an event notifies you that Amazon EC2 instance enters the running state.
- Direct specific API records from CloudTrail to an Amazon Kinesis data stream for detailed analysis of potential security or availability risks.
- Periodically invoke a built-in target to create a snapshot of an Amazon EBS volume.

For more information about the features of Amazon EventBridge, see the [Amazon EventBridge User Guide](#).

Usage

```
eventbridge(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • access_key_id: AWS access key ID • secret_access_key: AWS secret access key • session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials.
--------	--

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

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 <- eventbridge(
  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"
  )
)
```

Operations

activate_event_source	Activates a partner event source that has been deactivated
cancel_replay	Cancels the specified replay
create_api_destination	Creates an API destination, which is an HTTP invocation endpoint configured as a target
create_archive	Creates an archive of events with the specified settings
create_connection	Creates a connection
create_endpoint	Creates a global endpoint
create_event_bus	Creates a new event bus within your account
create_partner_event_source	Called by an SaaS partner to create a partner event source
deactivate_event_source	You can use this operation to temporarily stop receiving events from the specified partner
deauthorize_connection	Removes all authorization parameters from the connection
delete_api_destination	Deletes the specified API destination

<code>delete_archive</code>	Deletes the specified archive
<code>delete_connection</code>	Deletes a connection
<code>delete_endpoint</code>	Delete an existing global endpoint
<code>delete_event_bus</code>	Deletes the specified custom event bus or partner event bus
<code>delete_partner_event_source</code>	This operation is used by SaaS partners to delete a partner event source
<code>delete_rule</code>	Deletes the specified rule
<code>describe_api_destination</code>	Retrieves details about an API destination
<code>describe_archive</code>	Retrieves details about an archive
<code>describe_connection</code>	Retrieves details about a connection
<code>describe_endpoint</code>	Get the information about an existing global endpoint
<code>describe_event_bus</code>	Displays details about an event bus in your account
<code>describe_event_source</code>	This operation lists details about a partner event source that is shared with your account
<code>describe_partner_event_source</code>	An SaaS partner can use this operation to list details about a partner event source that the
<code>describe_replay</code>	Retrieves details about a replay
<code>describe_rule</code>	Describes the specified rule
<code>disable_rule</code>	Disables the specified rule
<code>enable_rule</code>	Enables the specified rule
<code>list_api_destinations</code>	Retrieves a list of API destination in the account in the current Region
<code>list_archives</code>	Lists your archives
<code>list_connections</code>	Retrieves a list of connections from the account
<code>list_endpoints</code>	List the global endpoints associated with this account
<code>list_event_buses</code>	Lists all the event buses in your account, including the default event bus, custom event b
<code>list_event_sources</code>	You can use this to see all the partner event sources that have been shared with your Am
<code>list_partner_event_source_accounts</code>	An SaaS partner can use this operation to display the Amazon Web Services account ID
<code>list_partner_event_sources</code>	An SaaS partner can use this operation to list all the partner event source names that the
<code>list_replays</code>	Lists your replays
<code>list_rule_names_by_target</code>	Lists the rules for the specified target
<code>list_rules</code>	Lists your Amazon EventBridge rules
<code>list_tags_for_resource</code>	Displays the tags associated with an EventBridge resource
<code>list_targets_by_rule</code>	Lists the targets assigned to the specified rule
<code>put_events</code>	Sends custom events to Amazon EventBridge so that they can be matched to rules
<code>put_partner_events</code>	This is used by SaaS partners to write events to a customer's partner event bus
<code>put_permission</code>	Running PutPermission permits the specified Amazon Web Services account or Amazon
<code>put_rule</code>	Creates or updates the specified rule
<code>put_targets</code>	Adds the specified targets to the specified rule, or updates the targets if they are already
<code>remove_permission</code>	Revokes the permission of another Amazon Web Services account to be able to put even
<code>remove_targets</code>	Removes the specified targets from the specified rule
<code>start_replay</code>	Starts the specified replay
<code>tag_resource</code>	Assigns one or more tags (key-value pairs) to the specified EventBridge resource
<code>test_event_pattern</code>	Tests whether the specified event pattern matches the provided event
<code>untag_resource</code>	Removes one or more tags from the specified EventBridge resource
<code>update_api_destination</code>	Updates an API destination
<code>update_archive</code>	Updates the specified archive
<code>update_connection</code>	Updates settings for a connection
<code>update_endpoint</code>	Update an existing endpoint

Examples

```
## Not run:
svc <- eventbridge()
svc$activate_event_source(
  Foo = 123
)

## End(Not run)
```

locationservice	<i>Amazon Location Service</i>
-----------------	--------------------------------

Description

"Suite of geospatial services including Maps, Places, Routes, Tracking, and Geofencing"

Usage

```
locationservice(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none">• access_key_id: AWS access key ID• secret_access_key: AWS secret access key• session_token: AWS temporary session token• profile: The name of a profile to use. If not given, then the default profile is used.• anonymous: Set anonymous credentials.• endpoint: The complete URL to use for the constructed client.• region: The AWS Region used in instantiating the client.• close_connection: Immediately close all HTTP connections.• timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds.• s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e., <code>http://s3.amazonaws.com/BUCKET/KEY</code>.
--------	---

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 <- locationsservice(
  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"
  )
)

