

# Package ‘clarifai’

April 12, 2017

**Title** Access to Clarifai API

**Version** 0.4.2

**Maintainer** Gaurav Sood <gsood07@gmail.com>

**Description** Get description of images from Clarifai API. For more information, see <<http://clarifai.com>>. Clarifai uses a large deep learning cloud to come up with descriptive labels of the things in an image. It also provides how confident it is about each of the labels.

**URL** <http://github.com/soodoku/clarifai>

**BugReports** <http://github.com/soodoku/clarifai/issues>

**Depends** R (>= 3.2.0)

**License** MIT + file LICENSE

**LazyData** true

**VignetteBuilder** knitr

**Imports** curl, jsonlite

**Suggests** knitr, rmarkdown, testthat

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Gaurav Sood [aut, cre]

**Repository** CRAN

**Date/Publication** 2017-04-12 13:41:15 UTC

## R topics documented:

clarifai-package . . . . .	2
clarifai_check_auth . . . . .	2
clarifai_check_results . . . . .	3
clarifai_check_token . . . . .	3
clarifai_GET . . . . .	3
clarifai_POST . . . . .	4
feedback . . . . .	4

get_color	5
get_color_urls	6
get_info	7
get_langs	8
get_token	8
get_usage	9
secret_id	10
tag_images	11
tag_image_urls	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

clarifai-package	<i>clarifai: Get Description of (Moving) Images</i>
------------------	---

---

### Description

Get Description of Images. Understand what is in the images. See <http://clarifai.com/>. Details about results of calls to the API can be found at <https://developer.clarifai.com>.

Your need credentials to use this application. If you haven't already, you can get this at <https://developer.clarifai.com/>.

### Author(s)

Gaurav Sood

### See Also

[get\\_token](#)

---

clarifai_check_auth	<i>Check if authentication information is in the environment</i>
---------------------	--

---

### Description

Check if authentication information is in the environment

### Usage

clarifai\_check\_auth()

---

clarifai\_check\_results  
*Check results*

---

**Description**

Check results

**Usage**

```
clarifai_check_results(res)
```

**Arguments**

res                results being returned by the API

---

clarifai\_check\_token    *Check if authentication token is in the environment*

---

**Description**

Check if authentication token is in the environment

**Usage**

```
clarifai_check_token()
```

---

clarifai\_GET            *GET*

---

**Description**

GET

**Usage**

```
clarifai_GET(path, query, ...)
```

**Arguments**

path                path to specific API request URL  
query               query list  
...                 Additional arguments passed to [curl\\_fetch\\_memory](#).

**Value**

list

---

clarifai_POST	<i>Base POST AND GET functions. Not exported.</i>
---------------	---

---

**Description**

POST

**Usage**

```
clarifai_POST(path, query = NULL, ...)
```

**Arguments**

path	path to specific API request URL
query	query list
...	Additional arguments passed to <a href="#">curl_fetch_memory</a> .

**Value**

list

---

feedback	<i>Provide Feedback on Tags of an Image</i>
----------	---

---

**Description**

Suggest tags, suggest removal of tags, suggest (dis)similar documents, or let clarifai know which tag was clicked on

**Usage**

```
feedback(file_path = "", feedback_type = "add_tags", feedback_value = "",
...)
```

**Arguments**

file_path	image on which the feedback is being provided
feedback_type	must be one of the following types: add_tags vector of suggested tags; default remove_tags vector of tags you suggest be removed search_click which tag(s) where clicked on similar_docids similar documents dissimilar_docids dissimilar documents
feedback_value	what tags to add, remove etc.
...	Additional arguments passed to <a href="#">clarifai_POST</a> .

## References

<https://developer.clarifai.com/guide/>

## Examples

```
## Not run:  
  
# Before calling the function, set API secret and id via secret_id(c("client_id", "secret"))  
# and get token via get_token()  
  
feedback(file_path="path_to_image", feedback_type="add_tags", feedback_value="suggested_tag")  
  
## End(Not run)
```

---

get\_color

*Get the dominant colors present in images or videos*

---

## Description

Get the dominant colors present in images or videos

## Usage

```
get_color(file_paths = NULL, ...)
```

## Arguments

file\_paths      a vector of paths to image file(s) for which you want tags  
...              Additional arguments passed to `clarifai_POST`.

## Value

Named list of length 3: `status_code`, `status_msg`, and `results`. `results` is a named list of length 12. Contains information about max. and min. image and video size allowed etc.

Prints `status_msg` by default

## References

<https://developer.clarifai.com/>

**Examples**

```
## Not run:

# Before calling the function, set API secret and id via secret_id(c("client_id", "secret"))
# and get token via get_token()

get_color(file_paths="path_to_image")

## End(Not run)
```

---

get_color_urls	<i>Get the dominant colors present in images or videos</i>
----------------	--

---

**Description**

Get the dominant colors present in images or videos

**Usage**

```
get_color_urls(img_urls = NULL, meta = FALSE, ...)
```

**Arguments**

img_urls	A string or character vector of URLs of images for which you want tags
meta	Boolean that toggles whether or not you want the entire object. Default is FALSE. When FALSE, the function returns just a data.frame of results
...	Additional arguments passed to <code>clarifai_POST</code> .

