

Package ‘investdatar’

August 21, 2026

Title Investment Data Access and Preparation Toolkit

Version 0.1.5

Description A toolkit for loading, transforming, and managing structured investment-related data. Includes functions for retrieving macroeconomic and public financial series, regulatory filings, futures positioning, energy and fiscal data, market open-high-low-close-volume data, fund holdings, narrative feeds, and crypto derivatives from multiple providers. It detects time-series gaps and synchronizes local '.rds' data stores with sidecar metadata for analysis and trading workflows. It also generates structured dataset descriptions for LLM-based analyst agents that produce R code for analysis and visualization.

License MIT + file LICENSE

URL <https://github.com/OliverLDS/investdatar>

BugReports <https://github.com/OliverLDS/investdatar/issues>

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

Imports data.table, jsonlite, curl, httr, xml2, yaml

Suggests binxr, okxr, quantmod, testthat (>= 3.0.0), xts, waldo, wbstats, withr, zoo

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Oliver Zhou [aut, cre]

Maintainer Oliver Zhou <oliver.yxzhou@gmail.com>

Repository CRAN

Date/Publication 2026-08-21 13:40:07 UTC

Contents

add_fred_registry_series	6
add_ishare_registry_ticker	6
add_sec_registry_company	7
add_wbstats_registry_series	8
bootstrap_yahoofinance_registry	9
describe_alphavantage_data	9
describe_bea_data	10
describe_binance_data	10
describe_bls_data	11
describe_census_data	12
describe_cftc_cot_data	12
describe_crypto_derivatives	13
describe_eia_data	13
describe_fiscaldata	14
describe_fred_data	14
describe_ishare_data	15
describe_market_ohlc_data	15
describe_okx_candle_data	16
describe_quantmod_data	16
describe_rss_data	17
describe_sdmx_data	18
describe_sec_companyfacts	18
describe_sec_frame	19
describe_sec_submissions	19
describe_treasury_rates	20
describe_wbstats_data	20
detect_time_gaps	21
detect_time_gaps_fred	21
detect_time_gaps_okx_candle	22
detect_time_gaps_wbstats	22
fetch_quantmod_OHLC	23
get_alphavantage_registry	24
get_alphavantage_registry_file_path	25
get_bea_registry	25
get_bea_registry_file_path	26
get_bls_registry	26
get_bls_registry_file_path	27
get_census_registry	27
get_census_registry_file_path	28
get_cftc_cot_registry	28
get_cftc_cot_registry_file_path	29
get_completed_local_quantmod_OHLC	29
get_crypto_derivatives_registry	30
get_crypto_derivatives_registry_file_path	30
get_eia_registry	31
get_eia_registry_file_path	31

get_fiscaldata_registry	32
get_fiscaldata_registry_file_path	32
get_fred_registry	33
get_fred_registry_file_path	33
get_investdata_config	34
get_ishare_registry	34
get_ishare_registry_file_path	35
get_latest_sync_run	35
get_local_alphavantage_data	36
get_local_bea_data	36
get_local_binance_klines	37
get_local_bls_data	37
get_local_census_data	38
get_local_cftc_cot	38
get_local_crypto_derivatives	39
get_local_data_meta	40
get_local_data_partitioned	40
get_local_data_utime	41
get_local_eia_data	41
get_local_fiscaldata	42
get_local_FRED_data	42
get_local_fund_holdings	43
get_local_ishare_data	43
get_local_ishare_holdings	44
get_local_ishare_holdings_standardized	44
get_local_ishare_mega_data	45
get_local_okx_candle	45
get_local_quantmod_OHLC	46
get_local_rss_data	46
get_local_sdmx_data	47
get_local_sec_companyfacts	47
get_local_sec_frame	48
get_local_sec_submissions	48
get_local_treasury_rates	49
get_local_wbstats_data	49
get_rss_registry	50
get_rss_registry_file_path	50
get_sdmx_registry	51
get_sdmx_registry_file_path	51
get_sec_company_tickers	52
get_sec_frames_registry	52
get_sec_frames_registry_file_path	53
get_sec_registry	53
get_sec_registry_file_path	54
get_source_config	54
get_source_data_alphavantage_ts_daily	55
get_source_data_bea	56
get_source_data_binance_klines	56

get_source_data_bls	57
get_source_data_census	58
get_source_data_cftc_cot	59
get_source_data_crypto_derivatives	60
get_source_data_eia	61
get_source_data_fiscaldata	61
get_source_data_fred	62
get_source_data_ishare	63
get_source_data_ishare_holdings	63
get_source_data_okx_candle	64
get_source_data_path	65
get_source_data_rss	65
get_source_data_sdmx	66
get_source_data_sec_companyfacts	67
get_source_data_sec_frame	68
get_source_data_sec_submissions	69
get_source_data_treasury_rates	70
get_source_data_wbstats	70
get_source_hist_data_okx_candle	71
get_source_metadata_fred	72
get_source_spec	73
get_source_etime_cftc_cot	73
get_source_etime_crypto_derivatives	74
get_source_etime_eia	75
get_source_etime_fiscaldata	75
get_source_etime_fred	76
get_source_etime_ishare	76
get_source_etime_okx_candle	77
get_source_etime_rss	77
get_source_etime_sdmx	78
get_source_etime_sec_companyfacts	79
get_source_etime_sec_submissions	79
get_source_etime_treasury_rates	80
get_source_etime_wbstats	80
get_wbstats_registry	81
get_wbstats_registry_file_path	81
get_yahoofinance_registry	82
get_yahoofinance_registry_file_path	82
get_yahoofinance_seed_registry_path	83
infer_source_etime_from_frequency	83
is_sync_run_successful	84
list_source_specs	84
load_investdata_config	85
repair_local_binance_klines_gaps	85
repair_local_okx_candle_gaps	86
resolve_sec_cik	87
standardize_fund_holdings	87
sync_all_alphavantage_registry_data	88

sync_all_bea_registry_data	89
sync_all_bls_registry_data	89
sync_all_census_registry_data	90
sync_all_cftc_cot_registry_data	91
sync_all_crypto_derivatives_registry_data	91
sync_all_eia_registry_data	92
sync_all_fiscaldata_registry_data	93
sync_all_fred_registry_data	93
sync_all_ishare_registry_data	94
sync_all_ishare_registry_holdings	95
sync_all_rss_registry_data	96
sync_all_sdmx_registry_data	96
sync_all_sec_companyfacts_registry_data	97
sync_all_sec_frames_registry_data	98
sync_all_sec_submissions_registry_data	98
sync_all_treasury_rates	99
sync_all_wbstats_registry_data	100
sync_all_yahoofinance_registry_data	100
sync_local_alphavantage_data	101
sync_local_bea_data	102
sync_local_binance_klines	103
sync_local_bls_data	104
sync_local_census_data	105
sync_local_cftc_cot	106
sync_local_crypto_derivatives	107
sync_local_crypto_liquidations	108
sync_local_data	108
sync_local_data_batches	109
sync_local_data_partitioned	110
sync_local_eia_data	111
sync_local_fiscaldata	112
sync_local_fred_data	113
sync_local_fund_holdings	113
sync_local_ishare_data	114
sync_local_ishare_holdings	115
sync_local_okx_candle	116
sync_local_quantmod_OHLC	117
sync_local_rss_data	118
sync_local_sdmx_data	118
sync_local_sec_bulk_archive	120
sync_local_sec_companyfacts	120
sync_local_sec_filing_document	121
sync_local_sec_frame	122
sync_local_sec_submissions	123
sync_local_treasury_rates	124
sync_local_wbstats_data	124
sync_sec_filing_documents	125
validate_yahoofinance_registry	126

Index[127](#)

`add_fred_registry_series`*Add Or Update One FRED Registry Entry*

Description

Add Or Update One FRED Registry Entry

Usage

```
add_fred_registry_series(  
    series_id,  
    main_group = NULL,  
    registry_path = get_fred_registry_file_path(),  
    config = NULL  
)
```

Arguments

<code>series_id</code>	FRED series identifier.
<code>main_group</code>	Optional grouping label. If NULL, read one line from stdin after showing existing <code>main_group</code> hints.
<code>registry_path</code>	Optional registry JSON path.
<code>config</code>	Optional FRED API config.

Value

The added or updated row as a `data.table`.

`add_ishare_registry_ticker`*Add Or Update One iShares Registry Entry*

Description

Add Or Update One iShares Registry Entry

Usage

```
add_ishare_registry_ticker(  
    ticker,  
    type = NULL,  
    registry_path = get_ishare_registry_file_path()  
)
```

Arguments

ticker	ETF ticker.
type	Optional ticker type. If NULL, read from stdin.
registry_path	Optional registry JSON path.

Value

The added or updated row as a `data.table`.

add_sec_registry_company	<i>Add Or Update An SEC Registry Company</i>
--------------------------	--

Description

Add Or Update An SEC Registry Company

Usage

```
add_sec_registry_company(  
  ticker,  
  cik = NULL,  
  company_name = NULL,  
  forms = c("10-K", "10-Q", "8-K"),  
  concepts = NULL,  
  active = TRUE,  
  registry_path = get_sec_registry_file_path(),  
  config = NULL  
)
```

Arguments

ticker	Security ticker.
cik	Optional CIK; resolved from the SEC ticker mapping when omitted.
company_name	Optional company name.
forms	Optional filing forms retained by submissions sync.
concepts	Optional XBRL concepts retained by Company Facts sync.
active	Logical registry flag.
registry_path	Optional registry path.
config	Optional SEC configuration.

Value

The stored registry row.

`add_wbstats_registry_series`*Add Or Update One World Bank Registry Entry*

Description

Add Or Update One World Bank Registry Entry

Usage

```
add_wbstats_registry_series(  
  indicator,  
  country = NULL,  
  freq = "Y",  
  main_group = NA_character_,  
  label = NA_character_,  
  notes = NA_character_,  
  active = TRUE,  
  registry_path = get_wbstats_registry_file_path()  
)
```

Arguments

<code>indicator</code>	World Bank indicator code.
<code>country</code>	Country or aggregate code.
<code>freq</code>	Frequency code.
<code>main_group</code>	Optional grouping label.
<code>label</code>	Optional display label.
<code>notes</code>	Optional free-text notes.
<code>active</code>	Logical flag stored in the registry.
<code>registry_path</code>	Optional registry JSON path.

Value

The added or updated row as a `data.table`.

`bootstrap_yahoofinance_registry`*Bootstrap Yahoo Finance Runtime Registry*

Description

Create an absent runtime registry from the tracked package seed. Existing registries are never overwritten; they are validated and an actionable error is raised if required fallback declarations have drifted.

Usage

```
bootstrap_yahoofinance_registry(  
  registry_path = get_yahoofinance_registry_file_path(),  
  seed_path = get_yahoofinance_seed_registry_path()  
)
```

Arguments

<code>registry_path</code>	Runtime registry JSON path.
<code>seed_path</code>	Package seed registry JSON path.

Value

Invisibly returns the runtime registry path.

`describe_alphavantage_data`*Describe AlphaVantage Data*

Description

Fetches AlphaVantage daily OHLCV data and creates a compact narrative for the resulting standardized market table.

Usage

```
describe_alphavantage_data(symbol, mode = c("compact", "full"), config = NULL)
```

Arguments

<code>symbol</code>	Market symbol.
<code>mode</code>	Output size passed to AlphaVantage.
<code>config</code>	Optional AlphaVantage API configuration.

Value

Character scalar narrative.

describe_bea_data	<i>Describe Local BEA Data</i>
-------------------	--------------------------------

Description

Describe Local BEA Data

Usage

```
describe_bea_data(series_id, local_path = NULL)
```

Arguments

series_id	Registered series identifier.
local_path	Optional BEA storage directory.

Value

Character scalar narrative.

describe_binance_data	<i>Describe Binance Data</i>
-----------------------	------------------------------

Description

Fetches Binance OHLCV data and creates a compact narrative for the resulting standardized market table.

Usage

```
describe_binance_data(
  symbol = "ETHUSD",
  interval = "1m",
  start_time = NULL,
  end_time = NULL,
  limit = 1500L,
  tz = "UTC",
  paginate = TRUE
)
```

Arguments

symbol	Trading pair symbol.
interval	Candlestick interval.
start_time	Optional start time.
end_time	Optional end time.
limit	Page size. Binance futures allows up to 1500.
tz	Time zone applied to returned timestamps.
paginate	Logical. Request multiple pages when needed.

Value

Character scalar narrative.

describe_bls_data	<i>Describe Local BLS Data</i>
-------------------	--------------------------------

Description

Describe Local BLS Data

Usage

```
describe_bls_data(series_id, local_path = NULL)
```

Arguments

series_id	BLS series identifier.
local_path	Optional BLS storage directory.

Value

Character scalar narrative.

describe_census_data	<i>Describe Local Census Data</i>
----------------------	-----------------------------------

Description

Describe Local Census Data

Usage

describe_census_data(series_id, local_path = NULL)

Arguments

- series_id Registered series identifier.
- local_path Optional Census storage directory.

Value

Character scalar narrative.

describe_cftc_cot_data	<i>Describe Local CFTC COT Data</i>
------------------------	-------------------------------------

Description

Describe Local CFTC COT Data

Usage

describe_cftc_cot_data(report_id, local_path = NULL)

Arguments

- report_id Registry report identifier.
- local_path Optional CFTC storage directory.

Value

Character scalar narrative.

```
describe_crypto_derivatives
```

Describe Local Crypto Derivatives Data

Description

Describe Local Crypto Derivatives Data

Usage

```
describe_crypto_derivatives(
  provider,
  dataset_type,
  symbol,
  interval = NULL,
  local_path = NULL
)
```

Arguments

provider	binance or okx.
dataset_type	Funding, open-interest, mark/index price, basis, or Binance long-short-ratio dataset identifier.
symbol	Provider instrument identifier.
interval	Required Binance open-interest period; otherwise a local label.
local_path	Optional derivatives storage directory.

Value

Character scalar narrative.

```
describe_eia_data
```

Describe Local EIA Data

Description

Describe Local EIA Data

Usage

```
describe_eia_data(series_id, local_path = NULL)
```

Arguments

series_id EIA series identifier.
 local_path Optional EIA storage directory.

Value

Character scalar narrative.

describe_fiscaldata *Describe Local Treasury Fiscal Data*

Description

Describe Local Treasury Fiscal Data

Usage

```
describe_fiscaldata(dataset_id, local_path = NULL)
```

Arguments

dataset_id Registry dataset identifier.
 local_path Optional Fiscal Data storage directory.

