Package: paws.game.development (via r-universe)

September 4, 2024

Title 'Amazon Web Services' Game Development Services

Version 0.7.0

Description Interface to 'Amazon Web Services' game development services, including 'GameLift' game server hosting <https://aws.amazon.com/gametech/>.

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 'gamelift_service.R' 'gamelift_interfaces.R' 'gamelift_operations.R'

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

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

RemoteRef HEAD

RemoteSha 8ab20b498262e0b343c0153b4f244483aff4987f

Contents

gamelift 2

Index **6**

`gamelift` *Amazon GameLift*

Description

Amazon GameLift Service

GameLift provides solutions for hosting session-based multiplayer game servers in the cloud, including tools for deploying, operating, and scaling game servers. Built on AWS global computing infrastructure, GameLift helps you deliver high-performance, high-reliability, low-cost game servers while dynamically scaling your resource usage to meet player demand.

About GameLift solutions

Get more information on these GameLift solutions in the Amazon GameLift Developer Guide.

- Managed GameLift – GameLift offers a fully managed service to set up and maintain computing machines for hosting, manage game session and player session life cycle, and handle security, storage, and performance tracking. You can use automatic scaling tools to balance hosting costs against meeting player demand., configure your game session management to minimize player latency, or add FlexMatch for matchmaking.
- Managed GameLift with Realtime Servers – With GameLift Realtime Servers, you can quickly configure and set up game servers for your game. Realtime Servers provides a game server framework with core Amazon GameLift infrastructure already built in.
- GameLift FleetIQ – Use GameLift FleetIQ as a standalone feature while managing your own EC2 instances and Auto Scaling groups for game hosting. GameLift FleetIQ provides optimizations that make low-cost Spot Instances viable for game hosting.

About this API Reference

This reference guide describes the low-level service API for Amazon GameLift. You can find links to language-specific SDK guides and the AWS CLI reference with each operation and data type topic. Useful links:

