

Package: paws.mobile (via r-universe)

September 4, 2024

Title 'Amazon Web Services' Mobile Services

Version 0.7.0

Description Interface to 'Amazon Web Services' mobile services, including the 'Amplify' library for mobile applications, 'AppSync' back-end for mobile applications, 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

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

RoxygenNote 7.1.1

Collate 'amplify_service.R' 'amplify_interfaces.R'
'amplify_operations.R' 'appsync_service.R'
'appsync_interfaces.R' 'appsync_operations.R'
'devicefarm_service.R' 'devicefarm_interfaces.R'
'devicefarm_operations.R' 'mobile_service.R'
'mobile_interfaces.R' 'mobile_operations.R'
'mobileanalytics_service.R' 'mobileanalytics_interfaces.R'
'mobileanalytics_operations.R'

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

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

RemoteRef HEAD

RemoteSha 8ab20b498262e0b343c0153b4f244483aff4987f

Contents

amplify	2
appsync	4
devicefarm	6
mobile	9
mobileanalytics	10
Index	12

amplify	<i>AWS Amplify</i>
---------	--------------------

Description

Amplify enables developers to develop and deploy cloud-powered mobile and web apps. The Amplify Console provides a continuous delivery and hosting service for web applications. For more information, see the [Amplify Console User Guide](#). The Amplify Framework is a comprehensive set of SDKs, libraries, tools, and documentation for client app development. For more information, see the [Amplify Framework](#).

Usage

```
amplify(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 <- amplify(
  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_app	Creates a new Amplify app
create_backend_environment	Creates a new backend environment for an Amplify app
create_branch	Creates a new branch for an Amplify app
create_deployment	Creates a deployment for a manually deployed Amplify app
create_domain_association	Creates a new domain association for an Amplify app
create_webhook	Creates a new webhook on an Amplify app
delete_app	Deletes an existing Amplify app specified by an app ID
delete_backend_environment	Deletes a backend environment for an Amplify app
delete_branch	Deletes a branch for an Amplify app
delete_domain_association	Deletes a domain association for an Amplify app
delete_job	Deletes a job for a branch of an Amplify app
delete_webhook	Deletes a webhook
generate_access_logs	Returns the website access logs for a specific time range using a presigned URL
get_app	Returns an existing Amplify app by appID
get_artifact_url	Returns the artifact info that corresponds to an artifact id
get_backend_environment	Returns a backend environment for an Amplify app
get_branch	Returns a branch for an Amplify app
get_domain_association	Returns the domain information for an Amplify app
get_job	Returns a job for a branch of an Amplify app
get_webhook	Returns the webhook information that corresponds to a specified webhook ID
list_apps	Returns a list of the existing Amplify apps
list_artifacts	Returns a list of artifacts for a specified app, branch, and job
list_backend_environments	Lists the backend environments for an Amplify app
list_branches	Lists the branches of an Amplify app
list_domain_associations	Returns the domain associations for an Amplify app
list_jobs	Lists the jobs for a branch of an Amplify app
list_tags_for_resource	Returns a list of tags for a specified Amazon Resource Name (ARN)
list_webhooks	Returns a list of webhooks for an Amplify app
start_deployment	Starts a deployment for a manually deployed app
start_job	Starts a new job for a branch of an Amplify app
stop_job	Stops a job that is in progress for a branch of an Amplify app
tag_resource	Tags the resource with a tag key and value
untag_resource	Untags a resource with a specified Amazon Resource Name (ARN)
update_app	Updates an existing Amplify app
update_branch	Updates a branch for an Amplify app
update_domain_association	Creates a new domain association for an Amplify app
update_webhook	Updates a webhook

Examples

```
## Not run:
```

```
svc <- amplify()
svc$create_app(
  Foo = 123
)

## End(Not run)
```

appsync

AWS AppSync

Description

AWS AppSync provides API actions for creating and interacting with data sources using GraphQL from your application.

Usage