Value

Character scalar narrative.

describe_fred_data *Describe FRED Data*

Description

Creates a compact narrative describing the structure and meaning of a locally stored FRED series for downstream analyst agents.

Usage

```
describe_fred_data(  
  series_id,  
  local_path = NULL,  
  registry_path = get_fred_registry_file_path()  
)
```

Arguments

series_id	FRED series identifier.
local_path	Optional local storage path used to read the local series.
registry_path	Optional registry path used to resolve series metadata.

Value

Character scalar narrative.

describe_ishare_data *Describe iShares Data*

Description

Creates a compact narrative for locally stored iShares historical fund data.

Usage

```
describe_ishare_data(ticker, local_path = NULL)
```

Arguments

ticker	ETF ticker.
local_path	Optional local storage path.

Value

Character scalar narrative.

describe_market_ohlcv_data
 Describe Market OHLCV Data

Description

Creates a compact narrative for standardized market OHLCV data.

Usage

```
describe_market_ohlcv_data(data, source_label = NULL)
```

Arguments

data	A standardized OHLCV data.table.
source_label	Optional source label override.

Value

Character scalar narrative.

describe_okx_candle_data

Describe OKX Candle Data

Description

Creates a compact narrative for locally stored OKX candle data.

Usage

```
describe_okx_candle_data(inst_id, bar, local_path = NULL)
```

Arguments

inst_id	Instrument identifier.
bar	Candle interval.
local_path	Optional local storage path.

Value

Character scalar narrative.

describe_quantmod_data

Describe quantmod Data

Description

Fetches market OHLCV data through quantmod and creates a compact narrative for the resulting standardized market table.

Usage

```
describe_quantmod_data(
  ticker,
  label = ticker,
  from = NULL,
  to = NULL,
  src = "yahoo",
  local_path = NULL
)
```

Arguments

ticker	Market symbol passed to <code>quantmod::getSymbols()</code> .
label	Optional label to store in the standardized symbol column.
from	Start date. If NULL, use the oldest date in local quantmod data for label.
to	End date. If NULL, use the newest date in local quantmod data for label.
src	quantmod source, default "yahoo".
local_path	Optional local storage path used when from or to is omitted.

Value

Character scalar narrative.

describe_rss_data	<i>Describe RSS Data</i>
-------------------	--------------------------

Description

Creates a compact narrative for locally stored RSS narrative data.

Usage

```
describe_rss_data(feed_id, local_path = NULL)
```

Arguments

feed_id	Local feed identifier.
local_path	Optional local storage path.

Value

Character scalar narrative.

describe_sdmx_data	<i>Describe Local SDMX Data</i>
--------------------	---------------------------------

Description

Describe Local SDMX Data

Usage

```
describe_sdmx_data(series_id, local_path = NULL)
```

Arguments

series_id	Registry series identifier.
local_path	Optional SDMX storage directory.

Value

Character scalar narrative.

describe_sec_companyfacts	<i>Describe Local SEC Company Facts</i>
---------------------------	---

Description

Describe Local SEC Company Facts

Usage

```
describe_sec_companyfacts(cik, local_path = NULL)
```

Arguments

cik	SEC Central Index Key.
local_path	Optional SEC Company Facts storage directory.

Value

Character scalar narrative.

describe_sec_frame	<i>Describe A Local SEC XBRL Frame</i>
--------------------	--

Description

Describe A Local SEC XBRL Frame

Usage

```
describe_sec_frame(taxonomy, tag, unit, period, local_path = NULL)
```

Arguments

taxonomy	XBRL taxonomy, such as us-gaap.
tag	XBRL concept tag.
unit	XBRL unit, such as USD.
period	SEC frame period, such as CY2025Q4I.
local_path	Optional SEC frame cache directory.

Value

Character scalar narrative.

describe_sec_submissions	<i>Describe Local SEC Submissions</i>
--------------------------	---------------------------------------

Description

Describe Local SEC Submissions

Usage

```
describe_sec_submissions(cik, local_path = NULL)
```

Arguments

cik	SEC Central Index Key.
local_path	Optional SEC submissions storage directory.

Value

Character scalar narrative.

`describe_treasury_rates`*Describe Treasury Rates*

Description

Creates a compact narrative for locally stored Treasury raw rates.

Usage

```
describe_treasury_rates(dataset, local_path = NULL)
```

Arguments

<code>dataset</code>	Treasury dataset key.
<code>local_path</code>	Optional local storage path.

Value

Character scalar narrative.

`describe_wbstats_data` *Describe World Bank Data*

Description

Creates a compact narrative for locally stored World Bank indicator data.

Usage

```
describe_wbstats_data(indicator, country, freq = "Y", local_path)
```

Arguments

<code>indicator</code>	World Bank indicator code.
<code>country</code>	Country or aggregate code.
<code>freq</code>	Frequency code.
<code>local_path</code>	Local storage path.

Value

Character scalar narrative.

detect_time_gaps	<i>Detect Time Gaps</i>
------------------	-------------------------

Description

Detects gaps in a local time series using either a fixed or calendar frequency.

Usage

```
detect_time_gaps(dt, time_col, frequency, tolerance = 1e-04, tz = "UTC")
```

Arguments

dt	A data.table-compatible object.
time_col	Name of the time column.
frequency	Frequency string or candle interval.
tolerance	Numeric tolerance applied to fixed-interval gaps.
tz	Time zone used when coercing timestamps.

Value

A data.table describing detected gaps.

detect_time_gaps_fred	<i>Detect Gaps In Local FRED Data</i>
-----------------------	---------------------------------------

Description

Detect Gaps In Local FRED Data

Usage

```
detect_time_gaps_fred(x, frequency = NULL, local_path = NULL)
```

Arguments

x	A FRED data.table, or a series id.
frequency	Optional explicit frequency string.
local_path	Optional local storage path if x is a series id.

Value

A data.table of gaps.

detect_time_gaps_okx_candle

Detect Time Gaps In OKX Candle Data

Description

Detect Time Gaps In OKX Candle Data

Usage

```
detect_time_gaps_okx_candle(dt, bar = "4H", tolerance = 1e-04)
```

Arguments

dt	A candle data.table.
bar	Candle interval.
tolerance	Numeric tolerance for fixed-width gap detection.

Value

A data.table.

detect_time_gaps_wbstats

Detect Gaps In Local World Bank Data

Description

Detect Gaps In Local World Bank Data

Usage

```
detect_time_gaps_wbstats(
  x = NULL,
  indicator = NULL,
  country = NULL,
  freq = "Y",
  local_path = NULL
)
```

Arguments

x	A World Bank data.table, or NULL when reading from local_path.
indicator	Optional indicator code when reading local data.
country	Optional country code when reading local data.
freq	Frequency code.
local_path	Optional local storage path.

Value

A data.table of gaps.

fetch_quantmod_OHLC	<i>Fetch Market OHLCV Through quantmod</i>
---------------------	--

Description

Returns a standardized OHLCV data.table with common market-schema columns: source, symbol, interval, datetime, date, open, high, low, close, and volume.

Usage

```
fetch_quantmod_OHLC(
  ticker,
  label = ticker,
  from,
  to,
  src = "yahoo",
  raw_data = FALSE,
  max_attempts = 3L,
  retry_delay_seconds = 1,
  fallback_source = NULL,
  fallback_ticker = ticker,
  require_start_coverage = FALSE
)
```

Arguments

ticker	Market symbol passed to quantmod::getSymbols().
label	Optional label to store in the standardized symbol column.
from	Start date.
to	End date.
src	quantmod source, default "yahoo".
raw_data	Logical. If TRUE, return the raw xts object.
max_attempts	Maximum bounded attempts for a transient source failure.
retry_delay_seconds	Initial retry delay in seconds; delays use exponential backoff.
fallback_source	Optional explicitly configured fallback provider. Currently supports "eastmoney" for daily OHLC data. For Yahoo sources, a failed dated quantmod request first retries Yahoo's chart endpoint with a bounded range before this external fallback is considered.

fallback_ticker

Optional provider-specific fallback identifier.

require_start_coverage

Logical. Require material coverage from from. sync_local_quantmod_OHLC() enables this only when valid local bars already establish the instrument's history.

Details

A row is usable only if open, high, low, and close are finite. Isolated invalid rows are discarded. A window is materially incomplete when its end is more than seven calendar days behind the requested end, or, for an instrument with valid local history, when its start is more than seven calendar days late or fewer than half of the requested weekdays are present. The start rule is not applied to a newly listed instrument, and the calendar-day grace prevents weekend and market-holiday false positives.

Value

data.table or raw xts object when raw_data = TRUE.

get_alphavantage_registry

Get Alpha Vantage Series Registry

Description

Get Alpha Vantage Series Registry

Usage

```
get_alphavantage_registry(
  registry_path = get_alphavantage_registry_file_path()
)
```

Arguments

registry_path Optional registry JSON path.

Value

A registry data.table.

`get_alphavantage_registry_file_path`*Get Alpha Vantage Registry File Path*

Description

Get Alpha Vantage Registry File Path

Usage

```
get_alphavantage_registry_file_path(config_dir = NULL)
```

Arguments

`config_dir` Optional configuration directory used for fallback.

Value

Character scalar path.

`get_bea_registry`*Get BEA Series Registry*

Description

Get BEA Series Registry

Usage

```
get_bea_registry(registry_path = get_bea_registry_file_path())
```

Arguments

`registry_path` Optional registry path.

Value

A registry data.table.

get_bea_registry_file_path	<i>Get BEA Registry File Path</i>
----------------------------	-----------------------------------

Description

Get BEA Registry File Path

Usage

get_bea_registry_file_path(config_dir = NULL)

Arguments

config_dir Optional configuration directory.

Value

Character scalar path.

get_bls_registry	<i>Get BLS Registry</i>
------------------	-------------------------

Description

Get BLS Registry

Usage

get_bls_registry(registry_path = get_bls_registry_file_path())

Arguments

registry_path Optional registry path.

Value

A registry data.table.

get_bls_registry_file_path
Get BLS Registry File Path

Description

Get BLS Registry File Path

Usage

```
get_bls_registry_file_path(config_dir = NULL)
```

Arguments

config_dir Optional configuration directory.

Value

Character scalar path.

get_census_registry *Get Census Series Registry*

Description

Get Census Series Registry

Usage

```
get_census_registry(registry_path = get_census_registry_file_path())
```

Arguments

registry_path Optional registry path.

Value

A registry data.table.

get_census_registry_file_path
Get Census Registry File Path

Description

Get Census Registry File Path

Usage

```
get_census_registry_file_path(config_dir = NULL)
```

Arguments

config_dir Optional configuration directory.

Value

Character scalar path.

get_cftc_cot_registry *Get CFTC COT Registry*

Description

Get CFTC COT Registry

Usage

```
get_cftc_cot_registry(registry_path = get_cftc_cot_registry_file_path())
```

Arguments

registry_path Optional JSON registry path.

Value

A registry data.table.

get_cftc_cot_registry_file_path

Get CFTC COT Registry File Path

Description

Get CFTC COT Registry File Path

Usage

```
get_cftc_cot_registry_file_path(config_dir = NULL)
```

Arguments

config_dir Optional configuration directory used for fallback.

Value

Character scalar path.

get_completed_local_quantmod_OHLC

Get Completed Local Daily quantmod OHLC Data

Description

Return only daily OHLC rows whose UTC date is strictly before an explicit cutoff. A finite row dated on the current UTC date is retained in the raw cache but is provisional until the next UTC date.

Usage

```
get_completed_local_quantmod_OHLC(
  label,
  src = "yahoo",
  interval = "1d",
  local_path = NULL,
  as_of = as.Date(Sys.time(), tz = "UTC")
)
```

Arguments

label	Local symbol label used in the stored data.
src	quantmod source, default "yahoo".
interval	Interval label. Only "1d" is supported.
local_path	Optional local storage path.
as_of	UTC timestamp used to determine the current UTC date.

Value

data.table or NULL.

get_crypto_derivatives_registry
<i>Get Crypto Derivatives Registry</i>

Description

Get Crypto Derivatives Registry

Usage

```
get_crypto_derivatives_registry(  
  registry_path = get_crypto_derivatives_registry_file_path()  
)
```

Arguments

registry_path Optional JSON registry path.

Value

A registry data.table.

get_crypto_derivatives_registry_file_path
<i>Get Crypto Derivatives Registry File Path</i>

Description

Get Crypto Derivatives Registry File Path

Usage

```
get_crypto_derivatives_registry_file_path(config_dir = NULL)
```

Arguments

config_dir Optional configuration directory used for fallback.

Value

Character scalar path.

get_eia_registry	<i>Get EIA Series Registry</i>
------------------	--------------------------------

Description

Get EIA Series Registry

Usage

```
get_eia_registry(registry_path = get_eia_registry_file_path())
```

Arguments

registry_path Optional JSON registry path.

Value

A registry data.table.

get_eia_registry_file_path	<i>Get EIA Registry File Path</i>
----------------------------	-----------------------------------

Description

Get EIA Registry File Path

Usage

```
get_eia_registry_file_path(config_dir = NULL)
```

Arguments

config_dir Optional configuration directory used for fallback.

Value

Character scalar path.

`get_fiscaldata_registry`*Get Treasury Fiscal Data Registry*

Description

Get Treasury Fiscal Data Registry

Usage

```
get_fiscaldata_registry(registry_path = get_fiscaldata_registry_file_path())
```

Arguments

`registry_path` Optional JSON registry path.

Value

A registry data.table.

`get_fiscaldata_registry_file_path`*Get Treasury Fiscal Data Registry File Path*

Description

Get Treasury Fiscal Data Registry File Path

Usage

```
get_fiscaldata_registry_file_path(config_dir = NULL)
```

Arguments

`config_dir` Optional configuration directory used for fallback.

Value

Character scalar path.

get_fred_registry	<i>Get FRED Registry</i>
-------------------	--------------------------

Description

Get FRED Registry

Usage

```
get_fred_registry(registry_path = get_fred_registry_file_path())
```

Arguments

registry_path Optional JSON registry path.

Value

data.table.

get_fred_registry_file_path	<i>Get FRED Registry File Path</i>
-----------------------------	------------------------------------

Description

Resolve the JSON registry path for FRED metadata. If no explicit registry_file is configured, the function falls back to a default filename in the package config directory.

Usage

```
get_fred_registry_file_path(config_dir = NULL)
```

Arguments

config_dir Optional configuration directory used for the fallback registry path.

Value

Character scalar path.

`get_investdatar_config`*Get Package Configuration*

Description

Returns the cached package configuration, loading it from YAML if needed.