- GameLift API operations listed by tasks
- GameLift tools and resources

Usage

```
gamelift(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 <- gamelift(
  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_match	Registers a player's acceptance or rejection of a proposed FlexMatch match
claim_game_server	This operation is used with the Amazon GameLift FleetIQ solution and game server
create_alias	Creates an alias for a fleet
create_build	Creates a new Amazon GameLift build resource for your game server binary files
create_fleet	Creates a new fleet to run your game servers
create_game_server_group	This operation is used with the Amazon GameLift FleetIQ solution and game server
create_game_session	Creates a multiplayer game session for players
create_game_session_queue	Establishes a new queue for processing requests to place new game sessions
create_matchmaking_configuration	Defines a new matchmaking configuration for use with FlexMatch
create_matchmaking_rule_set	Creates a new rule set for FlexMatch matchmaking
create_player_session	Reserves an open player slot in an active game session
create_player_sessions	Reserves open slots in a game session for a group of players
create_script	Creates a new script record for your Realtime Servers script
create_vpc_peering_authorization	Requests authorization to create or delete a peer connection between the VPC for yo
create_vpc_peering_connection	Establishes a VPC peering connection between a virtual private cloud (VPC) in an A
delete_alias	Deletes an alias
delete_build	Deletes a build
delete_fleet	Deletes everything related to a fleet
delete_game_server_group	This operation is used with the Amazon GameLift FleetIQ solution and game server
delete_game_session_queue	Deletes a game session queue
delete_matchmaking_configuration	Permanently removes a FlexMatch matchmaking configuration
delete_matchmaking_rule_set	Deletes an existing matchmaking rule set
delete_scaling_policy	Deletes a fleet scaling policy
delete_script	Deletes a Realtime script
delete_vpc_peering_authorization	Cancels a pending VPC peering authorization for the specified VPC
delete_vpc_peering_connection	Removes a VPC peering connection
deregister_game_server	This operation is used with the Amazon GameLift FleetIQ solution and game server
describe_alias	Retrieves properties for an alias
describe_build	Retrieves properties for a custom game build

describe_ec2_instance_limits	Retrieves the following information for the specified EC2 instance type:
describe_fleet_attributes	Retrieves core properties, including configuration, status, and metadata, for a fleet
describe_fleet_capacity	Retrieves the current capacity statistics for one or more fleets
describe_fleet_events	Retrieves entries from the specified fleet's event log
describe_fleet_port_settings	Retrieves a fleet's inbound connection permissions
describe_fleet_utilization	Retrieves utilization statistics for one or more fleets
describe_game_server	This operation is used with the Amazon GameLift FleetIQ solution and game server
describe_game_server_group	This operation is used with the Amazon GameLift FleetIQ solution and game server
describe_game_server_instances	This operation is used with the Amazon GameLift FleetIQ solution and game server
describe_game_session_details	Retrieves properties, including the protection policy in force, for one or more game s
describe_game_session_placement	Retrieves properties and current status of a game session placement request
describe_game_session_queues	Retrieves the properties for one or more game session queues
describe_game_sessions	Retrieves a set of one or more game sessions
describe_instances	Retrieves information about a fleet's instances, including instance IDs
describe_matchmaking	Retrieves one or more matchmaking tickets
describe_matchmaking_configurations	Retrieves the details of FlexMatch matchmaking configurations
describe_matchmaking_rule_sets	Retrieves the details for FlexMatch matchmaking rule sets
describe_player_sessions	Retrieves properties for one or more player sessions
describe_runtime_configuration	Retrieves a fleet's runtime configuration settings
describe_scaling_policies	Retrieves all scaling policies applied to a fleet
describe_script	Retrieves properties for a Realtime script
describe_vpc_peering_authorizations	Retrieves valid VPC peering authorizations that are pending for the AWS account
describe_vpc_peering_connections	Retrieves information on VPC peering connections
get_game_session_log_url	Retrieves the location of stored game session logs for a specified game session
get_instance_access	Requests remote access to a fleet instance
list_aliases	Retrieves all aliases for this AWS account
list_builds	Retrieves build resources for all builds associated with the AWS account in use
list_fleets	Retrieves a collection of fleet resources for this AWS account
list_game_server_groups	This operation is used with the Amazon GameLift FleetIQ solution and game server
list_game_servers	This operation is used with the Amazon GameLift FleetIQ solution and game server
list_scripts	Retrieves script records for all Realtime scripts that are associated with the AWS acc
list_tags_for_resource	Retrieves all tags that are assigned to a GameLift resource
put_scaling_policy	Creates or updates a scaling policy for a fleet
register_game_server	This operation is used with the Amazon GameLift FleetIQ solution and game server
request_upload_credentials	Retrieves a fresh set of credentials for use when uploading a new set of game build fi
resolve_alias	Retrieves the fleet ID that an alias is currently pointing to
resume_game_server_group	This operation is used with the Amazon GameLift FleetIQ solution and game server
search_game_sessions	Retrieves all active game sessions that match a set of search criteria and sorts them ir
start_fleet_actions	Resumes activity on a fleet that was suspended with StopFleetActions
start_game_session_placement	Places a request for a new game session in a queue (see CreateGameSessionQueue)
start_match_backfill	Finds new players to fill open slots in an existing game session
start_matchmaking	Uses FlexMatch to create a game match for a group of players based on custom matc
stop_fleet_actions	Suspends activity on a fleet
stop_game_session_placement	Cancels a game session placement that is in PENDING status
stop_matchmaking	Cancels a matchmaking ticket or match backfill ticket that is currently being process
suspend_game_server_group	This operation is used with the Amazon GameLift FleetIQ solution and game server
tag_resource	Assigns a tag to a GameLift resource
untag_resource	Removes a tag that is assigned to a GameLift resource

update_alias	Updates properties for an alias
update_build	Updates metadata in a build resource, including the build name and version
update_fleet_attributes	Updates fleet properties, including name and description, for a fleet
update_fleet_capacity	Updates capacity settings for a fleet
update_fleet_port_settings	Updates port settings for a fleet
update_game_server	This operation is used with the Amazon GameLift FleetIQ solution and game server
update_game_server_group	This operation is used with the Amazon GameLift FleetIQ solution and game server
update_game_session	Updates game session properties
update_game_session_queue	Updates settings for a game session queue, which determines how new game session
update_matchmaking_configuration	Updates settings for a FlexMatch matchmaking configuration
update_runtime_configuration	Updates the current runtime configuration for the specified fleet, which tells Amazon
update_script	Updates Realtime script metadata and content
validate_matchmaking_rule_set	Validates the syntax of a matchmaking rule or rule set

Examples

