Package ‘zoltr’

May 30, 2019

Title Interface to the ‘Zoltar’ Forecast Repository API

Version 0.2.2

Description ‘Zoltar’ <https://www.zoltardata.com/> is a website that provides a repository of model forecast results in a standardized format and a central location. It supports storing, retrieving, comparing, and analyzing time series forecasts for prediction challenges of interest to the modeling community. This package provides functions for working with the ‘Zoltar’ API, including connecting and authenticating, getting information about projects, models, and forecasts, deleting and uploading forecast data, and downloading scores.


BugReports https://github.com/reichlab/zoltr/issues

License GPL-3

Encoding UTF-8

LazyData true

Suggests testthat, knitr, rmarkdown

Imports httr, jsonlite, readr, mockery, webmockr, base64url

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Matthew Cornell [aut, cre], Nicholas Reich [aut, cph]

Maintainer Matthew Cornell <cornell@umass.edu>

Repository CRAN

Date/Publication 2019-05-30 18:50:03 UTC
R topics documented:

- delete_forecast ................................................. 2
- forecasts ........................................................ 3
- forecast_data ................................................... 3
- forecast_info ................................................... 4
- models ........................................................... 5
- model_info ....................................................... 5
- new_connection ................................................ 6
- projects .......................................................... 7
- project_info ..................................................... 7
- scores ............................................................ 8
- upload_forecast ................................................ 8
- upload_info ..................................................... 9
- zoltar_authenticate ............................................. 10

Index 11

---

**delete_forecast**  
*Delete a forecast*

**Description**

Deletes the forecast with the passed ID. This is permanent and cannot be undone.

**Usage**

```r
decomplete_forecast(zoltar_connection, forecast_id)
```

**Arguments**

- `zoltar_connection`
  
  A ‘ZoltarConnection’ object as returned by `new_connection`

- `forecast_id`
  
  ID of a forecast in `zoltar_connection`'s forecasts

**Value**

None

**Examples**

```r
## Not run:
decomplete_forecast(conn, T6L)
## end(not run)
```

```r
## End(Not run)
```
forecasts

Get a model’s forecasts

Description
Get a model’s forecasts

Usage
forecasts(zoltar_connection, model_id)

Arguments
  zoltar_connection
    A ‘ZoltarConnection’ object as returned by new_connection
  model_id
    ID of a model in zoltar_connection’s models

Value
A ‘data.frame’ of forecast information for the passed model

Examples
## Not run:
  the_forecasts <- forecasts(conn, 26L)
## End(Not run)

forecast_data

Gets a forecast’s data

Description
Gets a forecast’s data

Usage
forecast_data(zoltar_connection, forecast_id, is_json)

Arguments
  zoltar_connection
    A ‘ZoltarConnection’ object as returned by new_connection
  forecast_id
    ID of a forecast in zoltar_connection’s forecasts
  is_json
    A boolean specifying whether the forecast is in ‘list’ or ‘data.frame’ format
forecast_info

Value

Forecast data in the requested format - either a ‘list’ or a ‘data.frame’

Examples

```r
## not run:
forecast_data_json <- forecast_data(connL, 46L, is_json=TRUE)
forecast_data_csv <- forecast_data(connL, 46L, is_json=FALSE)
## End(not run)
```

Description

Gets a forecast’s information

Usage

```r
forecast_info(zoltar_connection, forecast_id)
```

Arguments

- `zoltar_connection`: A ‘ZoltarConnection’ object as returned by `new_connection`
- `forecast_id`: ID of a forecast in zoltar_connection’s forecasts

Value

A ‘list’ of forecast information for the passed forecast_id

Examples

```r
## not run:
the_forecast_info <- forecast_info(connL, 46L)
## End(Not run)
```
models

Get a project's models

Description
Get a project's models

Usage
models(zoltar_connection, project_id)

Arguments
zoltar_connection
   A ‘ZoltarConnection’ object as returned by new_connection
project_id
   ID of a project in zoltar_connection's projects

Value
A 'data.frame' of model contents for all models in the passed project

Examples
## not run:
the_models <- models(conn, 4L)
## End(Not run)

model_info

Get information about a model

Description
Get information about a model

Usage
model_info(zoltar_connection, model_id)

Arguments
zoltar_connection
   A ‘ZoltarConnection’ object as returned by new_connection
model_id
   ID of a model in zoltar_connection's models
new_connection

Value

A `list` of model information for the passed model_id

Examples

```r
## not run:
the_model_info <- model_info(conn, 26L)

## End(Not run)
```

new_connection

Get a connection to a Zoltar host

Description

Returns a new connection object, which is the starting point for working with the Zoltar API. Once you have the connection you can call `zoltar_authenticate` on it, and then call `projects` to get a list of Project objects to start working with.

