

Downloading data from Symbiota2 portals in R

Austin Koontz & William D. Pearse

09/24/2020

Overview

Symbiota is an open-source content management system built for the purpose of integrating virtual biodiversity databases. Currently used by over 700 natural history collections, containing more than 30 million specimens, Symbiota is an essential tool for digitizing biological specimen data. In an effort to expand modularity and accessibility, Symbiota2 is an improved, refactored version of the original Symbiota core code structure, designed based on user feedback. While packages do exist for accessing Symbiota portals (for instance, see the `rSymbiota` package), R users currently cannot access the data offered by Symbiota2. Here, we describe **SymbiotaR2**, a package built to address this need by allowing users to access Symbiota2 portals in an R environment.

Below, we provide a general workflow for using SymbiotaR2, a description of the command structure, code for installing the package, and examples of using SymbiotaR2 functions. The GitHub page for the Symbiota2 software can be found [here](#), and instructions for setting up a new Symbiota2 portal can be found on the documentation site. Finally, a review of the original Symbiota platform is offered in Gries et al., 2014.

Workflow

SymbiotaR2 allows R users to download data from specified Symbiota2 portals, granting access to thousands of digitized flora and fauna specimen records across the United States. It does this by querying endpoints in the Symbiota2 API, then downloading a JSON object containing the requested data to a temporary directory on the local computer. The JSON object is then converted into an R format that is straightforward and easy to use. The general argument structure of all SymbiotaR2 functions allows users to specify whether they want to pull a single resource or a collection of resources from the API.

The general workflow for using SymbiotaR2 is as follows:

1. Determine the URL of the Symbiota2 portal you wish to access data from. You will need to be approved access by the portal manager before being able to retrieve data from the portal.
2. Install the SymbiotaR2 package (see Installation below).
3. Find the function corresponding to the resource you wish to pull from the Symbiota2 portal.
4. Call the function, specifying the Symbiota2 portal using the `url` argument (see Example below).

Installation

SymbiotaR2 can be downloaded by calling:

```
library(devtools)
install_github("pearselab/SymbiotaR2")
```

Load the package using:

```
library(SymbiotaR2)
```

Portal Specification

SymbiotaR2_setup will save to your .Rprofile a default URL, for automatic reference. Specifying a different url argument will let you refer to a portal besides the default. The code below demonstrates this:

```
SymbiotaR2_setup("http://imaginary-symbiota-portal.com/api", append=TRUE)
```

```
Coordinates() # Download from http://imaginary-symbiota-portal.com/api
```

```
Coordinates("http://another-imaginary-portal.com/api") # Download from a different portal
```

Example

SymbiotaR2 consists of commands pulling from the Checklists, Collections, Crowdsourcing, Exsiccata, Glossary, ImageProcessor, Key, Media, Occurrence, Reference, Taxa, Traits, and UserRoles API families of the specified Symbiota2 portal. Note that because each Symbiota2 portal owner can load their own plugins into the API, it's possible that not every possible API endpoint from the specified Symbiota2 instance will be covered.

Below, we provide an example of pulling a single Taxa resource into the R environment, by specifying an id argument in the command call (using a random, nonexistent URL):

```
myURL <- "http://imaginary-symbiota2-portal.com/api"
```

```
myTaxa <- Taxa(id = 12, url = myURL)
```

```
str(myTaxa)
```

List of 23

```
$ @context      : chr "/api/contexts/Taxa"
$ @id           : chr "/api/taxa/12"
$ @type         : chr "Taxa"
$ id            : num 12
$ rankId        : chr "/api/taxa/ranks/31"
$ scientificName : chr "Polygonum bistortoides"
$ unitIndicator1 : logi NA
$ unitName1     : chr "Polygonum"
$ unitIndicator2 : logi NA
$ unitName2     : chr "bistortoides"
$ unitIndicator3 : logi NA
$ unitName3     : logi NA
$ author        : chr "Pursh"
$ phylogenySortSequence: logi NA
$ status        : chr "AZTT-USDA Plants consistent"
$ source        : logi NA
$ notes         : logi NA
$ hybrid        : logi NA
$ securityStatus : num 0
$ modifiedTimestamp : logi NA
$ initialTimestamp : chr "2019-01-11T21:44:39+00:00"
$ modifiedUserId  : logi NA
$ taxaAuthorityId : list()
```

If a collection of resources from the Symbiota2 API needs to come into the R environment, then the page argument can be specified in place of id to retrieve a list of resources (here, as a data.frame):

```
myURL <- "http://imaginary-symbiota2-portal.com/api"
```

```
myCoordinates <- Coordinates(page = 1, url = myURL)
```

```
str(my.Coordinates)
```

```
'data.frame':  5 obs. of  2 variables:
```

```
$ latitude : num 32.2 32.2 32.2 32.2 32.2
$ longitude: num -111 -111 -111 -111 -111
```

If neither an `id` or a `page` argument is provided, the functions are written to return the list of resources at `page = 1`. Once downloaded, these R objects can be taken and manipulated as needed for any downstream processes.

Troubleshooting

The code for SymbiotaR2 is structured hierarchically, and includes parameter type checking to ensure arguments are provided in the proper format. Additionally, all commands include a URL check (`.check.url`), which confirms the following:

1. URL provided refers to an accessible website, and
2. that website is a working Symbiota2 portal.

The 2nd step consists of an API call made at the end of the URL check. If either step fails, the error below will be triggered:

```
Error in .check.url(badURL) :
  URL http://incorrect-portal-address.com/api cannot be reached; is it a valid Symbiota2 portal API?
```

If this error is received, make sure your portal address is spelled correctly. Note that functions are designed such that a forward slash (/) at the end of the URL is optional. If your URL is correctly spelled, make sure that the Symbiota2 portal manager has allowed you access to the portal.

Testing

Tests for all SymbiotaR2 functions (using both an `id` and a `page` parameter) can be found in the `tests/testthat/` directory. Tests, like functions, are organized by API families. The `tests/fixtures` directory contains the `vcr` cassettes for the tests, which are used to speed up HTTP requests; information about the `vcr` package can be found [here](#).