```

Operations

associate_tracker_consumer	Creates an association between a geofence collection and a tracker resource
batch_delete_device_position_history	Deletes the position history of one or more devices from a tracker resource
batch_delete_geofence	Deletes a batch of geofences from a geofence collection
batch_evaluate_geofences	Evaluates device positions against the geofence geometries from a given geofence collection
batch_get_device_position	Lists the latest device positions for requested devices
batch_put_geofence	A batch request for storing geofence geometries into a given geofence collection, or updating existing geofences
batch_update_device_position	Uploads position update data for one or more devices to a tracker resource
calculate_route	Calculates a route given the following required parameters: DeparturePosition and DestinationPosition
calculate_route_matrix	Calculates a route matrix given the following required parameters: DeparturePosition and DestinationPosition
create_geofence_collection	Creates a geofence collection, which manages and stores geofences
create_map	Creates a map resource in your AWS account, which provides map tiles of different styles
create_place_index	Creates a place index resource in your AWS account
create_route_calculator	Creates a route calculator resource in your AWS account
create_tracker	Creates a tracker resource in your AWS account, which lets you retrieve current and historical device positions
delete_geofence_collection	Deletes a geofence collection from your AWS account
delete_map	Deletes a map resource from your AWS account
delete_place_index	Deletes a place index resource from your AWS account
delete_route_calculator	Deletes a route calculator resource from your AWS account
delete_tracker	Deletes a tracker resource from your AWS account
describe_geofence_collection	Retrieves the geofence collection details
describe_map	Retrieves the map resource details
describe_place_index	Retrieves the place index resource details
describe_route_calculator	Retrieves the route calculator resource details
describe_tracker	Retrieves the tracker resource details
disassociate_tracker_consumer	Removes the association between a tracker resource and a geofence collection
get_device_position	Retrieves a device's most recent position according to its sample time

get_device_position_history	Retrieves the device position history from a tracker resource within a specified range
get_geofence	Retrieves the geofence details from a geofence collection
get_map_glyphs	Retrieves glyphs used to display labels on a map
get_map_sprites	Retrieves the sprite sheet corresponding to a map resource
get_map_style_descriptor	Retrieves the map style descriptor from a map resource
get_map_tile	Retrieves a vector data tile from the map resource
list_device_positions	A batch request to retrieve all device positions
list_geofence_collections	Lists geofence collections in your AWS account
list_geofences	Lists geofences stored in a given geofence collection
list_maps	Lists map resources in your AWS account
list_place_indexes	Lists place index resources in your AWS account
list_route_calculators	Lists route calculator resources in your AWS account
list_tags_for_resource	Returns a list of tags that are applied to the specified Amazon Location resource
list_tracker_consumers	Lists geofence collections currently associated to the given tracker resource
list_trackers	Lists tracker resources in your AWS account
put_geofence	Stores a geofence geometry in a given geofence collection, or updates the geometry of
search_place_index_for_position	Reverse geocodes a given coordinate and returns a legible address
search_place_index_for_suggestions	Generates suggestions for addresses and points of interest based on partial or misspell
search_place_index_for_text	Geocodes free-form text, such as an address, name, city, or region to allow you to sea
tag_resource	Assigns one or more tags (key-value pairs) to the specified Amazon Location Service
untag_resource	Removes one or more tags from the specified Amazon Location resource
update_geofence_collection	Updates the specified properties of a given geofence collection
update_map	Updates the specified properties of a given map resource
update_place_index	Updates the specified properties of a given place index resource
update_route_calculator	Updates the specified properties for a given route calculator resource
update_tracker	Updates the specified properties of a given tracker resource

Examples

```
## Not run:
svc <- locationservice()
svc$associate_tracker_consumer(
  Foo = 123
)

## End(Not run)
```

Description

Amazon MQ is a managed message broker service for Apache ActiveMQ and RabbitMQ that makes it easy to set up and operate message brokers in the cloud. A message broker allows software applications and components to communicate using various programming languages, operating systems, and formal messaging protocols.

Usage

```
mq(config = list())
```

Arguments

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

- **access_key_id**: AWS access key ID
- **secret_access_key**: AWS secret access key
- **session_token**: AWS temporary 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`.

