

# Package ‘alterryx’

November 21, 2018

**Type** Package

**Title** An 'API' Client for the 'Alteryx' Gallery

**Version** 0.4.0

**Author** Michael Treadwell

**Maintainer** Michael Treadwell <michael.treadwell@interworks.com>

**Description** A tool to access each of the 'Alteryx' Gallery 'API' endpoints. Users can queue jobs, poll job status, and retrieve application output as a data frame. You will need an 'Alteryx' Server license and have 'Alteryx' Gallery running to utilize this package. The 'API' is accessed through the 'URL' that you setup for the server running 'Alteryx' Gallery and more information on the endpoints can be found at <<https://gallery.alteryx.com/api-docs/>>.

**URL** <https://github.com/mtreadwell/alterryx>,  
<https://gallery.alteryx.com/api-docs/>

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**Imports** httr, utils, digest, base64enc, jsonlite (>= 1.4)

**RoxygenNote** 6.1.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-11-21 18:40:03 UTC

## R topics documented:

alterryx . . . . .	2
app_jobs . . . . .	3
as.alteryx_app . . . . .	4

as.alterryx_job . . . . .	4
build_base_string . . . . .	5
build_base_url . . . . .	5
build_request_url . . . . .	6
build_signature . . . . .	6
check_class . . . . .	7
check_keys . . . . .	7
check_status . . . . .	7
download_app . . . . .	8
empty_answer . . . . .	8
encode_list . . . . .	9
generate_nonce . . . . .	9
generate_required_headers . . . . .	9
get_app . . . . .	10
get_app_questions . . . . .	11
get_gallery . . . . .	11
get_info . . . . .	12
get_job_log . . . . .	12
get_job_output . . . . .	13
get_priority . . . . .	13
normalize_request_params . . . . .	14
parse_request_response . . . . .	14
remove_bom . . . . .	15
running_jobs . . . . .	15
sign_base_string . . . . .	16
submit_get_request . . . . .	17
submit_post_request . . . . .	17
<b>Index</b>	<b>18</b>

---

alterryx

*alterryx: An API client for the Alteryx Gallery*


---

## Description

alterryx provides access to each of the Alteryx Gallery API endpoints. With alterryx users can:

## Details

\* Retrieve information on Alteryx Gallery resources like apps, workflows, and macros \* Queue jobs for applications \* Retrieve the status and output of jobs

For more information about the Alteryx Gallery API, see the documentation: <https://gallery.alteryx.com/api-docs/>

To learn more about alterryx, start with the vignettes: `browseVignettes(package = "alterryx")`

---

`app_jobs`*Get App Jobs*

---

## Description

A job is a single run of an app. To queue a job for an app use `queue_app`. Once queued, there are two ways to poll the gallery for an update on the job status: `get_app_jobs` or `get_job`.

`get_app_jobs` will return all jobs for a given app. `get_job` will return a single job and is mostly used for polling the status of a job.

## Usage

```
get_app_jobs(app, request_params = list(), gallery = get_gallery())
```

```
get_job(job, gallery = get_gallery())
```

## Arguments

<code>app</code>	A single <code>alteryx_app</code> returned from <code>get_app</code>
<code>request_params</code>	List of app search parameters. For more information on parameters, visit the Alteryx Gallery API documentation <a href="https://gallery.alteryx.com/api-docs/">https://gallery.alteryx.com/api-docs/</a> and see the parameters under the 'Find workflows in a subscription' section.
<code>gallery</code>	URL for your Alteryx Gallery
<code>job</code>	An Alteryx job returned from <code>get_app_jobs</code> or <code>queue_jobs</code>

## See Also

Once a job is complete, use `get_job_output` to retrieve the results.

## Examples

```
## Not run:
# get the five most recently queued jobs for an app
request_params <- list(
  sortField = "createdate",
  limit = "5"
)

get_app_jobs(app, request_params)

# queue a job and poll the job's status until it is "Completed"
job <- queue_job(app, answers)

while(job$status != "Completed") {
  job <- get_job(job)
  Sys.sleep(2)
}
```

```
## End(Not run)
```

---

```
as.alteryx_app      Alteryx Applications
```

---

### Description

Handling of application resources returned by the Alteryx Gallery API. For more information, see [get\\_app](#)

### Usage

```
as.alteryx_app(x)

is.alteryx_app(object)
```

### Arguments

x	A list of values pertaining to an Alteryx app returned from the Alteryx Gallery API
object	An R object

---

```
as.alteryx_job      Alteryx Jobs
```

---

### Description

Handling of job information returned by the Alteryx Gallery API. For more information, see [get\\_job](#).

