

Package: virionData (via r-universe)

October 7, 2026

Type Package

Title Access Versioned VIRION Data

Version 2.0.0

Description This package provides an ergonomic interface for accessing the VIRION dataset (<https://doi.org/10.5281/zenodo.15643003>).

License GPL (>= 3)

Encoding UTF-8

LazyData true

URL <https://github.com/viralemergence/virionData>,
<https://viralemergence.github.io/virionData/>

BugReports <https://github.com/viralemergence/virionData/issues>

Suggests dplyr, glue, httptest, kableExtra, knitr, rmarkdown, stringr,
testthat (>= 3.0.0)

Config/testthat/edition 3

Imports assertthat, curl, frictionless, fs, httr, jsonlite, purrr, R6,
readr, rlang

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Depends R (>= 4.1.0)

Config/roxygen2/version 8.0.0

RoxygenNote 8.0.0

Config/pak/sysreqs cmake make libuv1-dev libssl-dev libx11-dev

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

Date/Publication 2026-07-09 21:01:29 UTC

RemoteUrl <https://github.com/viralemergence/virionData>

RemoteRef HEAD

RemoteSha bfa1461ecfc6708d6dba35b25ef988d7d9b75ea8

Contents

batch_download_deposit_versions	2
deposit	3
deposit_summary	11
download_deposit_version	12
download_refresh	13
dummy_imports	13
export_deposit_bibtex	14
export_deposit_metadata	14
get_all_versions	15
get_data_dictionary	16
get_individual_dictionary	16
get_json	17
get_resource_name	17
get_version_citation	18
get_versioned_data	18
list_deposit_versions	19
make_url	20
sanitize_id	21
sanitize_version	21
Index	22

batch_download_deposit_versions
<i>Batch download deposit versions</i>

Description

This is download_deposit_version wrapped in a purrr::map call.

Usage

batch_download_deposit_versions(zenodo_ids = "all", dir_path, ...)

Arguments

- zenodo_ids Character. Either a vector of zenodo ids or "all"
- dir_path Character. Path to folder where files should be downloaded.
- ... Other arguments passed to download_deposit_version

Value

List of download locations.

Examples

```
## Not run:
# get all deposit versions
batch_download_deposit_versions(dir_path = "outputs")

# get select versions
batch_download_deposit_versions(zenodo_ids = c("15677137", "15643004"), dir_path = "outputs")

## End(Not run)
```

deposit

deposit

Description

An R6 client for getting virion data and metadata.

Value

A deposit class (R6 class)

invisibly returns self

dataframe of remote csv file.

String. Path to download location.

data.frame from `readr::read_csv()`

String. cleaned zenodo id

metadata in the specified format (invisible)

List. Named list of data frames with the following fields:

- name = field name
- type = field type
- description = characterization of the field

parent_id String. Identifier for a Zenodo deposit with multiple versions.

all_versions Character. All zenodo ids for versions of the deposit.

summary_df Data frame. A table of all zenodo ids for versions of the deposit.

latest_version String. Identifier for the latest version of the zenodo deposit.

working_version String. Identifier for the version of the zenodo deposit you are working with

working_bibtex String. Bibtext for the version of the zenodo deposit you are working with.

working_citation String. Citation for the version of the zenodo deposit you are working with.

working_files Data frame. Working files consist of key (file name) and url (url to file on zenodo)

working_json list. Metadata for the working version.

Methods

Public methods:

- `deposit$new()`
- `deposit$set_working_version()`
- `deposit$load_remote_csv_file()`
- `deposit$download_versioned_data()`
- `deposit$load_local_csv_file()`
- `deposit$check_zenodo_id()`
- `deposit$export_metadata()`
- `deposit$get_data_dictionary()`
- `deposit$get_parent_id()`
- `deposit$get_all_versions()`
- `deposit$get_summary_df()`
- `deposit$get_latest_version()`
- `deposit$get_working_version()`
- `deposit$get_working_bibtex()`
- `deposit$get_working_citation()`
- `deposit$get_working_files()`
- `deposit$get_working_json()`
- `deposit$clone()`

Method `new()`: Create a new deposit object, as an [R6](#) client. This object is connected to the zenodo deposit for virion and is the gateway for accessing different versions of the data.

Usage:

```
deposit$new(parent_id = "15643003")
```

Arguments:

`parent_id` String. Identifier for a Zenodo deposit with multiple versions.

Returns: A new deposit object

Examples:

```
\dontrun{
virion_deposit <- deposit$new()
}
```

Method `set_working_version()`: Set the version of the data you would like to work with. Note that the structure of Virion data may change through time.