```
## Not run:
svc <- gamelift()
svc$accept_match(
  Foo = 123
)

## End(Not run)
```

Index

accept_match, *3*

claim_game_server, *3*
create_alias, *3*
create_build, *3*
create_fleet, *3*
create_game_server_group, *3*
create_game_session, *3*
create_game_session_queue, *3*
create_matchmaking_configuration, *3*
create_matchmaking_rule_set, *3*
create_player_session, *3*
create_player_sessions, *3*
create_script, *3*
create_vpc_peering_authorization, *3*
create_vpc_peering_connection, *3*

delete_alias, *3*
delete_build, *3*
delete_fleet, *3*
delete_game_server_group, *3*
delete_game_session_queue, *3*
delete_matchmaking_configuration, *3*
delete_matchmaking_rule_set, *3*
delete_scaling_policy, *3*
delete_script, *3*
delete_vpc_peering_authorization, *3*
delete_vpc_peering_connection, *3*
deregister_game_server, *3*
describe_alias, *3*
describe_build, *3*
describe_ec2_instance_limits, *4*
describe_fleet_attributes, *4*
describe_fleet_capacity, *4*
describe_fleet_events, *4*
describe_fleet_port_settings, *4*
describe_fleet_utilization, *4*
describe_game_server, *4*
describe_game_server_group, *4*
describe_game_server_instances, *4*
describe_game_session_details, *4*
describe_game_session_placement, *4*
describe_game_session_queues, *4*
describe_game_sessions, *4*
describe_instances, *4*
describe_matchmaking, *4*
describe_matchmaking_configurations, *4*
describe_matchmaking_rule_sets, *4*
describe_player_sessions, *4*
describe_runtime_configuration, *4*
describe_scaling_policies, *4*
describe_script, *4*
describe_vpc_peering_authorizations, *4*
describe_vpc_peering_connections, *4*

gamelift, 2
get_game_session_log_url, *4*
get_instance_access, *4*

list_aliases, *4*
list_builds, *4*
list_fleets, *4*
list_game_server_groups, *4*
list_game_servers, *4*
list_scripts, *4*
list_tags_for_resource, *4*

put_scaling_policy, *4*

register_game_server, *4*
request_upload_credentials, *4*
resolve_alias, *4*
resume_game_server_group, *4*

search_game_sessions, *4*
start_fleet_actions, *4*
start_game_session_placement, *4*
start_match_backfill, *4*
start_matchmaking, *4*
stop_fleet_actions, *4*
stop_game_session_placement, *4*

stop_matchmaking, *4*
suspend_game_server_group, *4*

tag_resource, *4*

untag_resource, *4*
update_alias, *5*
update_build, *5*
update_fleet_attributes, *5*
update_fleet_capacity, *5*
update_fleet_port_settings, *5*
update_game_server, *5*
update_game_server_group, *5*
update_game_session, *5*
update_game_session_queue, *5*
update_matchmaking_configuration, *5*
update_runtime_configuration, *5*
update_script, *5*

validate_matchmaking_rule_set, *5*