

# Package: algorithmia (via r-universe)

October 13, 2024

**Type** Package

**Title** Allows you to Easily Interact with the Algorithmia Platform

**Version** 0.3.1

**Date** 2020-09-28

**Author** James Sutton, James Athappilly, Taylor Raack

**Maintainer** Robert Fulton <rfulton@algorithmia.com>

**Description** The company, Algorithmia, houses the largest marketplace of online algorithms. This package essentially holds a bunch of REST wrappers that make it very easy to call algorithms in the Algorithmia platform and access files and directories in the Algorithmia data API. To learn more about the services they offer and the algorithms in the platform visit <<http://algorithmia.com>>. More information for developers can be found at <<https://algorithmia.com/developers>>.

**Depends** R (>= 3.3.1), methods (>= 2.5.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Collate** 'Acl.r' 'Client.r' 'Algorithm.r' 'AlgorithmHandler.r'  
'DataDirectory.r' 'DataFile.r' 'data\_utilities.r'

**Imports** base64enc, httr, rjson, readr, here

**BugReports** <https://github.com/algorithmiaio/algorithmia-r/issues>

**RoxygenNote** 7.1.1

**Suggests** knitr, rmarkdown, RUnit

**VignetteBuilder** knitr

**Repository** <https://algorithmiaio.r-universe.dev>

**RemoteUrl** <https://github.com/algorithmiaio/algorithmia-r>

**RemoteRef** HEAD

**RemoteSha** 36e8b5b8168ed83d12a7d8ed8272d0758977ae8e

## Contents

AlgorithmiaAcl-class . . . . .	2
AlgorithmiaAlgorithm-class . . . . .	2
AlgorithmiaClient-class . . . . .	3
AlgorithmiaDataDirectory-class . . . . .	3
AlgorithmiaDataFile-class . . . . .	4
AlgorithmiaDirectoryIterator-class . . . . .	5
getAlgorithmHandler . . . . .	6
getAlgorithmiaClient . . . . .	7
ReadAcl.MY_ALGORITHMS . . . . .	7
ReadAcl.PRIVATE . . . . .	8
ReadAcl.PUBLIC . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

AlgorithmiaAcl-class *Internal class used to describe ACLs.*

---

### Description

Internal class used to describe ACLs.

---

AlgorithmiaAlgorithm-class  
*Algorithm object which enables you to call Algorithmia algorithms.  
 To create one, call: 'client\$algo("algoUrl")'*

---

### Description

Algorithm object which enables you to call Algorithmia algorithms. To create one, call: 'client\$algo("algoUrl")'

### Fields

`client` Reference to the AlgorithmiaClient object that has the credentials necessary to make API calls.

`algoUrl` The unique identifier for an algorithm. Follows the pattern: [Algorithm Author]/[Algorithm Name]/[Optional Version] like: 'demo/Hello/0.1.1'.

`queryParameters` Mutable list of parameters to use while making algorithm calls. These can be changed by calling `setOptions`.

### Methods

`pipe(input)` Calls an algorithm with the input provided.

`setOptions( timeout = 300, stdout = FALSE, output = "default", parameters = list() )` Allows you to set the timeout duration (in seconds), whether you want the stdout that was produced while running the algorithm (this only works when the algorithm author call it), and whether this should run as async (output = 'void') or in raw mode (output = 'raw').

---

**AlgorithmiaClient-class**

*Client object which makes it easy to interact with the Algorithmia REST API. To create one, call 'getAlgorithmiaClient("YOUR\_ALGORITHMIA\_API\_KEY")'*

---

**Description**

Client object which makes it easy to interact with the Algorithmia REST API. To create one, call 'getAlgorithmiaClient("YOUR\_ALGORITHMIA\_API\_KEY")'

**Fields**

`apiKey` The API key used when making REST calls to Algorithmia. This should NOT be set inside algorithms.

`apiAddress` The Algorithmia API address. In most cases you don't need to set this explicitly since the default will talk to the correct Algorithmia API server.

**Methods**

`algo(algoRef)` Takes an algorithm reference and returns an AlgorithmiaAlgorithm object. An algorithm reference is a string of the form [Algorithm Author]/[Algorithm Name]/[Optional Version] like: 'demo/Hello/0.1.1'. AlgorithmiaAlgorithm objects are used to call algorithms with data.

`dir(dataUrl)` Takes a path to a directory and returns a AlgorithmiaDataDirectory object. Data paths are described in detail at: <http://docs.algorithmia.com/?java#data-api-specification>. AlgorithmiaDataDirectory objects are used to interact with directories.

`file(dataUrl)` Takes a path to a file and returns a AlgorithmiaDataFile object. Data paths are described in detail at: <http://docs.algorithmia.com/?java#data-api-specification>. AlgorithmiaDataFile objects are used to read and write files.