### Usage

```
as.alteryx_job(x, parent)

is.alteryx_job(object)
```

### Arguments

x	A list of values pertaining to an Alteryx job returned from the Alteryx Gallery API
parent	An <code>alteryx_app</code> or <code>alteryx_job</code> containing the id and name information of the parent application for a job.
object	An R object

---

build_base_string	<i>Create the base string to be signed</i>
-------------------	--

---

**Description**

Create the base string to be signed

**Usage**

```
build_base_string(request_method, base_url, normalized_request_params)
```

**Arguments**

request_method	Character vector containing an HTTP request verb
base_url	Base URL created with build_base_url
normalized_request_params	Normalized request parameters created with normalize_request_params

---

build_base_url	<i>Create and percent encode the base URL, part of the base string to be signed</i>
----------------	---

---

**Description**

Create and percent encode the base URL, part of the base string to be signed

**Usage**

```
build_base_url(gallery, endpoint)
```

**Arguments**

gallery	URL of an Alteryx Gallery
endpoint	API endpoint beginning and ending with "/"

---

build_request_url	<i>Build Request URL</i>
-------------------	--------------------------

---

**Description**

Build Request URL

**Usage**

```
build_request_url(gallery, endpoint, request_method, request_params)
```

**Arguments**

gallery	URL for your Alteryx Gallery
endpoint	The api endpoint beginning and ending with "/"
request_method	HTTP request verb
request_params	List of request parameters

---

build_signature	<i>Make Signature</i>
-----------------	-----------------------

---

**Description**

Make Signature

**Usage**

```
build_signature(gallery, endpoint, request_method, required_headers,
               request_params)
```

**Arguments**

gallery	URL of an Alteryx Gallery
endpoint	API endpoint beginning and ending with "/"
request_method	HTTP request verb
required_headers	Required headers created with build_required_headers
request_params	List of request parameters

---

check_class	<i>Error handling for incorrect object inputs to api.R calls</i>
-------------	--

---

**Description**

Error handling for incorrect object inputs to api.R calls

**Usage**

```
check_class(object, type)
```

**Arguments**

object	An R object
type	Expected class of object

---

check_keys	<i>Check Keys</i>
------------	-------------------

---

**Description**

Check Keys

**Usage**

```
check_keys()
```

---

check_status	<i>Error handling for non-200 status codes</i>
--------------	--

---

**Description**

Error handling for non-200 status codes

**Usage**

```
check_status(response)
```

**Arguments**

response	Response generated by a request such as <code>httr::GET</code> or <code>httr::POST</code>
----------	---

---

download_app	<i>Download App</i>
--------------	---------------------

---

**Description**

Download an app as a .yxzp file

**Usage**

```
download_app(app, destfile, gallery = get_gallery())
```

**Arguments**

app	A single alteryx_app returned from get_app
destfile	A character string with the directory and name of where the downloaded file is to be saved.
gallery	URL for your Alteryx Gallery

---

empty_answer	<i>Empty Answer</i>
--------------	---------------------

---

**Description**

Utility function used to queue a job for an app that has no questions

**Usage**

```
empty_answer()
```

**Examples**

```
## Not run:  
job <- queue_job(app, answer = empty_answer())  
  
## End(Not run)
```



---

encode_list	<i>Percent encode all values in a list using <code>utils::URLencode</code></i>
-------------	--

---

**Description**

Percent encode all values in a list using `utils::URLencode`

**Usage**

```
encode_list(l, reserved = TRUE, repeated = TRUE)
```

**Arguments**

<code>l</code>	A list
<code>reserved</code>	logical: should reserved characters be encoded?
<code>repeated</code>	logical: should apparently already-encoded URLs be encoded again?

---

generate_nonce	<i>Generate Nonce</i>
----------------	-----------------------

---

**Description**

Generate Nonce

**Usage**

```
generate_nonce()
```

---

generate_required_headers	<i>Generate the request headers required by Alteryx Gallery</i>
---------------------------	---

---

**Description**

Generate the request headers required by Alteryx Gallery

**Usage**

```
generate_required_headers()
```

---

`get_app`*Get Applications*

---

### Description

An Alteryx application is a workflow that has been designed to run based on user input to questions. In order to run these resources using `alterryx` you must first obtain the application that you want to run.

To search the workflows you can access from your studio (subscription), use `get_app`. Subscription is tied to API key. You cannot request workflows for any other subscription without that subscription's API keys.

