Package ‘prevederer’

July 23, 2019

Type Package

Title Wrapper for the 'Prevedere' API

Version 0.0.1

Description Easy and efficient access to the API provided by 'Prevedere', an industry insights and predictive analytics company. Query and download indicators, models and workbenches built with 'Prevedere' for further analysis and reporting <https://www.prevedere.com/>.

License MIT + file LICENSE

URL `https://github.com/wkdavis/prevederer`, `https://api.prevedere.com/index.html`, `https://www.prevedere.com/`

BugReports `https://github.com/wkdavis/prevederer/issues`

Depends R (>= 3.2.0)

Imports httr

Suggests testthat (>= 2.1.0), knitr, covr

Encoding UTF-8

LazyData yes

RoxygenNote 6.1.1

NeedsCompilation no

Author Wil Davis [aut, cre] (<https://orcid.org/0000-0001-9780-2576>), Hiram Foster [ctb], Ted Dickinson [ctb]

Maintainer Wil Davis `<davis.3243@osu.edu>`

Repository CRAN

Date/Publication 2019-07-23 09:40:02 UTC

R topics documented:

`prevederer-package` *Prevederer*

Description

prevederer provides an R wrapper around the Prevedere Software[1] API. The package facilitates access to the main API components[2], including:

Details

- Enumeration (lists of acceptable values)
- Indicator
- ForecastModel
- Workbench

Direct calls can also be made to the API using `prevedere_fetch()`.

Author(s)

Maintainer: Wil Davis <davis.3243@osu.edu> (0000-0001-9780-2576)

Other contributors:

- Hiram Foster <hiramfoster.co@gmail.com> [contributor]
- Ted Dickinson <tmdickinson@gmail.com> [contributor]

[1] https://www.prevedere.com/
[2] https://api.prevedere.com/index.html

See Also

Useful links:

- https://github.com/wkdavis/prevederer
- https://api.prevedere.com/index.html
- https://www.prevedere.com/
- Report bugs at https://github.com/wkdavis/prevederer/issues

---

`prevedere_aggregations` *Aggregation methods*

Description

Retrieve a list of the aggregation methods currently supported by the Prevedere API.

Usage

```
prevedere_aggregations(key)
```

Arguments

`key`	A Prevedere API key.

Value

A character vector of aggregation methods.

See Also

Other enumeration functions: `prevedere_calculations`, `prevedere_frequencies`, `prevedere_seasonalities`

Examples

```
## Not run:
key <- "1235467abcdefg"
prevedere_aggregations(key)

## End(Not run)
```

prevedere_calculations *Calculations*

Description

Retrieve a list of the calculations currently supported by the Prevedere API.

Usage

```
prevedere_calculations(key)
```

Arguments

key	A Prevedere API key.

Value

A character vector of calculations.

See Also

Other enumeration functions: `prevedere_aggregations`, `prevedere_frequencies`, `prevedere_seasonalities`

Examples

```
## Not run:
key <- "1235467abcdefg"
prevedere_calculations(key)

## End(Not run)
```

prevedere_correlation *Correlation*

Description

Calculates Pearson's r and other statistics at different offsets between an endogenous and exogenous indicator.

Usage

```
prevedere_correlation(key, endog_provider, endog_provider_id,
  exog_provider, exog_provider_id, freq = prevedere_frequencies(key),
  calculation = prevedere_calculations(key), raw = FALSE)
```

Arguments

key	A Prevedere API key.
endog_provider	Code for the data provider of the endogenous indicator, can be hexadecimal or abbreviated name.
endog_provider_id	Specific ProviderID for the endogenous indicator.
exog_provider	Code for the data provider of the exogenous indicator, can be hexadecimal or abbreviated name.
exog_provider_id	Specific ProviderID for the exogenous indicator.
freq	Frequency of indicator to retrieve. For a list of supported frequencies, see `prevedere_frequencies()`.
calculation	Calculation to transform the indicator. For a list of supported calculations, see `prevedere_calculations()`.
raw	Logical value indicating if data should be returned in its raw form (typically nested lists) or formatted as appropriate, usually a dataframe.

Value

Model results and metadata, as a list.

See Also

Other indicator functions: `prevedere_indicator`

Examples

```
## Not run:
k <- "1235467abcdefg"

prevedere_correlation(
  key = k, endog_provider = "BLS", endog_provider_id = "CES3133231058",
  exog_provider = "FRED", exog_provider_id = "PCU332313332313", freq = "Monthly",
  calculation = "ThreePeriodMoving"
)

## End(Not run)
```

`prevedere_fetch` *Query the Prevedere API*

Description

Send a GET request to the Prevedere API. Most users should calling this function directly and instead use the appropriate wrapper for accessing each part of the API.

Usage

```
prevedere_fetch(key, path, payload = NULL)
```

Arguments

`key`	A Prevedere API key.
`path`	The path (within the API) to which the request will be sent.
`payload`	The payload for the request. This should be a named list.

Value

The result of the API request.

See Also

GET, content, response

Examples

```
## Not run:
prevedere_fetch(
  key = "1235467abcdefg",
  path = "/indicator/BLS/CES3133231058",
  payload = list(
    Frequency = "Annual",
    Calculation = "None",
    Offset = 0
  )
)

k <- "1235467abcdefg"
prevedere_fetch(
  key = k,
  path = "/indicator/BLS/CES3133231058",
  payload = list(
    Frequency = "Annual",
    Calculation = "None",
    Offset = 0
  )
)
```

```
## End(Not run)
```

`prevedere_forecast` *Forecast*

Description

Returns historical fit and forecasted values of a forecast model.

