Package 'zoltr'

May 8, 2026

Title Interface to the 'Zoltar' Forecast Repository API

Version 1.0.2

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 meta information (projects, models, and forecasts, and truth), and uploading, downloading, and deleting forecast and truth data.

URL https://github.com/reichlab/zoltr , http://reichlab.io/zoltr/

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

License GPL-3

Encoding UTF-8

Suggests testthat, knitr, rmarkdown, webmockr, mockery

Imports data.table, httr, jsonlite, readr, base64url, dplyr, MMWRweek, utils, rlang, magrittr, lubridate

RoxygenNote 7.3.1

VignetteBuilder knitr

NeedsCompilation no

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

Maintainer Matthew Cornell <cornell@umass.edu>

Repository CRAN

Date/Publication 2025-03-06 16:20:01 UTC

Contents

busy_poll_job *Poll job's status*

Description

A convenience function that polls the passed Job's status waiting for either FAILED, TIMEOUT, or SUCCESS.

Usage

```
busy_poll_job(zoltar_connection, job_url, verbose = TRUE)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
job_url	URL of a valid job in zoltar_connection
verbose	if TRUE, print messages on job status poll

Examples

```
## Not run:
  busy_poll_job(conn, "http://example.com/api/job/2/")

## End(Not run)
```

create_model *Create a model*

Description

Creates the model in the passed project using the passed list. Fails if a model with the passed name already exists.

Usage

```
create_model(zoltar_connection, project_url, model_config)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
project_url	url of a project in zoltar_connection's projects. this is the project the new model will be created in
model_config	A list containing a Zoltar model configuration. An example: example-model-config.json . Full documentation at https://docs.zoltardata.com/.

Value

model_url of the newly-created model

Examples

```
## Not run:
  new_model_url <- create_model(conn, "https://www.zoltardata.com/api/project/9/",
                      jsonlite::read_json("example-model-config.json"))

## End(Not run)
```

`create_project` *Create a project*

Description

Creates the project using the passed project configuration list. Fails if a project with the passed name already exists.

Usage

```
create_project(zoltar_connection, project_config)
```

Arguments

`zoltar_connection`	A `ZoltarConnection` object as returned by `new_connection()`
`project_config`	A `list` containing a Zoltar project configuration. note that this list validated by the server and not here. An example: cdc-project.json Full documentation at https://docs.zoltardata.com/.

Value

project_url of the newly-created project

Examples

```
## Not run:
  new_project_url <- create_project(conn, jsonlite::read_json("cdc-project.json"))

## End(Not run)
```

create_timezero *Create a timezero*

Description

Creates the timezero in the passed project using the passed list. Fails if a timezero with the passed timezero_date already exists.

Usage

```
create_timezero(
  zoltar_connection,
  project_url,
  timezero_date,
  data_version_date = NULL,
  is_season_start = FALSE,
  season_name = ""
)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
project_url	url of a project in zoltar_connection's projects. this is the project the new timezero will be created in
timezero_date	The timezero's date in YYYY-MM-DD format as documented at https://docs.zoltardata.com/fileformats/#creation-configuration-json
data_version_date	Optional data version date in the same format. Pass NULL if the timezero does not have one
is_season_start	TRUE if this starts a season, and FALSE otherwise
season_name	Applicable when is_season_start is true, names the season, e.g., "2010-2011"

Value

model_url of the newly-created timezero

Examples

```
## Not run:
  new_timezero_url <- create_timezero(conn, "https://www.zoltardata.com/api/project/9/",
                      "2022-11-08", "2022-11-09", TRUE, "2010-2011")

## End(Not run)
```

data_frame_from_forecast_data

Converts forecast data from Zoltar's native `list` *format to a* `data.frame`

Description

Converts forecast data from Zoltar's native `list` format to a `data.frame`

Usage

```
data_frame_from_forecast_data(forecast_data)
```

Arguments

`forecast_data`	Forecast data as a `list` in the Zoltar standard format

Value

A `data.frame` from forecast_data in zoltar-specific format. The columns are: 'unit', 'target', 'class', 'value', 'cat', 'prob', 'sample', 'quantile', 'family', 'param1', 'param2', 'param3'. They are documented at https://docs.zoltardata.com/fileformats/#forecast-data-format-csv . NB: columns are all character (i.e., data type information from forecast_data is lost). Also note that a retracted prediction element is represented as a single row with NA values for all but the first four columns.

Examples

