Package 'paws.analytics'

August 23, 2021

Title 'Amazon Web Services' Analytics Services

Version 0.1.12

Description Interface to 'Amazon Web Services' 'analytics' services, including 'Elastic MapReduce' 'Hadoop' and 'Spark' big data service, 'Elasticsearch' search engine, 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 (>= 0.3.0)

Suggests testthat

Encoding UTF-8

RoxygenNote 7.1.1

Collate 'athena_service.R' 'athena_interfaces.R' 'athena_operations.R' 'cloudsearch_service.R' 'cloudsearch_interfaces.R' 'cloudsearch_operations.R' 'cloudsearchdomain_service.R' 'cloudsearchdomain_interfaces.R' 'cloudsearchdomain_operations.R' 'datapipeline_service.R' 'datapipeline_interfaces.R' 'datapipeline_operations.R' 'elasticsearchservice_service.R' 'elasticsearchservice_interfaces.R' 'elasticsearchservice_operations.R' 'emr_service.R' 'emr_interfaces.R' 'emr_operations.R' 'firehose_service.R' 'firehose_interfaces.R' 'firehose_operations.R' 'glue_service.R' 'glue_interfaces.R' 'glue_operations.R' 'kafka_service.R' 'kafka_interfaces.R' 'kafka_operations.R' 'kinesis_service.R' 'kinesis_interfaces.R' 'kinesis_operations.R' 'kinesisanalytics_service.R' 'kinesisanalytics_interfaces.R' 'kinesisanalytics_operations.R' 'kinesisanalyticsv2_service.R' 'kinesisanalyticsv2_interfaces.R' 'kinesisanalyticsv2_operations.R' 'mturk_service.R' 'mturk_interfaces.R' 'mturk_operations.R'

'quicksight_service.R' 'quicksight_interfaces.R'
'quicksight_operations.R'

NeedsCompilation no

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

Maintainer David Kretch <david.kretch@gmail.com>

Repository CRAN

Date/Publication 2021-08-23 07:11:20 UTC

R topics documented:

`athena` *Amazon Athena*

Description

Amazon Athena is an interactive query service that lets you use standard SQL to analyze data directly in Amazon S3. You can point Athena at your data in Amazon S3 and run ad-hoc queries and get results in seconds. Athena is serverless, so there is no infrastructure to set up or manage. You pay only for the queries you run. Athena scales automatically—executing queries in parallel—so results are fast, even with large datasets and complex queries. For more information, see What is Amazon Athena in the *Amazon Athena User Guide*.

If you connect to Athena using the JDBC driver, use version 1.1.0 of the driver or later with the Amazon Athena API. Earlier version drivers do not support the API. For more information and to download the driver, see Accessing Amazon Athena with JDBC.

For code samples using the AWS SDK for Java, see Examples and Code Samples in the *Amazon Athena User Guide*.

Usage

```
athena(config = list())
```

Arguments

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

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 <- athena(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

batch_get_named_query	Returns the details of a single named query or a list of up to 50 queries, which you provide as ar
batch_get_query_execution	Returns the details of a single query execution or a list of up to 50 query executions, which you
create_data_catalog	Creates (registers) a data catalog with the specified name and properties
create_named_query	Creates a named query in the specified workgroup
create_work_group	Creates a workgroup with the specified name
delete_data_catalog	Deletes a data catalog
delete_named_query	Deletes the named query if you have access to the workgroup in which the query was saved
delete_work_group	Deletes the workgroup with the specified name
get_database	Returns a database object for the specfied database and data catalog
get_data_catalog	Returns the specified data catalog
get_named_query	Returns information about a single query
get_query_execution	Returns information about a single execution of a query if you have access to the workgroup in
get_query_results	Streams the results of a single query execution specified by QueryExecutionId from the Athena
get_table_metadata	Returns table metadata for the specified catalog, database, and table
get_work_group	Returns information about the workgroup with the specified name
list_databases	Lists the databases in the specified data catalog

list_data_catalogs	Lists the data catalogs in the current AWS account
list_named_queries	Provides a list of available query IDs only for queries saved in the specified workgroup
list_query_executions	Provides a list of available query execution IDs for the queries in the specified workgroup
list_table_metadata	Lists the metadata for the tables in the specified data catalog database
list_tags_for_resource	Lists the tags associated with an Athena workgroup or data catalog resource
list_work_groups	Lists available workgroups for the account
start_query_execution	Runs the SQL query statements contained in the Query
stop_query_execution	Stops a query execution
tag_resource	Adds one or more tags to an Athena resource
untag_resource	Removes one or more tags from a data catalog or workgroup resource
update_data_catalog	Updates the data catalog that has the specified name
update_work_group	Updates the workgroup with the specified name

Examples

```
## Not run:
svc <- athena()
svc$batch_get_named_query(
  Foo = 123
)

## End(Not run)
```

`cloudsearch` *Amazon CloudSearch*

Description

Amazon CloudSearch Configuration Service

You use the Amazon CloudSearch configuration service to create, configure, and manage search domains. Configuration service requests are submitted using the AWS Query protocol. AWS Query requests are HTTP or HTTPS requests submitted via HTTP GET or POST with a query parameter named Action.

The endpoint for configuration service requests is region-specific: cloudsearch.*region*.amazonaws.com. For example, cloudsearch.us-east-1.amazonaws.com. For a current list of supported regions and endpoints, see Regions and Endpoints.

Usage

```
cloudsearch(config = list())
```

Arguments

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

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 <- cloudsearch(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

build_suggesters	Indexes the search suggestions
create_domain	Creates a new search domain
define_analysis_scheme	Configures an analysis scheme that can be applied to a text or text-array field to define l
define_expression	Configures an Expression for the search domain
define_index_field	Configures an IndexField for the search domain
define_suggester	Configures a suggester for a domain
delete_analysis_scheme	Deletes an analysis scheme
delete_domain	Permanently deletes a search domain and all of its data
delete_expression	Removes an Expression from the search domain
delete_index_field	Removes an IndexField from the search domain
delete_suggester	Deletes a suggester
describe_analysis_schemes	Gets the analysis schemes configured for a domain
describe_availability_options	Gets the availability options configured for a domain
describe_domain_endpoint_options	Returns the domain's endpoint options, specifically whether all requests to the domain r
describe_domains	Gets information about the search domains owned by this account
describe_expressions	Gets the expressions configured for the search domain
describe_index_fields	Gets information about the index fields configured for the search domain
describe_scaling_parameters	Gets the scaling parameters configured for a domain
describe_service_access_policies	Gets information about the access policies that control access to the domain's document
describe_suggesters	Gets the suggesters configured for a domain
index_documents	Tells the search domain to start indexing its documents using the latest indexing options
list_domain_names	Lists all search domains owned by an account
update_availability_options	Configures the availability options for a domain
update_domain_endpoint_options	Updates the domain's endpoint options, specifically whether all requests to the domain r

update_scaling_parameters	Configures scaling parameters for a domain
update_service_access_policies	Configures the access rules that control access to the domain's document and search enc