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 <- mq(
  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"
  )
)
```


Operations

<code>create_broker</code>	Creates a broker
<code>create_configuration</code>	Creates a new configuration for the specified configuration name
<code>create_tags</code>	Add a tag to a resource
<code>create_user</code>	Creates an ActiveMQ user
<code>delete_broker</code>	Deletes a broker
<code>delete_tags</code>	Removes a tag from a resource
<code>delete_user</code>	Deletes an ActiveMQ user
<code>describe_broker</code>	Returns information about the specified broker
<code>describe_broker_engine_types</code>	Describe available engine types and versions
<code>describe_broker_instance_options</code>	Describe available broker instance options
<code>describe_configuration</code>	Returns information about the specified configuration
<code>describe_configuration_revision</code>	Returns the specified configuration revision for the specified configuration
<code>describe_user</code>	Returns information about an ActiveMQ user
<code>list_brokers</code>	Returns a list of all brokers
<code>list_configuration_revisions</code>	Returns a list of all revisions for the specified configuration
<code>list_configurations</code>	Returns a list of all configurations
<code>list_tags</code>	Lists tags for a resource
<code>list_users</code>	Returns a list of all ActiveMQ users
<code>reboot_broker</code>	Reboots a broker
<code>update_broker</code>	Adds a pending configuration change to a broker
<code>update_configuration</code>	Updates the specified configuration
<code>update_user</code>	Updates the information for an ActiveMQ user

Examples

```
## Not run:
svc <- mq()
svc$create_broker(
  Foo = 123
)

## End(Not run)
```

Description

Amazon Managed Workflows for Apache Airflow

This section contains the Amazon Managed Workflows for Apache Airflow (MWAA) API reference documentation. For more information, see [What Is Amazon MWAA?](#).

Endpoints

- `api.airflow.{region}.amazonaws.com` - This endpoint is used for environment management.
 - `create_environment`
 - `delete_environment`
 - `get_environment`
 - `list_environments`
 - `list_tags_for_resource`
 - `tag_resource`
 - `untag_resource`
 - `update_environment`
- `env.airflow.{region}.amazonaws.com` - This endpoint is used to operate the Airflow environment.
 - `create_cli_token`
 - `create_web_login_token`
- `ops.airflow.{region}.amazonaws.com` - This endpoint is used to push environment metrics that track environment health.
 - `publish_metrics`

Regions

For a list of regions that Amazon MWAA supports, see [Region availability](#) in the *Amazon MWAA User Guide*.

Usage

```
mwa(config = list())
```

Arguments

<code>config</code>	<p>Optional configuration of credentials, endpoint, and/or region.</p> <ul style="list-style-type: none"> • access_key_id: AWS access key ID • secret_access_key: AWS secret access key • session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to <code>true</code> to force the request to use path-style addressing, i.e., <code>http://s3.amazonaws.com/BUCKET/KEY</code>.
---------------------	--

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 <- mwaas(
  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"
  )
)
```

Operations

create_cli_token	Creates a CLI token for the Airflow CLI
create_environment	Creates an Amazon Managed Workflows for Apache Airflow (MWAA) environment
create_web_login_token	Creates a web login token for the Airflow Web UI
delete_environment	Deletes an Amazon Managed Workflows for Apache Airflow (MWAA) environment
get_environment	Describes an Amazon Managed Workflows for Apache Airflow (MWAA) environment
list_environments	Lists the Amazon Managed Workflows for Apache Airflow (MWAA) environments
list_tags_for_resource	Lists the key-value tag pairs associated to the Amazon Managed Workflows for Apache Airflow (MWAA) environment
publish_metrics	Internal only
tag_resource	Associates key-value tag pairs to your Amazon Managed Workflows for Apache Airflow (MWAA) environment
untag_resource	Removes key-value tag pairs associated to your Amazon Managed Workflows for Apache Airflow (MWAA) environment
update_environment	Updates an Amazon Managed Workflows for Apache Airflow (MWAA) environment

Examples

```
## Not run:
svc <- mwaas()
svc$create_cli_token(
  Foo = 123
```

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

 schemas

Schemas

Description

Amazon EventBridge Schema Registry

Usage

```
schemas(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • access_key_id: AWS access key ID • secret_access_key: AWS secret access key • session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to true to force the request to use path-style addressing, i.e., <code>http://s3.amazonaws.com/BUCKET/KEY</code>.
--------	--

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 <- schemas(
  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"
)
)

```

Operations

create_discoverer	Creates a discoverer
create_registry	Creates a registry
create_schema	Creates a schema definition
delete_discoverer	Deletes a discoverer
delete_registry	Deletes a Registry
delete_resource_policy	Delete the resource-based policy attached to the specified registry
delete_schema	Delete a schema definition
delete_schema_version	Delete the schema version definition
describe_code_binding	Describe the code binding URI
describe_discoverer	Describes the discoverer
describe_registry	Describes the registry
describe_schema	Retrieve the schema definition
export_schema	Export schema
get_code_binding_source	Get the code binding source URI
get_discovered_schema	Get the discovered schema that was generated based on sampled events
get_resource_policy	Retrieves the resource-based policy attached to a given registry
list_discoverers	List the discoverers
list_registries	List the registries
list_schemas	List the schemas
list_schema_versions	Provides a list of the schema versions and related information
list_tags_for_resource	Get tags for resource
put_code_binding	Put code binding URI
put_resource_policy	The name of the policy
search_schemas	Search the schemas
start_discoverer	Starts the discoverer
stop_discoverer	Stops the discoverer
tag_resource	Add tags to a resource
untag_resource	Removes tags from a resource
update_discoverer	Updates the discoverer
update_registry	Updates a registry
update_schema	Updates the schema definition

Examples

```
## Not run:
svc <- schemas()
svc$create_discoverer(
  Foo = 123
)

## End(Not run)
```

sfn

AWS Step Functions

Description

AWS Step Functions is a service that lets you coordinate the components of distributed applications and microservices using visual workflows.