Usage

```
get_investdatar_config(  
  reload = FALSE,  
  config_path = Sys.getenv("INVESTDATAR_CONFIG", unset = "")  
)
```

Arguments

<code>reload</code>	Logical. Reload from disk even if configuration is already cached.
<code>config_path</code>	Optional config path used when <code>reload = TRUE</code> or the cache is empty.

Value

A configuration list.

`get_ishare_registry`*Get iShares Registry*

Description

Get iShares Registry

Usage

```
get_ishare_registry(registry_path = get_ishare_registry_file_path())
```

Arguments

<code>registry_path</code>	Optional registry JSON path.
----------------------------	------------------------------

Value

`data.table`.

get_ishare_registry_file_path	
	<i>Get iShares Registry File Path</i>

Description

Get iShares Registry File Path

Usage

```
get_ishare_registry_file_path(config_dir = NULL)
```

Arguments

config_dir	Optional configuration directory used for fallback lookup.
------------	--

Value

Character scalar path.

get_latest_sync_run	<i>Get The Latest Batch Sync Run Log</i>
---------------------	--

Description

Get The Latest Batch Sync Run Log

Usage

```
get_latest_sync_run(source_id, local_path)
```

Arguments

source_id	Source identifier used for the sync log filename prefix.
local_path	Local source data path that contains the _sync_runs directory.

Value

A run-log list, or NULL when no matching log file exists.

get_local_alphavantage_data	
	<i>Read Local Alpha Vantage Daily Data</i>

Description

Read Local Alpha Vantage Daily Data

Usage

```
get_local_alphavantage_data(symbol, local_path = NULL)
```

Arguments

symbol	Market symbol.
local_path	Optional Alpha Vantage storage directory.

Value

A standardized market data .table, or NULL.

get_local_bea_data	<i>Read Local BEA Data</i>
--------------------	----------------------------

Description

Read Local BEA Data

Usage

```
get_local_bea_data(series_id, local_path = NULL)
```

Arguments

series_id	Registered series identifier.
local_path	Optional BEA storage directory.

Value

A data .table, or NULL.

get_local_binance_klines
Get Local Binance Kline Data

Description

Get Local Binance Kline Data

Usage

```
get_local_binance_klines(  
    symbol = "ETHUSD",  
    interval = "1m",  
    local_path = NULL,  
    storage = c("single", "monthly"),  
    from = NULL,  
    to = NULL  
)
```

Arguments

symbol	Trading pair symbol.
interval	Candlestick interval.
local_path	Optional Binance storage path.
storage	Local storage mode: monolithic "single" or monthly partitioned "monthly".
from, to	Optional bounds used to prune monthly partitions before read.

Value

data.table or NULL.

get_local_bls_data *Read Local BLS Data*

Description

Read Local BLS Data

Usage

```
get_local_bls_data(series_id, local_path = NULL)
```

Arguments

series_id	BLS series identifier.
local_path	Optional BLS storage directory.

Value

A data.table, or NULL.

get_local_census_data	<i>Read Local Census Data</i>
-----------------------	-------------------------------

Description

Read Local Census Data

Usage

```
get_local_census_data(series_id, local_path = NULL)
```

Arguments

- series_id Registered series identifier.
- local_path Optional Census storage directory.

Value

A data.table, or NULL.

get_local_cftc_cot	<i>Read Local CFTC COT Data</i>
--------------------	---------------------------------

Description

Read Local CFTC COT Data

Usage

```
get_local_cftc_cot(report_id, local_path = NULL)
```

Arguments

- report_id Registry report identifier.
- local_path Optional CFTC storage directory.

Value

A data.table, or NULL when no local file exists.

`get_local_crypto_derivatives`*Read Local Crypto Derivatives Data*

Description

Read Local Crypto Derivatives Data

Usage

```
get_local_crypto_derivatives(  
  provider,  
  dataset_type,  
  symbol,  
  interval = NULL,  
  local_path = NULL,  
  storage = c("single", "monthly"),  
  from = NULL,  
  to = NULL  
)
```

Arguments

<code>provider</code>	binance or okx.
<code>dataset_type</code>	Funding, open-interest, mark/index price, basis, or Binance long-short-ratio dataset identifier.
<code>symbol</code>	Provider instrument identifier.
<code>interval</code>	Required Binance open-interest period; otherwise a local label.
<code>local_path</code>	Optional derivatives storage directory.
<code>storage</code>	Local storage mode: monolithic "single" or monthly partitioned "monthly".
<code>from, to</code>	Optional inclusive UTC time bounds.

Value

A `data.table`, or `NULL`.

get_local_data_meta	<i>Get Local Data Metadata</i>
---------------------	--------------------------------

Description

Get Local Data Metadata

Usage

```
get_local_data_meta(local_file_path, fallback_to_file_mtime = TRUE)
```

Arguments

local_file_path	Path to a local .rds data file.
fallback_to_file_mtime	Logical. If no sidecar metadata file exists, use the data file modification time.

Value

A list of metadata, or NULL.

get_local_data_partitioned	<i>Read Monthly Partitioned Local Data</i>
----------------------------	--

Description

Reads a dataset stored as monthly RDS partitions. Time bounds are applied to partition discovery before files are loaded. If no partition directory exists, an existing monolithic RDS file is read for backward compatibility.

Usage

```
get_local_data_partitioned(
  local_file_path,
  time_col,
  from = NULL,
  to = NULL,
  order_cols = time_col,
  tz = "UTC"
)
```

Arguments

local_file_path	Virtual monolithic .rds path used to derive the sibling .parts directory and metadata sidecar.
time_col	Name of the partition timestamp column.
from, to	Optional inclusive time bounds.
order_cols	Optional output ordering columns.
tz	Time zone used to determine month boundaries.

Value

A data.table, or NULL when no local data exists.

get_local_data_untime	<i>Get Local Update Time</i>
-----------------------	------------------------------

Description

Get Local Update Time

Usage

```
get_local_data_untime(local_file_path)
```

Arguments

local_file_path	Path to a local .rds data file.
-----------------	---------------------------------

Value

POSIXct or NULL.

get_local_eia_data	<i>Read Local EIA Data</i>
--------------------	----------------------------

Description

Read Local EIA Data

Usage

```
get_local_eia_data(series_id, local_path = NULL)
```

Arguments

- `series_id` EIA series identifier.
- `local_path` Optional EIA storage directory.

Value

A `data.table`, or `NULL`.

<code>get_local_fiscaldata</code>	<i>Read Local Treasury Fiscal Data</i>
-----------------------------------	--

Description

Read Local Treasury Fiscal Data

Usage

```
get_local_fiscaldata(dataset_id, local_path = NULL)
```

Arguments

- `dataset_id` Registry dataset identifier.
- `local_path` Optional Fiscal Data storage directory.

Value

A `data.table`, or `NULL`.

<code>get_local_FRED_data</code>	<i>Get Local FRED Data</i>
----------------------------------	----------------------------

Description

Get Local FRED Data

Usage

```
get_local_FRED_data(series_id, local_path = NULL)
```

Arguments

- `series_id` FRED series identifier.
- `local_path` Optional local storage path.

Value

`data.table` or `NULL`.

`get_local_fund_holdings`*Read Provider-Neutral Local Fund Holdings*

Description

Read Provider-Neutral Local Fund Holdings

Usage

```
get_local_fund_holdings(provider, fund_id, local_path = NULL)
```

Arguments

<code>provider</code>	Fund issuer/provider identifier.
<code>fund_id</code>	Fund ticker or stable identifier.
<code>local_path</code>	Optional generic holdings storage root.

Value

A provider-neutral `data.table`, or `NULL`.

`get_local_ishare_data` *Get Local iShares Historical Data*

Description

Get Local iShares Historical Data

Usage

```
get_local_ishare_data(ticker, local_path = NULL)
```

Arguments

<code>ticker</code>	ETF ticker.
<code>local_path</code>	Optional local storage path.

Value

`data.table` or `NULL`.

`get_local_ishare_holdings`*Get Local iShares Holdings Data*

Description

Get Local iShares Holdings Data

Usage

```
get_local_ishare_holdings(ticker, local_path = NULL)
```

Arguments

<code>ticker</code>	ETF ticker.
<code>local_path</code>	Optional local storage path.

Value

`data.table` or `NULL`.

`get_local_ishare_holdings_standardized`*Convert Cached iShares Holdings To The Neutral Contract*

Description

Convert Cached iShares Holdings To The Neutral Contract

Usage

```
get_local_ishare_holdings_standardized(ticker, local_path = NULL)
```

Arguments

<code>ticker</code>	iShares fund ticker.
<code>local_path</code>	Optional iShares storage directory.

Value

A provider-neutral holdings data `table`, or `NULL`.

`get_local_ishare_mega_data`*Get Local iShares Metadata Snapshot*

Description

Get Local iShares Metadata Snapshot

Usage

```
get_local_ishare_mega_data(local_path = NULL)
```

Arguments

`local_path` Optional local storage path.

Value

`data.table` or `NULL`.

`get_local_okx_candle` *Get Local OKX Candle Data*

Description

Get Local OKX Candle Data

Usage

```
get_local_okx_candle(  
  inst_id,  
  bar,  
  local_path = NULL,  
  storage = c("single", "monthly"),  
  from = NULL,  
  to = NULL  
)
```

Arguments

`inst_id` Instrument identifier.
`bar` Candle interval.
`local_path` Optional OKX storage path.
`storage` Local storage mode: monolithic "single" or monthly partitioned "monthly".
`from, to` Optional bounds used to prune monthly partitions before read.

Value

data.table or NULL.

get_local_quantmod_OHLC	<i>Get Local quantmod OHLC Data</i>
-------------------------	-------------------------------------

Description

Get Local quantmod OHLC Data

Usage

```
get_local_quantmod_OHLC(  
  label,  
  src = "yahoo",  
  interval = "1d",  
  local_path = NULL  
)
```

Arguments

- label Local symbol label used in the stored data.
- src quantmod source, default "yahoo".
- interval Interval label, default "1d".
- local_path Optional local storage path.

Value

data.table or NULL.

get_local_rss_data	<i>Get Local RSS Data</i>
--------------------	---------------------------

Description

Get Local RSS Data

Usage

```
get_local_rss_data(feed_id, local_path = NULL)
```

Arguments

feed_id	Local feed identifier.
local_path	Optional local storage path.

Value

data.table or NULL.

get_local_sdmx_data	<i>Read Local SDMX Data</i>
---------------------	-----------------------------

Description

Read Local SDMX Data

Usage

```
get_local_sdmx_data(series_id, local_path = NULL)
```

Arguments

series_id	Registry series identifier.
local_path	Optional SDMX storage directory.

Value

A data.table, or NULL.

get_local_sec_companyfacts	<i>Read Local SEC Company Facts</i>
----------------------------	-------------------------------------

Description

Read Local SEC Company Facts

Usage

```
get_local_sec_companyfacts(cik, local_path = NULL)
```

Arguments

cik	SEC Central Index Key.
local_path	Optional SEC Company Facts storage directory.

Value

A data.table, or NULL.

get_local_sec_frame	<i>Read A Local SEC XBRL Frame</i>
---------------------	------------------------------------

Description

Read A Local SEC XBRL Frame

Usage

get_local_sec_frame(taxonomy, tag, unit, period, local_path = NULL)

Arguments

- taxonomy XBRL taxonomy, such as us-gaap.
- tag XBRL concept tag.
- unit XBRL unit, such as USD.
- period SEC frame period, such as CY2025Q4I.
- local_path Optional SEC frame cache directory.

Value

A cached data.table, or NULL.

get_local_sec_submissions	<i>Read Local SEC Submissions</i>
---------------------------	-----------------------------------

Description

Read Local SEC Submissions

Usage

get_local_sec_submissions(cik, local_path = NULL)

Arguments

- cik SEC Central Index Key.
- local_path Optional SEC submissions storage directory.

Value

A data.table, or NULL.

```
get_local_treasury_rates
```

Get Local Treasury Rates

Description

Get Local Treasury Rates

Usage

```
get_local_treasury_rates(dataset, local_path = NULL)
```

Arguments

dataset	Treasury dataset key.
local_path	Optional local storage path.

Value

data.table or NULL.

```
get_local_wbstats_data
```

Get Local World Bank Data

Description

Get Local World Bank Data

Usage

```
get_local_wbstats_data(indicator, country, freq = "Y", local_path = NULL)
```

Arguments

indicator	World Bank indicator code.
country	Country or aggregate code.
freq	Frequency code.
local_path	Optional local storage path.

Value

data.table or NULL.

get_rss_registry	<i>Get RSS Registry</i>
------------------	-------------------------

Description

Get RSS Registry

Usage

```
get_rss_registry(registry_path = get_rss_registry_file_path())
```

Arguments

registry_path Optional JSON registry path.

Value

data.table.

get_rss_registry_file_path	<i>Get RSS Registry File Path</i>
----------------------------	-----------------------------------

Description

Resolve the JSON registry path for RSS feed metadata. If no explicit registry_file is configured, the function falls back to a default filename in the package config directory.

Usage

```
get_rss_registry_file_path(config_dir = NULL)
```

Arguments

config_dir Optional configuration directory used for the fallback registry path.

Value

Character scalar path.

get_sdmx_registry	<i>Get SDMX Series Registry</i>
-------------------	---------------------------------

Description

Get SDMX Series Registry

Usage

```
get_sdmx_registry(registry_path = get_sdmx_registry_file_path())
```

Arguments

registry_path Optional JSON registry path.

Value

A registry data.table.

get_sdmx_registry_file_path	<i>Get SDMX Registry File Path</i>
-----------------------------	------------------------------------

Description

Get SDMX Registry File Path

Usage

```
get_sdmx_registry_file_path(config_dir = NULL)
```

Arguments

config_dir Optional configuration directory used for fallback.

Value

Character scalar path.

`get_sec_company_tickers`*Retrieve SEC Ticker-CIK Mappings*

Description

Retrieve SEC Ticker-CIK Mappings

Usage

```
get_sec_company_tickers(config = NULL)
```

Arguments

`config` Optional SEC configuration containing an identifiable user agent.

Value

A `data.table` with `cik`, `company_name`, `ticker`, and `exchange`.

`get_sec_frames_registry`*Get SEC Frames Registry*

Description

Get SEC Frames Registry

Usage

```
get_sec_frames_registry(registry_path = get_sec_frames_registry_file_path())
```

Arguments

`registry_path` Optional registry path.

Value

A registry `data.table`.