Examples

```
## Not run:
svc <- cloudsearch()
svc$build_suggesters(
  Foo = 123
)

## End(Not run)
```

`cloudsearchdomain` *Amazon CloudSearch Domain*

Description

You use the AmazonCloudSearch2013 API to upload documents to a search domain and search those documents.

The endpoints for submitting `upload_documents`, `search`, and `suggest` requests are domain-specific. To get the endpoints for your domain, use the Amazon CloudSearch configuration service `DescribeDomains` action. The domain endpoints are also displayed on the domain dashboard in the Amazon CloudSearch console. You submit suggest requests to the search endpoint.

For more information, see the Amazon CloudSearch Developer Guide.

Usage

```
cloudsearchdomain(config = list())
```

Arguments

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

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 <- cloudsearchdomain(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

search	Retrieves a list of documents that match the specified search criteria
suggest	Retrieves autocomplete suggestions for a partial query string
upload_documents	Posts a batch of documents to a search domain for indexing

Examples

```
## Not run:
svc <- cloudsearchdomain()
svc$search(
  Foo = 123
)

## End(Not run)
```

---

datapipeline *AWS Data Pipeline*

Description

AWS Data Pipeline configures and manages a data-driven workflow called a pipeline. AWS Data Pipeline handles the details of scheduling and ensuring that data dependencies are met so that your application can focus on processing the data.

AWS Data Pipeline provides a JAR implementation of a task runner called AWS Data Pipeline Task Runner. AWS Data Pipeline Task Runner provides logic for common data management scenarios, such as performing database queries and running data analysis using Amazon Elastic MapReduce

(Amazon EMR). You can use AWS Data Pipeline Task Runner as your task runner, or you can write your own task runner to provide custom data management.

AWS Data Pipeline implements two main sets of functionality. Use the first set to create a pipeline and define data sources, schedules, dependencies, and the transforms to be performed on the data. Use the second set in your task runner application to receive the next task ready for processing. The logic for performing the task, such as querying the data, running data analysis, or converting the data from one format to another, is contained within the task runner. The task runner performs the task assigned to it by the web service, reporting progress to the web service as it does so. When the task is done, the task runner reports the final success or failure of the task to the web service.

Usage

```
datapipeline(config = list())
```

Arguments

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

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 <- datapipeline(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

activate_pipeline	Validates the specified pipeline and starts processing pipeline tasks
add_tags	Adds or modifies tags for the specified pipeline
create_pipeline	Creates a new, empty pipeline
deactivate_pipeline	Deactivates the specified running pipeline
delete_pipeline	Deletes a pipeline, its pipeline definition, and its run history
describe_objects	Gets the object definitions for a set of objects associated with the pipeline

describe_pipelines	Retrieves metadata about one or more pipelines
evaluate_expression	Task runners call EvaluateExpression to evaluate a string in the context of the specified object
get_pipeline_definition	Gets the definition of the specified pipeline
list_pipelines	Lists the pipeline identifiers for all active pipelines that you have permission to access
poll_for_task	Task runners call PollForTask to receive a task to perform from AWS Data Pipeline
put_pipeline_definition	Adds tasks, schedules, and preconditions to the specified pipeline
query_objects	Queries the specified pipeline for the names of objects that match the specified set of condition
remove_tags	Removes existing tags from the specified pipeline
report_task_progress	Task runners call ReportTaskProgress when assigned a task to acknowledge that it has the task
report_task_runner_heartbeat	Task runners call ReportTaskRunnerHeartbeat every 15 minutes to indicate that they are opera
set_status	Requests that the status of the specified physical or logical pipeline objects be updated in the s
set_task_status	Task runners call SetTaskStatus to notify AWS Data Pipeline that a task is completed and prov
validate_pipeline_definition	Validates the specified pipeline definition to ensure that it is well formed and can be run witho

Examples

```
## Not run:
svc <- datapipeline()
svc$activate_pipeline(
  Foo = 123
)

## End(Not run)
```

`elasticsearchservice` *Amazon Elasticsearch Service*

Description

Amazon Elasticsearch Configuration Service

Use the Amazon Elasticsearch Configuration API to create, configure, and manage Elasticsearch domains.

For sample code that uses the Configuration API, see the Amazon Elasticsearch Service Developer Guide. The guide also contains sample code for sending signed HTTP requests to the Elasticsearch APIs.

The endpoint for configuration service requests is region-specific: es.*region*.amazonaws.com. For example, es.us-east-1.amazonaws.com. For a current list of supported regions and endpoints, see Regions and Endpoints.

Usage

```
elasticsearchservice(config = list())
```

Arguments

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

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 <- elasticsearchservice(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

accept_inbound_cross_cluster_search_connection	Allows the destination domain owner to accept an inbound cross-clus
add_tags	Attaches tags to an existing Elasticsearch domain
associate_package	Associates a package with an Amazon ES domain
cancel_elasticsearch_service_software_update	Cancels a scheduled service software update for an Amazon ES doma
create_elasticsearch_domain	Creates a new Elasticsearch domain
create_outbound_cross_cluster_search_connection	Creates a new cross-cluster search connection from a source domain
create_package	Create a package for use with Amazon ES domains
delete_elasticsearch_domain	Permanently deletes the specified Elasticsearch domain and all of its
delete_elasticsearch_service_role	Deletes the service-linked role that Elasticsearch Service uses to man
delete_inbound_cross_cluster_search_connection	Allows the destination domain owner to delete an existing inbound cr
delete_outbound_cross_cluster_search_connection	Allows the source domain owner to delete an existing outbound cross
delete_package	Delete the package
describe_elasticsearch_domain	Returns domain configuration information about the specified Elastic
describe_elasticsearch_domain_config	Provides cluster configuration information about the specified Elastic
describe_elasticsearch_domains	Returns domain configuration information about the specified Elastic
describe_elasticsearch_instance_type_limits	Describe Elasticsearch Limits for a given InstanceType and Elasticsea
describe_inbound_cross_cluster_search_connections	Lists all the inbound cross-cluster search connections for a destinatio
describe_outbound_cross_cluster_search_connections	Lists all the outbound cross-cluster search connections for a source d
describe_packages	Describes all packages available to Amazon ES
describe_reserved_elasticsearch_instance_offerings	Lists available reserved Elasticsearch instance offerings

