

Package ‘aws.lambda’

July 22, 2025

Type Package

Title AWS Lambda Client Package

Version 0.2.0

Description A simple client package for the Amazon Web Services ('AWS') Lambda API <<https://aws.amazon.com/lambda/>>.

Encoding UTF-8

LazyData true

Depends R (>= 3.4)

License GPL-3

Imports utils, http, jsonlite, aws.signature (>= 0.3.4), base64enc

Suggests testthat, aws.s3, aws.iam

URL <https://github.com/cloudyr/aws.lambda>

BugReports <https://github.com/cloudyr/aws.lambda/issues>

RoxygenNote 7.1.0

NeedsCompilation no

Author Thomas J. Leeper [aut] (ORCID: <<https://orcid.org/0000-0003-4097-6326>>),
Jon Harmon [ctb, cre] (ORCID: <<https://orcid.org/0000-0003-4781-4346>>)

Maintainer Jon Harmon <jonthegeek@gmail.com>

Repository CRAN

Date/Publication 2020-04-15 15:40:02 UTC

Contents

create_function	2
create_function_alias	4
get_function	5
get_function_name	7
get_lambda_account	7
invoke_function	8
lambdaHTTP	8

Index[11](#)

create_function	<i>Manage AWS Lambda Functions</i>
-----------------	------------------------------------

Description

Create, update, and version AWS Lambda functions

Usage

```
create_function(  
    name,  
    func,  
    handler,  
    role,  
    runtime = c("nodejs12.x", "nodejs10.x", "java11", "java8", "python3.8", "python3.7",  
        "python3.6", "python2.7", "dotnetcore3.1", "dotnetcore2.1", "go1.x", "ruby2.7",  
        "ruby2.5", "provided"),  
    timeout = 3L,  
    description,  
    ...  
)  
  
update_function_code(name, func, ...)  
  
update_function_config(  
    name,  
    description,  
    handler,  
    role,  
    runtime = c("nodejs12.x", "nodejs10.x", "java11", "java8", "python3.8", "python3.7",  
        "python3.6", "python2.7", "dotnetcore3.1", "dotnetcore2.1", "go1.x", "ruby2.7",  
        "ruby2.5", "provided"),  
    timeout = 3L,  
    ...  
)  
  
update_function(  
    name,  
    func,  
    description,  
    handler,  
    role,  
    runtime = c("nodejs12.x", "nodejs10.x", "java11", "java8", "python3.8", "python3.7",  
        "python3.6", "python2.7", "dotnetcore3.1", "dotnetcore2.1", "go1.x", "ruby2.7",  
        "ruby2.5", "provided"),  
    timeout = 3L,
```

```

    ...
)

publish_function_version(name, description, ...)

make_function_version(name, description, ...)

```

Arguments

name	A character string specifying the function name (either a full ARN or a max 64-character string). For functions other than <code>create_function</code> this can also be an object of class “aws_lambda_function”.
func	Either (1) a character string containing a url-style AWS S3 bucket and object key (e.g., “s3://bucketname/objectkey”) where the object is the .zip file containing the AWS Lambda deployment package; (2) a file string pointing to a .zip containing the deployment package; or (3) a single file (e.g., a javascript file) that will be zipped and used as the deployment. The third option is merely a convenience for very simple deployment packages.
handler	A character string specifying the function within your code that Lambda calls to begin execution.
role	A character string containing an IAM role or an object of class “iam_role”. This is the role that is used when the function is invoked, so it must have permissions over any AWS resources needed by the function.
runtime	A character string specifying the runtime environment for the function.
timeout	An integer specifying the timeout for the function, in seconds.
description	Optionally, a max 256-character description of the function for your own use.
...	Additional arguments passed to lambdaHTTP .

Details

`create_function` creates a new function from a deployment package. `update_function_code` updates the code within a function. `update_function_config` updates the configuration settings of a function. `publish_function_version` records a function version (see [list_function_versions](#); changes made between versioning are not recorded).

Value

A list of class “aws_lambda_function”.