```
## Not run:
  forecast_data <- jsonlite::read_json("docs-predictions.json")
  data_frame <- data_frame_from_forecast_data(forecast_data)

## End(Not run)
```

delete_forecast

Delete a forecast

Description

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

Usage

```
delete_forecast(zoltar_connection, forecast_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
forecast_url	URL of a forecast in zoltar_connection's forecasts

Value

A Job URL for the deletion

Examples

```
## Not run:
  delete_forecast(conn, "http://example.com/api/forecast/1/")

## End(Not run)
```

---

delete_model *Delete a model*

Description

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

Usage

```
delete_model(zoltar_connection, model_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
model_url	URL of a model in zoltar_connection's models

Value

None

Examples

```
## Not run:
  delete_model(conn, "http://www.zoltardata.com/api/model/1/")

## End(Not run)
```

delete_project *Delete a project*

Description

Deletes the project with the passed URL. This is permanent and cannot be undone.

Usage

```
delete_project(zoltar_connection, project_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
project_url	URL of a project in zoltar_connection's projects

Value

None

Examples

```
## Not run:
  delete_project(conn, "https://www.zoltardata.com/api/project/9/")

## End(Not run)
```

download_forecast *Gets a forecast's data*

Description

Gets a forecast's data

Usage

```
download_forecast(zoltar_connection, forecast_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
forecast_url	URL of a forecast in zoltar_connection's forecasts

Value

Forecast data as a `list` in the Zoltar standard format. meta information is ignored. Full documentation at https://docs.zoltardata.com/.

Examples

```
## Not run:
  forecast_data <- download_forecast(conn, "http://example.com/api/forecast/1/")

## End(Not run)
```

`do_zoltar_query` *A convenience function to construct and execute a Zoltar query for either forecast or truth data.*

Description

A convenience function to construct and execute a Zoltar query for either forecast or truth data.

Usage

```
do_zoltar_query(
  zoltar_connection,
  project_url,
  query_type,
  models = NULL,
  units = NULL,
  targets = NULL,
  timezeros = NULL,
  types = NULL,
  as_of = NULL,
  verbose = TRUE
)
```

Arguments

`zoltar_connection`	A `ZoltarConnection` object as returned by `new_connection()`
`project_url`	URL of a project in zoltar_connection's projects
`query_type`	A character indicating the type of query to run. Must be one of: "forecasts" or "truth".
`models`	Character vector of model abbreviations. Used for query_type = "forecasts".
`units`	Character vector of units to retrieve. Used for all query_types.
`targets`	Character vector of targets to retrieve. Used for all query_types.
`timezeros`	Character vector of timezeros to retrieve in YYYY_MM_DD_DATE_FORMAT, e.g., '2017-01-17'. Used for all query_types.

types	Character vector of prediction types to retrieve. Used for query_type = "forecasts".
as_of	a datetime used for either query_type that constrains based on forecast issued_at. must be a datetime as parsed by the dateutil python library https://dateutil.readthedocs.io/en/stable/index.h , which accepts a variety of styles.
verbose	if TRUE, print messages on job status poll

Value

A data.frame of Job's data. Full documentation at https://docs.zoltardata.com/.

Examples

```
## Not run:
  forecast_data <- do_zoltar_query(
    conn, "https://www.zoltardata.com/api/project/44/", "forecasts",
    models=c("CMU-TimeSeries", "UMass-MechBayes"), units=c("01003", "US"),
    targets=c("1 wk ahead inc death"), targets=c("2020-07-19", "2020-07-20"),
    types=c("quantile"), as_of="2020-07-10")
  truth_data <- do_zoltar_query(
    conn, "https://www.zoltardata.com/api/project/44/", "truth", c("01003", "US"),
    c("1 wk ahead inc death"), c("2020-07-19", "2020-07-20"))

## End(Not run)
```

---

edit_model *Edit a model*

Description

Edits the model in the passed project using the passed list. Fails if a model with the passed name already exists.

Usage

```
edit_model(zoltar_connection, model_url, model_config)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
model_url	url of a project in zoltar_connection's projects. this is the project the new model will be edited in
model_config	A list containing a Zoltar model configuration. An example: example-model-config.json . Full documentation at https://docs.zoltardata.com/.

