Package 'rjsonapi'

January 9, 2017

Title Consumer for APIs that Follow the JSON API Specification

Description Consumer for APIs that Follow the JSON API Specification (<http://jsonapi.org/>). Package mostly consumes data - with experimental support for serving JSON API data.

Version 0.1.0

License MIT + file LICENSE

URL https://github.com/ropensci/rjsonapi

BugReports https://github.com/ropensci/rjsonapi/issues

Imports crul (>= 0.2.0), R6 (>= 2.2.0), jsonlite (>= 1.2), plumber (>= 0.3.1)

Suggests testthat, covr

RoxygenNote 5.0.1

NeedsCompilation no

Author Scott Chamberlain [aut, cre]

Maintainer Scott Chamberlain <myrmecocystus@gmail.com>

Repository CRAN

Date/Publication 2017-01-09 01:47:26

R topics documented:

rjsonapi-package *rjsonapi client*

Description

rjsonapi client

JSON API

JSON API is a specification for building APIs (Application Programming Interfaces) in JSON (Javascript Object Notation).

The JSON API website is at http://jsonapi.org/. The specification itself is at http://jsonapi.org/format/, with implementations at http://jsonapi.org/implementations/.

Examples

There are very few examples to see in the real world. One example comes from CodeClimate. Their API docs are at https://docs.codeclimate.com/docs/api

Author(s)

Scott Chamberlain <myrmecocystus@gmail.com>

jsonapi_connect *Connection*

Description

Connection

Arguments

url	(character) Base url, without the version information, e.g., http://localhost:8088
version	(character) API version. Default: v1
content_type	(character) the content type to set in all request headers. Default: application/vnd.api+json
headers	(list) A list of headers to be applied to each requset.
...	Curl options passed on to HttpClient. You can set these for all requests, or on each request - see examples.

Details

Methods

`status(...)` Check server status with a HEAD request

- ... - curl options

`routes(...)` Get routes the server supports

- ... - curl options

`route(endpt, query, include, error_handler, ...)` Fetch a route, optional query parameters

- endpt - The endpoint to request data from. required.
- query - a set of query parameters. combined with include parameter
- include - A comma-separated list of relationship paths. combined with query parameter
- error_handler - A function for error handling
- ... - curl options

Examples

```
## Not run:
library("crul")
(conn <- jsonapi_connect("http://localhost:8088"))
conn$url
conn$version
conn$content_type
conn$status()
conn$routes()
conn$routes(verbose = TRUE)

# get data from speicific routes
conn$route("authors")
conn$route("chapters")
conn$route("authors/1")
conn$route("authors/1/books")
conn$route("chapters/5")
conn$route("chapters/5/book")
conn$route("chapters/5/relationships/book")

## include
conn$route("authors/1", include = "books")
conn$route("authors/1", include = "photos")
conn$route("authors/1", include = "photos.title")

## set curl options on jsonapi_connect() call
xx <- jsonapi_connect("http://localhost:8088", verbose = TRUE)
xx$opts
xx$status()

## set headers on initializing the client
(conn <- jsonapi_connect("http://localhost:8088", headers = list(foo = "bar")))
```

```
## errors
### route doesn't exist
# conn$route("foobar")

### document doesn't exist
# conn$route("authors/56")

## End(Not run)
```

`jsonapi_server` *Start a JSONAPI server*

Description

Start a JSONAPI server

Usage

```
jsonapi_server(port = 8000)
```

Arguments

`port`	(integer) the port to run the server on. default: 8000

Details

Right now, this function doesn't take arbitrary input, but instead serves the same content as this JSON API example https://github.com/endpoints/endpoints-example. Note that not all features are the same as the full `endpoints-example`, but most should work.

Right now, this function serves data from static, minified JSON files, so there's no dynamic database underneath.

Kill the server by CTRL+C, ESC or similar.

Examples

```
## Not run:
# start server in another R session
if (interactive()) {
  jsonapi_server()

  # Back in this session ...
  # Connect
  (conn <- jsonapi_connect("http://localhost:8000"))

  # Get data
  conn$url
  conn$version
  conn$content_type
  conn$routes()
```

```
  conn$route("authors")
  conn$route("chapters")
  conn$route("authors/1")
  conn$route("authors/1/books")
  conn$route("chapters/5")
}

## End(Not run)
```

Index