References

[API Reference: CreateFunction](#) [API Reference: UpdateFunctionCode](#) [API Reference: PublishVersion](#)

See Also

[invoke_function](#), [create_function_alias](#), [list_functions](#), [delete_function](#)

Examples

```
## Not run:
# 'hello world!' example code
hello <- system.file("templates", "helloworld.js", package = "aws.lambda")

# get IAM role for Lambda execution
library("aws.iam")
id <- get_caller_identity()[["Account"]]
# Note: This role may not work. We recommend copying the ARN of a
# Lambda-capable role from the console once until we're able to more
# smoothly integrate with aws.iam.
role <- paste0("arn:aws:iam::", id, ":role/lambda_basic_execution")

lambda_func <- create_function("helloworld",
  func = hello,
  handler = "helloworld.handler",
  role = role
)

# invoke function
invoke_function(lambda_func)

# delete function
delete_function(lambda_func)

## End(Not run)
```

create_function_alias *Alias Management*

Description

List, create, update, and delete function aliases

Usage

```
create_function_alias(name, alias, version, description, ...)

update_function_alias(name, alias, version, description, ...)

delete_function_alias(name, alias, ...)

get_function_alias(name, alias, ...)

list_function_aliases(name, version, marker, n, ...)
```

Arguments

name	A character string specifying the function name (either a full ARN or a max 64-character string). For functions other than create_function this can also be an object of class “aws_lambda_function”.
alias	A character string specifying a function alias
version	A character string specifying a function version to associate with this alias.
description	Optionally, a max 256-character description of the function for your own use.
...	Additional arguments passed to lambdaHTTP .
marker	A pagination marker from a previous request.
n	An integer specifying the number of results to return.

Details

create_function_alias creates a new function alias for a given version of a function. update_function_alias updates the association between a function alias and the function version. list_function_aliases lists all function aliases. get_function_alias retrieves a specific function alias. delete_function_alias deletes a function alias, but not the associated function version.

Value

An object of class “aws_lambda_function”.

References

[API Reference: GetAlias](#) [API Reference: CreateAlias](#) [API Reference: UpdateAlias](#) [API Reference: DeleteAlias](#) [API Reference: ListAliases](#)

See Also

[create_function](#), [list_functions](#)

get_function	<i>Function Management</i>
--------------	----------------------------

Description

List functions, function versions, and function policies

Usage

```
get_function(name, qualifier, ...)

list_functions(marker, n, ...)

list_function_versions(name, marker, n, ...)

delete_function(name, qualifier, ...)

get_function_policy(name, qualifier, ...)
```

Arguments

name	A character string specifying the function name (either a full ARN or a max 64-character string). For functions other than <code>create_function</code> this can also be an object of class “aws_lambda_function”.
qualifier	Optionally, either a function version or alias. If omitted, information about the latest version is returned.
...	Additional arguments passed to lambdaHTTP .
marker	A pagination marker from a previous request.
n	An integer specifying the number of results to return.

Details

`list_functions` lists all functions. `get_function` retrieves a specific function and `list_function_versions` retrieves all versions of that function. `get_function_policy` returns the resource-based IAM policy for a function. `delete_function` deletes a function, if you have permission to do so.

Value

An object of class “aws_lambda_function”.

References

[API Reference: GetFunction](#) [API Reference: ListVersionsByFunction](#) [API Reference: ListFunctions](#) [API Reference: GetPolicy](#)

See Also

[create_function](#), [update_function_code](#), [update_function_config](#)

get_function_name	<i>Get name of Lambda function</i>
-------------------	------------------------------------

Description

Extracts an AWS Lambda function's name

Usage

```
get_function_name(x, ...)  
  
## S3 method for class 'character'  
get_function_name(x, ...)  
  
## S3 method for class 'aws_lambda_function'  
get_function_name(x, ...)
```

Arguments

x	An object of class “aws_lambda_function”.
...	Additional arguments passed to methods

get_lambda_account	<i>AWS Lambda Account Settings</i>
--------------------	------------------------------------

Description

Get account settings

Usage

```
get_lambda_account(...)
```

Arguments

...	Additional arguments passed to lambdaHTTP .
-----	---

Value

A list.