describe_reserved_elasticsearch_instances	Returns information about reserved Elasticsearch instances for this ac
dissociate_package	Dissociates a package from the Amazon ES domain
get_compatible_elasticsearch_versions	Returns a list of upgrade compatible Elastisearch versions
get_package_version_history	Returns a list of versions of the package, along with their creation tim
get_upgrade_history	Retrieves the complete history of the last 10 upgrades that were perfo
get_upgrade_status	Retrieves the latest status of the last upgrade or upgrade eligibility ch
list_domain_names	Returns the name of all Elasticsearch domains owned by the current u
list_domains_for_package	Lists all Amazon ES domains associated with the package
list_elasticsearch_instance_types	List all Elasticsearch instance types that are supported for given Elast
list_elasticsearch_versions	List all supported Elasticsearch versions
list_packages_for_domain	Lists all packages associated with the Amazon ES domain
list_tags	Returns all tags for the given Elasticsearch domain
purchase_reserved_elasticsearch_instance_offering	Allows you to purchase reserved Elasticsearch instances
reject_inbound_cross_cluster_search_connection	Allows the destination domain owner to reject an inbound cross-clust
remove_tags	Removes the specified set of tags from the specified Elasticsearch do
start_elasticsearch_service_software_update	Schedules a service software update for an Amazon ES domain
update_elasticsearch_domain_config	Modifies the cluster configuration of the specified Elasticsearch doma
update_package	Updates a package for use with Amazon ES domains
upgrade_elasticsearch_domain	Allows you to either upgrade your domain or perform an Upgrade eli

Examples

```
## Not run:
svc <- elasticsearchservice()
svc$accept_inbound_cross_cluster_search_connection(
  Foo = 123
)

## End(Not run)
```

`emr` *Amazon Elastic MapReduce*

Description

Amazon EMR is a web service that makes it easier to process large amounts of data efficiently. Amazon EMR uses Hadoop processing combined with several AWS services to do tasks such as web indexing, data mining, log file analysis, machine learning, scientific simulation, and data warehouse management.

Usage

```
emr(config = list())
```

Arguments

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

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 <- emr(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

add_instance_fleet	Adds an instance fleet to a running cluster
add_instance_groups	Adds one or more instance groups to a running cluster
add_job_flow_steps	AddJobFlowSteps adds new steps to a running cluster
add_tags	Adds tags to an Amazon EMR resource
cancel_steps	Cancels a pending step or steps in a running cluster
create_security_configuration	Creates a security configuration, which is stored in the service and can be specified
create_studio	The Amazon EMR Studio APIs are in preview release for Amazon EMR and are su
create_studio_session_mapping	The Amazon EMR Studio APIs are in preview release for Amazon EMR and are su
delete_security_configuration	Deletes a security configuration
delete_studio	The Amazon EMR Studio APIs are in preview release for Amazon EMR and are su
delete_studio_session_mapping	The Amazon EMR Studio APIs are in preview release for Amazon EMR and are su
describe_cluster	Provides cluster-level details including status, hardware and software configuration,
describe_job_flows	This API is no longer supported and will eventually be removed
describe_notebook_execution	Provides details of a notebook execution
describe_security_configuration	Provides the details of a security configuration by returning the configuration JSON
describe_step	Provides more detail about the cluster step
describe_studio	The Amazon EMR Studio APIs are in preview release for Amazon EMR and are su
get_block_public_access_configuration	Returns the Amazon EMR block public access configuration for your AWS account
get_managed_scaling_policy	Fetches the attached managed scaling policy for an Amazon EMR cluster
get_studio_session_mapping	The Amazon EMR Studio APIs are in preview release for Amazon EMR and are su

list_bootstrap_actions	Provides information about the bootstrap actions associated with a cluster
list_clusters	Provides the status of all clusters visible to this AWS account
list_instance_fleets	Lists all available details about the instance fleets in a cluster
list_instance_groups	Provides all available details about the instance groups in a cluster
list_instances	Provides information for all active EC2 instances and EC2 instances terminated in t
list_notebook_executions	Provides summaries of all notebook executions
list_security_configurations	Lists all the security configurations visible to this account, providing their creation d
list_steps	Provides a list of steps for the cluster in reverse order unless you specify stepIds wit
list_studios	The Amazon EMR Studio APIs are in preview release for Amazon EMR and are su
list_studio_session_mappings	The Amazon EMR Studio APIs are in preview release for Amazon EMR and are su
modify_cluster	Modifies the number of steps that can be executed concurrently for the cluster speci
modify_instance_fleet	Modifies the target On-Demand and target Spot capacities for the instance fleet with
modify_instance_groups	ModifyInstanceGroups modifies the number of nodes and configuration settings of a
put_auto_scaling_policy	Creates or updates an automatic scaling policy for a core instance group or task inst
put_block_public_access_configuration	Creates or updates an Amazon EMR block public access configuration for your AW
put_managed_scaling_policy	Creates or updates a managed scaling policy for an Amazon EMR cluster
remove_auto_scaling_policy	Removes an automatic scaling policy from a specified instance group within an EM
remove_managed_scaling_policy	Removes a managed scaling policy from a specified EMR cluster
remove_tags	Removes tags from an Amazon EMR resource
run_job_flow	RunJobFlow creates and starts running a new cluster (job flow)
set_termination_protection	SetTerminationProtection locks a cluster (job flow) so the EC2 instances in the clus
set_visible_to_all_users	Sets the Cluster$VisibleToAllUsers value, which determines whether the cluster is v
start_notebook_execution	Starts a notebook execution
stop_notebook_execution	Stops a notebook execution
terminate_job_flows	TerminateJobFlows shuts a list of clusters (job flows) down
update_studio_session_mapping	The Amazon EMR Studio APIs are in preview release for Amazon EMR and are su

Examples

```
## Not run:
svc <- emr()
svc$add_instance_fleet(
  Foo = 123
)

## End(Not run)
```

`firehose` *Amazon Kinesis Firehose*

Description

Amazon Kinesis Data Firehose API Reference

Amazon Kinesis Data Firehose is a fully managed service that delivers real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3), Amazon Elasticsearch Service (Amazon ES), Amazon Redshift, and Splunk.