Usage:

```
deposit$set_working_version(
  zenodo_id = "latest",
  style = "apa",
  print_bibtex = TRUE,
  print_citation = TRUE
)
```

Arguments:

`zenodo_id` String. A zenodo id or latest.
`style` String. The citation style to be used.
`print_bibtex` Logical. Should the bibtex for the working version of the deposit be printed when setting the working version?
`print_citation` Logical. Should the citation for the working version of the deposit be printed?

Examples:

```
\dontrun{
# set the working version to the latest version of the deposit
virion_deposit$set_working_version("latest")

# set the working versino to an arbitrary deposit
virion_deposit$get_all_versions()
virion_deposit$set_working_version("19502921")
}
```

Method `load_remote_csv_file()`: Load a remote csv listed in the working files field directly into the environment.

Can handle either csv or csv.gz files. This function downloads a temporary file to your machine, reads it, then returns a dataframe.

The local file cache is removed when your R session restarts.

Consider using the `download_versioned_data` to save a persistent version of the virion dataset on your machine.

Usage:

```
deposit$load_remote_csv_file(file_key, refresh = FALSE, ...)
```

Arguments:

`file_key` String. A file key as seen in the `working_files` field
`refresh` Logical. Should files be re-downloaded.
`...` Additional arguments to pass to `readr::read_csv()`

Examples:

```
\dontrun{
## see the working files
working_files <- virion$get_working_files()
working_files$file_key
virion_data <- virion_deposit$load_remote_csv_file("virion.csv.gz")

}
```

Method `download_versioned_data()`: Download versioned data

Downloads all the files for a particular version of Virion to `dir/zenodo_id`.

This function does not change the working version!

Usage:

```
deposit$download_versioned_data(
  zenodo_id = "working",
  dir = "outputs",
  refresh = FALSE
)
```

Arguments:

zenodo_id String. A zenodo id, working or latest
 dir String. Storage location for files
 refresh Logical. Should files be re-downloaded.

Examples:

```
\dontrun{
# download files for the working version
virion_deposit$download_versioned_data()

# download files for the latest version
virion_deposit$download_versioned_data("latest")

# download files for some zenodo id
virion_deposit$download_versioned_data("19502921")
}
```

Method `load_local_csv_file()`: Load a local version of the csv.

Assumes you would like to load the file from the current working version.

If the csv file is not present, a persistent version of the file will be downloaded to your machine.

Usage:

```
deposit$load_local_csv_file(file_key, refresh, ...)
```

Arguments:

file_key string. Name of file from working_files
 refresh logical. Should the file be re-downloaded?
 ... Additional parameters to pass to `readr::read_csv()`

Examples:

```
\dontrun{

virion <- deposit$new()
virion$set_working_version()
working_files <- virion$get_working_files()
working_files$file_key
virion$load_local_csv_file("virion.csv.gz")
}
```

Method `check_zenodo_id()`: Check the zenodo ID

Look for key words like working or latest and convert them to integer ids OR pass the zenodo id directly to `sanitize_id()`

Usage:

```
deposit$check_zenodo_id(zenodo_id)
```

Arguments:

zenodo_id String. One of working, latest, or a zenodo id

Examples:

```
\dontrun{
  virion_deposit$check_zenodo_id("latest")
}
```

Method export_metadata(): Export metadata for a version of the deposit.

Usage:

```
deposit$export_metadata(zenodo_id, format, verbose)
```

Arguments:

zenodo_id String. One of "latest", "working", or a zenodo id

format String. One of "json", "json-ld", "csl", "datacite-json", "datacite-xml", "dublincore", "marcxml", "bibtex", "geojson", "ap", "codemeta", "cff"

verbose Logical. Print the metadata

Examples:

```
\dontrun{
  virion_deposit$export_metadata("working")
}
```

Method get_data_dictionary(): Create a list of data dictionaries.

Usage:

```
deposit$get_data_dictionary(
  file_key = "datapackage.json",
  dir = tempdir(),
  refresh
)
```

Arguments:

file_key string. Name of file from working_files

dir string. Name of directory

refresh refresh Logical. Should files be re-downloaded.

Examples:

```
\dontrun{
  virion <- deposit$new()
  virion$set_working_version()
  virion$get_data_dictionary()

}
```

Method get_parent_id(): Get a property of the object that is private

- `all_versions` Character. All zenodo ids for versions of the deposit.
- `summary_df` Data frame. A table of all zenodo ids for versions of the deposit.
- `latest_version` String. Identifier for the latest version of the zenodo deposit.
- `working_version` String. Identifier for the version of the zenodo deposit you are working with.
- `working_bibtex` String. Bibtext for the version of the zenodo deposit you are working with.,
- `working_citation` String. Citation for the version of the zenodo deposit you are working with.
- `working_files` Data frame. Working files consist of key (file name) and url (url to file on zenodo)
- `working_json` list. Metadata for the working version.