`get_sec_frames_registry_file_path`*Get SEC Frames Registry File Path*

Description

Get SEC Frames Registry File Path

Usage

```
get_sec_frames_registry_file_path(config_dir = NULL)
```

Arguments

`config_dir` Optional configuration directory.

Value

Character scalar path.

`get_sec_registry`*Get SEC Company Registry*

Description

Get SEC Company Registry

Usage

```
get_sec_registry(registry_path = get_sec_registry_file_path())
```

Arguments

`registry_path` Optional JSON registry path.

Value

A registry data.table.

`get_sec_registry_file_path`*Get SEC Company Registry File Path*

Description

Get SEC Company Registry File Path

Usage

```
get_sec_registry_file_path(config_dir = NULL)
```

Arguments

`config_dir` Optional configuration directory used for fallback.

Value

Character scalar path.

`get_source_config`*Get Source Configuration*

Description

Returns the configuration block for a named source.

Usage

```
get_source_config(source, config = get_investdatar_config())
```

Arguments

`source` Character source name.
`config` Optional package config list.

Value

A list, or an empty list if the source is not configured.

get_source_data_alphavantage_ts_daily

Fetch Daily Time Series from AlphaVantage

Description

Retrieves daily historical stock data (open, high, low, close, volume) for a given symbol using the AlphaVantage API. You can specify whether to retrieve only recent data ("compact") or the full history ("full"). The returned object follows the package's standardized market OHLCV schema.

Usage

```
get_source_data_alphavantage_ts_daily(
  symbol,
  mode = c("compact", "full"),
  config = NULL
)
```

Arguments

symbol	A character string for the stock ticker symbol (e.g., "AAPL", "TSLA").
mode	Character string; either "compact" (latest 100 days) or "full" (full history). Defaults to "compact".
config	A list of AlphaVantage API settings, typically retrieved via tool_set_config("alphavantage").

Value

A data.table with standardized columns including source, symbol, interval, datetime, date, open, high, low, close, and volume.

datetime Midnight timestamp derived from the trading date.

date Date of the observation.

open Opening price.

high Highest price of the day.

low Lowest price of the day.

close Closing price.

volume Volume of trades.

Examples

```
## Not run:
config <- list(api_key = Sys.getenv("ALPHAVANTAGE_API_KEY"))
df <- get_source_data_alphavantage_ts_daily("MSFT", mode = "compact", config = config)
head(df)

## End(Not run)
```

get_source_data_bea	Retrieve BEA Regional Data
---------------------	----------------------------

Description

Retrieve BEA Regional Data

Usage

```
get_source_data_bea(  
    series_id,  
    table_name,  
    line_code,  
    geofips = "STATE",  
    year = "ALL",  
    label = NULL,  
    config = NULL  
)
```

Arguments

series_id	Stable local series identifier.
table_name	BEA Regional table name.
line_code	BEA line code.
geofips	Geography selector such as STATE or COUNTY.
year	Year selector, default ALL.
label	Optional local label.
config	Optional BEA configuration.

Value

A standardized regional panel.

get_source_data_binance_klines	Fetch Kline Data from Binance Futures API
--------------------------------	---

Description

Retrieves candlestick data from Binance USDT-margined futures. If `paginate = TRUE`, the function keeps requesting pages until the requested time range is exhausted or the API returns fewer rows than `limit`.

Usage

```
get_source_data_binance_klines(  
  symbol = "ETHUSD",  
  interval = "1m",  
  start_time = NULL,  
  end_time = NULL,  
  limit = 1500L,  
  tz = "UTC",  
  paginate = TRUE  
)
```

Arguments

symbol	Trading pair symbol.
interval	Candlestick interval.
start_time	Optional start time.
end_time	Optional end time.
limit	Page size. Binance futures allows up to 1500.
tz	Time zone applied to returned timestamps.
paginate	Logical. Request multiple pages when needed.

Value

Standardized OHLCV data. table with source, symbol, interval, datetime, date, open, high, low, close, and volume, plus Binance-specific columns such as close_time and trade counts.

get_source_data_bls	<i>Retrieve One BLS Series</i>
---------------------	--------------------------------

Description

Retrieve One BLS Series

Usage

```
get_source_data_bls(  
  series_id,  
  label = NULL,  
  from = NULL,  
  to = NULL,  
  config = NULL  
)
```

Arguments

series_id	BLS series identifier.
label	Optional local label.
from, to	Optional year or date bounds.
config	Optional BLS configuration.

Value

A standardized monthly data .table.

get_source_data_census
<i>Retrieve A Selected Census Economic Indicator Series</i>

Description

Retrieve A Selected Census Economic Indicator Series

Usage

```
get_source_data_census(  
  series_id,  
  dataset,  
  data_type_code,  
  category_code,  
  seasonally_adj = "yes",  
  from = NULL,  
  to = NULL,  
  label = NULL,  
  config = NULL  
)
```

Arguments

series_id	Stable local series identifier.
dataset	Economic Indicators dataset suffix, such as marts.
data_type_code	Census item type code.
category_code	Census industry/category code.
seasonally_adj	Seasonal adjustment value.
from, to	Optional time bounds.
label	Optional local label.
config	Optional Census configuration.

Value

A standardized long data .table.

get_source_data_cftc_cot

Retrieve CFTC Commitments Of Traders Data

Description

Retrieves TFF, Disaggregated, or Legacy reports from the CFTC Public Reporting Environment.

Usage

```
get_source_data_cftc_cot(
  report_variant = c("futures_only", "combined"),
  report_type = "tff",
  report_id = NULL,
  dataset_id = NULL,
  market_codes = NULL,
  from = NULL,
  to = NULL,
  page_size = 5000L,
  max_pages = Inf
)
```

Arguments

report_variant	Report variant: "futures_only" or "combined".
report_type	Report family: "tff", "disaggregated", or "legacy".
report_id	Stable local report identifier.
dataset_id	Optional official Socrata dataset identifier.
market_codes	Optional CFTC contract-market codes.
from, to	Optional inclusive report-date bounds.
page_size	Number of source rows requested per page.
max_pages	Optional page limit, primarily useful for diagnostics.

Value

A standardized wide data . table, one row per market and report date.

`get_source_data_crypto_derivatives`*Retrieve Historical Crypto Derivatives Data*

Description

Retrieve Historical Crypto Derivatives Data

Usage

```
get_source_data_crypto_derivatives(  
  provider,  
  dataset_type,  
  symbol,  
  interval = NULL,  
  from = NULL,  
  to = NULL,  
  limit = NULL,  
  config = NULL  
)
```

Arguments

<code>provider</code>	binance or okx.
<code>dataset_type</code>	Funding, open-interest, mark/index price, basis, or Binance long-short-ratio dataset identifier.
<code>symbol</code>	Provider instrument identifier.
<code>interval</code>	Required Binance open-interest period; otherwise a local label.
<code>from, to</code>	Optional inclusive UTC time bounds.
<code>limit</code>	Provider page size.
<code>config</code>	Optional exchange configuration.

Value

A standardized `data.table` retaining provider fields.

get_source_data_eia	<i>Retrieve One EIA Series</i>
---------------------	--------------------------------

Description

Retrieve One EIA Series

Usage

```
get_source_data_eia(  
  series_id,  
  label = NULL,  
  config = NULL,  
  from = NULL,  
  to = NULL,  
  page_size = 5000L,  
  max_pages = Inf  
)
```

Arguments

series_id	EIA series identifier.
label	Optional local label.
config	Optional EIA API configuration.
from, to	Optional inclusive period bounds.
page_size	Number of observations requested per page.
max_pages	Optional page limit for diagnostics.

Value

A standardized long data.table.

get_source_data_fiscaldata	<i>Retrieve A U.S. Treasury Fiscal Data Table</i>
----------------------------	---

Description

Retrieve A U.S. Treasury Fiscal Data Table

Usage

```
get_source_data_fiscaldata(
  dataset_id,
  endpoint,
  date_col = "record_date",
  key_cols = date_col,
  fields = NULL,
  from = NULL,
  to = NULL,
  page_size = 10000L,
  max_pages = Inf
)
```

Arguments

dataset_id	Stable local dataset identifier.
endpoint	Fiscal Data endpoint relative to the fiscal-service base URL.
date_col	Source date column.
key_cols	Source columns that uniquely identify observations per date.
fields	Optional source fields to request.
from, to	Optional inclusive date bounds.
page_size	Number of source rows requested per page.
max_pages	Optional page limit for diagnostics.

Value

A typed, table-shaped data.table with source and dataset identity.

get_source_data_fred	<i>Get FRED Series Data</i>
----------------------	-----------------------------

Description

Download a FRED series and return dates and numeric values.

Usage

```
get_source_data_fred(series_id, config = NULL)
```

Arguments

series_id	Character. FRED series ID (e.g., "DGS10").
config	Optional list with fields: api_key, url, mode.

Value

data.table with columns: date (Date), value (numeric, NA for ".").

`get_source_data_ishare`*Get iShares Historical Data*

Description

Downloads and parses the historical sheet for a single iShares fund using the local iShares registry metadata.

Usage

```
get_source_data_ishare(  
    ticker,  
    ishare_mega_data = NULL,  
    cache_dir = NULL,  
    local_path = NULL  
)
```

Arguments

<code>ticker</code>	ETF ticker.
<code>ishare_mega_data</code>	Optional iShares registry table.
<code>cache_dir</code>	Optional download cache directory.
<code>local_path</code>	Optional local storage path used to resolve defaults.

Value

`data.table`.

`get_source_data_ishare_holdings`*Get iShares Holdings Data*

Description

Downloads and parses the holdings sheet for a single iShares fund using the local iShares registry metadata.

Usage

```
get_source_data_ishare_holdings(  
    ticker,  
    ishare_mega_data = NULL,  
    cache_dir = NULL,  
    local_path = NULL  
)
```

Arguments

ticker	ETF ticker.
ishare_mega_data	Optional iShares registry table.
cache_dir	Optional download cache directory.
local_path	Optional local storage path used to resolve defaults.

Value

data.table.

get_source_data_okx_candle
<i>Get Latest OKX Candle Data</i>

Description

Get Latest OKX Candle Data

Usage

```
get_source_data_okx_candle(  
  inst_id,  
  bar,  
  limit = 100L,  
  config = NULL,  
  tz = "UTC"  
)
```

Arguments

inst_id	Instrument identifier.
bar	Candle interval.
limit	Integer page size.
config	Optional OKX API config. If omitted, defaults from the package config and OKX_API_KEY / OKX_SECRET_KEY / OKX_PASSPHRASE environment variables are used.
tz	Output time zone.

Value

data.table or NULL.

get_source_data_path *Get Source Data Path*

Description

Resolves the local data directory for a source using the package config.

Usage

```
get_source_data_path(
  source,
  config = get_investdata_config(),
  subdir = NULL,
  create = FALSE
)
```

Arguments

source	Character source name.
config	Optional package config list.
subdir	Optional subdirectory appended to the source data path.
create	Logical. Create the directory if it does not exist.

Value

Character scalar path.

get_source_data_rss *Get RSS Feed Data*

Description

Download an RSS feed and return a normalized item-level data . table.

Usage

```
get_source_data_rss(
  feed_id,
  url,
  parser = c("plain", "gdpnow"),
  ca_bundle = NULL
)
```

Arguments

feed_id	Local feed identifier.
url	RSS feed URL.
parser	Parsing strategy. Currently supports "plain" and "gdpnow".
ca_bundle	Optional CA bundle path used only for this feed request.

Value

data.table.

get_source_data_sdmx	<i>Retrieve Data From An SDMX Provider</i>
----------------------	--

Description

Retrieve Data From An SDMX Provider

Usage

```
get_source_data_sdmx(  
  series_id,  
  provider,  
  base_url,  
  agency = NULL,  
  dataflow,  
  version = "latest",  
  key = "",  
  flow_ref = NULL,  
  format = NULL,  
  accept = "text/csv",  
  time_col = "TIME_PERIOD",  
  value_col = "OBS_VALUE",  
  dimension_cols = NULL,  
  label = NULL,  
  frequency = NULL,  
  from = NULL,  
  to = NULL,  
  last_n_observations = NULL  
)
```

Arguments

series_id	Stable local series identifier.
provider	Provider dialect: oecd, ecb, bis, eurostat, or the IMF DataMapper-compatible imf adapter.

base_url	Provider REST base URL.
agency	Optional SDMX agency identifier.
dataflow	SDMX dataflow identifier.
version	SDMX dataflow version.
key	SDMX series key.
flow_ref	Optional provider-specific flow reference.
format	Provider query format.
accept	HTTP response media type.
time_col, value_col	Source observation columns.
dimension_cols	Columns identifying distinct series in the response.
label, frequency	Optional local metadata.
from, to	Optional inclusive SDMX periods.
last_n_observations	Optional number of latest observations requested per matching SDMX series.

Value

A standardized long data.table retaining original SDMX columns.

```
get_source_data_sec_companyfacts
```

Retrieve SEC Company Facts

Description

Retrieve SEC Company Facts

Usage

```
get_source_data_sec_companyfacts(
  cik,
  ticker = NULL,
  company_name = NULL,
  concepts = NULL,
  forms = NULL,
  from = NULL,
  to = NULL,
  config = NULL
)
```

Arguments

cik	SEC Central Index Key.
ticker	Optional ticker label.
company_name	Optional company name label.
concepts	Optional concept names or taxonomy:concept identifiers.
forms	Optional filing forms to retain.
from, to	Optional inclusive filed-date bounds.
config	Optional SEC configuration.

Value

A standardized long XBRL fact data. table.

get_source_data_sec_frame
<i>Retrieve An SEC XBRL Frame</i>

Description

Retrieve An SEC XBRL Frame

Usage

get_source_data_sec_frame(taxonomy, tag, unit, period, config = NULL)

Arguments

taxonomy	XBRL taxonomy, such as us-gaap.
tag	XBRL concept tag.
unit	XBRL unit, such as USD.
period	SEC frame period, such as CY2025Q4I.
config	Optional SEC configuration.

Value

A normalized cross-company data. table.

get_source_data_sec_submissions
<i>Retrieve SEC Filing Submissions</i>

Description

Retrieve SEC Filing Submissions

Usage

```
get_source_data_sec_submissions(  
  cik,  
  ticker = NULL,  
  company_name = NULL,  
  forms = NULL,  
  from = NULL,  
  to = NULL,  
  include_history = TRUE,  
  config = NULL  
)
```

Arguments

cik	SEC Central Index Key.
ticker	Optional ticker label.
company_name	Optional company name label.
forms	Optional filing forms to retain.
from, to	Optional inclusive filing-date bounds.
include_history	Logical; retrieve older SEC submission files in addition to recent filings.
config	Optional SEC configuration.