Usage

```
firehose(config = list())
```

Arguments

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

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 <- firehose(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

create_delivery_stream	Creates a Kinesis Data Firehose delivery stream
delete_delivery_stream	Deletes a delivery stream and its data
describe_delivery_stream	Describes the specified delivery stream and its status
list_delivery_streams	Lists your delivery streams in alphabetical order of their names
list_tags_for_delivery_stream	Lists the tags for the specified delivery stream
put_record	Writes a single data record into an Amazon Kinesis Data Firehose delivery stream
put_record_batch	Writes multiple data records into a delivery stream in a single call, which can achieve high
start_delivery_stream_encryption	Enables server-side encryption (SSE) for the delivery stream
stop_delivery_stream_encryption	Disables server-side encryption (SSE) for the delivery stream
tag_delivery_stream	Adds or updates tags for the specified delivery stream
untag_delivery_stream	Removes tags from the specified delivery stream
update_destination	Updates the specified destination of the specified delivery stream

Examples

```
## Not run:
svc <- firehose()
svc$create_delivery_stream(
  Foo = 123
)

## End(Not run)
```

---

glue *AWS Glue*

Description

Defines the public endpoint for the AWS Glue service.

Usage

```
glue(config = list())
```

Arguments

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

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 <- glue(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

batch_create_partition	Creates one or more partitions in a batch operation
batch_delete_connection	Deletes a list of connection definitions from the Data Catalog
batch_delete_partition	Deletes one or more partitions in a batch operation
batch_delete_table	Deletes multiple tables at once
batch_delete_table_version	Deletes a specified batch of versions of a table
batch_get_crawlers	Returns a list of resource metadata for a given list of crawler names
batch_get_dev_endpoints	Returns a list of resource metadata for a given list of development endpoint nam
batch_get_jobs	Returns a list of resource metadata for a given list of job names
batch_get_partition	Retrieves partitions in a batch request
batch_get_triggers	Returns a list of resource metadata for a given list of trigger names
batch_get_workflows	Returns a list of resource metadata for a given list of workflow names
batch_stop_job_run	Stops one or more job runs for a specified job definition
batch_update_partition	Updates one or more partitions in a batch operation
cancel_ml_task_run	Cancels (stops) a task run
check_schema_version_validity	Validates the supplied schema
create_classifier	Creates a classifier in the user's account
create_connection	Creates a connection definition in the Data Catalog
create_crawler	Creates a new crawler with specified targets, role, configuration, and optional sc
create_database	Creates a new database in a Data Catalog
create_dev_endpoint	Creates a new development endpoint
create_job	Creates a new job definition
create_ml_transform	Creates an AWS Glue machine learning transform
create_partition	Creates a new partition
create_partition_index	Creates a specified partition index in an existing table
create_registry	Creates a new registry which may be used to hold a collection of schemas
create_schema	Creates a new schema set and registers the schema definition
create_script	Transforms a directed acyclic graph (DAG) into code
create_security_configuration	Creates a new security configuration
create_table	Creates a new table definition in the Data Catalog
create_trigger	Creates a new trigger
create_user_defined_function	Creates a new function definition in the Data Catalog
create_workflow	Creates a new workflow
delete_classifier	Removes a classifier from the Data Catalog
delete_column_statistics_for_partition	Delete the partition column statistics of a column
delete_column_statistics_for_table	Retrieves table statistics of columns
delete_connection	Deletes a connection from the Data Catalog
delete_crawler	Removes a specified crawler from the AWS Glue Data Catalog, unless the crawl
delete_database	Removes a specified database from a Data Catalog
delete_dev_endpoint	Deletes a specified development endpoint
delete_job	Deletes a specified job definition
delete_ml_transform	Deletes an AWS Glue machine learning transform
delete_partition	Deletes a specified partition
delete_partition_index	Deletes a specified partition index from an existing table
delete_registry	Delete the entire registry including schema and all of its versions
delete_resource_policy	Deletes a specified policy
delete_schema	Deletes the entire schema set, including the schema set and all of its versions
delete_schema_versions	Remove versions from the specified schema
delete_security_configuration	Deletes a specified security configuration

delete_table	Removes a table definition from the Data Catalog
delete_table_version	Deletes a specified version of a table
delete_trigger	Deletes a specified trigger
delete_user_defined_function	Deletes an existing function definition from the Data Catalog
delete_workflow	Deletes a workflow
get_catalog_import_status	Retrieves the status of a migration operation
get_classifier	Retrieve a classifier by name
get_classifiers	Lists all classifier objects in the Data Catalog
get_column_statistics_for_partition	Retrieves partition statistics of columns
get_column_statistics_for_table	Retrieves table statistics of columns
get_connection	Retrieves a connection definition from the Data Catalog
get_connections	Retrieves a list of connection definitions from the Data Catalog
get_crawler	Retrieves metadata for a specified crawler
get_crawler_metrics	Retrieves metrics about specified crawlers
get_crawlers	Retrieves metadata for all crawlers defined in the customer account
get_database	Retrieves the definition of a specified database
get_databases	Retrieves all databases defined in a given Data Catalog
get_data_catalog_encryption_settings	Retrieves the security configuration for a specified catalog
get_dataflow_graph	Transforms a Python script into a directed acyclic graph (DAG)
get_dev_endpoint	Retrieves information about a specified development endpoint
get_dev_endpoints	Retrieves all the development endpoints in this AWS account
get_job	Retrieves an existing job definition
get_job_bookmark	Returns information on a job bookmark entry
get_job_run	Retrieves the metadata for a given job run
get_job_runs	Retrieves metadata for all runs of a given job definition
get_jobs	Retrieves all current job definitions
get_mapping	Creates mappings
get_ml_task_run	Gets details for a specific task run on a machine learning transform
get_ml_task_runs	Gets a list of runs for a machine learning transform
get_ml_transform	Gets an AWS Glue machine learning transform artifact and all its corresponding
get_ml_transforms	Gets a sortable, filterable list of existing AWS Glue machine learning transforms
get_partition	Retrieves information about a specified partition
get_partition_indexes	Retrieves the partition indexes associated with a table
get_partitions	Retrieves information about the partitions in a table
get_plan	Gets code to perform a specified mapping
get_registry	Describes the specified registry in detail
get_resource_policies	Retrieves the security configurations for the resource policies set on individual r
get_resource_policy	Retrieves a specified resource policy
get_schema	Describes the specified schema in detail
get_schema_by_definition	Retrieves a schema by the SchemaDefinition
get_schema_version	Get the specified schema by its unique ID assigned when a version of the schem
get_schema_versions_diff	Fetches the schema version difference in the specified difference type between t
get_security_configuration	Retrieves a specified security configuration
get_security_configurations	Retrieves a list of all security configurations
get_table	Retrieves the Table definition in a Data Catalog for a specified table
get_tables	Retrieves the definitions of some or all of the tables in a given Database
get_table_version	Retrieves a specified version of a table
get_table_versions	Retrieves a list of strings that identify available versions of a specified table