Usage:

```
deposit$get_parent_id()
```

Method `get_all_versions()`: Get a property of the object that is private

Usage:

```
deposit$get_all_versions()
```

Method `get_summary_df()`: Get a property of the object that is private

- `latest_version` String. Identifier for the latest version of the zenodo deposit.
- `working_version` String. Identifier for the version of the zenodo deposit you are working with.
- `working_bibtex` String. Bibtext for the version of the zenodo deposit you are working with.,
- `working_citation` String. Citation for the version of the zenodo deposit you are working with.
- `working_files` Data frame. Working files consist of key (file name) and url (url to file on zenodo)
- `working_json` list. Metadata for the working version.

Usage:

```
deposit$get_summary_df()
```

Method `get_latest_version()`: Get a property of the object that is private

Usage:

```
deposit$get_latest_version()
```

Method `get_working_version()`: Get a property of the object that is private

Usage:

```
deposit$get_working_version()
```

Method `get_working_bibtex()`: Get a property of the object that is private

Usage:

```
deposit$get_working_bibtex()
```

Method `get_working_citation()`: Get a property of the object that is private

Usage:

```
deposit$get_working_citation()
```

Method `get_working_files()`: Get a property of the object that is private

Usage:

```
deposit$get_working_files()
```

Method `get_working_json()`: Get a property of the object that is private

Usage:

```
deposit$get_working_json()
```

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
deposit$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
## Not run:
# make a client
virion_deposit <- deposit$new()

# methods
virion_deposit$set_working_version("latest")
virion_deposit$get_working_files()

# load remote data directly into your session
virion_data <- virion_deposit$load_remote_csv_file(file_key = "virion.csv.gz")

# download remote data to disk - defaults to working version
virion_deposit$download_versioned_data()

## End(Not run)

## -----
## Method `deposit$new`
## -----

## Not run:
virion_deposit <- deposit$new()

## End(Not run)

## -----
## Method `deposit$set_working_version`
## -----

## Not run:
# set the working version to the latest version of the deposit
virion_deposit$set_working_version("latest")
```

```

# set the working versino to an arbitrary deposit
virion_deposit$get_all_versions()
virion_deposit$set_working_version("19502921")

## End(Not run)

## -----
## Method `deposit$load_remote_csv_file`
## -----

## Not run:
## see the working files
working_files <- virion$get_working_files()
working_files$file_key
virion_data <- virion_deposit$load_remote_csv_file("virion.csv.gz")

## End(Not run)

## -----
## Method `deposit$download_versioned_data`
## -----

## Not run:
# download files for the working version
virion_deposit$download_versioned_data()

# download files for the latest version
virion_deposit$download_versioned_data("latest")

# download files for some zenodo id
virion_deposit$download_versioned_data("19502921")

## End(Not run)

## -----
## Method `deposit$load_local_csv_file`
## -----

## Not run:

virion <- deposit$new()
virion$set_working_version()
working_files <- virion$get_working_files()
working_files$file_key
virion$load_local_csv_file("virion.csv.gz")

## End(Not run)

```

```
## -----
## Method `deposit$check_zenodo_id`
## -----

## Not run:
virion_deposit$check_zenodo_id("latest")

## End(Not run)

## -----
## Method `deposit$export_metadata`
## -----

## Not run:
virion_deposit$export_metadata("working")

## End(Not run)

## -----
## Method `deposit$get_data_dictionary`
## -----

## Not run:
virion <- deposit$new()
virion$set_working_version()
virion$get_data_dictionary()

## End(Not run)
```

deposit_summary

Simple summary data frame for the deposit

Description

Simple summary data frame for the deposit

Usage

```
deposit_summary()
```

Value

data frame. Simplified deposit metadata with the following fields:

- id = zenodo id

- latest_version = True or False indicating if that record is the latest version of the dataset.
- publication_date = YMD the item was published to zenodo
- doi_url = digital object identifier for that version of the record.

Examples

```
deposit_summary()
```

```
download_deposit_version
```

Download deposit version

Description

Downloads and extracts some version of the deposit.

Usage