```
appsync(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 <- appsync(
  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_api_cache	Creates a cache for the GraphQL API
create_api_key	Creates a unique key that you can distribute to clients who are executing your API
create_data_source	Creates a DataSource object
create_function	Creates a Function object
create_graphql_api	Creates a GraphqlApi object
create_resolver	Creates a Resolver object
create_type	Creates a Type object
delete_api_cache	Deletes an ApiCache object
delete_api_key	Deletes an API key
delete_data_source	Deletes a DataSource object
delete_function	Deletes a Function
delete_graphql_api	Deletes a GraphqlApi object
delete_resolver	Deletes a Resolver object
delete_type	Deletes a Type object
flush_api_cache	Flushes an ApiCache object
get_api_cache	Retrieves an ApiCache object
get_data_source	Retrieves a DataSource object
get_function	Get a Function
get_graphql_api	Retrieves a GraphqlApi object
get_introspection_schema	Retrieves the introspection schema for a GraphQL API
get_resolver	Retrieves a Resolver object
get_schema_creation_status	Retrieves the current status of a schema creation operation
get_type	Retrieves a Type object
list_api_keys	Lists the API keys for a given API
list_data_sources	Lists the data sources for a given API
list_functions	List multiple functions
list_graphql_apis	Lists your GraphQL APIs
list_resolvers	Lists the resolvers for a given API and type
list_resolvers_by_function	List the resolvers that are associated with a specific function
list_tags_for_resource	Lists the tags for a resource
list_types	Lists the types for a given API
start_schema_creation	Adds a new schema to your GraphQL API
tag_resource	Tags a resource with user-supplied tags
untag_resource	Untags a resource
update_api_cache	Updates the cache for the GraphQL API
update_api_key	Updates an API key
update_data_source	Updates a DataSource object
update_function	Updates a Function object
update_graphql_api	Updates a GraphqlApi object
update_resolver	Updates a Resolver object
update_type	Updates a Type object

Examples

Not run:

```
svc <- appsync()
svc$create_api_cache(
  Foo = 123
)

## End(Not run)
```

devicefarm

AWS Device Farm

Description

Welcome to the AWS Device Farm API documentation, which contains APIs for:

- Testing on desktop browsers
Device Farm makes it possible for you to test your web applications on desktop browsers using Selenium. The APIs for desktop browser testing contain TestGrid in their names. For more information, see [Testing Web Applications on Selenium with Device Farm](#).
- Testing on real mobile devices
Device Farm makes it possible for you to test apps on physical phones, tablets, and other devices in the cloud. For more information, see the [Device Farm Developer Guide](#).

Usage

```
devicefarm(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 <- devicefarm(
  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_device_pool	Creates a device pool
create_instance_profile	Creates a profile that can be applied to one or more private fleet device instances
create_network_profile	Creates a network profile
create_project	Creates a project
create_remote_access_session	Specifies and starts a remote access session
create_test_grid_project	Creates a Selenium testing project
create_test_grid_url	Creates a signed, short-term URL that can be passed to a Selenium RemoteWebDriver component
create_upload	Uploads an app or test scripts
create_vpce_configuration	Creates a configuration record in Device Farm for your Amazon Virtual Private Cloud (VPC) endpoint
delete_device_pool	Deletes a device pool given the pool ARN
delete_instance_profile	Deletes a profile that can be applied to one or more private device instances
delete_network_profile	Deletes a network profile
delete_project	Deletes an AWS Device Farm project, given the project ARN
delete_remote_access_session	Deletes a completed remote access session and its results
delete_run	Deletes the run, given the run ARN
delete_test_grid_project	Deletes a Selenium testing project and all content generated under it
delete_upload	Deletes an upload given the upload ARN
delete_vpce_configuration	Deletes a configuration for your Amazon Virtual Private Cloud (VPC) endpoint
get_account_settings	Returns the number of unmetered iOS or unmetered Android devices that have been purchased by an AWS account
get_device	Gets information about a unique device type
get_device_instance	Returns information about a device instance that belongs to a private device fleet
get_device_pool	Gets information about a device pool
get_device_pool_compatibility	Gets information about compatibility with a device pool
get_instance_profile	Returns information about the specified instance profile
get_job	Gets information about a job
get_network_profile	Returns information about a network profile
get_offering_status	Gets the current status and future status of all offerings purchased by an AWS account
get_project	Gets information about a project
get_remote_access_session	Returns a link to a currently running remote access session
get_run	Gets information about a run
get_suite	Gets information about a suite
get_test	Gets information about a test
get_test_grid_project	Retrieves information about a Selenium testing project
get_test_grid_session	A session is an instance of a browser created through a RemoteWebDriver with the URL of the test
get_upload	Gets information about an upload
get_vpce_configuration	Returns information about the configuration settings for your Amazon Virtual Private Cloud (VPC) endpoint
install_to_remote_access_session	Installs an application to the device in a remote access session
list_artifacts	Gets information about artifacts
list_device_instances	Returns information about the private device instances associated with one or more AWS accounts
list_device_pools	Gets information about device pools

<code>list_devices</code>	Gets information about unique device types
<code>list_instance_profiles</code>	Returns information about all the instance profiles in an AWS account
<code>list_jobs</code>	Gets information about jobs for a given test run
<code>list_network_profiles</code>	Returns the list of available network profiles
<code>list_offering_promotions</code>	Returns a list of offering promotions
<code>list_offerings</code>	Returns a list of products or offerings that the user can manage through the API
<code>list_offering_transactions</code>	Returns a list of all historical purchases, renewals, and system renewal transactions for an
<code>list_projects</code>	Gets information about projects
<code>list_remote_access_sessions</code>	Returns a list of all currently running remote access sessions
<code>list_runs</code>	Gets information about runs, given an AWS Device Farm project ARN
<code>list_samples</code>	Gets information about samples, given an AWS Device Farm job ARN
<code>list_suites</code>	Gets information about test suites for a given job
<code>list_tags_for_resource</code>	List the tags for an AWS Device Farm resource
<code>list_test_grid_projects</code>	Gets a list of all Selenium testing projects in your account
<code>list_test_grid_session_actions</code>	Returns a list of the actions taken in a TestGridSession
<code>list_test_grid_session_artifacts</code>	Retrieves a list of artifacts created during the session
<code>list_test_grid_sessions</code>	Retrieves a list of sessions for a TestGridProject
<code>list_tests</code>	Gets information about tests in a given test suite
<code>list_unique_problems</code>	Gets information about unique problems, such as exceptions or crashes
<code>list_uploads</code>	Gets information about uploads, given an AWS Device Farm project ARN
<code>list_vpce_configurations</code>	Returns information about all Amazon Virtual Private Cloud (VPC) endpoint configurations
<code>purchase_offering</code>	Immediately purchases offerings for an AWS account
<code>renew_offering</code>	Explicitly sets the quantity of devices to renew for an offering, starting from the effective
<code>schedule_run</code>	Schedules a run
<code>stop_job</code>	Initiates a stop request for the current job
<code>stop_remote_access_session</code>	Ends a specified remote access session
<code>stop_run</code>	Initiates a stop request for the current test run
<code>tag_resource</code>	Associates the specified tags to a resource with the specified resourceArn
<code>untag_resource</code>	Deletes the specified tags from a resource
<code>update_device_instance</code>	Updates information about a private device instance
<code>update_device_pool</code>	Modifies the name, description, and rules in a device pool given the attributes and the pool
<code>update_instance_profile</code>	Updates information about an existing private device instance profile
<code>update_network_profile</code>	Updates the network profile
<code>update_project</code>	Modifies the specified project name, given the project ARN and a new name
<code>update_test_grid_project</code>	Change details of a project
<code>update_upload</code>	Updates an uploaded test spec
<code>update_vpce_configuration</code>	Updates information about an Amazon Virtual Private Cloud (VPC) endpoint configuration

Examples