**Value**

Named list of length 3: `status_code`, `status_msg`, and `results`. `results` is a named list of length 12. Contains information about max. and min. image and video size allowed etc.

Prints `status_msg` by default

**References**

<https://developer.clarifai.com/>

**Examples**

```
## Not run:

# Before calling the function, set API secret and id via secret_id(c("client_id", "secret"))
# and get token via get_token()

get_color_urls("https://samples.clarifai.com/metro-north.jpg")
get_color_urls(c("https://samples.clarifai.com/metro-north.jpg",
"https://samples.clarifai.com/metro-north.jpg"))
```

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

---

get_info	<i>Get Information</i>
----------	------------------------

---

### Description

Basic information about the application — what are maximum sizes allowed, default language, max. and min. image and video size, max. batch size, etc.

### Usage

```
get_info(...)
```

### Arguments

... Additional arguments passed to `clarifai_GET`.

### Value

Named list of length 3: `status_code`, `status_msg`, and `results`. `results` is a named list of length 12. Contains information about max. and min. image and video size allowed etc.

Prints `status_msg` by default

### References

<https://developer.clarifai.com/>

### Examples

```
## Not run:  
  
# Before calling the function, set API secret and id via secret_id(c("client_id", "secret"))  
# and get token via get_token()  
  
get_info()  
  
## End(Not run)
```

---

`get_langs`*List all the languages that the tag API call supports*

---

**Description**

List all the languages that the tag API call supports

**Usage**

```
get_langs(...)
```

**Arguments**

... Additional arguments passed to `clarifai_GET`.

**Value**

character vector listing all the languages that tag API call supports. See `tag_images` and `codetag_image_urls`

**References**

<https://developer.clarifai.com/>

**Examples**

```
## Not run:  
  
# Before calling the function, set API secret and id via secret_id(c("client_id", "secret"))  
# and get token via get_token()  
  
get_langs()  
  
## End(Not run)
```

---

`get_token`*Get Access Token*

---

**Description**

Once you have set your Application ID and Secret via `secret_id`, get the access token.

**Usage**

```
get_token(...)
```



**Arguments**

... Additional arguments passed to [clarifai\\_POST](#).

**Value**

a list with 4 things: access\_token, token\_type, expires\_in, scope

**References**

<https://developer.clarifai.com/>

**Examples**

```
## Not run:  
  
# Before calling the function, set API secret and id via secret_id(c("client_id", "secret"))  
get_token()  
  
## End(Not run)
```

---

get_usage	<i>Get API usage for the current month and hour</i>
-----------	---

---

**Description**

Get API usage for the current month and hour

**Usage**

get\_usage(...)

**Arguments**

... Additional arguments passed to [clarifai\\_GET](#).

**Value**

named list with four items status\_code, status\_msg, results (user and application details — credits consumed, total credit)

**References**

<https://developer.clarifai.com/>

## Examples

```
## Not run:  
  
# Before calling the function, set API secret and id via secret_id(c("client_id", "secret"))  
# and get token via get_token()  
  
get_usage()  
  
## End(Not run)
```

---

secret_id	<i>Sets Application ID and Secret</i>
-----------	---------------------------------------

---

## Description

Set Client ID and Secret. Needed for interfacing with Clarifai. Run this before anything else.

## Usage

```
secret_id(appdetails = NULL, force = FALSE)
```

## Arguments

appdetails	A vector of client_id, client_secret. Get these from <a href="https://developer.clarifai.com/">https://developer.clarifai.com/</a> . Set them before you use other functions.
force	force reset client id and secret

## Details

The function looks for ClarifaiId and ClarifaiSecret in the environment. If it doesn't find them and if we don't want to force change in them, it looks for arguments. And if no arguments are passed, it asks for user to input the values.

## References

<https://developer.clarifai.com/>

## Examples

```
## Not run:  
setapp(c("client_id", "client_secret"))  
  
## End(Not run)
```

---

tag_images	<i>Tag Images on the Computer</i>
------------	-----------------------------------

---

## Description

Tag Images on the Computer

## Usage

```
tag_images(file_paths = NULL, model = NULL, language = NULL,
           meta = FALSE, simplify = TRUE, ...)
```

## Arguments

file_paths	a vector of paths to image file(s) for which you want tags
model	For topic specific answers. Optional. Can be one of the following: general-v1.3, nsfw-v1.0, weddings-v1.0, travel-v1.0, food-items-v0.1 Default is NULL
language	Return tag in another language than english. 2 digit code. Optional. See <code>list{get_langs}</code> for the supported languages and their 2 digit codes.
meta	Boolean that toggles whether or not you want the entire object.
simplify	Boolean that toggles whether or not you want a simplified data frame with each label and associated probability in a separate row. Default is TRUE.
...	Additional arguments passed to <code>clarifai_POST</code> .