Examples

```
## Not run:  
get_lambda_account()  
  
## End(Not run)
```

invoke_function	<i>Invoke Lambda Function</i>
-----------------	-------------------------------

Description

Invoke a lambda function

Usage

```
invoke_function(  
  name,  
  qualifier,  
  payload = NULL,  
  type = c("RequestResponse", "Event", "DryRun"),  
  log = c("None", "Tail"),  
  ...  
)
```

Arguments

name	A character string specifying the function name (either a full ARN or a max 64-character string). For functions other than <code>create_function</code> this can also be an object of class “aws_lambda_function”.
qualifier	Optionally, either a function version or alias. If omitted, information about the latest version is returned.
payload	Optionally, a list of parameters to send in the JSON body to the function.
type	A character string specifying one of: (1) “Event” (asynchronous execution), (2) “RequestResponse” (the default), or (3) “DryRun” to test the function without executing it.
log	A character string to control log response.
...	Additional arguments passed to lambdaHTTP .

See Also

[create_function](#), [list_functions](#),

lambdaHTTP	<i>Execute AWS Lambda API Request</i>
------------	---------------------------------------

Description

This is the workhorse function to execute calls to the Lambda API.

Usage

```
lambdaHTTP(  
  verb = "GET",  
  action,  
  query = NULL,  
  headers = list(),  
  body = NULL,  
  verbose = getOption("verbose", FALSE),  
  region = Sys.getenv("AWS_DEFAULT_REGION", "us-east-1"),  
  key = NULL,  
  secret = NULL,  
  session_token = NULL,  
  ...  
)
```

Arguments

verb	A character string specifying the HTTP verb to use.
action	A character string specifying the API version and endpoint.
query	An optional named list containing query string parameters and their character values.
headers	A list of headers to pass to the HTTP request.
body	The HTTP request body.
verbose	A logical indicating whether to be verbose. Default is given by <code>options("verbose")</code> .
region	A character string specifying an AWS region. See locate_credentials .
key	A character string specifying an AWS Access Key. See locate_credentials .
secret	A character string specifying an AWS Secret Key. See locate_credentials .
session_token	Optionally, a character string specifying an AWS temporary Session Token to use in signing a request. See locate_credentials .
...	Additional arguments passed to GET or another http request function.

Details

This function constructs and signs an AWS Lambda API request and returns the results thereof, or relevant debugging information in the case of error.

Value

If successful, a named list. Otherwise, a data structure of class “aws-error” containing any error message(s) from AWS and information about the request attempt.

Author(s)

Thomas J. Leeper

See Also

[get_lambda_account](#), which works well as a hello world for the package

Index

`create_function`, [2](#), [5](#), [6](#), [8](#)
`create_function_alias`, [3](#), [4](#)

`delete_function`, [3](#)
`delete_function (get_function)`, [5](#)
`delete_function_alias`
 (`create_function_alias`), [4](#)

`GET`, [9](#)
`get_function`, [5](#)
`get_function_alias`
 (`create_function_alias`), [4](#)
`get_function_name`, [7](#)
`get_function_policy (get_function)`, [5](#)
`get_lambda_account`, [7](#), [10](#)

`invoke_function`, [3](#), [8](#)

`lambdaHTTP`, [3](#), [5–8](#), [8](#)
`list_function_aliases`
 (`create_function_alias`), [4](#)
`list_function_versions`, [3](#)
`list_function_versions (get_function)`, [5](#)
`list_functions`, [3](#), [5](#), [8](#)
`list_functions (get_function)`, [5](#)
`locate_credentials`, [9](#)

`make_function_version`
 (`create_function`), [2](#)

`publish_function_version`
 (`create_function`), [2](#)

`update_function (create_function)`, [2](#)
`update_function_alias`
 (`create_function_alias`), [4](#)
`update_function_code`, [6](#)
`update_function_code (create_function)`,
 [2](#)
`update_function_config`, [6](#)
`update_function_config`
 (`create_function`), [2](#)