get_tags	Retrieves a list of tags associated with a resource
get_trigger	Retrieves the definition of a trigger
get_triggers	Gets all the triggers associated with a job
get_user_defined_function	Retrieves a specified function definition from the Data Catalog
get_user_defined_functions	Retrieves multiple function definitions from the Data Catalog
get_workflow	Retrieves resource metadata for a workflow
get_workflow_run	Retrieves the metadata for a given workflow run
get_workflow_run_properties	Retrieves the workflow run properties which were set during the run
get_workflow_runs	Retrieves metadata for all runs of a given workflow
import_catalog_to_glue	Imports an existing Amazon Athena Data Catalog to AWS Glue
list_crawlers	Retrieves the names of all crawler resources in this AWS account, or the resourc
list_dev_endpoints	Retrieves the names of all DevEndpoint resources in this AWS account, or the re
list_jobs	Retrieves the names of all job resources in this AWS account, or the resources w
list_ml_transforms	Retrieves a sortable, filterable list of existing AWS Glue machine learning transf
list_registries	Returns a list of registries that you have created, with minimal registry informati
list_schemas	Returns a list of schemas with minimal details
list_schema_versions	Returns a list of schema versions that you have created, with minimal informatio
list_triggers	Retrieves the names of all trigger resources in this AWS account, or the resource
list_workflows	Lists names of workflows created in the account
put_data_catalog_encryption_settings	Sets the security configuration for a specified catalog
put_resource_policy	Sets the Data Catalog resource policy for access control
put_schema_version_metadata	Puts the metadata key value pair for a specified schema version ID
put_workflow_run_properties	Puts the specified workflow run properties for the given workflow run
query_schema_version_metadata	Queries for the schema version metadata information
register_schema_version	Adds a new version to the existing schema
remove_schema_version_metadata	Removes a key value pair from the schema version metadata for the specified sc
reset_job_bookmark	Resets a bookmark entry
resume_workflow_run	Restarts selected nodes of a previous partially completed workflow run and resu
search_tables	Searches a set of tables based on properties in the table metadata as well as on th
start_crawler	Starts a crawl using the specified crawler, regardless of what is scheduled
start_crawler_schedule	Changes the schedule state of the specified crawler to SCHEDULED, unless the
start_export_labels_task_run	Begins an asynchronous task to export all labeled data for a particular transform
start_import_labels_task_run	Enables you to provide additional labels (examples of truth) to be used to teach t
start_job_run	Starts a job run using a job definition
start_ml_evaluation_task_run	Starts a task to estimate the quality of the transform
start_ml_labeling_set_generation_task_run	Starts the active learning workflow for your machine learning transform to impro
start_trigger	Starts an existing trigger
start_workflow_run	Starts a new run of the specified workflow
stop_crawler	If the specified crawler is running, stops the crawl
stop_crawler_schedule	Sets the schedule state of the specified crawler to NOT_SCHEDULED, but does
stop_trigger	Stops a specified trigger
stop_workflow_run	Stops the execution of the specified workflow run
tag_resource	Adds tags to a resource
untag_resource	Removes tags from a resource
update_classifier	Modifies an existing classifier (a GrokClassifier, an XMLClassifier, a JsonClassi
update_column_statistics_for_partition	Creates or updates partition statistics of columns
update_column_statistics_for_table	Creates or updates table statistics of columns
update_connection	Updates a connection definition in the Data Catalog

update_crawler	Updates a crawler
update_crawler_schedule	Updates the schedule of a crawler using a cron expression
update_database	Updates an existing database definition in a Data Catalog
update_dev_endpoint	Updates a specified development endpoint
update_job	Updates an existing job definition
update_ml_transform	Updates an existing machine learning transform
update_partition	Updates a partition
update_registry	Updates an existing registry which is used to hold a collection of schemas
update_schema	Updates the description, compatibility setting, or version checkpoint for a schem
update_table	Updates a metadata table in the Data Catalog
update_trigger	Updates a trigger definition
update_user_defined_function	Updates an existing function definition in the Data Catalog
update_workflow	Updates an existing workflow

Examples

```
## Not run:
svc <- glue()
svc$batch_create_partition(
  Foo = 123
)

## End(Not run)
```

---

kafka *Managed Streaming for Kafka*

Description

The operations for managing an Amazon MSK cluster.

Usage

```
kafka(config = list())
```

Arguments

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

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 <- kafka(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

