

Package: bambooHR (via r-universe)

September 12, 2024

Type Package

Title A Wrapper to the 'BambooHR' API

Version 0.1.1

Maintainer Tom Bowling <tom.bowling@ascent.io>

Description Enables a user to consume the 'BambooHR' API endpoints using R. The actual URL of the API will depend on your company domain, and will be handled by the package automatically once you setup the config file. The API documentation can be found here <<https://documentation.bamboohr.com/docs>>.

Depends R (>= 4.0.0)

Imports rlang, glue, httr, curl, jsonlite, magrittr, dplyr, purrr, tibble, tidyr, lubridate, janitor, stringr, cli, withr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

URL <https://mangothecat.github.io/bambooHR/>

BugReports <https://github.com/MangoTheCat/bambooHR/issues>

Suggests httpptest, testthat

Config/testthat/edition 3

Repository <https://mangothecat.r-universe.dev>

RemoteUrl <https://github.com/mangothecat/bamboohr>

RemoteRef HEAD

RemoteSha a1671b5cfea969513469370e6955578456694f7c

Contents

build_url	2
config_setup	3

get_company_file	3
get_employee	4
get_employee_file	5
get_meta	6
get_report	7
get_request	8
get_table	9
get_timeoff_policies	10
get_timeoff_requests	10
get_timeoff_types	12
get_whos_out	12
response_view	13
use_config	14

Index	15
--------------	-----------

build_url	<i>Build URL Endpoints with Company Domain and API Version</i>
-----------	--

Description

Used to generate the base API request URL. URLs will start with "https://api.bamboohr.com/api/gateway.php/company_domain" which forms the base URL where a query string can then be appended to make the full request URL.

Usage

```
build_url(
    company_domain = .get_company_name(),
    api_version = "v1",
    base_url = "https://api.bamboohr.com/api/gateway.php"
)
```

Arguments

company_domain	The subdomain used to access BambooHR. If you access BambooHR at https://mycompany.bamboohr.com then the company_domain is "mycompany". Default value is obtained from options("bamboohr.company_name"). This can also be set in the config file using config_setup("api key", "company_domain").
api_version	Version of API to use to make request. Default is "v1".
base_url	URL to BambooHR API. Default is "https://api.bamboohr.com/api/gateway.php".

Value

character vector of URL with API version and company domain to give the base URL for API requests.

config_setup	<i>Create a JSON config file</i>
--------------	----------------------------------

Description

Create a JSON config file

Usage

```
config_setup(apikey, companyname, conffile, verbose = TRUE)
```

Arguments

apikey	API key to access the bambooHR API. User will be prompted to enter one if not supplied
companyname	This is the subdomain used to access BambooHR. If you access BambooHR at https://mycompany.bamboohr.com , then companyname will be "mycompany"
conffile	A string giving the path where the configuration file will be written. This will automatically attempt to load from the default location <code>~/ .bambooHR_user_config.json</code> (which can be changed via a <code>bambooHR.config_file</code> entry in options).
verbose	whether to print path of conf file used, passed to <code>use_config</code>

Examples

```
## Not run:  
config_setup("secret api key", "cool company")  
  
## End(Not run)
```

get_company_file	<i>Retrieve A Company File.</i>
------------------	---------------------------------

Description

'get_company_file' takes a `file_id` (string) and `api version` (string) as arguments and then requests and returns data about the corresponding file from the BambooHR API, the file is then downloaded if possible.

Usage

```
get_company_file(file_id = "view", api_version = "v1", suppress_view = FALSE)
```

Arguments

`file_id` The ID of the file to get from BambooHR.
`api_version` version of the BambooHR API.
`suppress_view` prevent display of results when `file_id = "view"`, default is FALSE.

Value

returns a response object.

Author(s)

Harry Alexander, <harry.alexander@ascent.io>

Examples

```
## Not run:  
response <- get_company_file(  
  "480",  
  api_version = "v1"  
)  
  
## End(Not run)
```

get_employee	<i>Get employee data by specifying a set of fields.</i>
--------------	---

Description

This is suitable for getting basic employee information, including current values for fields that are part of a historical table, like job title, or compensation information. See the fields endpoint for a list of possible fields.

Usage

```
get_employee(  
  id = "directory",  
  fields = NULL,  
  only_current = FALSE,  
  verbose = FALSE,  
  api_version = "v1"  
)
```

Arguments

id	Employee IDs (character or numeric). The default of "directory" will give a directory of all employees, along with a table of possible fields. The special employee ID of zero (0) means to use the employee ID associated with the API key (if any).
fields	Character vector of fields to return. To see all available fields, see the column names of get_employee().
only_current	Setting to false will return future dated values from history table fields.
verbose	Logical, default FALSE. Whether to return the fields dataframe when [id = "directory"].
api_version	(optional) - Version of API to use to make request. Default is "v1".

Value

Tibble (dataframe), unless [verbose = TRUE] and [id = "directory"], in which case a list with the fields dataframe and the directory dataframe.

