Package ‘RZabbix’

April 26, 2016

Version 0.1.0
Type Package
Title R Module for Working with the 'Zabbix API'
Description R interface to the 'Zabbix API' data <https://www.zabbix.com/documentation/3.0/manual/api/reference>. Enables easy and direct communication with 'Zabbix API' from 'R'.
Repository CRAN
License GPL-2
LazyLoad yes
LazyData yes
Depends R (>= 3.2.0)
Imports httr (>= 1.1.0), jsonlite (>= 0.9.19)
BugReports https://github.com/MarcinKosinski/RZabbix/issues
RoxygenNote 5.0.1
NeedsCompilation no
Author Marcin Kosinski [aut, cre]
Maintainer Marcin Kosinski <m.p.kosinski@gmail.com>
Date/Publication 2016-04-26 08:44:09

R topics documented:

ZabbixAPI ................................................................. 2

Index 5
Description
ZabbixAPI function enables easy and direct communication with Zabbix API from R. Each request to the Zabbix API requires passing auth authentication token. Generation of the authentication token is explained here or can be seen in Examples section.

Usage
ZabbixAPI(server = "http://localhost/zabbix", body = list(),
user.agent = "RZabbix", content.type = "application/json-rpc",
encode = "json", ..., only.content = TRUE, fromJSON = TRUE,
content.only.result = TRUE)

Arguments
server A character with base URI for zabbix web interface (omitting /api_jsonrpc.php).
body A named list specifying method, user auth token and additional params (passed in jsonlite::unbox(data.frame())) to be used during request. It can also be passed as JSON through character with fromJSON('json_string') - see Examples. Available methods are described in Zabbix Documentation. One does not have to pass id and jsonrpc.
user.agent, content.type Character arguments passed to POST through user_agent and content_type.
encode The same as encode in POST.
... Further arguments passed to POST.
only.content, fromJSON, content.only.result
• only.content Should return results only for content part of the response.
• fromJSON Should apply fromJSON conversion for content when only.content = TRUE.
• content.only.result Should return only result part of fromJSON(rawToChar(response$content)) when only.content = TRUE, fromJSON = TRUE.

Details
Communication with the Zabbix API is described at Zabbix API Manual Reference and relies on specifying methods of call and additional parameters in the JSON-RPC format. An example of such specification can be found in the example of authenticating a user but in RZabbix this should be specified in parameter body where parameters are passed in the named list, where additional params should be passed in jsonlite::unbox(data.frame()). See examples.

Note
Bug reports and feature requests can be sent to https://github.com/MarcinKosinski/RZabbix/issues
Author(s)

Marcin Kosinski, <m.p.kosinski@gmail.com>

Examples

```r
## Not run:
# user authentication

ZabbixAPI('http://localhost/zabbix',
  body = list(method = "user.login",
     params = jsonlite::unbox(
       data.frame(user = "Admin",
                     password = "zabbix"))))) -> auth

## request to get history of an item of 'item_id' number

ZabbixAPI('http://localhost/zabbix',
  body = list(method = "history.get",
     params = jsonlite::unbox(
       data.frame(output = "extend",
                    itemids = "item_id",
                    history = 0,
                    sortfield = "clock",
                    sortorder = "DESC",
                    limit = 10,
                    auth = auth))

## API info

ZabbixAPI('http://localhost/zabbix',
  body = list(method = "apiinfo.version"))

## fromJSON example for get event data for object with 'object_id' number

library(jsonlite)
paste0('{
  "method": "event.get",
  "params": {
    "output": "extend",
    "select_acknowledges": "extend",
    "objectids": "object_id",
    "sortfield": ["clock", "eventid"],
    "sortorder": "DESC"
  },
  "auth": "", auth, ""
}
') -> json_rpc

ZabbixAPI('http://localhost/zabbix',
  body = fromJSON(json_rpc)) -> event.info

# colnames - https://www.zabbix.com/documentation/3.0/manual/api/reference/event/object
```
event.info %>%
  select(value, clock) %>%
  mutate(clock =
    as.POSIXct(as.numeric(clock),
      tz = "GMT",
      origin = "1970-01-01") -> clock2viz

list2bind <- list()
for(i in 1:(nrow(clock2viz)-1)) {
  data.frame(
    times =
      seq(from = clock2viz$clock[i+1],
      to = clock2viz$clock[i],
      by = "min") %>%
    head(-1),
    status = clock2viz$value[i+1],
    stringsAsFactors = FALSE) ->
  list2bind[[i]]
}

library(ggplot2)
do.call(bind_rows, list2bind) %>%
  ggplot(aes(x=times, y = status)) +
  geom_point(size = 0.1)

## End(Not run)
Index

ccontent_type, 2
fromJSON, 2
POST, 2
response, 2
user_agent, 2
ZabbixAPI, 2