### Usage

```
get_app(request_params = list(), gallery = get_gallery())
```

### Arguments

<code>request_params</code>	List of app search parameters. For more information on parameters, visit the Alteryx Gallery API documentation <a href="https://gallery.alteryx.com/api-docs/">https://gallery.alteryx.com/api-docs/</a> and see the parameters under the 'Find workflows in a subscription' section.
<code>gallery</code>	URL for your Alteryx Gallery

### Value

`get_app` returns a list of `alteryx_apps`

### WARNING

`get_app` will return all resources in the Gallery tied to your subscription including macros and workflows in addition to applications. You cannot run workflows or macros using the Alteryx Gallery API and if you try to do so you will receive an error.

### Examples

```
## Not run:
# get the five most recently uploaded apps in your studio

request_params <- list(
  packageType = "0",
  limit = "5",
  sortField = "uploaddate"
)

get_app(request_params)

## End(Not run)
```

---

get_app_questions	<i>Get App Questions</i>
-------------------	--------------------------

---

**Description**

Get the questions for the given Alteryx Analytic App. Only app workflows can be used.

**Usage**

```
get_app_questions(app, gallery = get_gallery())
```

**Arguments**

app	A single alteryx_app returned from get_app
gallery	URL for your Alteryx Gallery

**Details**

Most Alteryx apps have questions, user input that defines how the application should run. The answers to these questions need to be sent as answers when queueing an application on Gallery with queue\_job. get\_app\_questions returns the names, types, and default values of the questions for an app.

**WARNING**

Trees, Maps, and File Browse are unsupported question types. This is a limitation of Alteryx Gallery and not this API client.

---

get_gallery	<i>Get Gallery</i>
-------------	--------------------

---

**Description**

Remove '/' from the end of the Gallery URL

**Usage**

```
get_gallery()
```

---

`get_info`*Get Resource Information*

---

**Description**

Retrieves all metadata about an Alteryx resource including but not limited to upload date, version, and author

**Usage**

```
get_info(resource, full_info = FALSE)
```

**Arguments**

<code>resource</code>	An Alteryx app or job
<code>full_info</code>	If TRUE, returns all info retrieved from Gallery. The default, FALSE, removes extraneous information to make it easier to read.

**Examples**

```
## Not run:  
job <- queue_job(app, answers)  
get_info(job)  
  
## End(Not run)
```

---

`get_job_log`*Job Log*

---

**Description**

Retrieve the log messages for a "Completed" job

**Usage**

```
get_job_log(job)
```

**Arguments**

<code>job</code>	An completed Alteryx job returned from <code>get_job</code>
------------------	---

---

get_job_output	<i>Get Job Output</i>
----------------	-----------------------

---

**Description**

If an Alteryx app includes one or more output tools inside the application, the output will be available for download once the job is complete.

Use `get_job_output` to get the job outputs as a `data.frame`

**Usage**

```
get_job_output(job, gallery = get_gallery(), quiet = FALSE)
```

**Arguments**

job	An completed Alteryx job returned from <code>get_app_jobs</code>
gallery	URL for your Alteryx Gallery
quiet	Set to TRUE to ignore the warnings of skipped, invalid outputs

**Value**

A list containing a `data.frame` for each valid output of an Alteryx app. Valid means that the output can be converted as explained in the WARNING section.

**WARNING**

In order to retrieve the results for a job as a `data.frame`, the from the Alteryx app that is published to your gallery must have an output format of `csv` or `yxdb`. Otherwise, it cannot be properly converted. If you have multiple outputs, some of which cannot be converted, this function will issue a warning and skip the invalid outputs. Use `quiet = TRUE` to skip the outputs without warning.

---

get_priority	<i>Get Priority</i>
--------------	---------------------

---

**Description**

Translate Gallery priority text to appropriate value

**Usage**

```
get_priority(priority)
```

**Arguments**

priority	A value of 'low', 'medium', 'high', or 'critical' to be translated into the appropriate integer value for job priority
----------	--

---

normalize\_request\_params

*Combine, sort, and percent encode headers and param list, part of the base string to be signed*

---

### **Description**

Combine, sort, and percent encode headers and param list, part of the base string to be signed

### **Usage**

```
normalize_request_params(required_headers, request_params)
```

### **Arguments**

required\_headers

Required headers created with build\_required\_headers

request\_params List of request parameters

---

parse\_request\_response

*Parse Request Response*

---

### **Description**

Parse Request Response

### **Usage**

```
parse_request_response(response, as = "text", remove_bom = TRUE,
  parse_JSON = TRUE)
```

### **Arguments**

response Response generated by a request such as `httr::GET` or `httr::POST`

as Passed to `httr::content(as = as)`

remove\_bom Remove BOM from response content?

parse\_JSON Parse response content as JSON?

