

Package ‘Ohmage’

February 19, 2015

Type Package

Title R Client for Ohmage 2 server

Version 2.11-4

Date 2014-09-10

Author Jeroen Ooms

Maintainer Jeroen Ooms <jeroen.ooms@stat.ucla.edu>

Description R Client for Ohmage 2 server. Implements basic R functions to retrieve and process data.

License GPL

Depends R (>= 2.14), stats, methods

Imports RCurl, RJSONIO

Suggests XML

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-10 15:14:12

R topics documented:

loadtest	2
loadtest.wipe	3
multifactor	3
oh.campaign.create	4
oh.campaign.delete	4
oh.campaign.read	5
oh.class.create	5
oh.class.delete	6
oh.class.read	6
oh.class.update	6
oh.image.read	7
oh.login	7
oh.logout	8

oh.mobility.read	8
oh.survey.upload	9
oh.survey_response.function.read	9
oh.survey_response.read	10
oh.survey_response.update	11
oh.user.auth	11
oh.user.create	12
oh.user.delete	12
oh.user.read	13

Index	14
--------------	-----------

loadtest	<i>Data generations for loadtesting</i>
----------	---

Description

Data generations for loadtesting

Usage

```
loadtest(n.users = 10, n.days = 5, n.responses = 2,
  xmlfile = system.file(package = "Ohmage", "files/jeroen.xml"),
  recycle = TRUE, verbose = FALSE, shareall = TRUE,
  user.prefix = "loadtest")
```

Arguments

n.users	number of users to generate
n.days	number of days per user
n.responses	number of responses per day
xmlfile	xml file or string
recycle	if https connection should be kept alive (recommended)
verbose	verbose output
shareall	should data be shared? T/F
user.prefix	prefix used for campaign and user names.

loadtest.wipe	<i>Delete campaign generated by loadtest function</i>
---------------	---

Description

Delete campaign generated by loadtest function

Usage

```
loadtest.wipe(n.users = 10, xmlfile = system.file(package = "Ohmage",
  "files/jeroen.xml"), recycle = TRUE, user.prefix = "loadtest")
```

Arguments

n.users	number of users that was generated
xmlfile	path or string of original xml
recycle	to recycle the connection
user.prefix	a character string used in the usernames and campaign name. Make it short.

multifactor	<i>Multifactor is a datastructure for survey items in the form of 'check all that apply'. Every response has multiple values.</i>
-------------	---

Description

Multifactor is a datastructure for survey items in the form of 'check all that apply'. Every response has multiple values.

Usage

```
multifactor(values, levels = unique(unlist(values)), labels = levels,
  ordered = TRUE)
```

Arguments

values	a list of vectors with response values
levels	a vector with possible values
labels	a vector with labels
ordered	ordered or unordered factor

Value

a multifactor object

oh.campaign.create *Create a new campaign*

Description

Create a new campaign

Usage

```
oh.campaign.create(xml, running_state = "running", privacy_state = "shared",
  class_urn_list = "", description = "My campaign.", ...)
```

Arguments

xml	campaign.xml
running_state	if campaign is running or not
privacy_state	private or shared
class_urn_list	classes to add
description	a description
...	other stuff passed to the server

oh.campaign.delete *Delete a campaign from the server*

Description

Delete a campaign from the server

Usage

```
oh.campaign.delete(campaign_urn, ...)
```

Arguments

campaign_urn	campaign uuid
...	other arguments passed to ohmage.

oh.campaign.read *List campaigns available on the server*

Description

List campaigns available on the server

Usage

```
oh.campaign.read(output_format = "short", to.data.frame = TRUE, ...)
```

Arguments

output_format short or long
to.data.frame should response be converted to a dataframe
... other arguments passed to ohmage

oh.class.create *Create a new class*

Description

Create a new class

Usage

```
oh.class.create(class_urn, class_name, description = "My new class.", ...)
```

Arguments

class_urn class id
class_name class name
description class description
... other arguments passed to ohmage

oh.class.delete *Delete a class from the server*

Description

Delete a class from the server

Usage

oh.class.delete(class_urn, ...)

Arguments

class_urn	class id
...	other args passed to ohmage

oh.class.read *Read info about a class*

Description

Read info about a class

Usage

oh.class.read(class_urn_list = "urn:sys:andwellness", ...)

Arguments

class_urn_list	list of class id's
...	other arguments passed to ohmage

oh.class.update *Update a class*

Description

Update a class

Usage

oh.class.update(class_urn, ...)

