Package ‘genesysr’

October 13, 2022

Version 1.0.0
Title Genesys PGR Client
Description Access data on plant genetic resources from genebanks around the world published on Genesys (<https://www.genesys-pgr.org>). Your use of data is subject to terms and conditions available at <https://www.genesys-pgr.org/content/legal/terms>.
Maintainer Matija Obreza <matija.obreza@croptrust.org>
Depends R (>= 3.1.0)
Imports httr, jsonlite, dplyr
License Apache License 2.0
RoxygenNote 6.1.1
URL https://gitlab.croptrust.org/genesys-pgr/genesysr
BugReports https://gitlab.croptrust.org/genesys-pgr/genesysr/issues
Suggests knitr, rmarkdown, tidyverse
VignetteBuilder knitr
NeedsCompilation no
Author Global Crop Diversity Trust [cph], Matija Obreza [aut, cre], Nora Castaneda [ctb]
Repository CRAN
Date/Publication 2019-11-22 09:00:02 UTC

R topics documented:

api1_url ................................................................. 2
api2_url ................................................................. 3
authorization ............................................................. 3
check_country .......................................................... 4
check_landorsea ......................................................... 4
check_taxonomy ......................................................... 5
api1_url

Get full Genesys API v1 URL for a specific path

Description

Get full Genesys API v1 URL for a specific path

Usage

api1_url(path)

Arguments

path relative path of the API v1 endpoint (e.g. /me)

Value

Absolute URL to an API call

Examples

api1_url("/me")
**api2_url**

Get full Genesys API v1 URL for a specific path

**Description**
Get full Genesys API v1 URL for a specific path

**Usage**

api2_url(path)

**Arguments**

- **path** relative path of the API v1 endpoint (e.g. /me)

**Value**
Absolute URL to an API call

**Examples**

api2_url("/me")

---

**authorization**

Provide OAuth2 token to use for authorization with Genesys

**Description**
Provide OAuth2 token to use for authorization with Genesys

**Usage**

authorization(authorization)

**Arguments**

- **authorization** OAuth2 Authorization header obtained from somewhere else (e.g. an ENV variable)

**See Also**

user_login, client_login
**check_landorsea**

Run Land-or-Sea check on MCPD data using https://validator.genesys-pgr.org. Uploads only rows where DECLATITUDE and DECLONGITUDE are provided. In practice it is better to use `check_country` if ORIGCTY data exists.

**Description**

Run Land-or-Sea check on MCPD data using https://validator.genesys-pgr.org. Uploads only rows where DECLATITUDE and DECLONGITUDE are provided. In practice it is better to use `check_country` if ORIGCTY data exists.

**Usage**

```r
check_landorsea(mcpd)
```

**Arguments**

- `mcpd` Accession passport data in MCPD format

**Value**

Results from validator

**Examples**

```r
## Not run:
geoCheck <- genesysr::check_landorsea(mcpd)
## End(Not run)
```
check_taxonomy

Arguments

mcpd  
Accession passport data in MCPD format

Value

Results from validator

Examples

```r
## Not run:
waterCheck <- genesysr::check_landorsea(mcpd)

## End(Not run)
```

---

**check_taxonomy**  
*Check MCPD taxonomic data (GENUS, SPECIES, SPAUTHOR, SUB-TAXA, SUBTAUTHOR) using https://validator.genesys-pgr.org.*

**Description**

Duplicate input rows are removed using dplyr::distinct() and results are returned for unique rows.

**Usage**

```r
check_taxonomy(mcpd, toCurrentTaxa = FALSE)
```

**Arguments**

mcpd  
Accession passport data in MCPD format

toCurrentTaxa  
Should obsoleted names be reported?

**Value**

Results from validator

**Examples**

```r
## Not run:
taxaCheck <- genesysr::check_taxonomy(mcpd)

## End(Not run)
```
client_login

Login to Genesys as a service client (system-to-system)

Description
The client must be enabled for Client Credential grant on Genesys.

Usage
client_login()

See Also
setup

download_mcpd

Download all passport data for one genebank in Excel format and save it to disk

Description
Download all passport data for one genebank in Excel format and save it to disk

Usage
download_mcpd(instituteCode, file = NULL)

Arguments
instituteCode FAO WIEWS institute code
file Target file name. Defaults to Genesys-provided file name in the current working directory.

Value
HTTP response data

Examples
## Not run:
# Download MCPD passport data for NGA039
excelData <- download_mcpd("NGA039")

## End(Not run)
download_pdcxi

**Description**

Download PDCI data for one genebank in Excel format and save it to disk.

**Usage**

```
download_pdcxi(instituteCode, file = NULL)
```

**Arguments**

- `instituteCode`: FAO WIEWS institute code
- `file`: Target file name. Defaults to Genesys-provided file name in the current working directory.

**Value**

HTTP response data

**Examples**

```r
## Not run:
# Download PDCI data for NGA039
excelData <- download_pdcxi("NGA039")
## End(Not run)
```

---

**fetch_accessions**

*Fetch accession passport data*

**Description**

Fetch accession passport data

**Usage**

```
fetch_accessions(filters = list(), page = NULL, size = 1000, 
selector = NULL, at.least = NULL)
```
**Arguments**

- **filters**
  - an R structure with Genesys filters
- **page**
  - the page index (0-based)
- **size**
  - number of records to load per page (page size)
- **selector**
  - NULL or a function to "select" variables of interest
- **at.least**
  - stop fetching when at.least records are received from Genesys