```
download_deposit_version(
  zenodo_id,
  deposit_versions = list_deposit_versions(),
  dir_path,
  datapackage_only = FALSE
)
```

Arguments

zenodo_id	String. ID for a Zenodo deposit. Should correspond to the version of a deposit.
deposit_versions	data frame. Output from list_deposit_versions
dir_path	String. Path to directory where the files should be downloaded e.g. "inst/extdata/wdds_archive" note no trailing slash on the path.
datapackage_only	Logical. Download only the datapackage.json file

Value

String. Path to downloaded version.

download_refresh	<i>Download or redownload data</i>
------------------	------------------------------------

Description

Downloads a file or re-downloads a file to refresh the cached files.

Usage

```
download_refresh(file_url, local, refresh)
```

Arguments

file_url	String. URL for file you'd like to download
local	String. A character string with the name where the downloaded file is saved. Tilde-expansion is performed.
refresh	Logical. Should saved files be refreshed.

Value

String. Local File path

dummy_imports	<i>Dummy function to pass RMD check</i>
---------------	---

Description

Apparently a known issue in R6 as the class is not created until install.

Usage

```
dummy_imports()
```

Value

Character string explaining why this function exists.

`export_deposit_bibtex` *Get bibtex entry for a deposit.*

Description

Bibtex specific wrapper for `export_deposit_metadata`

Usage

```
export_deposit_bibtex(zenodo_id = the$working_version, verbose = TRUE)
```

Arguments

<code>zenodo_id</code>	String. ID for a Zenodo deposit. Should correspond to the version of a deposit.
<code>verbose</code>	Logical. Print the formatted metadata?

Value

Character. bibtex formatted entry

Examples

```
## Not run:  
export_deposit_bibtex(zenodo_id = "15692263" ) |>  
  writeLines(con = "outputs/citation.bib")  
  
## End(Not run)
```

`export_deposit_metadata`

Export Deposit Metadata in various formats

Description

Export Deposit Metadata in various formats

Usage

```
export_deposit_metadata(  
  zenodo_id = the$working_version,  
  format,  
  verbose = TRUE  
)
```

Arguments

zenodo_id	String. ID for a Zenodo deposit. Should correspond to the version of a deposit.
format	String. File format for export. One of "json", "json-ld", "csl", "datacite-json", "datacite-xml", "dublincore", "marcxml", "bibtex", "geojson", "dcap", "codemeta", or "cff"
verbose	Logical. Print the formatted metadata?

Value

Character. Returns text in selected format.

Examples

```
## Not run:
export_deposit_metadata(zenodo_id = "15692263", "json-ld" ) |>
  writeLines(con = "outputs/citation.jsonld")

## End(Not run)
```

get_all_versions	<i>Get all deposit versions</i>
------------------	---------------------------------

Description

Recursively traverse a deposit link to get all versions of a deposit.

Usage

```
get_all_versions(url)
```

Arguments

url	Character. URL for a version of the deposit.
-----	--

Value

Data.frame. Metadata for all deposit version

Examples

```
## Not run:

self_link <- "https://zenodo.org/api/records/21232661/versions?page=1&size=25&sort=version"
get_all_versions(self_link)

## End(Not run)
```

get_data_dictionary	<i>Get Data Dictionary for a data Package</i>
---------------------	---

Description

Creates a list of data frames that constitutes the data dictionary.

Usage

```
get_data_dictionary(datapackage_json)
```

Arguments

datapackage_json
Character. Path to datapackage.json file

Value

List. Named list of data frames with the following fields:

- name = field name
- type = field type
- description = characterization of the field

Examples

```
## Not run:  
get_data_dictionary(datapackage_json = "data/1235600/datapackage.json")  
  
## End(Not run)
```

get_individual_dictionary	<i>Get the dictionary for a single resource</i>
---------------------------	---

Description

Get the dictionary for a single resource

Usage

```
get_individual_dictionary(x)
```

Arguments

x
List. Resource object from the data package.

Value

dataframe with three fields.

get_json

Get json

Description

`jsonlite::fromJSON()` wrapped in `rlang::try_fetch()` to make calls smoother.

Usage

```
get_json(url)
```

Arguments

`url` String. A url fed to `jsonlite::fromJSON()`

Value

List. JSON as a list

Examples

```
## Not run:  
get_json("https://www.example.com/example.json")  
  
## End(Not run)
```

get_resource_name

Get the name of a resource

Description

Get the name of a resource

Usage

```
get_resource_name(x)
```

Arguments

`x` data frame with name field

Value

character. Name of resources

get_version_citation	<i>Get Version Citation</i>
----------------------	-----------------------------

Description

Get Version Citation

Usage