Examples

```
## Not run:
  edit_model(conn, "https://www.zoltardata.com/api/model/2/",
    jsonlite::read_json("example-model-config.json"))

## End(Not run)
```

---

forecasts *Get a model's forecasts*

Description

Get a model's forecasts

Usage

```
forecasts(zoltar_connection, model_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
model_url	URL 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, "http://www.zoltardata.com/api/model/1/")

## End(Not run)
```

---

forecast_data_from_cdc_csv_file *Loads and converts a CDC CSV file to Zoltar's native* list *format*

Description

Loads and converts a CDC CSV file to Zoltar's native list format

Usage

```
forecast_data_from_cdc_csv_file(season_start_year, cdc_csv_file)
```

Arguments

season_start_year	An integer specifying the "season" that cdc_csv_file is in. Used to convert EWs to YYYY_MM_DD_DATE_FORMAT. **zoltr** uses week 30 as the season breakpoint, e.g. the "2016/2017 season" starts with
cdc_csv_file	A CDC CSV file

Value

cdc_csv_file's data as Zoltar's native `list` format, but only the "predictions" item, and not "meta"

Examples

```
## Not run:
  forecast_data <- forecast_data_from_cdc_csv_file(2016, "my_forecast.cdc.csv")

## End(Not run)
```

---

forecast_data_from_cdc_data_frame *forecast_data_from_cdc_csv_file() helper*

Description

forecast_data_from_cdc_csv_file() helper

Usage

```
forecast_data_from_cdc_data_frame(season_start_year, cdc_data_frame)
```

Arguments

season_start_year	as passed to forecast_data_from_cdc_csv_file()
cdc_data_frame	""

Value

same as forecast_data_from_cdc_csv_file()

forecast_info *Gets a forecast's information*

Description

Gets a forecast's information

Usage

```
forecast_info(zoltar_connection, forecast_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
forecast_url	URL of a forecast in zoltar_connection's forecasts

Value

A list of forecast information for the passed forecast_url

Examples

```
## Not run:
  the_forecast_info <- forecast_info(conn, "http://example.com/api/forecast/1/")

## End(Not run)
```

get_resource *Get JSON for a resource (URL). Authenticates if necessary*

Description

Get JSON for a resource (URL). Authenticates if necessary

Usage

```
get_resource(zoltar_connection, url, col_types = NULL)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
url	A string of the resource's URL
col_types	Same as readr::read_csv takes

Value

A `list` that contains JSON information for the passed URL

`job_data` *Gets a job's file's data*

Description

Downloads the data for jobs that have an associated file, such as a query's results. Called on Jobs that are the results of a project forecast or truth queries via `submit_query()`. NB: It is a 404 Not Found error if this is called on a Job that has no underlying S3 data file, which can happen b/c: 1) 24 hours has passed (the expiration time) or 2) the Job is not complete and therefore has not saved the data file. For the latter you may use `busy_poll_job()` to ensure the job is done.

Usage

```
job_data(zoltar_connection, job_url, query_type)
```

Arguments

`zoltar_connection`	A `ZoltarConnection` object as returned by `new_connection()`
`job_url`	URL of a valid job in zoltar_connection that has a data file associated with it
`query_type`	A character indicating the type of query to run. Must be one of: "forecasts" or "truth".

Value

A `data.frame` of Job's data. The columns depend on query_type - see https://docs.zoltardata.com/fileformats/#truth-data-format-csv and https://docs.zoltardata.com/fileformats/#forecast-data-format-csv.

Examples

```
## Not run:
  the_job_data <- job_data(conn, "http://example.com/api/job/2/")

## End(Not run)
```

job_info *Get a job's information*

Description

Gets a job's information that can be used to track the job's progress. Jobs represent long-running asynchronous activities like uploading a file (e.g., a forecast or truth) or running a query.

Usage

```
job_info(zoltar_connection, job_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
job_url	URL of a valid job in zoltar_connection

Value

A list of job information for the passed job_url. it has these names: id, url, status, user, created_at, updated_at, failure_message, input_json, output_json

Examples

```
## Not run:
  the_job_info <- job_info(conn, "http://example.com/api/job/2/")

## End(Not run)
```

job_info_forecast_url *Get a new forecast upload's url*

Description

A helper function for jobs representing file uploads. Returns the URL of a newly-uploaded forecast from job_info.

Usage

```
job_info_forecast_url(zoltar_connection, the_job_info)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
the_job_info	a list object as returned by job_info()

Value

A URL of the new forecast

Examples

```
## Not run:
  new_forecast_url <- job_info_forecast_url(conn, "http://example.com/api/job/2/")

## End(Not run)
```

`latest_forecasts` *Get a project's latest_forecasts*

Description

Get a project's latest_forecasts

Usage

```
latest_forecasts(zoltar_connection, project_url)
```

Arguments

`zoltar_connection`	A `ZoltarConnection` object as returned by `new_connection()`
`project_url`	URL of a project in zoltar_connection's projects

Value

A `data.frame` of all of the latest forecasts for the passed project. columns: forecast_id, source. (Later we may generalize to allow passing specific columns to retrieve, such as 'forecast_model_id', 'time_zero_id', 'issued_at', 'created_at', 'source', and 'notes'.)