You can use Step Functions to build applications from individual components, each of which performs a discrete function, or *task*, allowing you to scale and change applications quickly. Step Functions provides a console that helps visualize the components of your application as a series of steps. Step Functions automatically triggers and tracks each step, and retries steps when there are errors, so your application executes predictably and in the right order every time. Step Functions logs the state of each step, so you can quickly diagnose and debug any issues.

Step Functions manages operations and underlying infrastructure to ensure your application is available at any scale. You can run tasks on AWS, your own servers, or any system that has access to AWS. You can access and use Step Functions using the console, the AWS SDKs, or an HTTP API. For more information about Step Functions, see the [AWS Step Functions Developer Guide](https://docs.aws.amazon.com/step-functions/latest/dg/welcome.html).

Usage

```
sfn(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none">• access_key_id: AWS access key ID• secret_access_key: AWS secret access key• session_token: AWS temporary session token• profile: The name of a profile to use. If not given, then the default profile is used.• anonymous: Set anonymous credentials.• endpoint: The complete URL to use for the constructed client.
--------	---

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

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 <- sfn(
  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"
  )
)
```

Operations

create_activity	Creates an activity
create_state_machine	Creates a state machine
delete_activity	Deletes an activity
delete_state_machine	Deletes a state machine
describe_activity	Describes an activity
describe_execution	Describes an execution
describe_state_machine	Describes a state machine
describe_state_machine_for_execution	Describes the state machine associated with a specific execution
get_activity_task	Used by workers to retrieve a task (with the specified activity ARN) which has been
get_execution_history	Returns the history of the specified execution as a list of events
list_activities	Lists the existing activities
list_executions	Lists the executions of a state machine that meet the filtering criteria

list_state_machines	Lists the existing state machines
list_tags_for_resource	List tags for a given resource
send_task_failure	Used by activity workers and task states using the callback pattern to report that the
send_task_heartbeat	Used by activity workers and task states using the callback pattern to report to Step F
send_task_success	Used by activity workers and task states using the callback pattern to report that the
start_execution	Starts a state machine execution
start_sync_execution	Starts a Synchronous Express state machine execution
stop_execution	Stops an execution
tag_resource	Add a tag to a Step Functions resource
untag_resource	Remove a tag from a Step Functions resource
update_state_machine	Updates an existing state machine by modifying its definition, roleArn, or loggingCo

Examples

```
## Not run:
svc <- sfn()
svc$create_activity(
  Foo = 123
)

## End(Not run)
```

sns

Amazon Simple Notification Service

Description

Amazon Simple Notification Service (Amazon SNS) is a web service that enables you to build distributed web-enabled applications. Applications can use Amazon SNS to easily push real-time notification messages to interested subscribers over multiple delivery protocols. For more information about this product see the [Amazon SNS product page](#). For detailed information about Amazon SNS features and their associated API calls, see the [Amazon SNS Developer Guide](#).

For information on the permissions you need to use this API, see [Identity and access management in Amazon SNS](#) in the *Amazon SNS Developer Guide*.

We also provide SDKs that enable you to access Amazon SNS from your preferred programming language. The SDKs contain functionality that automatically takes care of tasks such as: cryptographically signing your service requests, retrying requests, and handling error responses. For a list of available SDKs, go to [Tools for Amazon Web Services](#).

Usage

```
sns(config = list())
```


Arguments

config	Optional configuration of credentials, endpoint, and/or region. <ul style="list-style-type: none"> • access_key_id: AWS access key ID • secret_access_key: AWS secret access key • session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to <code>true</code> to force the request to use path-style addressing, i.e., <code>http://s3.amazonaws.com/BUCKET/KEY</code>.
--------	---

