Package ‘lobsteR’

March 3, 2026

Type Package

Title Access and Process 'LOBSTER' High-Frequency Data

Version 1.0.0

Description Provides tools to authenticate with 'LOBSTER' (Limit Order Book System - The Efficient Reconstruction, <`https://app.lobsterdata.com/`>), request, download, and process high-frequency limit order book data. Streamlines the end-to-end workflow from data request to analysis-ready datasets. For advanced high-frequency econometric analysis, see the 'highfrequency' package.

Depends R (>= 4.2.0)

License MIT + file LICENSE

URL `https://github.com/voigtstefan/lobsteR`,
`https://voigtstefan.github.io/lobsteR/`

BugReports `https://github.com/voigtstefan/lobsteR/issues`

Imports archive, assertthat, callr, dplyr, httr, lubridate, purrr, rvest, timeDate

Suggests knitr, rmarkdown, testthat (>= 3.0.0), xml2

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Karol Kulma [aut],
Stefan Voigt [aut, cre, cph]

Maintainer Stefan Voigt <`stefan.voigt@econ.ku.dk`>

Repository CRAN

Date/Publication 2026-03-03 09:50:02 UTC

Contents

`account_archive` *Retrieve LOBSTER data archive information*

Description

Fetches information about available datasets in your LOBSTER account archive. This includes details about symbols, date ranges, order book levels, file sizes, and download links for each available dataset.

Usage

```
account_archive(account)
```

Arguments

`account`	list Output from `account_login()` containing a valid authenticated session (`valid == TRUE`).

Details

The function navigates to the archive page of the authenticated session, scrapes the archive table, extracts download links, and returns a structured tibble. Only datasets with non-zero file sizes are included — datasets still being processed by LOBSTER will not appear until processing is complete.

Value

A tibble with one row per available dataset and columns:

id integer. Unique identifier for each dataset.

symbol character. Stock/ETF ticker symbol (e.g., `"SPY"`, `"AAPL"`).

start_date Date. First date of data coverage.

end_date Date. Last date of data coverage.

level integer. Order book depth (number of price levels).

size integer. File size in bytes.

download character. Direct download URL for the dataset.

Rows are ordered by `id` descending (most recently requested first). Datasets with zero file size (not yet processed by LOBSTER) are excluded.

See Also

account_login(), data_download()

Examples

```
## Not run:
acct <- account_login(
  login = Sys.getenv("LOBSTER_USER"),
  pwd   = Sys.getenv("LOBSTER_PWD")
)

archive <- account_archive(acct)
archive
#> # A tibble: 3 × 7
#>      id symbol start_date end_date   level    size download
#>   <int> <chr>  <date>     <date>     <int>   <int> <chr>
#> 1   102 AAPL   2023-01-03 2023-01-03     1  204800 https://. . .
#> 2   101 MSFT   2023-01-03 2023-01-05     2  512000 https://. . .
#> 3   100 SPY    2022-12-01 2022-12-31    10 1048576 https://. . .

# Filter to a single symbol before downloading
data_download(
  requested_data = archive[archive$symbol == "AAPL", ],
  account_login  = acct,
  path           = "data-lobster"
)

## End(Not run)
```

account_login *Authenticate with LOBSTER account*

Description

Logs into your LOBSTER account and creates a session object for subsequent data requests. This function handles the authentication process with lobsterdata.com and validates the login was successful.

Usage

```
account_login(login, pwd)
```

Arguments

login	character(1) Email address associated with the LOBSTER account.
pwd	character(1) Account password.

Details

The function submits the sign-in form using an AJAX header (`x-requested-with: XMLHttpRequest`) and confirms success by checking the redirect URL. Network connectivity and valid credentials are required.

Store credentials in your `.Renviron` file (`usethis::edit_r_environ()`) to avoid hardcoding them in scripts:

```
LOBSTER_USER=you@example.com
LOBSTER_PWD=your-password
```

Value

A named list with components:

valid logical(1) — `TRUE` when authentication succeeded.

session rvest session object used for further navigation.

submission rvest response returned after the sign-in form was submitted.

See Also

`account_archive()`, `request_submit()`

Examples

```
## Not run:
acct <- account_login(
  login = Sys.getenv("LOBSTER_USER"),
  pwd   = Sys.getenv("LOBSTER_PWD")
)

if (acct$valid) {
  # Retrieve available datasets in the archive
  archive <- account_archive(acct)

  # Build and submit a new data request
  req <- request_query("AAPL", "2023-01-03", "2023-01-05", level = 1)
  request_submit(acct, req)
}

## End(Not run)
```

data_download *Download requested archive files*

Description

Download one or more files listed in `requested_data` using the authenticated session in `account_login`. Files are written to `path`. The file write and optional extraction are performed in a background R process (via `callr::r_bg()`). If `unzip = TRUE` the original `.7z` archive is removed after extraction.

Usage

```
data_download(requested_data, account_login, path = ".", unzip = TRUE)
```

Arguments

`requested_data`	data.frame A tibble with archive metadata that must include at minimum a `download` column (full download URL) and an `id` column. Typically a (filtered) result from `account_archive()`.
`account_login`	list Output from `account_login()` containing the authenticated session used to fetch file content.
`path`	character(1) Directory where downloaded files will be written and (if `unzip = TRUE`) extracted. The directory must already exist; create it first with `dir.create(path, recursive = TRUE)` if needed.
`unzip`	logical(1) If `TRUE` (default) extract the downloaded `.7z` archive using `archive::archive_extract()` and delete the archive file afterwards. Set to `FALSE` to keep the raw archive.