Value

A standardized filing-event data . table.

get_source_data_treasury_rates
<i>Get Treasury Rates Data</i>

Description

Download U.S. Treasury raw daily rates and return a normalized long-format data .table.

Usage

```
get_source_data_treasury_rates(dataset, years = NULL)
```

Arguments

dataset	Treasury dataset key. Supported values are "bill_rates", "par_yield_curve", "long_term_rates", "real_yield_curve", and "real_long_term_rates".
years	Optional integer vector of calendar years to fetch. If omitted, the function combines the historical archive with current-year annual XML feeds from 2024 through the current year.

Value

data.table.

get_source_data_wbstats
<i>Get World Bank Data via wbstats</i>

Description

Downloads indicator data using the wbstats package and standardizes it to a package-native long schema.

Usage

```
get_source_data_wbstats(  
  indicator,  
  country = "countries_only",  
  start_date = NULL,  
  end_date = NULL,  
  mrv = NULL,  
  mrnev = NULL,  
  cache = NULL,  
  freq = "Y",  
  gapfill = FALSE,  
  date_as_class_date = FALSE,  
  lang = NULL  
)
```

Arguments

indicator	World Bank indicator code.
country	Country or aggregate code.
start_date	Optional start date.
end_date	Optional end date.
mrval	Optional number of most recent values.
mrnev	Optional number of most recent non-empty values.
cache	Optional cache object passed to <code>wbstats::wb_data()</code> .
freq	Frequency code: "Y", "Q", or "M".
gapfill	Logical passed to <code>wbstats::wb_data()</code> .
date_as_class_date	Logical passed to <code>wbstats::wb_data()</code> .
lang	Optional language code.

Value

Standardized data.table.

get_source_hist_data_okx_candle

Get Historical OKX Candle Data

Description

Get Historical OKX Candle Data

Usage

```
get_source_hist_data_okx_candle(  
  inst_id,  
  bar,  
  before = NULL,  
  limit = 100L,  
  config = NULL,  
  tz = "UTC"  
)
```

Arguments

inst_id	Instrument identifier.
bar	Candle interval.
before	Optional pagination cursor.
limit	Integer page size.

config	Optional OKX API config. If omitted, defaults from the package config and OKX_API_KEY / OKX_SECRET_KEY / OKX_PASSPHRASE environment variables are used.
tz	Output time zone.

Value

data.table or NULL.

get_source_metadata_fred

Retrieve metadata for a FRED series

Description

Queries the FRED API to obtain basic metadata of a given series, such as title, observation range, frequency, units, and seasonal adjustment.

Usage

```
get_source_metadata_fred(series_id, config = NULL)
```

Arguments

series_id	A character string. The FRED series ID (e.g., "AMERIBOR").
config	A list containing API configuration with elements: api_key Your FRED API key as a character string. url Base URL of the FRED API endpoint for series (e.g., "https://api.stlouisfed.org/fred/series"). mode File type to request, usually "json".

Value

A list with elements:

- title — Series title
- start — Observation start date
- end — Observation end date
- freq — Data frequency
- units — Measurement units
- season — Seasonal adjustment type

Examples

```
## Not run:
config <- list(
  api_key = "your_api_key",
  url = "https://api.stlouisfed.org/fred/series",
  mode = "json"
)
get_source_metadata_fred("AMERIBOR", config)

## End(Not run)
```

get_source_spec	<i>Get Source Specification</i>
-----------------	---------------------------------

Description

Returns a formal source-spec object describing one provider module. The spec is intentionally lightweight so modules can later move into separate packages without changing the high-level contract.

Usage

```
get_source_spec(source)
```

Arguments

source	Character provider key.
--------	-------------------------

Value

An object of class `investdatar_source_spec`.

get_source_etime_cftc_cot	<i>Get CFTC Dataset Update Time</i>
---------------------------	-------------------------------------

Description

Get CFTC Dataset Update Time

Usage

```
get_source_etime_cftc_cot(
  report_variant = c("futures_only", "combined"),
  report_type = "tff",
  dataset_id = NULL
)
```

Arguments

report_variant Report variant: "futures_only" or "combined".
 report_type Report family: "tff", "disaggregated", or "legacy".
 dataset_id Optional official Socrata dataset identifier.

Value

A UTC POSIXct update time, or NULL when unavailable.

get_source_utime_crypto_derivatives
Get Latest Crypto Derivatives Observation Time

Description

Get Latest Crypto Derivatives Observation Time

Usage

```
get_source_utime_crypto_derivatives(
  provider,
  dataset_type,
  symbol,
  interval = NULL,
  config = NULL
)
```

Arguments

provider binance or okx.
 dataset_type Funding, open-interest, mark/index price, basis, or Binance long-short-ratio dataset identifier.
 symbol Provider instrument identifier.
 interval Required Binance open-interest period; otherwise a local label.
 config Optional exchange configuration.

Value

UTC POSIXct, or NULL.

get_source_untime_eia *Get Latest EIA Series Update Date*

Description

Get Latest EIA Series Update Date

Usage

```
get_source_untime_eia(series_id, config = NULL)
```

Arguments

series_id	EIA series identifier.
config	Optional EIA API configuration.

Value

A UTC POSIXct, or NULL.

get_source_untime_fiscaldata
 Get The Latest Treasury Fiscal Data Date

Description

Get The Latest Treasury Fiscal Data Date

Usage

```
get_source_untime_fiscaldata(dataset_id, endpoint, date_col = "record_date")
```

Arguments

dataset_id	Stable local dataset identifier.
endpoint	Fiscal Data endpoint relative to the fiscal-service base URL.
date_col	Source date column.

Value

A UTC POSIXct inferred from the newest source record, or NULL.

get_source_untime_fred *Get FRED Series Last Update Time*

Description

Return the last update time of a FRED series as POSIXct in UTC.

Usage

```
get_source_untime_fred(
  series_id,
  config = NULL,
  from_server = FALSE,
  tz = "America/Chicago"
)
```

Arguments

series_id	Character. FRED series ID.
config	Optional list with fields: api_key, url, mode.
from_server	Logical. If TRUE, query the FRED series endpoint for the reported update time. Otherwise infer it from the registry frequency.
tz	Time zone used when parsing or inferring the update time.

Value

POSIXct (UTC).

get_source_untime_ishare
Get iShares Source Update Time

Description

Returns a best-effort upstream update time for iShares data. When check_online = TRUE, the function scrapes the product page; otherwise it falls back to the latest midnight in tz.

Usage

```
get_source_untime_ishare(tz = "America/New_York", check_online = TRUE)
```

Arguments

tz	Time zone used for the returned timestamp.
check_online	Logical. Scrape iShares online when TRUE.

Value

POSIXct.

get_source_untime_okx_candle	<i>Get Last Completed Candle Time</i>
------------------------------	---------------------------------------

Description

Floor current time to the last completed candle start for the given bar.

Usage

```
get_source_untime_okx_candle(bar, tz = "UTC")
```

Arguments

bar	Character. OKX timeframe tag (e.g., "1m", "15m", "4H", "1D").
tz	Character. IANA timezone.

Value

POSIXct.

get_source_untime_rss	<i>Get RSS Feed Last Update Time</i>
-----------------------	--------------------------------------

Description

Get RSS Feed Last Update Time

Usage

```
get_source_untime_rss(  
  feed_id,  
  url,  
  parser = c("plain", "gdpnow"),  
  ca_bundle = NULL  
)
```

Arguments

feed_id	Local feed identifier.
url	RSS feed URL.
parser	Parsing strategy. Currently supports "plain" and "gdpnow".
ca_bundle	Optional CA bundle path used only for this feed request.

Value

POSIXct or NULL.

get_source_untime_sdmx	<i>Get Latest SDMX Observation Time</i>
------------------------	---

Description

Get Latest SDMX Observation Time

Usage

```
get_source_untime_sdmx(  
  series_id,  
  provider,  
  base_url,  
  agency = NULL,  
  dataflow,  
  version = "latest",  
  key = "",  
  flow_ref = NULL,  
  format = NULL,  
  accept = "text/csv",  
  time_col = "TIME_PERIOD",  
  value_col = "OBS_VALUE",  
  dimension_cols = NULL,  
  label = NULL,  
  frequency = NULL  
)
```

Arguments

series_id	Stable local series identifier.
provider	Provider dialect: oecd, ecb, bis, eurostat, or the IMF DataMapper-compatible imf adapter.
base_url	Provider REST base URL.
agency	Optional SDMX agency identifier.
dataflow	SDMX dataflow identifier.
version	SDMX dataflow version.
key	SDMX series key.
flow_ref	Optional provider-specific flow reference.
format	Provider query format.
accept	HTTP response media type.

time_col, value_col
Source observation columns.

dimension_cols Columns identifying distinct series in the response.

label, frequency
Optional local metadata.

Value

A UTC POSIXct, or NULL.

get_source_etime_sec_companyfacts
<i>Get Latest SEC Company Facts Filing Date</i>

Description

Get Latest SEC Company Facts Filing Date

Usage

get_source_etime_sec_companyfacts(cik, config = NULL)

Arguments

cik SEC Central Index Key.

config Optional SEC configuration.

Value

A UTC POSIXct, or NULL.

get_source_etime_sec_submissions
<i>Get Latest SEC Submission Time</i>

Description

Get Latest SEC Submission Time

Usage

get_source_etime_sec_submissions(cik, config = NULL)

Arguments

- cik SEC Central Index Key.
- config Optional SEC configuration.

Value

A UTC-compatible POSIXct, or NULL.

get_source_untime_treasury_rates
<i>Get Treasury Rates Last Update Time</i>

Description

Get Treasury Rates Last Update Time

Usage

```
get_source_untime_treasury_rates(dataset, years = NULL)
```

Arguments

- dataset Treasury dataset key. Supported values are "bill_rates", "par_yield_curve", "long_term_rates", "real_yield_curve", and "real_long_term_rates".
- years Optional integer vector of calendar years to fetch. If omitted, the function combines the historical archive with current-year annual XML feeds from 2024 through the current year.

Value

POSIXct or NULL.

get_source_untime_wbstats
<i>Get Fallback Source Update Time for wbstats Data</i>

Description

Get Fallback Source Update Time for wbstats Data

Usage

```
get_source_untime_wbstats(freq = "Y", tz = "UTC")
```

Arguments

freq	Frequency code: "Y", "Q", or "M".
tz	Time zone used for the inferred timestamp.

Value

POSIXct.

get_wbstats_registry	<i>Get World Bank Registry</i>
----------------------	--------------------------------

Description

Get World Bank Registry

Usage

```
get_wbstats_registry(registry_path = get_wbstats_registry_file_path())
```

Arguments

registry_path	Optional JSON registry path.
---------------	------------------------------

Value

data.table.

get_wbstats_registry_file_path	<i>Get World Bank Registry File Path</i>
--------------------------------	--

Description

Resolve the JSON registry path for World Bank indicator metadata. If no explicit `registry_file` is configured, the function falls back to a default filename in the package config directory.

Usage

```
get_wbstats_registry_file_path(config_dir = NULL)
```

Arguments

config_dir	Optional configuration directory used for the fallback registry path.
------------	---

Value

Character scalar path.

```
get_yahoofinance_registry
```

Get Yahoo Finance Registry

Description

Get Yahoo Finance Registry

Usage

```
get_yahoofinance_registry(  
  registry_path = get_yahoofinance_registry_file_path()  
)
```

Arguments

registry_path Optional registry JSON path.

Value

data.table.

```
get_yahoofinance_registry_file_path
```

Get Yahoo Finance Registry File Path

Description

Resolve the JSON registry path for Yahoo Finance ticker metadata. If no explicit registry_file is configured, the function falls back to a default filename in the package config directory.

Usage

```
get_yahoofinance_registry_file_path(config_dir = NULL)
```

Arguments

config_dir Optional configuration directory used for the fallback registry path.

Value

Character scalar path.

`get_yahoofinance_seed_registry_path`*Get Yahoo Finance Seed Registry Path*

Description

Get Yahoo Finance Seed Registry Path

Usage

```
get_yahoofinance_seed_registry_path()
```

Value

Path to the package-managed Yahoo Finance registry seed.

`infer_source_ftime_from_frequency`*Infer Source Update Time From Frequency*

Description

Provides a fallback source update time for resources that do not expose a server-side update timestamp.

Usage

```
infer_source_ftime_from_frequency(  
  frequency,  
  reference_time = Sys.time(),  
  tz = "UTC"  
)
```

Arguments

<code>frequency</code>	Frequency string or candle interval.
<code>reference_time</code>	POSIXct reference time. Defaults to <code>Sys.time()</code> .
<code>tz</code>	Time zone used for period flooring.

Value

POSIXct.

is_sync_run_successful	<i>Check Whether A Batch Sync Run Succeeded</i>
------------------------	---

Description

A run is successful when its stored summary contains no row with an error status or non-empty error message. Empty summaries are successful because an intentionally empty registry has no failed work.

Usage

is_sync_run_successful(run)

Arguments

run A run-log object returned by get_latest_sync_run().

Value

Logical scalar.

list_source_specs	<i>List Source Specifications</i>
-------------------	-----------------------------------

Description

List Source Specifications

Usage

list_source_specs()

Value

Named list of investdatar_source_spec objects.

`load_investdatar_config`*Load Package Configuration*

Description

Loads the package YAML configuration and caches it in package options.

Usage

```
load_investdatar_config(  
  config_path = Sys.getenv("INVESTDATAR_CONFIG", unset = "")  
)
```

Arguments

<code>config_path</code>	Optional path to the YAML config file. If omitted, the function uses the INVESTDATAR_CONFIG environment variable.
--------------------------	---

Value

A normalized configuration list.

`repair_local_binance_klines_gaps`*Repair Local Binance Kline Data From Multiple Windows*

Description

Fetches all requested Binance kline windows in memory and writes the merged repair result to local storage with one `sync_local_data()` call.

Usage

```
repair_local_binance_klines_gaps(  
  symbol = "ETHUSD",  
  interval = "1m",  
  windows,  
  limit = 1500L,  
  tz = "UTC",  
  local_path = NULL,  
  storage = c("single", "monthly")  
)
```

Arguments

symbol	Trading pair symbol.
interval	Candlestick interval.
windows	A data.frame/data.table with start_time and end_time columns, or a list of objects containing those fields.
limit	Page size. Binance futures allows up to 1500.
tz	Time zone applied to returned timestamps.
local_path	Optional Binance storage path.
storage	Local storage mode: monolithic "single" or monthly partitioned "monthly".

