Package ‘edgarWebR’

October 13, 2022

Title SEC Filings Access

Description A set of methods to access and parse live filing information from the U.S. Securities and Exchange Commission (SEC - <https://www.sec.gov/>) including company and fund filings along with all associated metadata.

Version 1.1.0

Maintainer Micah J Waldstein <micah@waldste.in>

Date 2021-04-18

Depends R (>= 3.4.0)

Imports xml2 (>= 1.3.2), methods, httr, jsonlite

Suggests covr, ggplot2, knitr, lintr, purrr, rmarkdown, htptest, tokenizers, devtools, dplyr, tidyr, roxygen2, pkgdown

VignetteBuilder knitr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1


BugReports https://github.com/mwaldstein/edgarWebR/issues

NeedsCompilation no

Author Micah J Waldstein [aut, cre]

Repository CRAN

Date/Publication 2021-04-24 12:10:06 UTC

R topics documented:

cik_search ................................................................. 2
company_details ...................................................... 3
company_filings ....................................................... 4
cik_search

Description

Provides access to the SEC CIK search tool from here

Usage

cik_search(company)

Arguments

- **company**
  
  Search term to search for CIK

Value

A dataframe with one row per company with Includes the following columns -

- cik
- company_href
- company_name
company_details

Examples

```
try(cik_search("cloudera"))
```

Description

For a given company, either by ticker, CIK, or pre-fetched page, we extract 2 sets of information:

**Company Information**  Filing date, accepted date, etc.

**Filings**  Companies included in the filing

Usage

```
company_details(
  x,  
  ownership = FALSE,  
  type = "",  
  before = "",  
  count = 40,  
  page = 1
)
```

Arguments

- **x**  
  either a stock ticker, CIK number, or XML document for a company page

- **ownership**  
  boolean for inclusion of company change filings

- **type**  
  Type of filing to fetch. NOTE: due to the way the SEC EDGAR system works, it is actually is a 'starts-with' search, so for instance specifying 'type = "10-K" will return "10-K/A" and "10-K405" filings as well. To ensure you only get the type you want, best practice would be to filter the results.

- **before**  
  yyyyymmdd format of latest filing to fetch

- **count**  
  Number of filings to fetch per page. Valid options are 10, 20, 40, 80, or 100. Other values will result in the closest count.

- **page**  
  Which page of results to return.

Value

A list with the following components

- **information**  data.frame as returned by `company_information`

- **filings**  data.frame as returned by `company_filings`
Examples

try(company_details("AAPL", before = "20170810"))

company_filings SEC Company Filings

Description
SEC Company Filings

Usage
company_filings(
  x,
  ownership = FALSE,
  type = "",
  before = "",
  count = 40,
  page = 1
)

Arguments
  x either a stock ticker, CIK number, or XML document for a company page
  ownership boolean for inclusion of company change filings
  type Type of filing to fetch. NOTE: due to the way the SEC EDGAR system works, it
       is actually a 'starts-with' search, so for instance specifying 'type = "10-K"
       will return "10-K/A" and "10-K405" filings as well. To ensure you only get the
       type you want, best practice would be to filter the results.
  before yyyyymmdd format of latest filing to fetch
  count Number of filings to fetch per page. Valid options are 10, 20, 40, 80, or 100.
          Other values will result in the closest count.
  page Which page of results to return.

Value
A dataframe of company filings

Examples

try(company_filings("AAPL", before = "20170810"))
**company_href**  
*Company URL for a CIK*

---

**Description**

Given a CIK, provide a link to the company information page.

**Usage**

```r
company_href(cik, ownership = FALSE, atom = FALSE)
```

**Arguments**

- **cik**: Company code
- **ownership**: (default: FALSE) boolean for inclusion of company change filings
- **atom**: (default: FALSE) if the link should be to the atom XML feed

**Value**

A string with URL requested

**Examples**

```r
company_href("0000037912")
```

---

**company_information**  
*SEC Company Info*

---

**Description**

Fetches basic information on a given company from the SEC site

**Usage**

```r
company_information(x)
```

**Arguments**

- **x**: Either a stock symbol (for the 10,000 largest companies) or CIK code

**Value**

a dataframe with all SEC company information
Examples

```r
try(company_information("INTC"))
```

---

**Description**

Provides access to the SEC Company Name Search from here using a company’s formal name rather than its common name.