```
## Not run:
svc <- devicefarm()
# The following example creates a new device pool named MyDevicePool
# inside an existing project.
svc$create_device_pool(
  name = "MyDevicePool",
  description = "My Android devices",
```



```
    projectArn = "arn:aws:devicefarm:us-west-2:123456789101:project:EXAMPLE-GUID-123-456",
    rules = list()
  )

  ## End(Not run)
```

mobile

AWS Mobile

Description

AWS Mobile Service provides mobile app and website developers with capabilities required to configure AWS resources and bootstrap their developer desktop projects with the necessary SDKs, constants, tools and samples to make use of those resources.

Usage

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

Operations

<code>create_project</code>	Creates an AWS Mobile Hub project
<code>delete_project</code>	Deletes a project in AWS Mobile Hub
<code>describe_bundle</code>	Get the bundle details for the requested bundle id
<code>describe_project</code>	Gets details about a project in AWS Mobile Hub
<code>export_bundle</code>	Generates customized software development kit (SDK) and or tool packages used to integrate mobile web
<code>export_project</code>	Exports project configuration to a snapshot which can be downloaded and shared
<code>list_bundles</code>	List all available bundles
<code>list_projects</code>	Lists projects in AWS Mobile Hub
<code>update_project</code>	Update an existing project

Examples

```
## Not run:
svc <- mobile()
svc$create_project(
  Foo = 123
)

## End(Not run)
```

mobileanalytics

Amazon Mobile Analytics

Description

Amazon Mobile Analytics is a service for collecting, visualizing, and understanding app usage data at scale.

Usage

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

Operations

[put_events](#) The PutEvents operation records one or more events

Examples

```
## Not run:  
svc <- mobileanalytics()  
svc$put_events(  
  Foo = 123  
)  
  