Details

For each row in `requested_data` the function fetches the file content via the authenticated session and spawns a background process to write and optionally extract the file. Because extraction runs in the background, the function returns before the files are fully written to disk.

Value

Invisibly returns `NULL`. Files are written to `path` by background R processes launched via `callr::r_bg()`. These processes are not monitored after launch; verify that the expected files exist in `path` before proceeding with analysis.

See Also

`account_login()`, `account_archive()`

Examples

```
## Not run:
acct    <- account_login(Sys.getenv("LOBSTER_USER"), Sys.getenv("LOBSTER_PWD"))
archive <- account_archive(acct)

# Download all AAPL files to a local directory
dir.create("data-lobster", showWarnings = FALSE)
data_download(
  requested_data = archive[archive$symbol == "AAPL", ],
  account_login  = acct,
  path           = "data-lobster"
)

# Keep the raw .7z archives without extracting
data_download(
  requested_data = archive,
  account_login  = acct,
  path           = "data-lobster",
  unzip          = FALSE
)

## End(Not run)
```

`request_query` *Create a LOBSTER data request (one row per trading day)*

Description

Construct a request describing which trading-day files to ask LOBSTER for. For each symbol and date range the function expands the range to one row per calendar day, converts the level to integer, and (optionally) validates the requested days by removing weekends, NYSE holidays and any days already present in the provided account archive.

Usage

```
request_query(
  symbol,
  start_date,
  end_date,
  level,
  validate = TRUE,
  account_archive = NULL,
  frequency = "1 day"
)
```

Arguments

`symbol`	character vector One or more ticker symbols (e.g. `"AAPL"`). Each element is paired with the corresponding elements of `start_date`, `end_date` and `level`. Recycling follows base R rules; mismatched lengths should be avoided.
`start_date`	Date-like (Date or character) Start date(s) for the requested range(s). Converted with `as.Date()`.
`end_date`	Date-like (Date or character) End date(s) for the requested range(s). Converted with `as.Date()`.
`level`	integer(1) Required order-book snapshot level (e.g. `1`, `2`, `10`).
`validate`	logical(1) If `TRUE` (default) remove weekend days and NYSE holidays. When `account_archive` is also supplied, days already present in the archive are additionally removed to avoid duplicate requests.
`account_archive`	data.frame or tibble, optional Archive table as returned by `account_archive()`. When provided, rows already present in the archive (matched on symbol, start_date, end_date, and level) are excluded.
`frequency`	character(1) Frequency string passed to `seq.Date()`. Defaults to `"1 day"` (one row per trading day). Use `"1 month"` for large date ranges to reduce the number of individual requests sent to the LOBSTER server. Validation (`validate = TRUE`) is only applied when `frequency == "1 day"`.

Details

This function performs no network activity. Use `request_submit()` to send the generated request to an authenticated LOBSTER session.

Value

A data.frame with one row per period and columns:

- `symbol`: character
- `start_date`: Date — start of the period (equal to `end_date` for daily requests)
- `end_date`: Date — end of the period
- `level`: integer

When `validate = TRUE` and `frequency = "1 day"`, weekend days and NYSE holidays are silently removed, so the output typically contains fewer rows than the full calendar span of the requested date range.

See Also

`request_submit()`, `account_archive()`, `account_login()`

Examples

```
# Single symbol, one-week range (weekends and holidays removed automatically)
request_query("AAPL", "2023-01-02", "2023-01-06", level = 1)

# Multiple symbols with paired date ranges
request_query(
  symbol     = c("AAPL", "MSFT"),
  start_date = c("2023-01-03", "2023-02-01"),
  end_date   = c("2023-01-05", "2023-02-03"),
  level      = 1
)

# Monthly frequency for a large date range (no per-day expansion)
request_query(
  symbol     = "SPY",
  start_date = "2022-01-01",
  end_date   = "2022-12-31",
  level      = 10,
  frequency  = "1 month"
)

## Not run:
# Exclude days already in the archive to avoid duplicate requests
acct    <- account_login(Sys.getenv("LOBSTER_USER"), Sys.getenv("LOBSTER_PWD"))
archive <- account_archive(acct)

req <- request_query(
  symbol          = "AAPL",
  start_date      = "2023-01-02",
  end_date        = "2023-01-31",
  level           = 1,
  account_archive = archive
)

## End(Not run)
```

request_submit — *Submit one or more requests to an authenticated LOBSTER account*

Description

Send the prepared request rows to lobsterdata.com using the authenticated session contained in `account_login`. Each row in `request` is submitted as a separate HTTP request. The function performs network side effects and returns invisibly.

Usage

```
request_submit(account_login, request)
```

Arguments

`account_login`	list Output from `account_login()` that contains a successful authenticated session (`valid` == `TRUE`) and the submission response required for navigation.
`request`	data.frame A tibble as returned by `request_query()` with columns: symbol, start_date, end_date, level.

Value

Invisibly returns `NULL`. The primary effect is to queue requests on the LOBSTER server; processing happens server-side and may take some time. Use `account_archive()` afterwards to check when files become available.

See Also

`account_login()`, `request_query()`, `account_archive()`

Examples

```
## Not run:
acct <- account_login(
  login = Sys.getenv("LOBSTER_USER"),
  pwd   = Sys.getenv("LOBSTER_PWD")
)

# Build a request and submit it
req <- request_query("AAPL", "2023-01-03", "2023-01-05", level = 1)
request_submit(acct, req)

# LOBSTER processes the request server-side; this may take several minutes.
# Once done, the files appear in the account archive.
archive <- account_archive(acct)

## End(Not run)
```

Index