Examples

```
## Not run:
 the_latest_forecasts <- latest_forecasts(conn, "https://www.zoltardata.com/api/project/9/")

## End(Not run)
```

models *Get a project's models*

Description

Get a project's models

Usage

```
models(zoltar_connection, project_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
project_url	URL 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, "https://www.zoltardata.com/api/project/9/")

## End(Not run)
```

model_info *Get information about a model*

Description

Get information about a model

Usage

```
model_info(zoltar_connection, model_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
model_url	URL of a model in zoltar_connection's models

Value

A `list` of model information for the passed model_url

Examples

```
## Not run:
  the_model_info <- model_info(conn, "http://www.zoltardata.com/api/model/1/")

## 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

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

Arguments

`host`	The Zoltar site to connect to. Does *not* include a trailing slash ('/'). Defaults to `https://zoltardata.com`

Details

A note on URLs: We require a trailing slash ('/') on all URLs. The only exception is the host arg passed to this function. This convention matches Django REST framework one, which is what Zoltar is written in.

Value

A `ZoltarConnection` object

Examples

```
## 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_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
project_url	URL of a project in zoltar_connection's projects

Value

A list of project information for the passed project_url

Examples

```
## Not run:
  the_project_info <- project_info(conn, "https://www.zoltardata.com/api/project/9/")

## End(Not run)
```

`quantile_data_frame_from_forecast_data`
Converts forecast data from Zoltar's native `list` *format to a quantile* `data.frame`

Description

Converts forecast data from Zoltar's native `list` format to a quantile `data.frame`

Usage

```
quantile_data_frame_from_forecast_data(forecast_data)
```

Arguments

`forecast_data`	Forecast data as a `list` in the Zoltar standard format

Value

A `data.frame` from forecast_data that's the same as `data_frame_from_forecast_data()` does except only includes point and quantile rows, and with this header: 'location', 'target', 'type', 'quantile', 'value', i.e., 'unit' -> 'location' and 'class' -> 'type'

Examples

```
## Not run:
  forecast_data <- jsonlite::read_json("docs-predictions.json")
  data_frame <- quantile_data_frame_from_forecast_data(forecast_data)

## End(Not run)
```

`submit_query` *Submit a query*

Description

Submits a request for the execution of a query of either forecasts or truth in this Project.

Usage

```
submit_query(zoltar_connection, project_url, query_type, query)
```

Arguments

`zoltar_connection`	A `ZoltarConnection` object as returned by `new_connection()`
`project_url`	URL of a project in zoltar_connection's projects
`query_type`	A character indicating the type of query to run. Must be one of: "forecasts" or "truth".
`query`	A `list` of character `lists` that constrains the queried data. It is the analog of the JSON object documented at `https://docs.zoltardata.com/`. The keys vary depending on query_type. References to models, units, targets, and timezeros are strings that name the objects, and not IDs.

Value

a Job URL for tracking the query and getting its results when it successfully completes

Examples

```
## Not run:
  job_url <- submit_query(conn, "https://www.zoltardata.com/api/project/9/", "forecasts",
                     list("models"=list("60-contact", "CovidIL_100"), "units"=list("US"),
                  "targets"=list(1894, 1897), "timezeros"=list("2020-05-14", "2020-05-09"),
                           "types"=list("point", "quantile")))

## End(Not run)
```

`targets` *Get a project's targets*

Description

Get a project's targets

Usage

```
targets(zoltar_connection, project_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
project_url	URL of a project in zoltar_connection's projects

Value

A data.frame of target contents for the passed project

Examples

```
## Not run:
  the_targets <- targets(conn, "https://www.zoltardata.com/api/project/9/")

## End(Not run)
```

---

target_info *Get information about a target*

Description

Get information about a target

Usage

```
target_info(zoltar_connection, target_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
target_url	URL of a target in zoltar_connection's targets

Value

A list of target information for the passed target_url

Examples