**Usage**

```r
company_search(
  x,
  match = "start",
  file_number = FALSE,
  state = "",
  country = "",
  sic = "",
  ownership = FALSE,
  type = "",
  count = 40,
  page = 1
)
```

**Arguments**

- `x` Name of company to search or file number
- `match` (default: 'start') Either 'start' or 'contains' for where in the company name to search
- `file_number` (default: FALSE) if set to TRUE, x is treated as a file number
- `state` (default: "") Limit to a specific state of registration using 2-letter state abbreviations. Special values:
  - X1 The United States
  - A0 Alberta, Canada
  - A1 British Columbia, Canada
  - A2 Manitoba, Canada
  - A3 New Brunswick, Canada
  - A4 Newfoundland, Canada
  - A5 Nova Scotia, Canada
  - A6 Ontario, Canada
company_search

A7 Prince Edward Island, Canada
A8 Quebec, Canada
A9 Saskatchewan, Canada
B0 Yukon, Canada

country

2-character country code. The mapping is non-obvious, so unfortunately the best way to find it is to examine the company search page.

sic

SIC Code

ownership

boolean for inclusion of company change filings

type

Limit to companies with a given filing type - e.g. 'N-PX'

count

Number of filings to fetch per page. Valid options are 10, 20, 40, 80, or 100. Other values will result in the closest count.

page

Which page of results to return.

Details

Note On 'Fast Search' – The SEC Company Search page also includes a ‘Fast Search’ function to "search" by CIK or Stock Ticker. This doesn’t actually search, but rather goes directly to the company details page if found. If you have a company’s CIK or Ticker, use the company_information, company_filings, or company_details functions.

Value

A dataframe of companies

- cik
- company_href
- name
- location
- location_href
- formerly
- sic
- sic_description
- sic_href

Examples

try(company_search("Intel"))
**current_events**

**SEC Current Events**

**Description**

Provides access to the SEC Current Events search tool from [here](#).

**Usage**

```r
current_events(day, form)
```

**Arguments**

- **day** (0-5) Day to search for current forms. e.g. '2' returns forms from 2 business days ago.
- **form** Form to return filings (e.g. '10-K')

**Value**

A dataframe with one row per company with the following columns -

- cik
- type
- href
- company_name
- company_href
- filing_date

**Examples**

```r
try(current_events(0, "10-K")[1:5,])
```

---

**effectiveness**

**SEC Notice of Effectiveness**

**Description**

Returns the current Notice of Effectiveness from the most recently completed business day from [here](#).

**Usage**

```r
effectiveness()
```
Details

You can also see the same filings going further back by using 'latest_filings()' specifying the type = "EFFECT"

Value

a data.frame with each row as a submission with the following columns:

registration_number
file_href
registrant
registrant_href
filing_date
effective_date
division
type

Examples

try(effectiveness())

filing_details | SEC Filing Details

Description

The SEC generates a html page as an index for every filing it receives containing all the meta-information about the filing. We extract 3 main types of information:

Filing Information  Filing date, accepted date, etc.
Documents          All the documents included in the filing
Filers              Companies included in the filing
Funds              Funds included in the filing

Usage

filing_details(x)

## S3 method for class 'character'
filing_details(x)

## S3 method for class 'xml_node'
filing_details(x)
Arguments

x URL to a SEC filing index page

Details

For a company, there is typically a single filer and no funds, but many filings for funds get more complicated - e.g. 400+ funds with 100's of companies

NOTE: This can get process intensive for large fund pages. If you don’t need all components, try just using filing_info

Value

A list with the following components:

- information A data.frame as returned by filing_information
- documents A data.frame as returned by filing_documents
- filers A data.frame as returned by filing_filers
- funds A data.frame as returned by filing_funds

Examples

# Typically you’d get the URL from one of the search functions
x <- paste0("https://www.sec.gov/Archives/edgar/data/",
           "712515/000071251517000063/0000712515-17-000063-index.htm")
try(filing_details(x))
Arguments

x URL or xml_document for a SEC filing index page

Details

Information returned:

- seq
- description
- document
- href
- type
- size

Value

A dataframe with all the documents in the filing along with their meta info

Examples

# Typically you'd get the URL from one of the search functions
x <- paste0("https://www.sec.gov/Archives/edgar/data/",
"712515/000071251517000063/0000712515-17-000063-index.htm")
try(filing_documents(x))

