Package: orderlyweb (via r-universe)

October 14, 2026

Title Orderly Support for 'OrderlyWeb'

Version 0.1.16

Description Client for 'OrderlyWeb' for use from the 'orderly1' package. Allows downloading of reports, running remote reports and other interaction with the remote report repository.

Imports R6, gert, httr, jsonlite, orderly1 (>= 1.2.19), progress, zip

Suggests mockery, testthat

RoxygenNote 6.1.1

License MIT + file LICENSE

URL https://github.com/vimc/orderlyweb

BugReports https://github.com/vimc/orderlyweb/issues

Encoding UTF-8

LazyData true

Remotes vimc/orderly1

Config/pak/sysreqs cmake git make libgit2-dev libuv1-dev libssl-dev

Repository https://ncov-ic.r-universe.dev

Date/Publication 2025-07-14 14:41:48 UTC

RemoteUrl https://github.com/vimc/orderlyweb

RemoteRef master

RemoteSha e98a44f6c8f0769133037294678a5239ef3be3b3

Contents

orderlyweb 2
orderlyweb_api_client 2
orderlyweb_remote 3

Index **5**

orderlyweb *Create an orderlyweb object*

Description

Create an orderlyweb object, for interaction with the OrderlyWeb API at a high level. For low level access, see `orderlyweb_api_client` or the `$api_client` field of the created object.

Usage

```
orderlyweb(..., api_client = NULL)
```

Arguments

`...`	Arguments passed through to `orderlyweb_api_client`
`api_client`	A pre-constructed `orderlyweb_api_client`, in which case `...` is ignored

Examples

```
cl <- orderlyweb::orderlyweb(host = "example.com", port = 443,
                             token = "mytoken")
cl$api_client$is_authorised()
```

orderlyweb_api_client *Create a low-level OrderlyWeb client*

Description

Create a low-level HTTP API client for use with OrderlyWeb.

Usage

```
orderlyweb_api_client(host, port, token, name = NULL, https = TRUE,
  prefix = NULL, api_version = 2, insecure = FALSE,
  verbose = FALSE)
```

Arguments

`host`	Fully qualified hostname for the OrderlyWeb instance
`port`	Port to use
`token`	Your application token for authentication. The appropriate value here will depend on the authentication support that is built into the OrderlyWeb server that you are communicating with. Provide the token directly (as a string) or provide a callback function that takes no arguments and returns token.

name	A friendly name for the server (e.g, "production" or "testing") which may be printed when using the remote, or when authenticating. If not provided then a name will be constructed from host, port and (if provided) prefix.
https	Optional logical, indicating if this is an https connection - this should be TRUE in all production settings or credentials will be sent in the clear!
prefix	A prefix, if your OrderlyWeb server is mounted at a path within some larger website.
api_version	The API version to request - this should be left as 2.
insecure	Avoid SSL certificate testing - this is completely insecure (as bad as http) and exists only for testing.
verbose	Be verbose in all http exchanges. This will be very noisy.

Examples

```
cl <- orderlyweb::orderlyweb_api_client(host = "example.com", port = 443,
                                        token = "mytoken")
cl$is_authorised()
```

---

orderlyweb_remote *Create orderly remote*

Description

Implements an orderly "remote" using OrderlyWeb as a backend. Use this within an orderly_config.yml configuration.

Usage

```
orderlyweb_remote(host, port, token, https = TRUE, prefix = NULL,
  name = NULL)
```

Arguments

host	Fully qualified hostname for the OrderlyWeb instance
port	Port to use
token	Your application token for authentication. The appropriate value here will depend on the authentication support that is built into the OrderlyWeb server that you are communicating with. Provide the token directly (as a string) or provide a callback function that takes no arguments and returns token.
https	Optional logical, indicating if this is an https connection - this should be TRUE in all production settings or credentials will be sent in the clear!
prefix	A prefix, if your OrderlyWeb server is mounted at a path within some larger website.
name	A friendly name for the server (e.g, "production" or "testing") which may be printed when using the remote, or when authenticating. If not provided then a name will be constructed from host, port and (if provided) prefix.

Examples

```
remote <- orderlyweb::orderlyweb_remote("example.com", 443, "mytoken")
remote
```

Index

orderlyweb, 2
orderlyweb_api_client, 2, *2*
orderlyweb_remote, 3