**Value**

Paged data structure

**Examples**

```r
## Not run:
# Retrieve all accession data by country of origin
accessions <- fetch_accessions(mcpd_filter(ORIGCTY = c("DEU", "SVN")))

# Fetch Musa
musa <- genesysr::fetch_accessions(list(taxonomy.genus = c('Musa'))) 

# Apply selector function
accessions <- fetch_accessions(mcpd_filter(ORIGCTY = c("DEU", "SVN")), selector = function(x) {
  list(id = x$id, acceNumb = x$acceNumb, instCode = x$institute$code) })

## End(Not run)
```

---

**filter.DOI**

**Add filter on accession DOI**

**Description**

Add filter on accession DOI

**Usage**

```r
filter.DOI(filter = list(), DOI)
```

**Arguments**

- **filter**
  - Existing filters (or blank list if not provided)
- **DOI**
  - Accession DOI
filter_GENUS  

Add filter by genus

Description
Add filter by genus

Usage
filter_GENUS(filter = list(), GENUS)

Arguments
- filter: Existing filters (or blank list if not provided)
- GENUS: List of genera

filter_ORIGCTY  

Add filter on Country of origin of material

Description
Add filter on Country of origin of material

Usage
filter_ORIGCTY(filter = list(), ORIGCTY)

Arguments
- filter: Existing filters (or blank list if not provided)
- ORIGCTY: Country of origin

filter_SAMPSTAT  

Add filter on Biological status of sample

Description
Add filter on Biological status of sample

Usage
filter_SAMPSTAT(filter = list(), SAMPSTAT)

Arguments
- filter: Existing filters (or blank list if not provided)
- SAMPSTAT: Biological status of sample
filter_SPECIES  

* Add filter on specific epithet *

**Description**

Add filter on specific epithet

**Usage**

```
filter_SPECIES(filter = list(), SPECIES)
```

**Arguments**

- `filter`: Existing filters (or blank list if not provided)
- `SPECIES`: List of specific epithets

---

get_accessions  

* Fetch accession passport data *

**Description**

Fetch accession passport data

**Usage**

```
get_accessions(filters = list(), page = 0, size = 1000, 
               fields = NULL, selector = NULL, at.least = NULL)
```

**Arguments**

- `filters`: an R structure with Genesys filters
- `page`: the page index (0-based)
- `size`: number of records to load per page (page size)
- `fields`: list of fields to fetch from Genesys
- `selector`: NULL or a function to "select" variables of interest
- `at.least`: stop fetching when at.least records are received from Genesys

**Value**

Paged data structure

**See Also**

`mcpd_filter`
mcpd_filter

Examples

## Not run:

# Retrieve all accession data by country of origin (Slovenia, Ivory Coast)
accessions <- genesysr::get_accessions(list(countryOfOrigin = list(code3 = c('SVN', 'CIV'))))

# Fetch Musa, but only geographic data and accessionNumber
musa <- genesysr::get_accessions(list(taxonomy = list(genus = c('Musa'))),
fields = c("accessionNumber", "geo"))

# Apply selector function
accessions <- get_accessions(mcpd_filter(ORIGCTY = c('DEU', 'SVN')),
selector = function(x) {
  list(id = x$id, acceNumb = x$accessionNumber, instCode = x$instituteCode)
}, at.least = 100)

## End(Not run)

mcpd_filter

Make or adjust filter using MCPD terminology

Description

See FAO/Bioversity Multi-Crop Passport Descriptors.

Usage

mcpd_filter(filter = list(), DOI = NULL, ORIGCTY = NULL,
SAMPSTAT = NULL, GENUS = NULL, SPECIES = NULL)

Arguments

filter Existing filters (or blank list if not provided)
DOI Accession DOI
ORIGCTY Country of origin
SAMPSTAT Biological status of sample
GENUS List of genera
SPECIES List of specific epithets (within specified genera)

Examples

# Filter accessions from Mexico and Slovenia
mcpd_filter(ORIGCTY = c("MEX", "SVN"))
me

Who am i?

Description
Who am i?

Usage
me()

print_setup
Print Genesys client configuration

Description
Print Genesys client configuration

Usage
print_setup()

setup
Configure the Genesys environment

Description
Configure the Genesys environment

Usage
setup(server = NULL, client_id = NULL, client_secret = NULL)

Arguments
- server: Server base URL (e.g. "https://api.genesys-pgr.org" or "https://api.sandbox.genesys-pgr.org")
- client_id: OAuth client ID
- client_secret: OAuth client secret

See Also
See utility methods setup_production, setup_sandbox
## Examples

```r
# Link with sandbox
setup_sandbox()
```

### Description

Use the Genesys R Client with <https://api.genesys-pgr.org> requiring `user_login`.

### Usage

```r
setup_production()
```

## setup_sandbox

### Description

Use the Genesys R Client with <https://api.sandbox.genesys-pgr.org> requiring `user_login`.

### Usage

```r
setup_sandbox()
```

## user_login

### Description

Login to Genesys as a user.

The authorization URL will open in a browser, ask the user to grant permissions to R and the verification code must be copy-pasted after you grant access to the client.

### Usage

```r
user_login()
```

## See Also

`setup`
Index

api1_url, 2
api2_url, 3
authorization, 3

check_country, 4
check_lanorsea, 4
check_taxonomy, 5
client_login, 3, 6

download_mcpd, 6
download_pdc, 7

fetch_accessions, 7
filter_DOO, 8
filter_GENUS, 9
filter_ORIGCTY, 9
filter_SAMPSTAT, 9
filter_SPECIES, 10

get_accessions, 10

mcpd_filter, 10, 11
me, 12

print_setup, 12

setup, 6, 12, 13
setup_production, 12, 13
setup_sandbox, 12, 13

user_login, 3, 13, 13