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 <- sns(
  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"
  )
)
```

Operations

[add_permission](#)

Adds a statement to a topic's access control policy, granting access for the specified

check_if_phone_number_is_opted_out	Accepts a phone number and indicates whether the phone holder has opted out of receiving messages
confirm_subscription	Verifies an endpoint owner's intent to receive messages by validating the token sent by the endpoint
create_platform_application	Creates a platform application object for one of the supported push notification services
create_platform_endpoint	Creates an endpoint for a device and mobile app on one of the supported push notification services
create_sms_sandbox_phone_number	Adds a destination phone number to an Amazon Web Services account in the SMS sandbox
create_topic	Creates a topic to which notifications can be published
delete_endpoint	Deletes the endpoint for a device and mobile app from Amazon SNS
delete_platform_application	Deletes a platform application object for one of the supported push notification services
delete_sms_sandbox_phone_number	Deletes an Amazon Web Services account's verified or pending phone number from the SMS sandbox
delete_topic	Deletes a topic and all its subscriptions
get_endpoint_attributes	Retrieves the endpoint attributes for a device on one of the supported push notification services
get_platform_application_attributes	Retrieves the attributes of the platform application object for the supported push notification services
get_sms_attributes	Returns the settings for sending SMS messages from your Amazon Web Services account
get_sms_sandbox_account_status	Retrieves the SMS sandbox status for the calling Amazon Web Services account in the SMS sandbox
get_subscription_attributes	Returns all of the properties of a subscription
get_topic_attributes	Returns all of the properties of a topic
list_endpoints_by_platform_application	Lists the endpoints and endpoint attributes for devices in a supported push notification service
list_origination_numbers	Lists the calling Amazon Web Services account's dedicated origination numbers and their status
list_phone_numbers_opted_out	Returns a list of phone numbers that are opted out, meaning you cannot send SMS messages to them
list_platform_applications	Lists the platform application objects for the supported push notification services, sorted by creation time
list_sms_sandbox_phone_numbers	Lists the calling Amazon Web Services account's current verified and pending destination phone numbers in the SMS sandbox
list_subscriptions	Returns a list of the requester's subscriptions
list_subscriptions_by_topic	Returns a list of the subscriptions to a specific topic
list_tags_for_resource	List all tags added to the specified Amazon SNS topic
list_topics	Returns a list of the requester's topics
opt_in_phone_number	Use this request to opt in a phone number that is opted out, which enables you to send messages to it
publish	Sends a message to an Amazon SNS topic, a text message (SMS message) directly to a phone number, or an email message
publish_batch	Publishes up to ten messages to the specified topic
remove_permission	Removes a statement from a topic's access control policy
set_endpoint_attributes	Sets the attributes for an endpoint for a device on one of the supported push notification services
set_platform_application_attributes	Sets the attributes of the platform application object for the supported push notification services
set_sms_attributes	Use this request to set the default settings for sending SMS messages and receiving messages to your Amazon Web Services account
set_subscription_attributes	Allows a subscription owner to set an attribute of the subscription to a new value
set_topic_attributes	Allows a topic owner to set an attribute of the topic to a new value
subscribe	Subscribes an endpoint to an Amazon SNS topic
tag_resource	Add tags to the specified Amazon SNS topic
unsubscribe	Deletes a subscription
untag_resource	Remove tags from the specified Amazon SNS topic
verify_sms_sandbox_phone_number	Verifies a destination phone number with a one-time password (OTP) for the calling Amazon Web Services account in the SMS sandbox

Examples

```
## Not run:
svc <- sns()
svc$add_permission(
  Foo = 123
)
```

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

sqs

Amazon Simple Queue Service

Description

Welcome to the *Amazon SQS API Reference*.

Amazon SQS is a reliable, highly-scalable hosted queue for storing messages as they travel between applications or microservices. Amazon SQS moves data between distributed application components and helps you decouple these components.

For information on the permissions you need to use this API, see [Identity and access management](#) in the *Amazon SQS Developer Guide*.

You can use [Amazon Web Services SDKs](#) to access Amazon SQS using your favorite programming language. The SDKs perform tasks such as the following automatically:

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

Additional information

- [Amazon SQS Product Page](#)
- *Amazon SQS Developer Guide*
 - [Making API Requests](#)
 - [Amazon SQS Message Attributes](#)
 - [Amazon SQS Dead-Letter Queues](#)
- [Amazon SQS in the Command Line Interface](#)
- *Amazon Web Services General Reference*
 - [Regions and Endpoints](#)

Usage

```
sqs(config = list())
```

Arguments

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

- **access_key_id**: AWS access key ID
- **secret_access_key**: AWS secret access key
- **session_token**: AWS temporary 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`.

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 <- sqs(
  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"
  )
)
```

Operations

add_permission	Adds a permission to a queue for a specific principal
change_message_visibility	Changes the visibility timeout of a specified message in a queue to a new value
change_message_visibility_batch	Changes the visibility timeout of multiple messages
create_queue	Creates a new standard or FIFO queue
delete_message	Deletes the specified message from the specified queue
delete_message_batch	Deletes up to ten messages from the specified queue
delete_queue	Deletes the queue specified by the <code>QueueUrl</code> , regardless of the queue's contents
get_queue_attributes	Gets attributes for the specified queue

get_queue_url	Returns the URL of an existing Amazon SQS queue
list_dead_letter_source_queues	Returns a list of your queues that have the RedrivePolicy queue attribute configured with a
list_queues	Returns a list of your queues in the current region
list_queue_tags	List all cost allocation tags added to the specified Amazon SQS queue
purge_queue	Deletes the messages in a queue specified by the QueueURL parameter
receive_message	Retrieves one or more messages (up to 10), from the specified queue
remove_permission	Revokes any permissions in the queue policy that matches the specified Label parameter
send_message	Delivers a message to the specified queue
send_message_batch	Delivers up to ten messages to the specified queue
set_queue_attributes	Sets the value of one or more queue attributes
tag_queue	Add cost allocation tags to the specified Amazon SQS queue
untag_queue	Remove cost allocation tags from the specified Amazon SQS queue

Examples

```
## Not run:
svc <- sqs()
svc$add_permission(
  Foo = 123
)

## End(Not run)
```

swf

Amazon Simple Workflow Service

Description

The Amazon Simple Workflow Service (Amazon SWF) makes it easy to build applications that use Amazon's cloud to coordinate work across distributed components. In Amazon SWF, a *task* represents a logical unit of work that is performed by a component of your workflow. Coordinating tasks in a workflow involves managing intertask dependencies, scheduling, and concurrency in accordance with the logical flow of the application.