`reportInsights(insights)` Takes a list of Algorithmia Insights and reports them for this algorithm execution.

---

**AlgorithmiaDataDirectory-class**

*DataDirectory object to interact with directories. Supports Algorithmia data directories, S3, Dropbox and more coming soon! To create one, call: 'client\$dir("directory\_path")'*

---

**Description**

DataDirectory object to interact with directories. Supports Algorithmia data directories, S3, Dropbox and more coming soon! To create one, call: 'client\$dir("directory\_path")'

**Fields**

- `client` Reference to the AlgorithmiaClient object that has the credentials necessary to make API calls.
- `dataDirectoryUrl` Convenience field that holds `"/v1/data/" + dataDirectoryPath`
- `dataDirectoryPath` The path of to the directory

**Methods**

- `create(acl = NULL)` Creates the directory with the ACL provided.
- `delete(force = FALSE)` Deletes the directory. If the directory is not empty, it will fail unless `'force'` is set to TRUE.
- `dir(name)` Returns an AlgorithmiaDataDirectory object for the child directory.
- `dirs()` Returns an AlgorithmiaDirectoryIterator of all the child directories.
- `exists()` Returns TRUE if this directory exists. FALSE, if it does not.
- `file(name)` Returns an AlgorithmiaDataFile object for the child file.
- `files()` Returns an AlgorithmiaDirectoryIterator of all the child files.
- `getName()` Returns the name of the innermost directory.
- `getParent()` Returns the url for the parent directory (everything but the innermost directory).
- `getPermissions()` Returns the AlgorithmiaAcl object representing the permissions of the directory.
- `updatePermissions(acl)` Updates the permissions of the directory. Currently supported ACLs are: `'ReadAcl.PUBLIC'`, `'ReadAcl.PRIVATE'`, and `'ReadAcl.MY_ALGORITHMS'`.

---

**AlgorithmiaDataFile-class**

*DataFile object to interact with files. Supports Algorithmia data files, S3, Dropbox and more coming soon! To create one, call: `'client$file("file_path")'`*

---

**Description**

DataFile object to interact with files. Supports Algorithmia data files, S3, Dropbox and more coming soon! To create one, call: `'client$file("file_path")'`

**Fields**

- `client` Reference to the AlgorithmiaClient object that has the credentials necessary to make API calls.
- `dataFileUrl` Url to access the file.
- `last_modified` A timestamp for the last modified time. Only gets set when `setAttributes` is called.
- `size` Size (in bytes) of the file. Only gets set when `setAttributes` is called.

**Methods**

- delete() Deletes the file.
- exists() Returns TRUE if this file exists. FALSE, if it does not.
- getFile() Copies the file to a tempfile and returns the path to that file.
- getJson() Returns the contents of the file after it has been JSON decoded.
- getName() Returns the name of the file without the parent directory path.
- getRaw() Returns the raw contents of the file.
- getString() Returns the contents of the file as a string.
- put(data) Writes the file with the current data passed in.
- putFile(fileName) Takes a local file path and writes its contents to this file.
- putJson(data) Encodes the data to a JSON object and writes that to the file.
- setAttributes(attributes) Sets the last\_modified time and size (in bytes) of the file.

---

**AlgorithmiaDirectoryIterator-class**

*DirectoryIterator object to iterate over child files or directories of a parent directory. To create one, call: 'dataDirectory\$dirs()' or 'dataDirectory\$files()'*

---

**Description**

DirectoryIterator object to iterate over child files or directories of a parent directory. To create one, call: 'dataDirectory\$dirs()' or 'dataDirectory\$files()'

**Fields**

- dataDirectory Parent directory whose children we are iterating over.
- typeFilter Either DATA\_OBJECT\_TYPE\_FILE or DATA\_OBJECT\_TYPE\_DIRECTORY.
- marker Page marker while iterating over a directory with lots of entries.
- first Whether we have asked the server for anything for the first time.
- index The current position of the directoryEntries list.
- directoryEntries The list of children entries we are iterating over.

**Methods**

- getNext() Returns the next element or stops if you have asked for a next element that does not exist.
- hasNext() Returns TRUE if there are any more elements. FALSE otherwise.

---

getAlgorithmHandler	<i>Creates a new Algorithm Handler which registers the onLoad method (which loads the model or other dependencies for the algorithm) and the apply method which takes in the result of the onLoad method and the input from the user.</i>
---------------------	---

---

## Description

Creates a new Algorithm Handler which registers the onLoad method (which loads the model or other dependencies for the algorithm) and the apply method which takes in the result of the onLoad method and the input from the user.