## End(Not run)
```

Index

amplify, [2](#)
appsync, [4](#)

create_api_cache, [5](#)
create_api_key, [5](#)
create_app, [3](#)
create_backend_environment, [3](#)
create_branch, [3](#)
create_data_source, [5](#)
create_deployment, [3](#)
create_device_pool, [7](#)
create_domain_association, [3](#)
create_function, [5](#)
create_graphql_api, [5](#)
create_instance_profile, [7](#)
create_network_profile, [7](#)
create_project, [7](#), [10](#)
create_remote_access_session, [7](#)
create_resolver, [5](#)
create_test_grid_project, [7](#)
create_test_grid_url, [7](#)
create_type, [5](#)
create_upload, [7](#)
create_vpce_configuration, [7](#)
create_webhook, [3](#)

delete_api_cache, [5](#)
delete_api_key, [5](#)
delete_app, [3](#)
delete_backend_environment, [3](#)
delete_branch, [3](#)
delete_data_source, [5](#)
delete_device_pool, [7](#)
delete_domain_association, [3](#)
delete_function, [5](#)
delete_graphql_api, [5](#)
delete_instance_profile, [7](#)
delete_job, [3](#)
delete_network_profile, [7](#)
delete_project, [7](#), [10](#)
delete_remote_access_session, [7](#)
delete_resolver, [5](#)
delete_run, [7](#)
delete_test_grid_project, [7](#)
delete_type, [5](#)
delete_upload, [7](#)
delete_vpce_configuration, [7](#)
delete_webhook, [3](#)
describe_bundle, [10](#)
describe_project, [10](#)
devicefarm, [6](#)

export_bundle, [10](#)
export_project, [10](#)

flush_api_cache, [5](#)

generate_access_logs, [3](#)
get_account_settings, [7](#)
get_api_cache, [5](#)
get_app, [3](#)
get_artifact_url, [3](#)
get_backend_environment, [3](#)
get_branch, [3](#)
get_data_source, [5](#)
get_device, [7](#)
get_device_instance, [7](#)
get_device_pool, [7](#)
get_device_pool_compatibility, [7](#)
get_domain_association, [3](#)
get_function, [5](#)
get_graphql_api, [5](#)
get_instance_profile, [7](#)
get_introspection_schema, [5](#)
get_job, [3](#), [7](#)
get_network_profile, [7](#)
get_offering_status, [7](#)
get_project, [7](#)
get_remote_access_session, [7](#)
get_resolver, [5](#)

- get_run, 7
- get_schema_creation_status, 5
- get_suite, 7
- get_test, 7
- get_test_grid_project, 7
- get_test_grid_session, 7
- get_type, 5
- get_upload, 7
- get_vpce_configuration, 7
- get_webhook, 3

- install_to_remote_access_session, 7

- list_api_keys, 5
- list_apps, 3
- list_artifacts, 3, 7
- list_backend_environments, 3
- list_branches, 3
- list_bundles, 10
- list_data_sources, 5
- list_device_instances, 7
- list_device_pools, 7
- list_devices, 8
- list_domain_associations, 3
- list_functions, 5
- list_graphql_apis, 5
- list_instance_profiles, 8
- list_jobs, 3, 8
- list_network_profiles, 8
- list_offering_promotions, 8
- list_offering_transactions, 8
- list_offerings, 8
- list_projects, 8, 10
- list_remote_access_sessions, 8
- list_resolvers, 5
- list_resolvers_by_function, 5
- list_runs, 8
- list_samples, 8
- list_suites, 8
- list_tags_for_resource, 3, 5, 8
- list_test_grid_projects, 8
- list_test_grid_session_actions, 8
- list_test_grid_session_artifacts, 8
- list_test_grid_sessions, 8
- list_tests, 8
- list_types, 5
- list_unique_problems, 8
- list_uploads, 8
- list_vpce_configurations, 8

- list_webhooks, 3

- mobile, 9
- mobileanalytics, 10

- purchase_offering, 8
- put_events, 11

- renew_offering, 8

- schedule_run, 8
- start_deployment, 3
- start_job, 3
- start_schema_creation, 5
- stop_job, 3, 8
- stop_remote_access_session, 8
- stop_run, 8

- tag_resource, 3, 5, 8

- untag_resource, 3, 5, 8
- update_api_cache, 5
- update_api_key, 5
- update_app, 3
- update_branch, 3
- update_data_source, 5
- update_device_instance, 8
- update_device_pool, 8
- update_domain_association, 3
- update_function, 5
- update_graphql_api, 5
- update_instance_profile, 8
- update_network_profile, 8
- update_project, 8, 10
- update_resolver, 5
- update_test_grid_project, 8
- update_type, 5
- update_upload, 8
- update_vpce_configuration, 8
- update_webhook, 3