Amazon SWF gives you full control over implementing tasks and coordinating them without worrying about underlying complexities such as tracking their progress and maintaining their state.

This documentation serves as reference only. For a broader overview of the Amazon SWF programming model, see the <https://docs.aws.amazon.com/amazonswf/latest/developerguide/> *Amazon SWF Developer Guide* .

Usage

```
swf(config = list())
```

Arguments

config	<p>Optional configuration of credentials, endpoint, and/or region.</p> <ul style="list-style-type: none"> • access_key_id: AWS access key ID • secret_access_key: AWS secret access key • session_token: AWS temporary session token • profile: The name of a profile to use. If not given, then the default profile is used. • anonymous: Set anonymous credentials. • endpoint: The complete URL to use for the constructed client. • region: The AWS Region used in instantiating the client. • close_connection: Immediately close all HTTP connections. • timeout: The time in seconds till a timeout exception is thrown when attempting to make a connection. The default is 60 seconds. • s3_force_path_style: Set this to <code>true</code> to force the request to use path-style addressing, i.e., <code>http://s3.amazonaws.com/BUCKET/KEY</code>.
--------	--

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 <- swf(
  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"
  )
)
```

Operations

[count_closed_workflow_executions](#) Returns the number of closed workflow executions within the given domain that meet t

<code>count_open_workflow_executions</code>	Returns the number of open workflow executions within the given domain that meet the filter
<code>count_pending_activity_tasks</code>	Returns the estimated number of activity tasks in the specified task list
<code>count_pending_decision_tasks</code>	Returns the estimated number of decision tasks in the specified task list
<code>deprecate_activity_type</code>	Deprecates the specified activity type
<code>deprecate_domain</code>	Deprecates the specified domain
<code>deprecate_workflow_type</code>	Deprecates the specified workflow type
<code>describe_activity_type</code>	Returns information about the specified activity type
<code>describe_domain</code>	Returns information about the specified domain, including description and status
<code>describe_workflow_execution</code>	Returns information about the specified workflow execution including its type and some other details
<code>describe_workflow_type</code>	Returns information about the specified workflow type
<code>get_workflow_execution_history</code>	Returns the history of the specified workflow execution
<code>list_activity_types</code>	Returns information about all activities registered in the specified domain that match the filter
<code>list_closed_workflow_executions</code>	Returns a list of closed workflow executions in the specified domain that meet the filter
<code>list_domains</code>	Returns the list of domains registered in the account
<code>list_open_workflow_executions</code>	Returns a list of open workflow executions in the specified domain that meet the filter
<code>list_tags_for_resource</code>	List tags for a given domain
<code>list_workflow_types</code>	Returns information about workflow types in the specified domain
<code>poll_for_activity_task</code>	Used by workers to get an ActivityTask from the specified activity taskList
<code>poll_for_decision_task</code>	Used by deciders to get a DecisionTask from the specified decision taskList
<code>record_activity_task_heartbeat</code>	Used by activity workers to report to the service that the ActivityTask represented by the taskToken is still running
<code>register_activity_type</code>	Registers a new activity type along with its configuration settings in the specified domain
<code>register_domain</code>	Registers a new domain
<code>register_workflow_type</code>	Registers a new workflow type and its configuration settings in the specified domain
<code>request_cancel_workflow_execution</code>	Records a WorkflowExecutionCancelRequested event in the currently running workflow execution history
<code>respond_activity_task_canceled</code>	Used by workers to tell the service that the ActivityTask identified by the taskToken was canceled
<code>respond_activity_task_completed</code>	Used by workers to tell the service that the ActivityTask identified by the taskToken completed
<code>respond_activity_task_failed</code>	Used by workers to tell the service that the ActivityTask identified by the taskToken has failed
<code>respond_decision_task_completed</code>	Used by deciders to tell the service that the DecisionTask identified by the taskToken has completed
<code>signal_workflow_execution</code>	Records a WorkflowExecutionSignaled event in the workflow execution history and creates a new execution
<code>start_workflow_execution</code>	Starts an execution of the workflow type in the specified domain using the provided workflow type
<code>tag_resource</code>	Add a tag to a Amazon SWF domain
<code>terminate_workflow_execution</code>	Records a WorkflowExecutionTerminated event and forces closure of the workflow execution
<code>undeprecate_activity_type</code>	Undeprecates a previously deprecated activity type
<code>undeprecate_domain</code>	Undeprecates a previously deprecated domain
<code>undeprecate_workflow_type</code>	Undeprecates a previously deprecated workflow type
<code>untag_resource</code>	Remove a tag from a Amazon SWF domain

Examples

```
## Not run:
svc <- swf()
svc$count_closed_workflow_executions(
  Foo = 123
)