Examples

```
## Not run:
get_employee()
get_employee(0, fields = c("firstName", "lastName"))
get_employee(c(0, 1, 2), fields = c("firstName", "lastName", "jobTitle"))

## End(Not run)
```

get_employee_file	<i>Retrieve An Employee File</i>
-------------------	----------------------------------

Description

'get_employee_file' takes 'id' (string), 'file_id', (string), and 'api_version' (string) and then requests and returns data about the corresponding file from the BambooHR API. The file will then be written to the user's working directory if possible.

Usage

```
get_employee_file(
  id = "0",
  file_id = "view",
  api_version = "v1",
  suppress_view = FALSE
)
```

Arguments

id The employee id of the employee.
file_id The file id of the file to be returned.
api_version The version of BambooHR API to be used.
suppress_view prevent display of results when file_id = "view", default is FALSE.

Value

returns a response object.

Author(s)

Harry Alexander, <harry.alexander@ascent.io>

Examples

```
## Not run:  
response <- get_employee_file(  
  id = 0,  
  file_id = "480",  
  api_version = "v1"  
)  
  
## End(Not run)
```

get_meta

Get Meta Information

Description

Submits a get request to retrieve account meta information that can be used to help discover the types of field/table etc. available in your account.

Usage

```
get_meta(  
  query = c("fields", "tables", "lists", "users", "time_off/types", "time_off/policies"),  
  api_version = "v1"  
)
```

Arguments

query One of: "fields", "tables", "lists", "users", "time_off/types", "time_off/policies".
api_version (optional) - Version of API to use to make request. Default is "v1".

Value

A [tibble::tibble()] object.

References

<https://documentation.bamboohr.com/reference/account-information-1>

Examples

```
## Not run:
res <- get_meta("fields")

res2 <- get_meta("users")

## End(Not run)
```

get_report	<i>Get Company Report</i>
------------	---------------------------

Description

Request one of your pre-existing company reports from the reporting section. At present, only reports from the Company Reports section are supported.

Usage

```
get_report(
  id,
  format = c("CSV", "PDF", "XLS", "XML", "JSON"),
  field_filtering = TRUE,
  only_current = TRUE,
  api_version = "v1"
)
```

Arguments

id	The report ID. You can get the report number by hovering over the report name on the reports page and grabbing the ID.
format	The output format for the report. Supported formats: CSV, PDF, XLS, XML, JSON.
field_filtering	Logical, default [TRUE]. Whether to apply duplicate field filtering.
only_current	Logical default [TRUE]. Should the report be limited to only current employees.
api_version	(optional) - Version of API to use to make request. Default is "v1".

Value

Currently the function simply returns a parsed response object.

get_request	<i>Bamboo HR API get request wrapper</i>
-------------	--

Description

Submits get requests to the Bamboo API.

Usage

```
get_request(url, key = .get_api_key(), verbose = FALSE)
```

Arguments

url	Character url of the get request.
key	Character api key for bambooHR. Register in your account, see authentication for details. Default value is obtained from <code>options("bambooHR.api_key")</code> . This can also be set in the config file using <code>config_setup("api key", "company domain")</code> .
verbose	Logical. Should the function provide verbose messaging back on each step?

Value

A `httr::response()` object.

Examples

```
## Not run:  
res <- get_request("https://api.bamboohr.com/api/gateway.php/your_company/v1/employees/0/")  
httr::content(res, "text")  
  
## End(Not run)
```

`get_table`*Bamboo API get request wrapper for BambooHR tables*

Description

Returns a data structure representing all the table rows for a given employee and table combination.

Usage

```
get_table(employee_id, table_name, api_version = "v1")
```

Arguments

<code>employee_id</code>	Character of employee ID to return. A special employee ID of "all", can also be used which will cause the API to return all rows for all employees in the table. When specifying "all" employees, the result will include inactive and terminated employees. In addition, the special employee ID of 0 is the employee ID associated with the API key being used.
<code>table_name</code>	Character of table name. Valid table names can be found in the "parent_alias" column return by <code>get_meta("tables")</code> .
<code>api_version</code>	(optional) - Version of API to use to make request. Default is "v1".

Value

A `tibble::tibble()` object.

References

<https://documentation.bamboohr.com/reference#get-employee-table-row-1>

Examples

```
## Not run:  
get_table(employee_id = "all", table_name = "jobInfo")  
  
get_table(employee_id = 0, table_name = "jobInfo")  
  
get_table(0, "compensation")  
  
## End(Not run)
```

get_timeoff_policies *Get Time Off policies*

Description

Wrapper to get_meta function that returns a list of time off policies.

Usage

```
get_timeoff_policies()
```

Value

A `httr::response()` object where the content is a JSON format that contains the list of time off policies.