batch_associate_scram_secret	Associates one or more Scram Secrets with an Amazon MSK cluster
batch_disassociate_scram_secret	Disassociates one or more Scram Secrets from an Amazon MSK cluster
create_cluster	Creates a new MSK cluster
create_configuration	Creates a new MSK configuration
delete_cluster	Deletes the MSK cluster specified by the Amazon Resource Name (ARN) in the request
delete_configuration	Deletes an MSK Configuration
describe_cluster	Returns a description of the MSK cluster whose Amazon Resource Name (ARN) is specifi
describe_cluster_operation	Returns a description of the cluster operation specified by the ARN
describe_configuration	Returns a description of this MSK configuration
describe_configuration_revision	Returns a description of this revision of the configuration
get_bootstrap_brokers	A list of brokers that a client application can use to bootstrap
get_compatible_kafka_versions	Gets the Apache Kafka versions to which you can update the MSK cluster
list_cluster_operations	Returns a list of all the operations that have been performed on the specified MSK cluster
list_clusters	Returns a list of all the MSK clusters in the current Region
list_configuration_revisions	Returns a list of all the MSK configurations in this Region
list_configurations	Returns a list of all the MSK configurations in this Region
list_kafka_versions	Returns a list of Kafka versions
list_nodes	Returns a list of the broker nodes in the cluster
list_scram_secrets	Returns a list of the Scram Secrets associated with an Amazon MSK cluster
list_tags_for_resource	Returns a list of the tags associated with the specified resource
reboot_broker	Reboots brokers
tag_resource	Adds tags to the specified MSK resource
untag_resource	Removes the tags associated with the keys that are provided in the query
update_broker_count	Updates the number of broker nodes in the cluster
update_broker_storage	Updates the EBS storage associated with MSK brokers
update_cluster_configuration	Updates the cluster with the configuration that is specified in the request body
update_cluster_kafka_version	Updates the Apache Kafka version for the cluster
update_configuration	Updates an MSK configuration
update_monitoring	Updates the monitoring settings for the cluster

Examples

```
## Not run:
svc <- kafka()
svc$batch_associate_scram_secret(
  Foo = 123
)

## End(Not run)
```

`kinesis` *Amazon Kinesis*

Description

Amazon Kinesis Data Streams Service API Reference

Amazon Kinesis Data Streams is a managed service that scales elastically for real-time processing of streaming big data.

Usage

```
kinesis(config = list())
```

Arguments

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

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 <- kinesis(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
```

```
    endpoint = "string",
    region = "string"
  )
)
```

Operations

add_tags_to_stream	Adds or updates tags for the specified Kinesis data stream
create_stream	Creates a Kinesis data stream
decrease_stream_retention_period	Decreases the Kinesis data stream's retention period, which is the length of time data rec
delete_stream	Deletes a Kinesis data stream and all its shards and data
deregister_stream_consumer	To deregister a consumer, provide its ARN
describe_limits	Describes the shard limits and usage for the account
describe_stream	Describes the specified Kinesis data stream
describe_stream_consumer	To get the description of a registered consumer, provide the ARN of the consumer
describe_stream_summary	Provides a summarized description of the specified Kinesis data stream without the shard
disable_enhanced_monitoring	Disables enhanced monitoring
enable_enhanced_monitoring	Enables enhanced Kinesis data stream monitoring for shard-level metrics
get_records	Gets data records from a Kinesis data stream's shard
get_shard_iterator	Gets an Amazon Kinesis shard iterator
increase_stream_retention_period	Increases the Kinesis data stream's retention period, which is the length of time data reco
list_shards	Lists the shards in a stream and provides information about each shard
list_stream_consumers	Lists the consumers registered to receive data from a stream using enhanced fan-out, and
list_streams	Lists your Kinesis data streams
list_tags_for_stream	Lists the tags for the specified Kinesis data stream
merge_shards	Merges two adjacent shards in a Kinesis data stream and combines them into a single sha
put_record	Writes a single data record into an Amazon Kinesis data stream
put_records	Writes multiple data records into a Kinesis data stream in a single call (also referred to as
register_stream_consumer	Registers a consumer with a Kinesis data stream
remove_tags_from_stream	Removes tags from the specified Kinesis data stream
split_shard	Splits a shard into two new shards in the Kinesis data stream, to increase the stream's cap
start_stream_encryption	Enables or updates server-side encryption using an AWS KMS key for a specified stream
stop_stream_encryption	Disables server-side encryption for a specified stream
update_shard_count	Updates the shard count of the specified stream to the specified number of shards

Examples

```
## Not run:
svc <- kinesis()
svc$add_tags_to_stream(
  Foo = 123
)

## End(Not run)
```

`kinesisanalytics` *Amazon Kinesis Analytics*

Description

Overview

This documentation is for version 1 of the Amazon Kinesis Data Analytics API, which only supports SQL applications. Version 2 of the API supports SQL and Java applications. For more information about version 2, see Amazon Kinesis Data Analytics API V2 Documentation.

This is the *Amazon Kinesis Analytics v1 API Reference*. The Amazon Kinesis Analytics Developer Guide provides additional information.

Usage

```
kinesisanalytics(config = list())
```

Arguments

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

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 <- kinesisanalytics(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

add_application_cloud_watch_logging_option	This documentation is for version 1 of the Amazon Kinesis Data Analyt
add_application_input	This documentation is for version 1 of the Amazon Kinesis Data Analyt
add_application_input_processing_configuration	This documentation is for version 1 of the Amazon Kinesis Data Analyt
add_application_output	This documentation is for version 1 of the Amazon Kinesis Data Analyt
add_application_reference_data_source	This documentation is for version 1 of the Amazon Kinesis Data Analyt
create_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt
delete_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt
delete_application_cloud_watch_logging_option	This documentation is for version 1 of the Amazon Kinesis Data Analyt
delete_application_input_processing_configuration	This documentation is for version 1 of the Amazon Kinesis Data Analyt
delete_application_output	This documentation is for version 1 of the Amazon Kinesis Data Analyt
delete_application_reference_data_source	This documentation is for version 1 of the Amazon Kinesis Data Analyt
describe_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt
discover_input_schema	This documentation is for version 1 of the Amazon Kinesis Data Analyt
list_applications	This documentation is for version 1 of the Amazon Kinesis Data Analyt
list_tags_for_resource	Retrieves the list of key-value tags assigned to the application
start_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt
stop_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt
tag_resource	Adds one or more key-value tags to a Kinesis Analytics application
untag_resource	Removes one or more tags from a Kinesis Analytics application
update_application	This documentation is for version 1 of the Amazon Kinesis Data Analyt

Examples

```
## Not run:
svc <- kinesisanalytics()
svc$add_application_cloud_watch_logging_option(
  Foo = 123
)

## End(Not run)
```

`kinesisanalyticsv2` *Amazon Kinesis Analytics*

Description

Amazon Kinesis Data Analytics is a fully managed service that you can use to process and analyze streaming data using Java, SQL, or Scala. The service enables you to quickly author and run Java, SQL, or Scala code against streaming sources to perform time series analytics, feed real-time dashboards, and create real-time metrics.

Usage