Value

A sync result list.

```
repair_local_okx_candle_gaps
```

Repair Local OKX Candle Data From Multiple History Pages

Description

Fetches multiple OKX historical candle pages in memory and writes the merged repair result to local storage with one sync_local_data() call.

Usage

```
repair_local_okx_candle_gaps(
  inst_id,
  bar,
  before,
  config = NULL,
  local_path = NULL,
  limit = 100L,
  tz = "UTC",
  storage = c("single", "monthly")
)
```

Arguments

inst_id	Instrument identifier.
bar	Candle interval.
before	Character/numeric vector of OKX history pagination cursors.
config	Optional OKX API config. If omitted, defaults from the package config and OKX_API_KEY / OKX_SECRET_KEY / OKX_PASSPHRASE environment variables are used.

local_path	Optional OKX storage path.
limit	Integer page size.
tz	Output time zone.
storage	Local storage mode: monolithic "single" or monthly partitioned "monthly".

Value

A sync result list.

resolve_sec_cik	<i>Resolve An SEC CIK From A Ticker</i>
-----------------	---

Description

Resolve An SEC CIK From A Ticker

Usage

```
resolve_sec_cik(ticker, mappings = NULL, config = NULL)
```

Arguments

ticker	Security ticker.
mappings	Optional table returned by get_sec_company_tickers().
config	Optional SEC configuration used when mappings are not supplied.

Value

Unpadded CIK character scalar.

standardize_fund_holdings	<i>Standardize Fund Holdings Across Issuers</i>
---------------------------	---

Description

Converts issuer-specific snapshots to the provider-neutral holdings contract.

Usage

```
standardize_fund_holdings(
  data,
  provider,
  fund_id,
  as_of_date = NULL,
  column_map = list()
)
```

Arguments

data	Holdings table.
provider	Stable provider identifier, such as ishare.
fund_id	Stable fund ticker or identifier.
as_of_date	Optional snapshot date overriding the source column.
column_map	Optional named list mapping canonical names to source columns.

Value

A provider-neutral long data.table.

sync_all_alphavantage_registry_data
<i>Synchronize Registered Alpha Vantage Series</i>

Description

Synchronize Registered Alpha Vantage Series

Usage

```
sync_all_alphavantage_registry_data(  
  registry = get_alphavantage_registry(),  
  config = NULL,  
  local_path = NULL,  
  ...  
)
```

Arguments

registry	Optional Alpha Vantage registry.
config	Optional API configuration.
local_path	Optional storage directory.
...	Passed to sync_local_alphavantage_data().

Value

A standardized batch summary data.table.

`sync_all_bea_registry_data`*Synchronize Registered BEA Series*

Description

Synchronize Registered BEA Series

Usage

```
sync_all_bea_registry_data(  
    registry = get_bea_registry(),  
    config = NULL,  
    local_path = NULL  
)
```

Arguments

registry	Optional BEA registry.
config	Optional BEA configuration.
local_path	Optional storage directory.

Value

A batch summary.

`sync_all_bls_registry_data`*Synchronize Registered BLS Series*

Description

Synchronize Registered BLS Series

Usage

```
sync_all_bls_registry_data(  
    registry = get_bls_registry(),  
    config = NULL,  
    local_path = NULL,  
    ...  
)
```

Arguments

registry	Optional BLS registry.
config	Optional BLS configuration.
local_path	Optional BLS storage directory.
...	Passed to sync_local_bls_data().

Value

A batch summary.

sync_all_census_registry_data
<i>Synchronize Registered Census Series</i>

Description

Synchronize Registered Census Series

Usage

```
sync_all_census_registry_data(  
  registry = get_census_registry(),  
  config = NULL,  
  local_path = NULL  
)
```

Arguments

registry	Optional Census registry.
config	Optional Census configuration.
local_path	Optional storage directory.

Value

A batch summary.

sync_all_cftc_cot_registry_data
<i>Synchronize All Registered CFTC COT Reports</i>

Description

Synchronize All Registered CFTC COT Reports

Usage

```
sync_all_cftc_cot_registry_data(  
  registry = get_cftc_cot_registry(),  
  local_path = NULL,  
  ...  
)
```

Arguments

- registry Optional CFTC registry table.
- local_path Optional CFTC storage directory.
- ... Passed to sync_local_cftc_cot().

Value

A standardized batch summary data.table.

sync_all_crypto_derivatives_registry_data
<i>Synchronize All Registered Crypto Derivatives Data</i>

Description

Synchronize All Registered Crypto Derivatives Data

Usage

```
sync_all_crypto_derivatives_registry_data(  
  registry = get_crypto_derivatives_registry(),  
  local_path = NULL,  
  ...  
)
```

Arguments

- registry Optional derivatives registry.
- local_path Optional derivatives storage directory.
- ... Passed to sync_local_crypto_derivatives().

Value

A standardized batch summary data.table.

sync_all_eia_registry_data
<i>Synchronize All Registered EIA Series</i>

Description

Synchronize All Registered EIA Series

Usage

```
sync_all_eia_registry_data(  
  registry = get_eia_registry(),  
  config = NULL,  
  local_path = NULL,  
  ...  
)
```

Arguments

- registry Optional EIA registry table.
- config Optional EIA API configuration.
- local_path Optional EIA storage directory.
- ... Passed to sync_local_eia_data().

Value

A standardized batch summary data.table.

sync_all_fiscaldata_registry_data
<i>Synchronize All Registered Treasury Fiscal Data</i>

Description

Synchronize All Registered Treasury Fiscal Data

Usage

```
sync_all_fiscaldata_registry_data(  
  registry = get_fiscaldata_registry(),  
  local_path = NULL,  
  ...  
)
```

Arguments

- registry Optional Fiscal Data registry table.
- local_path Optional Fiscal Data storage directory.
- ... Passed to sync_local_fiscaldata().

Value

A standardized batch summary data.table.

sync_all_fred_registry_data
<i>Synchronize All FRED Series In The Registry</i>

Description

Synchronize All FRED Series In The Registry

Usage

```
sync_all_fred_registry_data(  
  registry = get_fred_registry(),  
  config = NULL,  
  local_path = NULL,  
  from_server = FALSE,  
  tz = "America/Chicago"  
)
```

Arguments

registry	Optional FRED registry table.
config	Optional FRED API config.
local_path	Optional local storage path.
from_server	Logical. If TRUE, use FRED server-reported update time.
tz	Time zone used for source update time inference.

Value

Summary data.table.

sync_all_ishare_registry_data
<i>Synchronize All iShares Tickers In The Registry</i>

Description

Synchronize All iShares Tickers In The Registry

Usage

```
sync_all_ishare_registry_data(  
  registry = get_ishare_registry(),  
  local_path = NULL,  
  cache_dir = NULL,  
  source_ftime = NULL,  
  ishare_mega_data = NULL  
)
```

Arguments

registry	Optional iShares registry table.
local_path	Optional local storage path.
cache_dir	Optional cache directory.
source_ftime	Optional upstream update time shared across the batch.
ishare_mega_data	Optional metadata table used to resolve fund URLs.

Value

Summary data.table.

sync_all_ishare_registry_holdings
<i>Synchronize All iShares Holdings In The Registry</i>

Description

Synchronize All iShares Holdings In The Registry

Usage

```
sync_all_ishare_registry_holdings(  
  registry = NULL,  
  tickers = NULL,  
  local_path = NULL,  
  cache_dir = NULL,  
  source_ftime = NULL,  
  ishare_mega_data = NULL  
)
```

Arguments

registry	Optional iShares registry table. When NULL, the function reads the full iShares registry and filters it to the configured holdings ticker list.
tickers	Optional character vector of holdings tickers to sync. When NULL, the function uses iShare.holdings_tickers from package config and falls back to c("DYNF", "THRO", "BAI", "BDYN", "BDVL").
local_path	Optional local storage path.
cache_dir	Optional cache directory.
source_ftime	Optional upstream update time shared across the batch.
ishare_mega_data	Optional metadata table used to resolve fund URLs.

Value

Summary data.table.

sync_all_rss_registry_data

Synchronize All RSS Registry Feeds

Description

Synchronize All RSS Registry Feeds

Usage

```
sync_all_rss_registry_data(
  registry = get_rss_registry(),
  local_path = NULL,
  ca_bundles = NULL
)
```

Arguments

registry	Optional RSS registry table.
local_path	Optional local storage path.
ca_bundles	Optional named list or character vector mapping feed IDs to request-scoped CA bundle paths. When omitted, RSS.feed_ca_bundles is read from package configuration.

Value

Summary data.table.

sync_all_sdmx_registry_data

Synchronize All Registered SDMX Series

Description

Synchronize All Registered SDMX Series

Usage

```
sync_all_sdmx_registry_data(
  registry = get_sdmx_registry(),
  local_path = NULL,
  ...
)
```

Arguments

registry	Optional SDMX series registry.
local_path	Optional SDMX storage directory.
...	Passed to sync_local_sdmx_data().

Value

A standardized batch summary data.table.

sync_all_sec_companyfacts_registry_data
<i>Synchronize Registered SEC Company Facts</i>

Description

Synchronize Registered SEC Company Facts

Usage

```
sync_all_sec_companyfacts_registry_data(  
  registry = get_sec_registry(),  
  config = NULL,  
  local_path = NULL,  
  ...  
)
```

Arguments

registry	Optional SEC company registry.
config	Optional SEC configuration.
local_path	Optional SEC Company Facts storage directory.
...	Passed to sync_local_sec_companyfacts().

Value

A standardized batch summary data.table.

`sync_all_sec_frames_registry_data`*Synchronize Registered SEC XBRL Frames*

Description

Synchronize Registered SEC XBRL Frames

Usage

```
sync_all_sec_frames_registry_data(  
    registry = get_sec_frames_registry(),  
    config = NULL,  
    local_path = NULL  
)
```

Arguments

registry	Optional Frames registry.
config	Optional SEC configuration.
local_path	Optional frame storage directory.

Value

A standardized batch summary.

`sync_all_sec_submissions_registry_data`*Synchronize Registered SEC Submissions*

Description

Synchronize Registered SEC Submissions

Usage

```
sync_all_sec_submissions_registry_data(  
    registry = get_sec_registry(),  
    config = NULL,  
    local_path = NULL,  
    ...  
)
```

Arguments

- registry Optional SEC company registry.
- config Optional SEC configuration.
- local_path Optional SEC submissions storage directory.
- ... Passed to sync_local_sec_submissions().

Value

A standardized batch summary data.table.

sync_all_treasury_rates
<i>Synchronize All Treasury Rate Datasets</i>

Description

Synchronize All Treasury Rate Datasets

Usage

```
sync_all_treasury_rates(  
  datasets = names(.treasury_dataset_map()),  
  years = NULL,  
  local_path = NULL  
)
```

Arguments

- datasets Optional dataset vector. If omitted, sync all supported Treasury rate datasets.
- years Optional year filter passed to sync_local_treasury_rates(). When omitted, each dataset uses a full-history backfill only if no local file exists; otherwise it syncs from the latest local year through the current year.
- local_path Optional local storage path.

Value

Summary data.table.

`sync_all_wbstats_registry_data`*Synchronize All World Bank Registry Series*

Description

Synchronize All World Bank Registry Series

Usage

```
sync_all_wbstats_registry_data(  
    registry = get_wbstats_registry(),  
    local_path = NULL,  
    ...  
)
```

Arguments

<code>registry</code>	Optional World Bank registry table.
<code>local_path</code>	Optional local storage path.
<code>...</code>	Passed to <code>sync_local_wbstats_data()</code> .

Value

Summary data.table.

`sync_all_yahoofinance_registry_data`*Synchronize All Yahoo Finance Tickers In The Registry*

Description

Synchronize All Yahoo Finance Tickers In The Registry

Usage

```
sync_all_yahoofinance_registry_data(  
    from = NULL,  
    to = Sys.Date(),  
    registry = get_yahoofinance_registry(),  
    local_path = NULL,  
    src = "yahoo",  
    overlap_days = 10L,  
    initial_lookback_days = 400L,  
    max_attempts = 3L,
```

```

    retry_delay_seconds = 1,
    fallback_source = NULL,
    fallback_ticker = NULL
)

```

Arguments

from	Optional start date passed to <code>quantmod::getSymbols()</code> . When omitted, the function derives a per-ticker start date from the latest local record minus <code>overlap_days</code> , or falls back to <code>to - initial_lookback_days</code> for tickers without local data.
to	End date passed to <code>quantmod::getSymbols()</code> .
registry	Optional Yahoo Finance registry table. When omitted, the configured runtime registry is validated against the package seed before any provider request.
local_path	Optional local storage path.
src	quantmod source, default "yahoo".
overlap_days	Integer safety overlap used when deriving per-ticker incremental start dates from local data.
initial_lookback_days	Integer fallback lookback for tickers without local data when <code>from</code> is omitted.
max_attempts	Maximum bounded attempts for a transient source failure.
retry_delay_seconds	Initial retry delay in seconds; delays use exponential backoff.
fallback_source	Optional explicitly configured fallback provider. Currently supports "eastmoney" for daily OHLC data. For Yahoo sources, a failed dated quantmod request first retries Yahoo's chart endpoint with a bounded range before this external fallback is considered.
fallback_ticker	Optional provider-specific fallback identifier.

Value

Summary `data.table`.

`sync_local_alphavantage_data`

Synchronize One Alpha Vantage Daily Series

Description

Synchronize One Alpha Vantage Daily Series

Usage

```
sync_local_alphavantage_data(  
    symbol,  
    mode = NULL,  
    config = NULL,  
    local_path = NULL  
)
```

Arguments

- symbol A character string for the stock ticker symbol (e.g., "AAPL", "TSLA").
- mode Character string; either "compact" (latest 100 days) or "full" (full history). Defaults to "compact".
- config A list of AlphaVantage API settings, typically retrieved via tool_set_config("alphavantage").
- local_path Optional Alpha Vantage storage directory.

Value

A standard synchronization result.

sync_local_bea_data	<i>Synchronize One BEA Regional Series</i>
---------------------	--

Description

Synchronize One BEA Regional Series

Usage

```
sync_local_bea_data(  
    series_id,  
    table_name,  
    line_code,  
    geofips = "STATE",  
    year = "ALL",  
    label = NULL,  
    config = NULL,  
    local_path = NULL  
)
```

Arguments

- series_id Stable local series identifier.
- table_name BEA Regional table name.
- line_code BEA line code.

geofips	Geography selector such as STATE or COUNTY.
year	Year selector, default ALL.
label	Optional local label.
config	Optional BEA configuration.
local_path	Optional BEA storage directory.