```
get_version_citation(
  zenodo_id = the$working_version,
  style = c("havard-cite-them-right", "apa", "modern-language-association", "vancouver",
    "chicago-fullnote-bibliography", "ieee"),
  verbose = TRUE
)
```

Arguments

zenodo_id	String. ID for a Zenodo deposit. Should correspond to the version of a deposit.
style	Character. One of "havard-cite-them-right", "apa", "modern-language-association", "vancouver", "chicago-fullnote-bibliography", or "ieee"
verbose	Logical. Print the citation?

Value

Character. Text for a citation

Examples

```
## Not run:
get_version_citation(zenodo_id = "15692263",
  style = "apa")

## End(Not run)
```

get_versioned_data	<i>Get a specific version of the data</i>
--------------------	---

Description

This is a convenience function that downloads a specific version of the data and returns attribution information.

Usage

```

get_versioned_data(
  version = "latest",
  style = "apa",
  dir_path,
  refresh_deposits_versions = TRUE,
  verbose = TRUE,
  ...
)

```

Arguments

version	Character. identifier for a version e.g. "15643003" or "latest"
style	Character. One of "havard-cite-them-right", "apa", "modern-language-association", "vancouver", "chicago-fullnote-bibliography", or "ieee"
dir_path	Character. Path to folder where files should be downloaded.
refresh_deposits_versions	Logical. Should the function check for new versions of the deposit?
verbose	Logical. Include print statements?
...	Additional arguments to pass to download_deposit_version

Details

If you would like to download a bibtex entry use `export_deposit_bibtex`. For other metadata types (CSL, CFF, JSON-LD, etcetera) use `export_deposit_metadata`.

Value

Character. path to versioned data.

`list_deposit_versions` *Get Versions of a Deposit on Zenodo*

Description

This function gets the metadata for all the versions of a deposit associated with a parent id. The parent id is used to identify a set of works that are different versions of the same work. The parent id is provided by the Zenodo API. To find the parent ID, download a JSON representation of the deposit (export to json on the webpage or use `export_deposit_metadata`), there will be an attribute called `parent` that looks like `"https://zenodo.org/api/records/15020049"`. The 8 digit string at the end of the url is the parent id.

Usage

```
list_deposit_versions(parent_id = "15643003")
```

Arguments

parent_id String. Identifier for a Zenodo deposit with multiple versions.

Details

This function also sets package variables

Value

Data frame. Invisible. The data frame contains the Zenodo id for each version of the deposit, as well as the version name, and logical field called latest that indicates if this is the latest version.

make_url	<i>Make Zenodo URLs</i>
----------	-------------------------

Description

Appends a zenodo id to the zenodo api url.

Usage

```
make_url(base_url = "https://zenodo.org/api/records/%s", id)
```

Arguments

base_url string. url for zenodo api
id string. zenodo id.

Value

String. URL for zenodo api.

Examples

```
make_url(id = "1235600")
```

sanitize_id	<i>Sanitize a zenodo id</i>
-------------	-----------------------------

Description

Remove any white space and check that it conforms to the zenodo id pattern.

Usage

```
sanitize_id(zenodo_id)
```

Arguments

zenodo_id	String. A zenodo id.
-----------	----------------------

Value

String. A cleaned zenodo id.

Examples

```
sanitize_id(" 2948598")
```

sanitize_version	<i>Check version ids</i>
------------------	--------------------------

Description

Checks that version ids are properly formatted. IDs should either be integers ("15643003") OR "latest"

Usage

```
sanitize_version(version)
```

Arguments

version	Character. Version identifier.
---------	--------------------------------

Value

Character. Version identifier from Zenodo.

Examples

```
sanitize_version("latest")  
sanitize_version(" 15643003")
```

Index

`batch_download_deposit_versions`, [2](#)

`deposit`, [3](#)

`deposit_summary`, [11](#)

`download_deposit_version`, [12](#)

`download_refresh`, [13](#)

`dummy_imports`, [13](#)

`export_deposit_bibtex`, [14](#)

`export_deposit_metadata`, [14](#)

`get_all_versions`, [15](#)

`get_data_dictionary`, [16](#)

`get_individual_dictionary`, [16](#)

`get_json`, [17](#)

`get_resource_name`, [17](#)

`get_version_citation`, [18](#)

`get_versioned_data`, [18](#)

`jsonlite::fromJSON()`, [17](#)

`list_deposit_versions`, [19](#)

`make_url`, [20](#)

`R6`, [4](#)

`readr::read_csv()`, [3](#), [5](#), [6](#)

`rlang::try_fetch()`, [17](#)

`sanitize_id`, [21](#)

`sanitize_version`, [21](#)