Usage

```
prevedere_forecast(key, model_id, as_of_date = NULL, raw = FALSE)
```

Arguments

`key`	A Prevedere API key.
`model_id`	UUID for the forecast model.
`as_of_date`	Get the model only using data up to the specified date (YYYY-MM-DD). Used for backtesting.
`raw`	Logical value indicating if data should be returned in its raw form (typically nested lists) or formatted as appropriate, usually a dataframe.

Value

A dataframe of forecasted values and metadata.

See Also

Other forecast model functions: `prevedere_raw_model`

Examples

```
## Not run:
k <- "1235467abcdefg"

prevedere_forecast(key = k, model_id = "1b1878399833c7f38b094e54dd43d374")

## End(Not run)
```

prevedere_frequencies *Frequencies*

Description

Retrieve a list of the time frequencies currently supported by the Prevedere API.

Usage

```
prevedere_frequencies(key)
```

Arguments

key	A Prevedere API key.

Value

A character vector of frequencies.

See Also

Other enumeration functions: prevedere_aggregations, prevedere_calculations, prevedere_seasonalities

Examples

```
## Not run:
key <- "1235467abcdefg"
prevedere_frequencies(key)

## End(Not run)
```

prevedere_indicator *Indicators*

Description

Access indicator data and metadata.

Usage

```
prevedere_indicator(key, provider, provider_id)

prevedere_indicator_series(key, provider, provider_id,
  freq = prevedere_frequencies(key),
  calculation = prevedere_calculations(key), start_date = NULL,
  end_date = NULL, offset_periods = 0, raw = FALSE)
```

Arguments

key	A Prevedere API key.
provider	Code for a data provider, can be hexadecimal or abbreviated name.
provider_id	Specific ProviderID for the indicator.
freq	Frequency of indicator to retrieve. For a list of supported frequencies, see prevedere_frequencies().
calculation	Calculation to transform the indicator. For a list of supported calculations, see prevedere_calculations().
start_date, end_date	Start and end dates for the indicator. Each should be either a date or a character string capable of being coerced to a date. Setting a date to NULL will result in the historical data being unbounded in that direction.
offset_periods	Number of periods to offset.
raw	Logical value indicating if data should be returned in its raw form (typically nested lists) or formatted as appropriate, usually a dataframe.

Value

A list. prevedere_indicator returns metadata for the target indicator, while prevedere_indicator_series returns the actual data for the indicator (in addition to the metadata).

See Also

Other indicator functions: prevedere_correlation

Examples

```
## Not run:

k <- "1235467abcdefg"

## Return indicator metadata
prevedere_indicator(key = k, provider = "BLS", provider_id = "CES3133231058")

## Return indicator data
prevedere_indicator_series(
  key = k, provider = "BLS", provider_id = "CES3133231058",
  freq = "Monthly", calculation = "None", start_date = "2010-01-01",
  offset_periods = 0
)

## Return indicator data unformatted
prevedere_indicator_series(
  key = k, provider = "BLS", provider_id = "CES3133231058",
  freq = "Monthly", calculation = "None", start_date = "2010-01-01",
  offset_periods = 0, raw = TRUE
)
```

```
## End(Not run)
```

`prevedere_providers` *Providers*

Description

Retrieve a list of entities whose data is available in Prevedere

Usage

```
prevedere_providers(key, raw = FALSE)
```

Arguments

`key`	A Prevedere API key.
`raw`	Logical value indicating if data should be returned in its raw form (typically nested lists) or formatted as appropriate, usually a dataframe.

Value

A dataframe (or list if `raw` = `TRUE`) of provider metadata.

Examples

```
## Not run:
k <- "1235467abcdefg"

prevedere_providers(key = k)

## End(Not run)
```

`prevedere_raw_model` *Raw model*

Description

Returns all information about a forecast model.

Usage

```
prevedere_raw_model(key, model_id, exclude_indicators = TRUE,
  as_of_date = NULL, raw = FALSE)
```

Arguments

key	A Prevedere API key.
model_id	UUID for the forecast model.
exclude_indicators	Whether to return only indicators used in model (TRUE), or all associated indicators.
as_of_date	Get the model only using data up to the specified date (YYYY-MM-DD). Used for backtesting.
raw	Logical value indicating if data should be returned in its raw form (typically nested lists) or formatted as appropriate, usually a dataframe.

Value

A list of model components and metadata, including indicators, coefficients, and the model start date.

See Also

Other forecast model functions: prevedere_forecast

Examples

```
## Not run:
k <- "1235467abcdefg"

prevedere_raw_model(key = k, model_id = "1b1878399833c7f38b094e54dd43d374")

## Backtest
prevedere_raw_model(key = k,
                    model_id = "1b1878399833c7f38b094e54dd43d374",
                    as_of_data = "2019-05-01")

## End(Not run)
```

---

prevedere_seasonalities *Seasonalities*

Description

Retrieve a list of the seasonalities currently supported by the Prevedere API.

Usage

```
prevedere_seasonalities(key)
```

Arguments

key	A Prevedere API key.

Value

A character vector of seasonalities.

See Also

Other enumeration functions: prevedere_aggregations, prevedere_calculations, prevedere_frequencies

Examples

```
## Not run:
key <- "1235467abcdefg"
prevedere_seasonalities(key)

## End(Not run)
```

---

prevedere_workbench *Workbench*

Description

Returns the indicators used in a workbench. Typically includes workbench metadata and associated indicator metadata.

Usage

```
prevedere_workbench(key, workbench_id)
```

Arguments

key	A Prevedere API key.
workbench_id	UUID for workbench.

Value

A list of workbench metadata.

Examples

```
## Not run:
k <- "1235467abcdefg"

prevedere_workbench(key = k, workbench_id = "b8da829f7a1d4509ca5125e4699d6f0e")

## End(Not run)
```