Note

Currently do not have the permissions to test this so returning the raw response object

References

<https://documentation.bamboohr.com/reference/get-time-off-policies>

Examples

```
## Not run:  
res <- get_timeoff_policies()  
httr::content(res, "parsed")  
  
## End(Not run)
```

get_timeoff_requests *Get Time Off Requests*

Description

Get Time Off Requests

Usage

```
get_timeoff_requests(
  start,
  end,
  id = NULL,
  action = c("view", "approve"),
  employee_id = NULL,
  type = NULL,
  status = c("approved", "denied", "superseded", "requested", "canceled"),
  api_version = "v1"
)
```

Arguments

start	Character in format "YYYY-MM-DD". Only show time off that occurs on/after the specified start date.
end	Character in format "YYYY-MM-DD". Only show time off that occurs on/before the specified end date.
id	(optional) Integer - A particular request ID to limit the response to.
action	(optional) - Limit to requests that the user has a particular level of access to. Legal values are: "view" or "approve".
employee_id	(optional) Character - A particular employee ID to limit the response to.
type	(optional) - A vector of time off types IDs to include limit the response to. Default is all types are included.
status	(optional) - A vector of request status values to include. Legal values are "approved", "denied", "superseded", "requested", "canceled". Default is all status types.
api_version	(optional) - Version of API to use to make request. Default is "v1".

Value

A `tibble::tibble()` object.

References

<https://documentation.bamboohr.com/reference/time-off-1>

Examples

```
## Not run:
# find valid valid types
types <- get_timeoff_types()
type_ids <- types %>% tidyr::drop_na(id) %>% dplyr::pull(id)

res <- get_timeoff_requests("2022-01-01", "2022-02-01", type = type_ids)

res2 <- get_timeoff_requests("2022-01-01", "2022-02-01", action = "approve",
```

```
status = c("approved", "denied"))

res3 <- get_timeoff_requests("2022-01-01", "2022-02-01", employee_id = "4")

## End(Not run)
```

get_timeoff_types *Get Time Off Types*

Description

Wrapper to get_meta function that returns time off types

Usage

```
get_timeoff_types()
```

Value

A vector of type ids

References

<https://documentation.bamboohr.com/reference/get-time-off-types>

Examples

```
## Not run:
types <- get_timeoff_types()
type_ids <- types %>% tidyr::drop_na(id) %>% dplyr::pull(id)

## End(Not run)
```

get_whos_out *Get a list of Who's Out*

Description

Get a list of Who's Out

Usage

```
get_whos_out(start = "", end = "", api_version = "v1")
```

Arguments

- start (optional) - a date in the form YYYY-MM-DD - defaults to the current date.
- end (optional) - a date in the form YYYY-MM-DD - defaults to 14 days from the start date.
- api_version (optional) - Version of API to use to make request. Default is "v1".

Value

A `tibble::tibble()` object.

References

<https://documentation.bamboohr.com/reference/get-a-list-of-whos-out-1>

Examples

```
## Not run:
res <- get_whos_out()

res2 <- get_whos_out(start = "2022-01-12")

res3 <- get_whos_out(start = "2022-01-01", end = "2022-04-01")

## End(Not run)
```

response_view	<i>View Data About a Response.</i>
---------------	------------------------------------

Description

This function uses the ‘View’ function from Rstudio if available, otherwise it uses the `utils::View` function.

Usage

```
response_view(x, title)
```

Arguments

- x An R object which can be coerced to a data frame with non-zero numbers of rows and columns.
- title Title for the viewer window.

Value

NULL.

`use_config`*Use Config File*

Description

Choose a specific configuration file to supply the API key and the company domain.

Usage

```
use_config(config = "~/bambooHR_user_config.json", verbose = TRUE)
```

Arguments

<code>config</code>	Filepath to the config file.
<code>verbose</code>	Print location of config file being used, default = TRUE

Details

In rare circumstances, it may be useful to swap between multiple configuration files, for example when comparing results on different accounts possibly with different permissions. In this case a user may set-up two config files with `[config_setup]` in different locations (see `[conffile]` argument), and swap between them with `[use_config]`. The chosen config file will then be used for subsequent API calls, as `[use_config]` sets global options and variables in the package environment. Of course, `[config_setup]` does all of this too, however `[use_config]` does not expose the API key when called.

Value

[NULL] invisibly.

Examples

```
## Not run:
# Setup 2 configuration files
config_setup(apikey = "key", companyname = "Ascent", conffile = "example.json")
config_setup(apikey = "key2", companyname = "Descent", conffile = "example2.json")

# Jump between them
use_config("example.json")
use_config("example2.json")

# Remove them when finished
unlink("example.json")
unlink("example2.json")

## End(Not run)
```

Index

`build_url`, [2](#)

`config_setup`, [3](#)

`get_company_file`, [3](#)

`get_employee`, [4](#)

`get_employee_file`, [5](#)

`get_meta`, [6](#)

`get_report`, [7](#)

`get_request`, [8](#)

`get_table`, [9](#)

`get_timeoff_policies`, [10](#)

`get_timeoff_requests`, [10](#)

`get_timeoff_types`, [12](#)

`get_whos_out`, [12](#)

`httr::response()`, [8](#), [10](#)

`response_view`, [13](#)

`tibble::tibble()`, [9](#), [11](#), [13](#)

`use_config`, [14](#)