Usage

```r
new_connection(host = "https://zoltardata.com")
```

Arguments

- `host`: The Zoltar site to connect to. Defaults to `https://zoltardata.com`

Value

A `ZoltarConnection` object

Examples

```r
## Not run:
conn <- new_connection()

## End(Not run)
```
projects

Get information about all projects

Description
Get information about all projects

Usage
projects(zoltar_connection)

Arguments
zoltar_connection
A ‘ZoltarConnection’ object as returned by new_connection

Value
A ‘data.frame’ of all projects’ contents

Examples
## Not run:
the_projects <- projects(conn)
## End(Not run)

project_info

Get information about a project

Description
Get information about a project

Usage
project_info(zoltar_connection, project_id)

Arguments
zoltar_connection
A ‘ZoltarConnection’ object as returned by new_connection
project_id
ID of a project in zoltar_connection’s projects

Value
A ‘list’ of project information for the passed project_id
## Not run:

```r
the_project_info <- project_info(connL Tl)
```

## End(Not run)

### scores

Get a project’s scores

### Usage

```r
scores(zoltar_connection, project_id)
```

### Arguments

- `zoltar_connection`  
  A `ZoltarConnection` object as returned by `new_connection`
- `project_id`  
  ID of a project in `zoltar_connection`’s projects

### Value

A `data.frame` of score data for all models in the passed project ID

### Examples

```r
## Not run:
the_scores <- scores(connL Tl)
## End(Not run)
```

## upload_forecast

Upload a forecast

### Description

This function submits a forecast file to the server for uploading. Returns an UploadFileJob object that can be used to track the upload’s progress. (Uploads are processed in a queue, which means they are delayed until their turn comes up, which depends on the number of current uploads in the queue. Zoltar tracks these via `UploadFileJob` objects.)
**upload_info**

**Usage**

```r
upload_forecast(zoltar_connection, model_id, timezero_date, forecast_csv_file)
```

**Arguments**

- **zoltar_connection**
  - A `ZoltarConnection` object as returned by `new_connection`
- **model_id**
  - ID of a model in `zoltar_connection`'s projects
- **timezero_date**
  - The date of the project timezero you are uploading for. It is a string in format `YYYYMMDD`
- **forecast_csv_file**
  - A CSV file in the Zoltar standard format - see [https://www.zoltardata.com/docs#forecasts](https://www.zoltardata.com/docs#forecasts)

**Value**

An `UploadFileJob` id for the upload

**Examples**

```r
## Not run:
upload_file_job_id <- upload_forecast(conn, 26L, "20170117", "/tmp/EN1-KoTsarma-2017-01-17.csv")
## End(Not run)
```

---

**upload_info**  
*Get an upload's information*

**Description**

Gets an upload’s information that can be used to track the upload’s progress. (Uploads are processed in a queue,

**Usage**

```r
upload_info(zoltar_connection, upload_file_job_id)
```

**Arguments**

- **zoltar_connection**
  - A `ZoltarConnection` object as returned by `new_connection`
- **upload_file_job_id**
  - ID of a job in `zoltar_connection` that was uploaded via `upload_forecast`
Value
A `list` of upload information for the passed `upload_file_job_id`. It has these names: `id`, `url`, `status`, `user`, `created_at`, `updated_at`, `failure_message`, `filename`, `input_json`, `output_json`.

Examples
```r
## not run:
the_upload_info <- upload_info(conn, 287L)

## End(Not run)
```

---

**zoltar_authenticate**: Log in to a Zoltar host

Description
Returns a new `ZoltarConnection` object, which is the starting point for working with the Zoltar API. Once you have the connection you can call `zoltar_authenticate()` on it, and call `projects()` to get a list of objects to start working with.

Usage
```
zoltar_authenticate(zoltar_connection, username, password)
```

Arguments
- `zoltar_connection`: A `ZoltarConnection` object as returned by `new_connection`.
- `username`: Username for the account to use on the connection’s host.
- `password`: Password `""`.

Value
None

Examples
```r
## Not run:
zoltar_authenticate(conn, "USERNAME", "PASSWORD")

## End(Not run)
```
Index

delete_forecast, 2
forecast_data, 3
forecast_info, 4
forecasts, 3
model_info, 5
models, 5
new_connection, 2–5, 6, 7–10
project_info, 7
projects, 6, 7
scores, 8
upload_forecast, 8, 9
upload_info, 9
zoltar_authenticate, 6, 10