Value

A standard synchronization result.

sync_local_binance_klines

Synchronize Local Binance Kline Data

Description

Synchronize Local Binance Kline Data

Usage

```
sync_local_binance_klines(
    symbol = "ETHUSD",
    interval = "1m",
    start_time = NULL,
    end_time = NULL,
    limit = 1500L,
    tz = "UTC",
    paginate = TRUE,
    local_path = NULL,
    storage = c("single", "monthly")
)
```

Arguments

symbol	Trading pair symbol.
interval	Candlestick interval.
start_time	Optional start time.
end_time	Optional end time.
limit	Page size. Binance futures allows up to 1500.
tz	Time zone applied to returned timestamps.
paginate	Logical. Request multiple pages when needed.
local_path	Optional Binance storage path.
storage	Local storage mode: monolithic "single" or monthly partitioned "monthly".

Value

A sync result list.

sync_local_bls_data	<i>Synchronize One BLS Series</i>
---------------------	-----------------------------------

Description

Synchronize One BLS Series

Usage

```
sync_local_bls_data(  
    series_id,  
    label = NULL,  
    from = NULL,  
    to = NULL,  
    config = NULL,  
    local_path = NULL,  
    overlap_years = 2L  
)
```

Arguments

- | | |
|---------------|---------------------------------|
| series_id | BLS series identifier. |
| label | Optional local label. |
| from, to | Optional year or date bounds. |
| config | Optional BLS configuration. |
| local_path | Optional BLS storage directory. |
| overlap_years | Years re-fetched for revisions. |

Value

A standard synchronization result.

sync_local_census_data
<i>Synchronize One Census Series</i>

Description

Synchronize One Census Series

Usage

```
sync_local_census_data(  
  series_id,  
  dataset,  
  data_type_code,  
  category_code,  
  seasonally_adj = "yes",  
  from = NULL,  
  to = NULL,  
  label = NULL,  
  config = NULL,  
  local_path = NULL,  
  overlap_months = 24L  
)
```

Arguments

- series_id Stable local series identifier.
- dataset Economic Indicators dataset suffix, such as marts.
- data_type_code Census item type code.
- category_code Census industry/category code.
- seasonally_adj Seasonal adjustment value.
- from, to Optional time bounds.
- label Optional local label.
- config Optional Census configuration.
- local_path Optional Census storage directory.
- overlap_months Months re-fetched for revisions.

Value

A standard synchronization result.

sync_local_cftc_cot	<i>Synchronize One CFTC COT Report</i>
---------------------	--

Description

Synchronize One CFTC COT Report

Usage

```
sync_local_cftc_cot(  
  report_variant = c("futures_only", "combined"),  
  report_type = "tff",  
  report_id = NULL,  
  dataset_id = NULL,  
  market_codes = NULL,  
  from = NULL,  
  to = NULL,  
  local_path = NULL,  
  overlap_days = 14L,  
  page_size = 5000L  
)
```

Arguments

report_variant	Report variant: "futures_only" or "combined".
report_type	Report family: "tff", "disaggregated", or "legacy".
report_id	Stable local report identifier.
dataset_id	Optional official Socrata dataset identifier.
market_codes	Optional CFTC contract-market codes.
from, to	Optional inclusive report-date bounds.
local_path	Optional CFTC storage directory.
overlap_days	Number of days re-fetched around the latest local report.
page_size	Number of source rows requested per page.

Value

A local synchronization result list.

`sync_local_crypto_derivatives`*Synchronize One Crypto Derivatives Dataset*

Description

Synchronize One Crypto Derivatives Dataset

Usage

```
sync_local_crypto_derivatives(  
  provider,  
  dataset_type,  
  symbol,  
  interval = NULL,  
  from = NULL,  
  to = NULL,  
  limit = NULL,  
  local_path = NULL,  
  overlap_days = 2L,  
  config = NULL,  
  storage = c("single", "monthly")  
)
```

Arguments

<code>provider</code>	binance or okx.
<code>dataset_type</code>	Funding, open-interest, mark/index price, basis, or Binance long-short-ratio dataset identifier.
<code>symbol</code>	Provider instrument identifier.
<code>interval</code>	Required Binance open-interest period; otherwise a local label.
<code>from, to</code>	Optional inclusive UTC time bounds.
<code>limit</code>	Provider page size.
<code>local_path</code>	Optional derivatives storage directory.
<code>overlap_days</code>	Days re-fetched before the latest local observation.
<code>config</code>	Optional exchange configuration.
<code>storage</code>	Local storage mode: monolithic "single" or monthly partitioned "monthly".

Value

A local synchronization result list.

sync_local_crypto_liquidations	<i>Store Captured Crypto Liquidation Events</i>
--------------------------------	---

Description

Exchange REST APIs do not provide trustworthy market-wide liquidation history. This function persists events captured from Binance or OKX public liquidation streams without presenting private force-order history as market data.

Usage

```
sync_local_crypto_liquidations(  
  events,  
  provider,  
  symbol,  
  local_path = NULL,  
  storage = c("single", "monthly")  
)
```

Arguments

events	Event rows containing a time column and price/quantity fields.
provider	Exchange identifier.
symbol	Exchange instrument identifier.
local_path	Optional derivatives storage directory.
storage	Local storage mode.

Value

A local synchronization result list.

sync_local_data	<i>Synchronize Local Data</i>
-----------------	-------------------------------

Description

Merges freshly retrieved source data into a local .rds file and stores a sidecar metadata file containing the local update time and source update time.

Usage

```
sync_local_data(  
  new_data,  
  local_file_path,  
  key_cols,  
  order_cols = key_cols,  
  source_ftime = NULL,  
  local_updated_at = Sys.time()  
)
```

Arguments

new_data	A data.table-compatible object.
local_file_path	Output .rds file path.
key_cols	Character vector of key columns used for de-duplication.
order_cols	Character vector used to sort the merged data.
source_ftime	Optional upstream update time recorded in metadata.
local_updated_at	Optional local update time override.

Value

A list describing the sync result.

sync_local_data_batches

Synchronize Local Data From In-Memory Batches

Description

Combines multiple freshly retrieved data batches in memory, de-duplicates them by key_cols, then calls sync_local_data() once. This is useful for gap repair workflows where calling sync_local_data() once per small page would repeatedly rewrite the same local .rds file.

Usage

```
sync_local_data_batches(  
  batches,  
  local_file_path,  
  key_cols,  
  order_cols = key_cols,  
  source_ftime = NULL,  
  local_updated_at = Sys.time()  
)
```

Arguments

batches	A list of data.table-compatible batches.
local_file_path	Output .rds file path.
key_cols	Character vector of key columns used for de-duplication.
order_cols	Character vector used to sort the merged data.
source_ftime	Optional upstream update time recorded in metadata.
local_updated_at	Optional local update time override.

Value

A sync result list.

sync_local_data_partitioned
Synchronize Monthly Partitioned Local Data

Description

Upserts only monthly partitions touched by new_data. On the first partitioned sync, an existing monolithic RDS cache is copied into monthly partitions so callers can opt in without a separate migration step.

Usage

```
sync_local_data_partitioned(
  new_data,
  local_file_path,
  time_col,
  key_cols,
  order_cols = key_cols,
  source_ftime = NULL,
  local_updated_at = Sys.time(),
  tz = "UTC"
)
```

Arguments

new_data	A data.table-compatible object.
local_file_path	Output .rds file path.
time_col	Name of the timestamp column used for monthly partitioning.
key_cols	Character vector of key columns used for de-duplication.

order_cols	Character vector used to sort the merged data.
source_ftime	Optional upstream update time recorded in metadata.
local_updated_at	Optional local update time override.
tz	Time zone used to determine month boundaries.

Value

A sync result list. Its data element contains the touched rows, rather than the complete potentially large dataset.

sync_local_eia_data	<i>Synchronize One EIA Series</i>
---------------------	-----------------------------------

Description

Synchronize One EIA Series

Usage

```
sync_local_eia_data(  
  series_id,  
  label = NULL,  
  config = NULL,  
  from = NULL,  
  to = NULL,  
  local_path = NULL,  
  overlap_days = 31L,  
  page_size = 5000L  
)
```

Arguments

series_id	EIA series identifier.
label	Optional local label.
config	Optional EIA API configuration.
from, to	Optional inclusive period bounds.
local_path	Optional EIA storage directory.
overlap_days	Days re-fetched around the latest local observation.
page_size	Number of observations requested per page.

Value

A local synchronization result list.

sync_local_fiscaldata	<i>Synchronize One Treasury Fiscal Data Table</i>
-----------------------	---

Description

Synchronize One Treasury Fiscal Data Table

Usage

```
sync_local_fiscaldata(  
  dataset_id,  
  endpoint,  
  date_col = "record_date",  
  key_cols = date_col,  
  fields = NULL,  
  from = NULL,  
  to = NULL,  
  local_path = NULL,  
  overlap_days = 14L,  
  page_size = 10000L  
)
```

Arguments

dataset_id	Stable local dataset identifier.
endpoint	Fiscal Data endpoint relative to the fiscal-service base URL.
date_col	Source date column.
key_cols	Source columns that uniquely identify observations per date.
fields	Optional source fields to request.
from, to	Optional inclusive date bounds.
local_path	Optional Fiscal Data storage directory.
overlap_days	Days re-fetched around the latest local date.
page_size	Number of source rows requested per page.

Value

A local synchronization result list.

sync_local_fred_data	<i>Synchronize Local FRED Data</i>
----------------------	------------------------------------

Description

Synchronize Local FRED Data

Usage

```
sync_local_fred_data(  
  series_id,  
  config = NULL,  
  local_path = NULL,  
  from_server = FALSE,  
  tz = "America/Chicago"  
)
```

Arguments

series_id	FRED series identifier.
config	Optional FRED API config.
local_path	Optional local storage path.
from_server	Logical. If TRUE, use the FRED server-reported update time. Otherwise use a frequency-based fallback.
tz	Time zone used for source update time inference.

Value

A sync result list.

sync_local_fund_holdings	<i>Synchronize Provider-Neutral Fund Holdings</i>
--------------------------	---

Description

Synchronize Provider-Neutral Fund Holdings

Usage

```
sync_local_fund_holdings(  
  data,  
  provider,  
  fund_id,  
  local_path = NULL,  
  as_of_date = NULL,  
  column_map = list(),  
  source_ftime = NULL  
)
```

Arguments

data	Holdings table, standardized internally.
provider	Fund issuer/provider identifier.
fund_id	Fund ticker or stable identifier.
local_path	Optional generic holdings storage root.
as_of_date	Optional snapshot-date override.
column_map	Optional canonical-to-source column mapping.
source_ftime	Optional upstream update time.

Value

A standard synchronization result.

sync_local_ishare_data
<i>Synchronize Local iShares Historical Data</i>

Description

Synchronize Local iShares Historical Data

Usage

```
sync_local_ishare_data(  
  ticker,  
  ishare_mega_data = NULL,  
  local_path = NULL,  
  cache_dir = NULL,  
  source_ftime = NULL  
)
```

Arguments

ticker	ETF ticker.
ishare_mega_data	Optional iShares registry table.
local_path	Optional local storage path.
cache_dir	Optional XLS cache directory.
source_untime	Optional upstream update time.

Value

A sync result list.

sync_local_ishare_holdings
<i>Synchronize Local iShares Holdings Data</i>

Description

Synchronize Local iShares Holdings Data

Usage

```
sync_local_ishare_holdings(  
    ticker,  
    ishare_mega_data = NULL,  
    local_path = NULL,  
    cache_dir = NULL,  
    source_untime = NULL  
)
```

Arguments

ticker	ETF ticker.
ishare_mega_data	Optional iShares registry table.
local_path	Optional local storage path.
cache_dir	Optional XLS cache directory.
source_untime	Optional upstream update time.

Value

A sync result list.

sync_local_okx_candle	<i>Synchronize Local OKX Candle Data</i>
-----------------------	--

Description

Synchronize Local OKX Candle Data

Usage

```
sync_local_okx_candle(  
    inst_id,  
    bar,  
    config = NULL,  
    local_path = NULL,  
    mode = c("latest", "history"),  
    before = NULL,  
    limit = 100L,  
    tz = "UTC",  
    storage = c("single", "monthly")  
)
```

Arguments

inst_id	Instrument identifier.
bar	Candle interval.
config	Optional OKX API config. If omitted, defaults from the package config and OKX_API_KEY / OKX_SECRET_KEY / OKX_PASSPHRASE environment variables are used.
local_path	Optional OKX storage path.
mode	Either "latest" or "history".
before	Optional history cursor.
limit	Integer page size.
tz	Output time zone.
storage	Local storage mode: monolithic "single" or monthly partitioned "monthly".

Value

A sync result list.

sync_local_quantmod_OHLC

Synchronize Local quantmod OHLC Data

Description

Synchronize Local quantmod OHLC Data

Usage

```
sync_local_quantmod_OHLC(
  ticker,
  label = ticker,
  from,
  to,
  src = "yahoo",
  local_path = NULL,
  max_attempts = 3L,
  retry_delay_seconds = 1,
  fallback_source = NULL,
  fallback_ticker = ticker
)
```

Arguments

ticker	Market symbol passed to <code>quantmod::getSymbols()</code> .
label	Optional label to store in the standardized symbol column.
from	Start date.
to	End date.
src	quantmod source, default "yahoo".
local_path	Optional local storage path.
max_attempts	Maximum bounded attempts for a transient source failure.
retry_delay_seconds	Initial retry delay in seconds; delays use exponential backoff.
fallback_source	Optional explicitly configured fallback provider. Currently supports "eastmoney" for daily OHLC data. For Yahoo sources, a failed dated quantmod request first retries Yahoo's chart endpoint with a bounded range before this external fallback is considered.
fallback_ticker	Optional provider-specific fallback identifier.

Details

An external fallback never replaces a finite existing primary-source OHLC bar. It only fills keys whose local bars are missing or invalid.