The entire object returned by the API contains a lot of meta data. By default a simplified data.frame with `img_url`, associated labels, and probabilities is returned.

## Value

If meta is TRUE, a named list of length 4 with following elements is returned: `status_code` OK or not `status_msg` Successful or not `meta` Named list of 1 containing another list named `tag` Sublist `tag` has three elements: `timestamp`, `model` and `config` `results` is a data.frame of length 6 and 1 row. Column names are: `docid`, `status_code`, `status_msg`, `local_id` and a data.frame named `tag` which has a data.frame result which contains two columns: labels and probabilities

If meta is FALSE and simplify is TRUE, a data.frame with three columns: `file_paths`, labels and probs returned

If meta is FALSE and simplify is FALSE, a data.frame with two columns carrying a vector of labels, vector of probs is returned for each image

## References

<https://developer.clarifai.com/>

## See Also

[tag\\_image\\_urls](#)

**Examples**

```
## Not run:

# Before calling the function, set API secret and id via secret_id(c("client_id", "secret"))
# and get token via get_token()

tag_images(file_paths="path_to_image")

## End(Not run)
```

---

tag_image_urls	<i>Tag Remote Images</i>
----------------	--------------------------

---

**Description**

Tag Remote Images

**Usage**

```
tag_image_urls(img_urls = NULL, model = NULL, language = NULL,
               meta = FALSE, simplify = TRUE, ...)
```

**Arguments**

img_urls	A string or character vector of URLs of images for which you want tags
model	For topic specific answers. Optional. Can be one of the following: general-v1.3, nsfw-v1.0, weddings-v1.0, travel-v1.0, food-items-v0.1 Default is NULL
language	Return tag in another language than english. 2 digit code. Optional. See <code>list{get_langs}</code> for the supported languages and their 2 digit codes.
meta	Boolean that toggles whether or not you want the entire object.
simplify	Boolean that toggles whether or not you want a simplified data frame with each label and associated probability in a separate row. Default is TRUE.
...	Additional arguments passed to <code>clarifai_POST</code> .

The entire object returned by the API contains a lot of meta data. By default a simplified data.frame with `img_url`, associated labels, and probabilities is returned.

**Value**

If `meta` is TRUE, a named list of length 4 with following elements is returned: `status_code` OK or not `status_msg` Successful or not `meta` Named list of 1 containing another list named `tag` Sublist `tag` has three elements: `timestamp`, `model` and `config` `results` is a data.frame of length 6 and 1 row. Column names are: `docid`, `status_code`, `status_msg`, `local_id` and a data.frame named `tag` which has a data.frame result which contains two columns: `labels` and `probabilities`

If `meta` is FALSE and `simplify` is TRUE, a data.frame with three columns: `img_urls`, `labels` and `probs` returned

If `meta` is `FALSE` and `simplify` is `FALSE`, a `data.frame` with two columns carrying a vector of labels, vector of probs is returned for each image

**References**

<https://developer.clarifai.com/>

**See Also**

[tag\\_images](#)

**Examples**

```
## Not run:  
  
# Before calling the function, set API secret and id via secret_id(c("client_id", "secret"))  
# and get token via get_token()  
  
tag_image_urls(img_urls="url_of_image")  
tag_image_urls("https://samples.clarifai.com/metro-north.jpg")  
tag_image_urls("https://samples.clarifai.com/metro-north.jpg", language="es")  
tag_image_urls("https://samples.clarifai.com/nsfw.jpg", model="nsfw-v1.0")  
  
## End(Not run)
```

# Index

\*Topic **Client**

secret\_id, [10](#)

\*Topic **ID**

secret\_id, [10](#)

\*Topic **Secret**

secret\_id, [10](#)

\*Topic **Sets**

secret\_id, [10](#)

\*Topic **and**

secret\_id, [10](#)

clarifai (clarifai-package), [2](#)

clarifai-package, [2](#)

clarifai\_check\_auth, [2](#)

clarifai\_check\_results, [3](#)

clarifai\_check\_token, [3](#)

clarifai\_GET, [3](#), [7-9](#)

clarifai\_POST, [4](#), [4](#), [5](#), [6](#), [9](#), [11](#), [12](#)

curl\_fetch\_memory, [3](#), [4](#)

feedback, [4](#)

get\_color, [5](#)

get\_color\_urls, [6](#)

get\_info, [7](#)

get\_langs, [8](#)

get\_token, [2](#), [8](#)

get\_usage, [9](#)

secret\_id, [8](#), [10](#)

tag\_image\_urls, [8](#), [11](#), [12](#)

tag\_images, [8](#), [11](#), [13](#)