## End(Not run)
```

Index

activate_event_source, [3](#)
add_permission, [17, 20](#)
associate_tracker_consumer, [6](#)

batch_delete_device_position_history, [6](#)
batch_delete_geofence, [6](#)
batch_evaluate_geofences, [6](#)
batch_get_device_position, [6](#)
batch_put_geofence, [6](#)
batch_update_device_position, [6](#)

calculate_route, [6](#)
calculate_route_matrix, [6](#)
cancel_replay, [3](#)
change_message_visibility, [20](#)
change_message_visibility_batch, [20](#)
check_if_phone_number_is_opted_out, [18](#)
confirm_subscription, [18](#)
count_closed_workflow_executions, [22](#)
count_open_workflow_executions, [23](#)
count_pending_activity_tasks, [23](#)
count_pending_decision_tasks, [23](#)
create_activity, [15](#)
create_api_destination, [3](#)
create_archive, [3](#)
create_broker, [9](#)
create_cli_token, [10, 11](#)
create_configuration, [9](#)
create_connection, [3](#)
create_discoverer, [13](#)
create_endpoint, [3](#)
create_environment, [10, 11](#)
create_event_bus, [3](#)
create_geofence_collection, [6](#)
create_map, [6](#)
create_partner_event_source, [3](#)
create_place_index, [6](#)
create_platform_application, [18](#)
create_platform_endpoint, [18](#)

create_queue, [20](#)
create_registry, [13](#)
create_route_calculator, [6](#)
create_schema, [13](#)
create_sms_sandbox_phone_number, [18](#)
create_state_machine, [15](#)
create_tags, [9](#)
create_topic, [18](#)
create_tracker, [6](#)
create_user, [9](#)
create_web_login_token, [10, 11](#)

deactivate_event_source, [3](#)
deauthorize_connection, [3](#)
delete_activity, [15](#)
delete_api_destination, [3](#)
delete_archive, [4](#)
delete_broker, [9](#)
delete_connection, [4](#)
delete_discoverer, [13](#)
delete_endpoint, [4, 18](#)
delete_environment, [10, 11](#)
delete_event_bus, [4](#)
delete_geofence_collection, [6](#)
delete_map, [6](#)
delete_message, [20](#)
delete_message_batch, [20](#)
delete_partner_event_source, [4](#)
delete_place_index, [6](#)
delete_platform_application, [18](#)
delete_queue, [20](#)
delete_registry, [13](#)
delete_resource_policy, [13](#)
delete_route_calculator, [6](#)
delete_rule, [4](#)
delete_schema, [13](#)
delete_schema_version, [13](#)
delete_sms_sandbox_phone_number, [18](#)
delete_state_machine, [15](#)
delete_tags, [9](#)

- delete_topic, [18](#)
- delete_tracker, [6](#)
- delete_user, [9](#)
- deprecate_activity_type, [23](#)
- deprecate_domain, [23](#)
- deprecate_workflow_type, [23](#)
- describe_activity, [15](#)
- describe_activity_type, [23](#)
- describe_api_destination, [4](#)
- describe_archive, [4](#)
- describe_broker, [9](#)
- describe_broker_engine_types, [9](#)
- describe_broker_instance_options, [9](#)
- describe_code_binding, [13](#)
- describe_configuration, [9](#)
- describe_configuration_revision, [9](#)
- describe_connection, [4](#)
- describe_discoverer, [13](#)
- describe_domain, [23](#)
- describe_endpoint, [4](#)
- describe_event_bus, [4](#)
- describe_event_source, [4](#)
- describe_execution, [15](#)
- describe_geofence_collection, [6](#)
- describe_map, [6](#)
- describe_partner_event_source, [4](#)
- describe_place_index, [6](#)
- describe_registry, [13](#)
- describe_replay, [4](#)
- describe_route_calculator, [6](#)
- describe_rule, [4](#)
- describe_schema, [13](#)
- describe_state_machine, [15](#)
- describe_state_machine_for_execution, [15](#)
- describe_tracker, [6](#)
- describe_user, [9](#)
- describe_workflow_execution, [23](#)
- describe_workflow_type, [23](#)
- disable_rule, [4](#)
- disassociate_tracker_consumer, [6](#)

- enable_rule, [4](#)
- eventbridge, [2](#)
- export_schema, [13](#)

- get_activity_task, [15](#)
- get_code_binding_source, [13](#)
- get_device_position, [6](#)
- get_device_position_history, [7](#)
- get_discovered_schema, [13](#)
- get_endpoint_attributes, [18](#)
- get_environment, [10, 11](#)
- get_execution_history, [15](#)
- get_geofence, [7](#)
- get_map_glyphs, [7](#)
- get_map_sprites, [7](#)
- get_map_style_descriptor, [7](#)
- get_map_tile, [7](#)
- get_platform_application_attributes, [18](#)
- get_queue_attributes, [20](#)
- get_queue_url, [21](#)
- get_resource_policy, [13](#)
- get_sms_attributes, [18](#)
- get_sms_sandbox_account_status, [18](#)