## Usage

```
getAlgorithmHandler(  
  applyfunc,  
  onLoadMethod = function() { NULL },  
  pipe = "stdin"  
)
```

## Arguments

applyfunc	The method that we will call synchronously for each algorithm call. The first argument it will be called with is the user input, followed with the output of the onLoad function.
onLoadMethod	This optional method that is run once when the process first starts. It loads the data required and other shared state for each algorithm call.
pipe	The file which we will read line-by-line to get user input

## Value

A new AlgorithmHandler object

## Examples

```
loadPrefix <- function() { "Hello" }  
algorithm <- function(input, prefix) { paste(prefix, input) }  
  
#To create an algorithm that returns "Hello" + input:  
algo <- getAlgorithmHandler(algorithm, loadPrefix)
```

---

`getAlgorithmiaClient` *Creates a new Algorithmia Client which you can use to call algorithms and interact with directories and files in the Algorithmia data API.*

---

### Description

Creates a new Algorithmia Client which you can use to call algorithms and interact with directories and files in the Algorithmia data API.

### Usage

```
getAlgorithmiaClient(apiKey = NA_character_, apiAddress = NA_character_)
```

### Arguments

`apiKey` The Algorithmia API key. You need to set this when you are interacting with Algorithmia outside of an algorithm. To find your Algorithmia API key visit: [https://algorithmia.com/users/\[YOUR\\_USER\\_NAME\]](https://algorithmia.com/users/[YOUR_USER_NAME])

`apiAddress` The Algorithmia API address. Normal users should not set this. This defaults to "https://api.algorithmia.com" when it is not explicitly set.

### Value

A new AlgorithmiaClient object

### Examples

```
client <- algorithmia::getAlgorithmiaClient() # Inside an Algorithmia algorithm
client <- algorithmia::getAlgorithmiaClient("YOUR_ALGORITHMIA_API_KEY") # Everywhere else
```

---

`ReadAcl.MY_ALGORITHMS` *The ACL that allows your algorithms to read an item. Sample usage: `dataDirectory$create(ReadAcl.MY_ALGORITHMS)` `dataDirectory$updatePermissions(ReadAcl.MY_ALGORITHMS)`*

---

### Description

The ACL that allows your algorithms to read an item. Sample usage: `dataDirectory$create(ReadAcl.MY_ALGORITHMS)` `dataDirectory$updatePermissions(ReadAcl.MY_ALGORITHMS)`

### Usage

```
ReadAcl.MY_ALGORITHMS
```

### Format

An object of class `AlgorithmiaAcl` of length 1.

---

ReadAcl.PRIVATE	<i>The ACL that allows only you to read an item. Sample usage: dataDirectory\$create(ReadAcl.PRIVATE) dataDirectory\$updatePermissions(ReadAcl.PRIVATE)</i>
-----------------	---

---

**Description**

The ACL that allows only you to read an item. Sample usage: dataDirectory\$create(ReadAcl.PRIVATE) dataDirectory\$updatePermissions(ReadAcl.PRIVATE)

**Usage**

ReadAcl.PRIVATE

**Format**

An object of class AlgorithmiaAcl of length 1.

---

ReadAcl.PUBLIC	<i>The ACL that allows anyone to read an item. Sample usage: dataDirectory\$create(ReadAcl.PUBLIC) dataDirectory\$updatePermissions(ReadAcl.PUBLIC)</i>
----------------	---

---

**Description**

The ACL that allows anyone to read an item. Sample usage: dataDirectory\$create(ReadAcl.PUBLIC) dataDirectory\$updatePermissions(ReadAcl.PUBLIC)

**Usage**

ReadAcl.PUBLIC

**Format**

An object of class AlgorithmiaAcl of length 1.

# Index

## \* datasets

- ReadAcl.MY\_ALGORITHMS, 7
- ReadAcl.PRIVATE, 8
- ReadAcl.PUBLIC, 8

  

- AlgorithmiaAcl (AlgorithmiaAcl-class), 2
- AlgorithmiaAcl-class, 2
- AlgorithmiaAlgorithm
  - (AlgorithmiaAlgorithm-class), 2
- AlgorithmiaAlgorithm-class, 2
- AlgorithmiaClient
  - (AlgorithmiaClient-class), 3
- AlgorithmiaClient-class, 3
- AlgorithmiaDataDirectory
  - (AlgorithmiaDataDirectory-class),
  - 3
- AlgorithmiaDataDirectory-class, 3
- AlgorithmiaDataFile
  - (AlgorithmiaDataFile-class), 4
- AlgorithmiaDataFile-class, 4
- AlgorithmiaDirectoryIterator
  - (AlgorithmiaDirectoryIterator-class),
  - 5
- AlgorithmiaDirectoryIterator-class, 5

  

- getAlgorithmHandler, 6
- getAlgorithmiaClient, 7

  

- ReadAcl.MY\_ALGORITHMS, 7
- ReadAcl.PRIVATE, 8
- ReadAcl.PUBLIC, 8