```
kinesisanalyticsv2(config = list())
```

Arguments

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

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 <- kinesisanalyticsv2(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

add_application_cloud_watch_logging_option	Adds an Amazon CloudWatch log stream to monitor application configu
add_application_input	Adds a streaming source to your SQL-based Kinesis Data Analytics app
add_application_input_processing_configuration	Adds an InputProcessingConfiguration to a SQL-based Kinesis Data An
add_application_output	Adds an external destination to your SQL-based Kinesis Data Analytics
add_application_reference_data_source	Adds a reference data source to an existing SQL-based Kinesis Data An
add_application_vpc_configuration	Adds a Virtual Private Cloud (VPC) configuration to the application
create_application	Creates a Kinesis Data Analytics application
create_application_presigned_url	Creates and returns a URL that you can use to connect to an application'
create_application_snapshot	Creates a snapshot of the application's state data
delete_application	Deletes the specified application
delete_application_cloud_watch_logging_option	Deletes an Amazon CloudWatch log stream from an Kinesis Data Analy
delete_application_input_processing_configuration	Deletes an InputProcessingConfiguration from an input
delete_application_output	Deletes the output destination configuration from your SQL-based Kines
delete_application_reference_data_source	Deletes a reference data source configuration from the specified SQL-ba
delete_application_snapshot	Deletes a snapshot of application state
delete_application_vpc_configuration	Removes a VPC configuration from a Kinesis Data Analytics applicatio

describe_application	Returns information about a specific Kinesis Data Analytics application
describe_application_snapshot	Returns information about a snapshot of application state data
discover_input_schema	Infers a schema for a SQL-based Kinesis Data Analytics application by
list_applications	Returns a list of Kinesis Data Analytics applications in your account
list_application_snapshots	Lists information about the current application snapshots
list_tags_for_resource	Retrieves the list of key-value tags assigned to the application
start_application	Starts the specified Kinesis Data Analytics application
stop_application	Stops the application from processing data
tag_resource	Adds one or more key-value tags to a Kinesis Data Analytics applicatior
untag_resource	Removes one or more tags from a Kinesis Data Analytics application
update_application	Updates an existing Kinesis Data Analytics application

Examples

```
## Not run:
svc <- kinesisanalyticsv2()
svc$add_application_cloud_watch_logging_option(
  Foo = 123
)

## End(Not run)
```

`mturk` *Amazon Mechanical Turk*

Description

Amazon Mechanical Turk API Reference

Usage

```
mturk(config = list())
```

Arguments

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

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 <- mturk(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

accept_qualification_request	The AcceptQualificationRequest operation approves a Worker's request for a Quali
approve_assignment	The ApproveAssignment operation approves the results of a completed assignment
associate_qualification_with_worker	The AssociateQualificationWithWorker operation gives a Worker a Qualification
create_additional_assignments_for_hit	The CreateAdditionalAssignmentsForHIT operation increases the maximum numbe
create_hit	The CreateHIT operation creates a new Human Intelligence Task (HIT)
create_hit_type	The CreateHITType operation creates a new HIT type
create_hit_with_hit_type	The CreateHITWithHITType operation creates a new Human Intelligence Task (HI
create_qualification_type	The CreateQualificationType operation creates a new Qualification type, which is re
create_worker_block	The CreateWorkerBlock operation allows you to prevent a Worker from working or
delete_hit	The DeleteHIT operation is used to delete HIT that is no longer needed
delete_qualification_type	The DeleteQualificationType deletes a Qualification type and deletes any HIT types
delete_worker_block	The DeleteWorkerBlock operation allows you to reinstate a blocked Worker to worl
disassociate_qualification_from_worker	The DisassociateQualificationFromWorker revokes a previously granted Qualificati
get_account_balance	The GetAccountBalance operation retrieves the amount of money in your Amazon
get_assignment	The GetAssignment operation retrieves the details of the specified Assignment
get_file_upload_url	The GetFileUploadURL operation generates and returns a temporary URL
get_hit	The GetHIT operation retrieves the details of the specified HIT
get_qualification_score	The GetQualificationScore operation returns the value of a Worker's Qualification f
get_qualification_type	The GetQualificationTypeoperation retrieves information about a Qualification type
list_assignments_for_hit	The ListAssignmentsForHIT operation retrieves completed assignments for a HIT
list_bonus_payments	The ListBonusPayments operation retrieves the amounts of bonuses you have paid f
list_hi_ts	The ListHITs operation returns all of a Requester's HITs
list_hi_ts_for_qualification_type	The ListHITsForQualificationType operation returns the HITs that use the given Qu
list_qualification_requests	The ListQualificationRequests operation retrieves requests for Qualifications of a pa
list_qualification_types	The ListQualificationTypes operation returns a list of Qualification types, filtered by
list_reviewable_hi_ts	The ListReviewableHITs operation retrieves the HITs with Status equal to Reviewa
list_review_policy_results_for_hit	The ListReviewPolicyResultsForHIT operation retrieves the computed results and t
list_worker_blocks	The ListWorkersBlocks operation retrieves a list of Workers who are blocked from
list_workers_with_qualification_type	The ListWorkersWithQualificationType operation returns all of the Workers that ha
notify_workers	The NotifyWorkers operation sends an email to one or more Workers that you speci

reject_assignment	The RejectAssignment operation rejects the results of a completed assignment
reject_qualification_request	The RejectQualificationRequest operation rejects a user's request for a Qualificatio
send_bonus	The SendBonus operation issues a payment of money from your account to a Work
send_test_event_notification	The SendTestEventNotification operation causes Amazon Mechanical Turk to send
update_expiration_for_hit	The UpdateExpirationForHIT operation allows you update the expiration time of a
update_hit_review_status	The UpdateHITReviewStatus operation updates the status of a HIT
update_hit_type_of_hit	The UpdateHITTypeOfHIT operation allows you to change the HITType properties
update_notification_settings	The UpdateNotificationSettings operation creates, updates, disables or re-enables n
update_qualification_type	The UpdateQualificationType operation modifies the attributes of an existing Quali

Examples

```
## Not run:
svc <- mturk()
svc$accept_qualification_request(
  Foo = 123
)

## End(Not run)
```

`quicksight` *Amazon QuickSight*

Description

Amazon QuickSight API Reference

Amazon QuickSight is a fully managed, serverless business intelligence service for the AWS Cloud that makes it easy to extend data and insights to every user in your organization. This API reference contains documentation for a programming interface that you can use to manage Amazon QuickSight.

Usage