```
## Not run:
  the_target_info <- target_info(conn, "https://www.zoltardata.com/api/target/3/")

## End(Not run)
```

timezeros *Get a project's timezeros*

Description

Get a project's timezeros

Usage

```
timezeros(zoltar_connection, project_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
project_url	URL of a project in zoltar_connection's projects

Value

A data.frame of timezero contents for the passed project

Examples

```
## Not run:
  the_timezeros <- timezeros(conn, "https://www.zoltardata.com/api/project/9/")

## End(Not run)
```

timezero_info *Get information about a timezero*

Description

Get information about a timezero

Usage

```
timezero_info(zoltar_connection, timezero_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
timezero_url	URL of a timezero in zoltar_connection's timezeros

Value

A `list` of timezero information for the passed timezero_url

Examples

```
## Not run:
  the_timezero_info <- timezero_info(conn, "https://www.zoltardata.com/api/timezero/3/")

## End(Not run)
```

`truth_info` *Get information about a project's truth*

Description

Get information about a project's truth

Usage

```
truth_info(zoltar_connection, project_url)
```

Arguments

`zoltar_connection`	A `ZoltarConnection` object as returned by `new_connection()`
`project_url`	URL of a project in zoltar_connection's projects

Value

A `list` of project truth information for the passed project_url

Examples

```
## Not run:
  the_truth_info <- truth_info(conn, "https://www.zoltardata.com/api/project/9/")

## End(Not run)
```

unit_info *Get information about a unit*

Description

Get information about a unit

Usage

```
unit_info(zoltar_connection, unit_url)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
unit_url	URL of a unit in zoltar_connection's zoltar_units

Value

A list of unit information for the passed unit_url

Examples

```
## Not run:
  the_unit_info <- unit_info(conn, "https://www.zoltardata.com/api/unit/3/")

## End(Not run)
```

upload_forecast *Upload a forecast*

Description

This function submits forecast data to the server for uploading. Returns a Job object that can be used to up, which depends on the number of current uploads in the queue. Zoltar tracks these via Job objects.)

Usage

```
upload_forecast(
  zoltar_connection,
  model_url,
  timezero_date,
  forecast_data,
  is_json = TRUE,
  notes = ""
)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
model_url	URL 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_data	Forecast data to upload data to upload, either a list (if is_json==TRUE) or a dataframe otherwise. formats are documented at https://docs.zoltardata.com/
is_json	TRUE if forecast_data is JSON (list) format, and FALSE if it is CSV (dataframe) format
notes	Optional user notes for the new forecast

Value

A Job URL for the upload

Examples

```
## Not run:
  forecast_data <- jsonlite::read_json("docs-predictions.json")
  job_url <- upload_forecast(conn, "http://www.zoltardata.com/api/model/1/",
                             "2017-01-17", forecast_data, TRUE, "a mid-January forecast")

## End(Not run)
```

---

upload_truth *Upload truth data*

Description

Uploads the data in truth_csv_file to the project identified by project_url.

Usage

```
upload_truth(zoltar_connection, project_url, truth_csv_file, issued_at = NULL)
```

Arguments

zoltar_connection	A ZoltarConnection object as returned by new_connection()
project_url	URL of a project in zoltar_connection's projects
truth_csv_file	A CSV file as documented at https://docs.zoltardata.com/fileformats/#truth-data-format-csv
issued_at	optional datetime to use for the uploaded truth forecasts' issued_at value in ISO 8601 format. NB: it must include timezone information. (the default issued_at is the time of upload.) the value must obey the constraints documented at https://docs.zoltardata.com/forecastversions/#forecast-version-rules

Value

A Job URL for the upload

Examples

```
## Not run:
  job_url <- upload_truth(conn, "http://www.zoltardata.com/api/project/1/", "truth.csv")

## 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

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

## End(Not run)
```

`zoltar_units` *Get a project's zoltar_units*

Description

Get a project's zoltar_units

Usage

```
zoltar_units(zoltar_connection, project_url)
```

Arguments

`zoltar_connection`	A `ZoltarConnection` object as returned by `new_connection()`
`project_url`	URL of a project in zoltar_connection's projects

Value

A `data.frame` of unit contents for the passed project

Examples

```
## Not run:
  the_units <- zoltar_units(conn, "https://www.zoltardata.com/api/project/9/")

## End(Not run)
```

Index