Value

A sync result list.

sync_local_rss_data	<i>Synchronize Local RSS Data</i>
---------------------	-----------------------------------

Description

Synchronize Local RSS Data

Usage

```
sync_local_rss_data(  
  feed_id,  
  url,  
  parser = c("plain", "gdpnow"),  
  local_path = NULL,  
  ca_bundle = NULL  
)
```

Arguments

- | | |
|------------|--|
| feed_id | Local feed identifier. |
| url | RSS feed URL. |
| parser | Parsing strategy. Currently supports "plain" and "gdpnow". |
| local_path | Optional local storage path. |
| ca_bundle | Optional CA bundle path used only for this feed request. |

Value

A sync result list.

sync_local_sdmx_data	<i>Synchronize One SDMX Registry Entry</i>
----------------------	--

Description

Synchronize One SDMX Registry Entry

Usage

```

sync_local_sdmx_data(
  series_id,
  provider,
  base_url,
  agency = NULL,
  dataflow,
  version = "latest",
  key = "",
  flow_ref = NULL,
  format = NULL,
  accept = "text/csv",
  time_col = "TIME_PERIOD",
  value_col = "OBS_VALUE",
  dimension_cols = NULL,
  label = NULL,
  frequency = NULL,
  from = NULL,
  to = NULL,
  local_path = NULL,
  overlap_days = 62L
)

```

Arguments

series_id	Stable local series identifier.
provider	Provider dialect: oecd, ecb, bis, eurostat, or the IMF DataMapper-compatible imf adapter.
base_url	Provider REST base URL.
agency	Optional SDMX agency identifier.
dataflow	SDMX dataflow identifier.
version	SDMX dataflow version.
key	SDMX series key.
flow_ref	Optional provider-specific flow reference.
format	Provider query format.
accept	HTTP response media type.
time_col, value_col	Source observation columns.
dimension_cols	Columns identifying distinct series in the response.
label, frequency	Optional local metadata.
from, to	Optional inclusive SDMX periods.
local_path	Optional SDMX storage directory.
overlap_days	Days re-fetched around the latest local period.

Value

A local synchronization result list.

sync_local_sec_bulk_archive
<i>Download An Optional SEC Bulk Archive</i>

Description

Download An Optional SEC Bulk Archive

Usage

```
sync_local_sec_bulk_archive(  
    archive = c("companyfacts", "submissions"),  
    local_path = NULL,  
    config = NULL,  
    overwrite = FALSE,  
    extract = FALSE  
)
```

Arguments

archive	One of companyfacts or submissions.
local_path	Optional SEC bulk archive directory.
config	Optional SEC configuration.
overwrite	Replace the existing ZIP.
extract	Extract the ZIP into a same-named directory.

Value

Download metadata.

sync_local_sec_companyfacts
<i>Synchronize SEC Company Facts For One Company</i>

Description

Synchronize SEC Company Facts For One Company

Usage

```
sync_local_sec_companyfacts(  
  cik,  
  ticker = NULL,  
  company_name = NULL,  
  concepts = NULL,  
  forms = NULL,  
  from = NULL,  
  to = NULL,  
  config = NULL,  
  local_path = NULL,  
  overlap_days = 31L  
)
```

Arguments

cik	SEC Central Index Key.
ticker	Optional ticker label.
company_name	Optional company name label.
concepts	Optional concept names or taxonomy:concept identifiers.
forms	Optional filing forms to retain.
from, to	Optional inclusive filed-date bounds.
config	Optional SEC configuration.
local_path	Optional SEC Company Facts storage directory.
overlap_days	Filed-date overlap used for local upserts.

Value

A local synchronization result list.

sync_local_sec_filing_document

Download A Selected SEC Filing Document

Description

Download A Selected SEC Filing Document

Usage

```
sync_local_sec_filing_document(  
  cik,  
  accession_number,  
  primary_document,  
  local_path = NULL,  
  config = NULL,  
  overwrite = FALSE  
)
```

Arguments

- cik SEC Central Index Key.
- accession_number Filing accession number.
- primary_document Primary filing document name from submissions data.
- local_path Optional document cache directory.
- config Optional SEC configuration.
- overwrite Replace an existing local document.

Value

A one-row metadata data.table.

sync_local_sec_frame	<i>Synchronize One SEC XBRL Frame</i>
----------------------	---------------------------------------

Description

Synchronize One SEC XBRL Frame

Usage

```
sync_local_sec_frame(  
  taxonomy,  
  tag,  
  unit,  
  period,  
  config = NULL,  
  local_path = NULL  
)
```

Arguments

taxonomy	XBRL taxonomy, such as us-gaap.
tag	XBRL concept tag.
unit	XBRL unit, such as USD.
period	SEC frame period, such as CY2025Q4I.
config	Optional SEC configuration.
local_path	Optional SEC frame cache directory.

Value

A standard synchronization result.

sync_local_sec_submissions
<i>Synchronize SEC Submissions For One Company</i>

Description

Synchronize SEC Submissions For One Company

Usage

```
sync_local_sec_submissions(  
  cik,  
  ticker = NULL,  
  company_name = NULL,  
  forms = NULL,  
  from = NULL,  
  to = NULL,  
  include_history = TRUE,  
  config = NULL,  
  local_path = NULL,  
  overlap_days = 14L  
)
```

Arguments

cik	SEC Central Index Key.
ticker	Optional ticker label.
company_name	Optional company name label.
forms	Optional filing forms to retain.
from, to	Optional inclusive filing-date bounds.
include_history	Logical; retrieve older SEC submission files in addition to recent filings.

config	Optional SEC configuration.
local_path	Optional SEC submissions storage directory.
overlap_days	Filing-date overlap for incremental synchronization.

Value

A local synchronization result list.

sync_local_treasury_rates
<i>Synchronize Local Treasury Rates</i>

Description

Synchronize Local Treasury Rates

Usage

sync_local_treasury_rates(dataset, years = NULL, local_path = NULL)

Arguments

dataset	Treasury dataset key. Supported values are "bill_rates", "par_yield_curve", "long_term_rates", "real_yield_curve", and "real_long_term_rates".
years	Optional integer vector of calendar years to fetch. If omitted, the function combines the historical archive with current-year annual XML feeds from 2024 through the current year.
local_path	Optional local storage path.

Value

A sync result list.

sync_local_wbstats_data
<i>Synchronize Local World Bank Data</i>

Description

Synchronize Local World Bank Data

Usage

sync_local_wbstats_data(indicator, country, freq = "Y", local_path = NULL, ...)

Arguments

indicator	World Bank indicator code.
country	Country or aggregate code.
freq	Frequency code.
local_path	Optional local storage path.
...	Passed to get_source_data_wbstats().

Value

A sync result list.

sync_sec_filing_documents
<i>Download Selected Documents From Local SEC Submissions</i>

Description

Download Selected Documents From Local SEC Submissions

Usage

```
sync_sec_filing_documents(  
  cik,  
  forms = NULL,  
  from = NULL,  
  to = NULL,  
  limit = 20L,  
  submissions_path = NULL,  
  local_path = NULL,  
  config = NULL  
)
```

Arguments

cik	SEC Central Index Key.
forms	Optional filing-form filter.
from, to	Optional filing-date bounds.
limit	Maximum number of newest documents.
submissions_path	Optional submissions cache directory.
local_path	Optional document cache directory.
config	Optional SEC configuration.

Value

One metadata row per selected document.

`validate_yahoofinance_registry`*Validate Yahoo Finance Runtime Registry*

Description

Compare fallback declarations in a runtime registry with the package-managed seed. Additional runtime-only ticker metadata is permitted; every seed fallback declaration must be present unchanged.

Usage

```
validate_yahoofinance_registry(  
  registry_path = get_yahoofinance_registry_file_path(),  
  seed_path = get_yahoofinance_seed_registry_path()  
)
```

Arguments

<code>registry_path</code>	Runtime registry JSON path.
<code>seed_path</code>	Package seed registry JSON path.

Value

A list with valid, missing, and mismatched data tables.

Index

add_fred_registry_series, 6
add_ishare_registry_ticker, 6
add_sec_registry_company, 7
add_wbstats_registry_series, 8

bootstrap_yahoofinance_registry, 9

describe_alphavantage_data, 9
describe_bea_data, 10
describe_binance_data, 10
describe_bls_data, 11
describe_census_data, 12
describe_cftc_cot_data, 12
describe_crypto_derivatives, 13
describe_eia_data, 13
describe_fiscaldata, 14
describe_fred_data, 14
describe_ishare_data, 15
describe_market_ohlc_data, 15
describe_okx_candle_data, 16
describe_quantmod_data, 16
describe_rss_data, 17
describe_sdmx_data, 18
describe_sec_companyfacts, 18
describe_sec_frame, 19
describe_sec_submissions, 19
describe_treasury_rates, 20
describe_wbstats_data, 20
detect_time_gaps, 21
detect_time_gaps_fred, 21
detect_time_gaps_okx_candle, 22
detect_time_gaps_wbstats, 22

fetch_quantmod_OHLC, 23

get_alphavantage_registry, 24
get_alphavantage_registry_file_path, 25
get_bea_registry, 25
get_bea_registry_file_path, 26

get_bls_registry, 26
get_bls_registry_file_path, 27
get_census_registry, 27
get_census_registry_file_path, 28
get_cftc_cot_registry, 28
get_cftc_cot_registry_file_path, 29
get_completed_local_quantmod_OHLC, 29
get_crypto_derivatives_registry, 30
get_crypto_derivatives_registry_file_path, 30
get_eia_registry, 31
get_eia_registry_file_path, 31
get_fiscaldata_registry, 32
get_fiscaldata_registry_file_path, 32
get_fred_registry, 33
get_fred_registry_file_path, 33
get_investdatar_config, 34
get_ishare_registry, 34
get_ishare_registry_file_path, 35
get_latest_sync_run, 35
get_local_alphavantage_data, 36
get_local_bea_data, 36
get_local_binance_klines, 37
get_local_bls_data, 37
get_local_census_data, 38
get_local_cftc_cot, 38
get_local_crypto_derivatives, 39
get_local_data_meta, 40
get_local_data_partitioned, 40
get_local_data_utime, 41
get_local_eia_data, 41
get_local_fiscaldata, 42
get_local_FRED_data, 42
get_local_fund_holdings, 43
get_local_ishare_data, 43
get_local_ishare_holdings, 44
get_local_ishare_holdings_standardized, 44
get_local_ishare_mega_data, 45

- get_local_okx_candle, 45
- get_local_quantmod_OHLC, 46
- get_local_rss_data, 46
- get_local_sdmx_data, 47
- get_local_sec_companyfacts, 47
- get_local_sec_frame, 48
- get_local_sec_submissions, 48
- get_local_treasury_rates, 49
- get_local_wbstats_data, 49
- get_rss_registry, 50
- get_rss_registry_file_path, 50
- get_sdmx_registry, 51
- get_sdmx_registry_file_path, 51
- get_sec_company_tickers, 52
- get_sec_frames_registry, 52
- get_sec_frames_registry_file_path, 53
- get_sec_registry, 53
- get_sec_registry_file_path, 54
- get_source_config, 54
- get_source_data_alphavantage_ts_daily, 55
- get_source_data_bea, 56
- get_source_data_binance_klines, 56
- get_source_data_bls, 57
- get_source_data_census, 58
- get_source_data_cftc_cot, 59
- get_source_data_crypto_derivatives, 60
- get_source_data_eia, 61
- get_source_data_fiscaldata, 61
- get_source_data_fred, 62
- get_source_data_ishare, 63
- get_source_data_ishare_holdings, 63
- get_source_data_okx_candle, 64
- get_source_data_path, 65
- get_source_data_rss, 65
- get_source_data_sdmx, 66
- get_source_data_sec_companyfacts, 67
- get_source_data_sec_frame, 68
- get_source_data_sec_submissions, 69
- get_source_data_treasury_rates, 70
- get_source_data_wbstats, 70
- get_source_hist_data_okx_candle, 71
- get_source_metadata_fred, 72
- get_source_spec, 73
- get_source_untime_cftc_cot, 73
- get_source_untime_crypto_derivatives, 74
- get_source_untime_eia, 75
- get_source_untime_fiscaldata, 75
- get_source_untime_fred, 76
- get_source_untime_ishare, 76
- get_source_untime_okx_candle, 77
- get_source_untime_rss, 77
- get_source_untime_sdmx, 78
- get_source_untime_sec_companyfacts, 79
- get_source_untime_sec_submissions, 79
- get_source_untime_treasury_rates, 80
- get_source_untime_wbstats, 80
- get_wbstats_registry, 81
- get_wbstats_registry_file_path, 81
- get_yahoofinance_registry, 82
- get_yahoofinance_registry_file_path, 82
- get_yahoofinance_seed_registry_path, 83
- infer_source_untime_from_frequency, 83
- is_sync_run_successful, 84
- list_source_specs, 84
- load_investdata_config, 85
- repair_local_binance_klines_gaps, 85
- repair_local_okx_candle_gaps, 86
- resolve_sec_cik, 87
- standardize_fund_holdings, 87
- sync_all_alphavantage_registry_data, 88
- sync_all_bea_registry_data, 89
- sync_all_bls_registry_data, 89
- sync_all_census_registry_data, 90
- sync_all_cftc_cot_registry_data, 91
- sync_all_crypto_derivatives_registry_data, 91
- sync_all_eia_registry_data, 92
- sync_all_fiscaldata_registry_data, 93
- sync_all_fred_registry_data, 93
- sync_all_ishare_registry_data, 94
- sync_all_ishare_registry_holdings, 95
- sync_all_rss_registry_data, 96
- sync_all_sdmx_registry_data, 96
- sync_all_sec_companyfacts_registry_data, 97
- sync_all_sec_frames_registry_data, 98
- sync_all_sec_submissions_registry_data, 98

`sync_all_treasury_rates`, 99
`sync_all_wbstats_registry_data`, 100
`sync_all_yahoofinance_registry_data`,
100
`sync_local_alphavantage_data`, 101
`sync_local_bea_data`, 102
`sync_local_binance_klines`, 103
`sync_local_bls_data`, 104
`sync_local_census_data`, 105
`sync_local_cftc_cot`, 106
`sync_local_crypto_derivatives`, 107
`sync_local_crypto_liquidations`, 108
`sync_local_data`, 108
`sync_local_data_batches`, 109
`sync_local_data_partitioned`, 110
`sync_local_eia_data`, 111
`sync_local_fiscaldata`, 112
`sync_local_fred_data`, 113
`sync_local_fund_holdings`, 113
`sync_local_ishare_data`, 114
`sync_local_ishare_holdings`, 115
`sync_local_okx_candle`, 116
`sync_local_quantmod_OHLC`, 117
`sync_local_rss_data`, 118
`sync_local_sdmx_data`, 118
`sync_local_sec_bulk_archive`, 120
`sync_local_sec_companyfacts`, 120
`sync_local_sec_filing_document`, 121
`sync_local_sec_frame`, 122
`sync_local_sec_submissions`, 123
`sync_local_treasury_rates`, 124
`sync_local_wbstats_data`, 124
`sync_sec_filing_documents`, 125

`validate_yahoofinance_registry`, 126