- get_subscription_attributes, [18](#)
- get_topic_attributes, [18](#)
- get_workflow_execution_history, [23](#)

- list_activities, [15](#)
- list_activity_types, [23](#)
- list_api_destinations, [4](#)
- list_archives, [4](#)
- list_brokers, [9](#)
- list_closed_workflow_executions, [23](#)
- list_configuration_revisions, [9](#)
- list_configurations, [9](#)
- list_connections, [4](#)
- list_dead_letter_source_queues, [21](#)
- list_device_positions, [7](#)
- list_discoverers, [13](#)
- list_domains, [23](#)
- list_endpoints, [4](#)
- list_endpoints_by_platform_application, [18](#)
- list_environments, [10, 11](#)
- list_event_buses, [4](#)
- list_event_sources, [4](#)
- list_executions, [15](#)
- list_geofence_collections, [7](#)
- list_geofences, [7](#)
- list_maps, [7](#)
- list_open_workflow_executions, [23](#)
- list_origination_numbers, [18](#)
- list_partner_event_source_accounts, [4](#)
- list_partner_event_sources, [4](#)
- list_phone_numbers_opted_out, [18](#)

- list_place_indexes, [7](#)
- list_platform_applications, [18](#)
- list_queue_tags, [21](#)
- list_queues, [21](#)
- list_registries, [13](#)
- list_replays, [4](#)
- list_route_calculators, [7](#)
- list_rule_names_by_target, [4](#)
- list_rules, [4](#)
- list_schema_versions, [13](#)
- list_schemas, [13](#)
- list_sms_sandbox_phone_numbers, [18](#)
- list_state_machines, [16](#)
- list_subscriptions, [18](#)
- list_subscriptions_by_topic, [18](#)
- list_tags, [9](#)
- list_tags_for_resource, [4](#), [7](#), [10](#), [11](#), [13](#),
[16](#), [18](#), [23](#)
- list_targets_by_rule, [4](#)
- list_topics, [18](#)
- list_tracker_consumers, [7](#)
- list_trackers, [7](#)
- list_users, [9](#)
- list_workflow_types, [23](#)
- locationService, [5](#)

- mq, [7](#)
- mWaa, [9](#)

- opt_in_phone_number, [18](#)

- poll_for_activity_task, [23](#)
- poll_for_decision_task, [23](#)
- publish, [18](#)
- publish_batch, [18](#)
- publish_metrics, [10](#), [11](#)
- purge_queue, [21](#)
- put_code_binding, [13](#)
- put_events, [4](#)
- put_geofence, [7](#)
- put_partner_events, [4](#)
- put_permission, [4](#)
- put_resource_policy, [13](#)
- put_rule, [4](#)
- put_targets, [4](#)

- reboot_broker, [9](#)
- receive_message, [21](#)
- record_activity_task_heartbeat, [23](#)

- register_activity_type, [23](#)
- register_domain, [23](#)
- register_workflow_type, [23](#)
- remove_permission, [4](#), [18](#), [21](#)
- remove_targets, [4](#)
- request_cancel_workflow_execution, [23](#)
- respond_activity_task_canceled, [23](#)
- respond_activity_task_completed, [23](#)
- respond_activity_task_failed, [23](#)
- respond_decision_task_completed, [23](#)

- schemas, [12](#)
- search_place_index_for_position, [7](#)
- search_place_index_for_suggestions, [7](#)
- search_place_index_for_text, [7](#)
- search_schemas, [13](#)
- send_message, [21](#)
- send_message_batch, [21](#)
- send_task_failure, [16](#)
- send_task_heartbeat, [16](#)
- send_task_success, [16](#)
- set_endpoint_attributes, [18](#)
- set_platform_application_attributes,
[18](#)
- set_queue_attributes, [21](#)
- set_sms_attributes, [18](#)
- set_subscription_attributes, [18](#)
- set_topic_attributes, [18](#)
- sfn, [14](#)
- signal_workflow_execution, [23](#)
- sns, [16](#)
- sqs, [19](#)
- start_discoverer, [13](#)
- start_execution, [16](#)
- start_replay, [4](#)
- start_sync_execution, [16](#)
- start_workflow_execution, [23](#)
- stop_discoverer, [13](#)
- stop_execution, [16](#)
- subscribe, [18](#)
- swf, [21](#)

- tag_queue, [21](#)
- tag_resource, [4](#), [7](#), [10](#), [11](#), [13](#), [16](#), [18](#), [23](#)
- terminate_workflow_execution, [23](#)
- test_event_pattern, [4](#)

- undeprecate_activity_type, [23](#)
- undeprecate_domain, [23](#)

undeprecate_workflow_type, 23
unsubscribe, 18
untag_queue, 21
untag_resource, 4, 7, 10, 11, 13, 16, 18, 23
update_api_destination, 4
update_archive, 4
update_broker, 9
update_configuration, 9
update_connection, 4
update_discoverer, 13
update_endpoint, 4
update_environment, 10, 11
update_geofence_collection, 7
update_map, 7
update_place_index, 7
update_registry, 13
update_route_calculator, 7
update_schema, 13
update_state_machine, 16
update_tracker, 7
update_user, 9

verify_sms_sandbox_phone_number, 18