<table>
<thead>
<tr>
<th>filing_filers</th>
<th>SEC Filing Included Filers</th>
</tr>
</thead>
</table>

Description

SEC Filing Included Filers

Usage

filing_filers(x)

# S3 method for class 'character'
filing_filers(x)

# S3 method for class 'xml_node'
filing_filers(x)

Arguments

x URL to a SEC filing index page
Value

A dataframe with all the filers in the filing along with their info

Examples

```
# Typically you'd get the URL from one of the search functions
x <- paste0("https://www.sec.gov/Archives/edgar/data/",
             "712515/000071251517000063/0000712515-17-000063-index.htm")
try(filing_filers(x))
```

---

**filing_funds**

*SEC Filing Funds*

Description

SEC Filing Funds

Usage

```
filing_funds(x)
```

## S3 method for class 'character'
filing_funds(x)

## S3 method for class 'xml_node'
filing_funds(x)

Arguments

- `x` URL to a SEC filing index page

Value

A dataframe with all the funds associated with a given filing

Examples

```
# Typically you'd get the URL from one of the search functions
x <- paste0("https://www.sec.gov/Archives/edgar/data/",
             "933691/000119312517247698/0001193125-17-247698-index.htm")
try(filing_funds(x))
```
Description

The SEC generates a html page as an index for every filing it receives containing all the meta-information about the filing.

Usage

filing_information(x)

```r
## S3 method for class 'character'
filing_information(x)
```

```r
## S3 method for class 'xml_node'
filing_information(x)
```

Arguments

- `x` URL or xml_document for a SEC filing index page

Details

Information returned:

- type
- description
- accession_number
- filing_date
- accepted_date
- documents
- period_date
- changed_date
- effective_date
- filing_bytes

Not all details are valid for all filings, but the column will always be present.

If you know you’re going to want all the details of a filing, including documents funds andfilers, look at ‘filing_details’

Value

A dataframe with all the parsed meta-info on the filing
Examples

# Typically you'd get the URL from one of the search functions
x <- paste0("https://www.sec.gov/Archives/edgar/data/",
    "933691/000119312517247698/0001193125-17-247698-index.htm")
try(filing_information(x))

full_text  SEC Full-Text Search

Description

Provides access to the SEC filings full-text search tool.

Usage

full_text(
  q = "*",
  type = "",
  reverse_order = FALSE,
  count = 100,
  page = 1,
  stemming = TRUE,
  name = "",
  cik = "",
  sic = "",
  from = "",
  to = "",
  location = "",
  incorporated_location = FALSE
)

Arguments

q   Search query. For details on special formatting, see the FAQ.
type  Type of forms to search - e.g. '10-K'. Can also be a list of types - e.g. c("10-K", "10-Q")
reverse_order  [DEP] If true, order by oldest first instead of newest first
count  [DEP] Number of results to return - will always try to return 100
page  Which page of results to return
stemming  [DEP] Search by base words(default) or exactly as entered
name   Company name OR individual’s name. Cannot be combined with 'cik' or 'sic'.
cik    Company code to search. Cannot be combined with 'name' or 'sic'
**fund_search**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sic</td>
<td>[DEP] Standard Industrial Classification of filer to search for. Cannot be combined with 'cik' or 'name'. Special options - 1: all, 0: Unspecified.</td>
</tr>
<tr>
<td>from</td>
<td>Start date. Must be in the form of 'mm/dd/yyyy'. Must also specify 'to'</td>
</tr>
<tr>
<td>to</td>
<td>End date. Must be in the form of 'mm/dd/yyyy'. Must also specify 'from'</td>
</tr>
<tr>
<td>location</td>
<td>Filter based on company’s location</td>
</tr>
<tr>
<td>incorporated_location</td>
<td>boolean to use location of incorporation rather than location of HQ</td>
</tr>
</tbody>
</table>

**Value**

A dataframe list of results including the following columns -

- filing_date
- name
- href
- company_name
- cik
- sic
- content
- parent_href
- index_href

**Examples**

```python
try(full_text('intel'))
```

---

**Description**

Provides access to the results of the SEC’s Mutual fund search tool available [here](#).