---

remove_bom	<i>Remove Byte Order Mark</i>
------------	-------------------------------

---

**Description**

Remove Byte Order Mark

**Usage**

```
remove_bom(content)
```

**Arguments**

content	Content from response generated by <code>httpr::content</code>
---------	--

---

running_jobs	<i>Queue Job for an App</i>
--------------	-----------------------------

---

**Description**

Each app has a set of questions that require an answer in order to be run. The answers to the app questions are formatted using `build_answers`. To see the required questions for an app use `get_app_questions`.

Use `queue_job` to queue a job for an app. A job is a single run of an app run according to the answers submitted.

**Usage**

```
queue_job(app, answers, priority = "low", track_job = FALSE,
  sleep = 10, timeout = 3600, gallery = get_gallery())
```

```
build_answers(name_value, ...)
```

**Arguments**

app	A single <code>alteryx_app</code> returned from <code>get_app</code>
answers	Answers to required app questions created using <code>build_answers</code>
priority	Assign a priority level to jobs to control which jobs are run by each worker, or to reserve specific workers for higher priority requests. When running a workflow, users (if enabled by the Server Admin) can select a priority level of 'low', 'medium', 'high', or 'critical' to ensure certain jobs always take priority over others. If multiple jobs are queued, jobs run in priority order starting with the highest priority.
track_job	If TRUE this function will not return a value until the job completes on Alteryx Gallery

sleep	Amount of time to wait between job polls. Ignored if track_job = FALSE
timeout	Maximum amount of time to track job. Ignored if track_job = FALSE
gallery	URL for your Alteryx Gallery
name_value	list containing an app question name and value pair
...	Additional name_value pairs

**Value**

An alteryx job with status "Queued"

**See Also**

Use get\_app to find apps to queue.

Once a job has been queued, use get\_job to poll the job and check its status.

**Examples**

```
## Not run:
first_question <- list(name = "a", value = "1")
second_question <- list(name = "b", value = "2")
answers <- build_answers(first_question, second_question)

new_job <- queue_job(app, answers)

## End(Not run)
```

---

sign_base_string	<i>Sign Base String</i>
------------------	-------------------------

---

**Description**

Sign Base String

**Usage**

```
sign_base_string(base_string)
```

**Arguments**

base\_string      Base string created with build\_base\_string



---

submit\_get\_request      *Submit GET Request*

---

**Description**

Submit GET Request

**Usage**

```
submit_get_request(gallery, endpoint, request_params, ...)
```

**Arguments**

gallery	URL for your Alteryx Gallery
endpoint	The api endpoint beginning and ending with "/"
request_params	List of request parameters
...	Additional arguments passed to parse_request_response

---

submit\_post\_request      *Submit POST Request*

---

**Description**

Submit POST Request

**Usage**

```
submit_post_request(gallery, endpoint, request_params, request_body, ...)
```

**Arguments**

gallery	URL for your Alteryx Gallery
endpoint	The api endpoint beginning and ending with "/"
request_params	List of request parameters
request_body	JSON body of request containing values for app questions
...	Additional arguments passed to parse_request_response

# Index

alterryx, 2  
alterryx-package (alterryx), 2  
app\_jobs, 3  
as.alterryx\_app, 4  
as.alterryx\_job, 4  
  
build\_answers (running\_jobs), 15  
build\_base\_string, 5  
build\_base\_url, 5  
build\_request\_url, 6  
build\_signature, 6  
  
check\_class, 7  
check\_keys, 7  
check\_status, 7  
  
download\_app, 8  
  
empty\_answer, 8  
encode\_list, 9  
  
generate\_nonce, 9  
generate\_required\_headers, 9  
get\_app, 4, 10  
get\_app\_jobs (app\_jobs), 3  
get\_app\_questions, 11  
get\_gallery, 11  
get\_info, 12  
get\_job, 4  
get\_job (app\_jobs), 3  
get\_job\_log, 12  
get\_job\_output, 13  
get\_priority, 13  
  
is.alterryx\_app (as.alterryx\_app), 4  
is.alterryx\_job (as.alterryx\_job), 4  
  
normalize\_request\_params, 14  
  
parse\_request\_response, 14  
  
queue\_job (running\_jobs), 15  
  
remove\_bom, 15  
running\_jobs, 15  
  
sign\_base\_string, 16  
submit\_get\_request, 17  
submit\_post\_request, 17