```
quicksight(config = list())
```

Arguments

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

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 <- quicksight(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

cancel_ingestion	Cancels an ongoing ingestion of data into SPICE
create_account_customization	Creates Amazon QuickSight customizations the current AWS Region
create_analysis	Creates an analysis in Amazon QuickSight
create_dashboard	Creates a dashboard from a template
create_data_set	Creates a dataset
create_data_source	Creates a data source
create_group	Creates an Amazon QuickSight group
create_group_membership	Adds an Amazon QuickSight user to an Amazon QuickSight group
create_iam_policy_assignment	Creates an assignment with one specified IAM policy, identified by its Amazon Reso
create_ingestion	Creates and starts a new SPICE ingestion on a dataset
create_namespace	(Enterprise edition only) Creates a new namespace for you to use with Amazon Quic
create_template	Creates a template from an existing QuickSight analysis or template
create_template_alias	Creates a template alias for a template
create_theme	Creates a theme
create_theme_alias	Creates a theme alias for a theme
delete_account_customization	Deletes all Amazon QuickSight customizations in this AWS Region for the specified
delete_analysis	Deletes an analysis from Amazon QuickSight
delete_dashboard	Deletes a dashboard
delete_data_set	Deletes a dataset
delete_data_source	Deletes the data source permanently
delete_group	Removes a user group from Amazon QuickSight
delete_group_membership	Removes a user from a group so that the user is no longer a member of the group
delete_iam_policy_assignment	Deletes an existing IAM policy assignment
delete_namespace	Deletes a namespace and the users and groups that are associated with the namespace
delete_template	Deletes a template
delete_template_alias	Deletes the item that the specified template alias points to
delete_theme	Deletes a theme
delete_theme_alias	Deletes the version of the theme that the specified theme alias points to
delete_user	Deletes the Amazon QuickSight user that is associated with the identity of the AWS
delete_user_by_principal_id	Deletes a user identified by its principal ID

describe_account_customization	Describes the customizations associated with the provided AWS account and Amazo
describe_account_settings	Describes the settings that were used when your QuickSight subscription was first cr
describe_analysis	Provides a summary of the metadata for an analysis
describe_analysis_permissions	Provides the read and write permissions for an analysis
describe_dashboard	Provides a summary for a dashboard
describe_dashboard_permissions	Describes read and write permissions for a dashboard
describe_data_set	Describes a dataset
describe_data_set_permissions	Describes the permissions on a dataset
describe_data_source	Describes a data source
describe_data_source_permissions	Describes the resource permissions for a data source
describe_group	Returns an Amazon QuickSight group's description and Amazon Resource Name (A
describe_iam_policy_assignment	Describes an existing IAM policy assignment, as specified by the assignment name
describe_ingestion	Describes a SPICE ingestion
describe_namespace	Describes the current namespace
describe_template	Describes a template's metadata
describe_template_alias	Describes the template alias for a template
describe_template_permissions	Describes read and write permissions on a template
describe_theme	Describes a theme
describe_theme_alias	Describes the alias for a theme
describe_theme_permissions	Describes the read and write permissions for a theme
describe_user	Returns information about a user, given the user name
get_dashboard_embed_url	Generates a session URL and authorization code that you can use to embed an Amaz
get_session_embed_url	Generates a session URL and authorization code that you can use to embed the Amaz
list_analyses	Lists Amazon QuickSight analyses that exist in the specified AWS account
list_dashboards	Lists dashboards in an AWS account
list_dashboard_versions	Lists all the versions of the dashboards in the QuickSight subscription
list_data_sets	Lists all of the datasets belonging to the current AWS account in an AWS Region
list_data_sources	Lists data sources in current AWS Region that belong to this AWS account
list_group_memberships	Lists member users in a group
list_groups	Lists all user groups in Amazon QuickSight
list_iam_policy_assignments	Lists IAM policy assignments in the current Amazon QuickSight account
list_iam_policy_assignments_for_user	Lists all the IAM policy assignments, including the Amazon Resource Names (ARN
list_ingestions	Lists the history of SPICE ingestions for a dataset
list_namespaces	Lists the namespaces for the specified AWS account
list_tags_for_resource	Lists the tags assigned to a resource
list_template_aliases	Lists all the aliases of a template
list_templates	Lists all the templates in the current Amazon QuickSight account
list_template_versions	Lists all the versions of the templates in the current Amazon QuickSight account
list_theme_aliases	Lists all the aliases of a theme
list_themes	Lists all the themes in the current AWS account
list_theme_versions	Lists all the versions of the themes in the current AWS account
list_user_groups	Lists the Amazon QuickSight groups that an Amazon QuickSight user is a member o
list_users	Returns a list of all of the Amazon QuickSight users belonging to this account
register_user	Creates an Amazon QuickSight user, whose identity is associated with the AWS Iden
restore_analysis	Restores an analysis
search_analyses	Searches for analyses that belong to the user specified in the filter
search_dashboards	Searches for dashboards that belong to a user
tag_resource	Assigns one or more tags (key-value pairs) to the specified QuickSight resource

untag_resource	Removes a tag or tags from a resource
update_account_customization	Updates Amazon QuickSight customizations the current AWS Region
update_account_settings	Updates the Amazon QuickSight settings in your AWS account
update_analysis	Updates an analysis in Amazon QuickSight
update_analysis_permissions	Updates the read and write permissions for an analysis
update_dashboard	Updates a dashboard in an AWS account
update_dashboard_permissions	Updates read and write permissions on a dashboard
update_dashboard_published_version	Updates the published version of a dashboard
update_data_set	Updates a dataset
update_data_set_permissions	Updates the permissions on a dataset
update_data_source	Updates a data source
update_data_source_permissions	Updates the permissions to a data source
update_group	Changes a group description
update_iam_policy_assignment	Updates an existing IAM policy assignment
update_template	Updates a template from an existing Amazon QuickSight analysis or another templat
update_template_alias	Updates the template alias of a template
update_template_permissions	Updates the resource permissions for a template
update_theme	Updates a theme
update_theme_alias	Updates an alias of a theme
update_theme_permissions	Updates the resource permissions for a theme
update_user	Updates an Amazon QuickSight user

Examples

```
## Not run:
svc <- quicksight()
svc$cancel_ingestion(
  Foo = 123
)

## End(Not run)
```

Index