

# Package ‘genesysr’

November 22, 2019

**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](mailto:matija.obreza@croptrust.org)>

**Depends** R (>= 3.1.0)

**Imports** htrr, 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

client_login . . . . .	6
download_mcpd . . . . .	6
download_pdcj . . . . .	7
fetch_accessions . . . . .	7
filter_DOI . . . . .	8
filter_GENUS . . . . .	9
filter_ORIGCTY . . . . .	9
filter_SAMPSTAT . . . . .	9
filter_SPECIES . . . . .	10
get_accessions . . . . .	10
mcpd_filter . . . . .	11
me . . . . .	12
print_setup . . . . .	12
setup . . . . .	12
setup_production . . . . .	13
setup_sandbox . . . . .	13
user_login . . . . .	13
<b>Index</b>	<b>14</b>

---

api1_url	<i>Get full Genesys API v1 URL for a specific path</i>
----------	--

---

### 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	<i>Get full Genesys API v1 URL for a specific path</i>
----------	--

---

**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	<i>Provide OAuth2 token to use for authorization with Genesys</i>
---------------	---

---

**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_country	<i>Run Land-or-Sea check on MCPD data. Uploads only rows where ORIGCTY, DECLATITUDE and DECLONGITUDE are provided.</i>
---------------	--

---

**Description**

Run Land-or-Sea check on MCPD data. Uploads only rows where ORIGCTY, DECLATITUDE and DECLONGITUDE are provided.

**Usage**

```
check_country(mcpd)
```

**Arguments**

mcpd	Accession passport data in MCPD format
------	--

**Value**

Results from validator

**Examples**

```
## Not run:
  geoCheck <- genesysr:::check_country(mcpd)

## End(Not run)
```

---

check_landorsea	<i>Run Land-or-Sea check on MCPD data using <a href="https://validator.genesys-pgr.org">https://validator.genesys-pgr.org</a>. Uploads only rows where DECLATITUDE and DECLONGITUDE are provided. In practice it is better to use 'check_country' if ORIGCTY data exists.</i>
-----------------	---

---

**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**

```
check_landorsea(mcpd)
```

**Arguments**

mcpd                    Accession passport data in MCPD format

**Value**

Results from validator

**Examples**

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

---

check_taxonomy	<i>Check MCPD taxonomic data (GENUS, SPECIES, SPAUTHOR, SUB-TAXA, SUBTAUTHOR) using <a href="https://validator.genesys-pgr.org">https://validator.genesys-pgr.org</a>.</i>
----------------	--

---

**Description**

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

**Usage**

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

**Arguments**

mcpd                    Accession passport data in MCPD format  
toCurrentTaxa        Should obsoleted names be reported?

**Value**

Results from validator

**Examples**

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

---

client_login	<i>Login to Genesys as a service client (system-to-system)</i>
--------------	--

---

**Description**

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

**Usage**

```
client_login()
```

**See Also**

[setup](#)

---

download_mcpd	<i>Download all passport data for one genebank in Excel format and save it to disk</i>
---------------	--

---

**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_pdc	<i>Download PDCI data for one genebank in Excel format and save it to disk.</i>
--------------	---

---

**Description**

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

**Usage**

```
download_pdc(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 PDCI data for NGA039  
excelData <- download_pdc("NGA039")  
  
## End(Not run)
```

---

fetch_accessions	<i>Fetch accession passport data</i>
------------------	--------------------------------------

---

**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**

```
## 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**

```
filter_DOI(filter = list(), DOI)
```

**Arguments**

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



---

filter_GENUS	<i>Add filter by genus</i>
--------------	----------------------------

---

**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	<i>Add filter on Country of origin of material</i>
----------------	--

---

**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	<i>Add filter on Biological status of sample</i>
-----------------	--

---

**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	<i>Add filter on specific epithet</i>
----------------	---------------------------------------

---

**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	<i>Fetch accession passport data</i>
----------------	--------------------------------------

---

**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](#)

**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	<i>Who am i?</i>
----	------------------

---

**Description**

Who am i?

**Usage**

me()

---

print_setup	<i>Print Genesys client configuration</i>
-------------	---

---

**Description**

Print Genesys client configuration

**Usage**

print\_setup()

---

setup	<i>Configure the Genesys environment</i>
-------	--

---

**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**

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

---

setup_production	<i>Setup for Genesys Production</i>
------------------	-------------------------------------

---

**Description**

Use the Genesys R Client with <<https://api.genesys-pgr.org>> requiring [user\\_login](#)

**Usage**

```
setup_production()
```

---

setup_sandbox	<i>Setup for Genesys Sandbox</i>
---------------	----------------------------------

---

**Description**

Use the Genesys R Client with <<https://api.sandbox.genesys-pgr.org>> requiring [user\\_login](#)

**Usage**

```
setup_sandbox()
```

---

user_login	<i>Login to Genesys as a user</i>
------------	-----------------------------------

---

**Description**

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**

```
user_login()
```

**See Also**

[setup](#)

# Index

api1\_url, [2](#)  
api2\_url, [3](#)  
authorization, [3](#)

check\_country, [4](#)  
check\_landorsea, [4](#)  
check\_taxonomy, [5](#)  
client\_login, [3, 6](#)

download\_mcpd, [6](#)  
download\_pdc, [7](#)

fetch\_accessions, [7](#)  
filter\_DOI, [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](#)