Arguments

class_urn	class id
...	arguments passed to ohmage

oh.image.read	<i>Download an image</i>
---------------	--------------------------

Description

Download an image

Usage

```
oh.image.read(campaign_urn, owner, id, ...)
```

Arguments

campaign_urn	campaign id
owner	owner of the image
id	photo uuid
...	other parameters

Value

path to image

oh.login	<i>Authenticate with an ohmage server</i>
----------	---

Description

Authenticate with an ohmage server

Usage

```
oh.login(user, password, serverurl, ...)
```

Arguments

user	ohmage username
password	ohmage passwd
serverurl	url to the ohmage server
...	extra parameters for oh.call

Examples

```

library(Ohmage)
## Not run:
#authentication works like a cookie.
#oh.login("ohmage.admin", "ohmage.passwd", "https://example.com/app")

#list campaigns you are in
#oh.campaign.read()

#read some data
#oh.survey_response.read("urn:ohmage:campaign:mycampaign");

## End(Not run)

```

oh.logout	<i>Logout from an ohmage session</i>
-----------	--------------------------------------

Description

Logout from an ohmage session

Usage

```
oh.logout()
```

oh.mobility.read	<i>Read Mobility data</i>
------------------	---------------------------

Description

Read Mobility data

Usage

```
oh.mobility.read(date = today(), username = getOption("ohmage_username"),
  column_list, ...)
```

Arguments

date	date in ISO format
username	name of user to query (only works when server allows to see others data).
column_list	variables to be returned
...	stuff passed to oh.call

Value

a dataframe with mobility data

oh.survey.upload *Upload a survey response*

Description

Upload a survey response

Usage

```
oh.survey.upload(campaign_urn, user, password, surveys,  
                 campaign_creation_timestamp, ...)
```

Arguments

campaign_urn	campaign id
user	username
password	password
surveys	surveys
campaign_creation_timestamp	ISO 8601
...	other arguments passed to ohmage

Value

data frame with responses

oh.survey_response.function.read
Survey Response Function Read

Description

Allows users with the appropriate permissions to read aggregate data about survey responses. The API takes an id parameter that tells the system which function to perform on the data.

Usage

```
oh.survey_response.function.read(campaign_urn, id = "privacy_state",  
                                privacy_state = NULL, privacy_state_item_list = "date,survey", ...)
```

Arguments

campaign_urn	campaign urn
id	Function id. The currently supported id is privacy_state
privacy_state	depricated. Should be NULL.
privacy_state_item_list	comma seperated list of output
...	other stuff passed to oh.call

Value

a list

oh.survey_response.read
Read survey responses

Description

Read survey responses

Usage

```
oh.survey_response.read(campaign_urn,
prompt_id_list = "urn:ohmage:special:all", privacy_state = "both",
column_list="urn:ohmage:user:id,urn:ohmage:prompt:response,urn:ohmage:context:timestamp",
output_format = "json-columns", user_list = "urn:ohmage:special:all",
to.data.frame = TRUE, ...)
```

Arguments

campaign_urn	campaign id
prompt_id_list	comma seperated list of prompt id's
privacy_state	shared or private
column_list	ohmage columns
output_format	should be json-columns
user_list	list of users
to.data.frame	if data should be converted to a dataframe
...	other args passed to ohmage

`oh.survey_response.update`*Update the privacy status of a survey response*

Description

Update the privacy status of a survey response

Usage

```
oh.survey_response.update(campaign_urn, survey_key, privacy_state = "shared",
...)
```

Arguments

<code>campaign_urn</code>	campaign id
<code>survey_key</code>	survey id
<code>privacy_state</code>	Must be one of shared or private.
<code>...</code>	other arguments passed to oh.call

`oh.user.auth`*Retrieve hashed passwd for a user*

Description

Retrieve hashed passwd for a user

Usage

```
oh.user.auth(user, password, ...)
```

Arguments

<code>user</code>	username
<code>password</code>	passwd
<code>...</code>	other argus

Value

pw hash

oh.user.create *Create a new user*

Description

Create a new user

Usage

```
oh.user.create(username, password, admin = FALSE, enabled = TRUE,  
               new_account = FALSE, campaign_creation_privilege = FALSE, ...)
```

Arguments

username	username
password	password
admin	if the user should be an admin
enabled	if new account should be enabled
new_account	user marked as new_account have to change their passwd
campaign_creation_privilege	if user can create campaigns
...	other arguments passed on to ohmage

oh.user.delete *Delete users from the server*

Description

Delete users from the server

Usage

```
oh.user.delete(user_list, ...)
```

Arguments

user_list	comma seperated list of users
...	other arguments passed on to ohmage

oh.user.read	<i>Read info about a user</i>
--------------	-------------------------------

Description

Read info about a user

Usage

oh.user.read(user_list, ...)

Arguments

user_list	username
...	other args passed to ohmage

Index

[.multifactor (multifactor), 3
[[.multifactor (multifactor), 3

as.vector.multifactor (multifactor), 3

expand.multifactor (multifactor), 3

facdim(multifactor), 3

is.multifactor (multifactor), 3

loadtest, 2
loadtest.wipe, 3

multifactor, 3

oh.campaign.create, 4
oh.campaign.delete, 4
oh.campaign.read, 5
oh.class.create, 5
oh.class.delete, 6
oh.class.read, 6
oh.class.update, 6
oh.image.read, 7
oh.login, 7
oh.logout, 8
oh.mobility.read, 8
oh.survey.upload, 9
oh.survey_response.function.read, 9
oh.survey_response.read, 10
oh.survey_response.update, 11
oh.user.auth, 11
oh.user.create, 12
oh.user.delete, 12
oh.user.read, 13

rep.multifactor (multifactor), 3