**Usage**

```python
fund_search(term)
```

```python
fund_fast_search(identifier)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>term</td>
<td>Search term to search for in a fund name</td>
</tr>
<tr>
<td>identifier</td>
<td>A Series, Class/Contract ID, Ticker Symbol or CIK</td>
</tr>
</tbody>
</table>
Details

NOTE: This is really a specific version of the Variable Insurance search tool.

Value

A dataframe of funds found including the following columns -

- class_id
- class_filings_href
- class_name
- class_ticker
- series_id
- series_filings_href
- series_name
- series_funds_href
- cik
- cik_name
- cik_filings_href
- cik_funds_href

Functions

- fund_fast_search: Performs a ‘Fast Search’ based on a fund identifier

Examples

try(fund_search("precious metals"))
try(fund_fast_search("VMFVX"))

header_search SEC Header Search

Description

Searches filing headers going back to 1994 excluding the most recent day using the interface here

Usage

header_search(q, page = 1, from = 1994, to = 2017)
Arguments

q          The search string. Documentation here
page      Which results page to return (default: 1)
from      Start year (default: 1994)
to        End year (default: Current year)

Value

A dataframe of funds found including the following columns -

- company_name
- filing_href
- form
- filing_date
- size

Examples

try(header_search("company-name = Apple"))

latest_filings  SEC Latest Filings

Description

Provides access to the latest SEC filings from here

Usage

latest_filings(
    name = "", 
    cik = "", 
    type = "", 
    owner = "include", 
    count = 40, 
    page = 1 
)
Arguments

- **name**: Optional company name to limit filing results
- **cik**: Optional company cik to limit filing results
- **type**: Optional form type to limit filing results
- **owner**: How to include ownership filings. Options are
  - include (default)
  - exclude
  - only
- **count**: Number of results to return
- **page**: Which page of results to return

Value

- A dataframe list of recent results, ordered by descending accepted date. Includes the following columns -
  - type
  - href
  - company_name
  - company_type
  - cik
  - filing_date
  - accepted_date
  - accession_number
  - size

Examples

```r
try(latest_filings())
```

Description

Given a link to filing document (e.g. the 10-K, 8-K) in HTML, process the file into parts and items. This enables follow-up processing of a desired section - e.g. just the Risk Factors. 'item.name' and 'part.name' are taken directly from the document without any attempt to normalize.

Usage

```r
parse_filing(x, strip = TRUE, include.raw = FALSE, fix.errors = TRUE)
```
parse_submission

Arguments

- **x** - URL to a filing HTML document, html text or xml_document
- **strip** - Should non-text elements be removed? Default: true
- **include.raw** - Include unprocessed nodes in result? Default: false
- **fix.errors** - Try to fix document errors (e.g. missing part labels). WIP. Default: true

Details

**NOTE:** This has been tested on a range of documents, but formatting differences could cause failures. Please report an issue for any document that isn’t parsed correctly.

**FURTHER NOTE:** Not all filings are well formed - missing headings, bad spacing, etc. These can all throw the parsing off!

Value

a dataframe with one row per paragraph

- **part.name** Detected name of the Part
- **item.name** Detected name of the Item
- **text** Text of the paragraph / node
- **raw** Raw HTML of the node if include.raw = TRUE

Examples

```r
try(head(parse_filing(paste0("https://www.sec.gov/Archives/edgar/data/",
712515/000071251517000010/ea12312016-q3fy1710qdoc.htm")), 6))
```

---

**parse_submission** | **Parse Submission**

Description

Raw SEC filings are sent in a SGML file - this parses that master submission into component documents, with content lines in list column ‘TEXT’.

Usage

```r
parse_submission(x, include.binary = T, include.content = T)
```
### Arguments

- **x** - Input submission to parse. May be one of the following:
  - **URI** URL to a SEC complete submission text file
  - **Text** String with the full submission
  - **File path** Path to local file containing the submission

- **include.binary** - Default TRUE, determines if the content of binary documents is returned.
- **include.content** - Default TRUE, determines if the content of documents is returned.

### Details

Most of the time the information you need along with the specific files will be available by using `filing_documents`, but there are scenarios where you may want to access the full contents of the master submission:

- **Old Submissions** Older submissions are not parsed into component documents by the SEC so access requires parsing the main filing
- **Full Document List** The SEC only provides what it considers the relevant documents, but filings often include many more ancillary files
- **Efficient Downloading** If you’re fetching many documents from a filing over many filings, there can be efficiency gains from just downloading a single file.

*NOTE: non-text documents are uuencoded and need a separate decoder to be viewed.*

### Value

A dataframe with one row per document. For the metadata (TYPE, DESCRIPTION, FILENAME) it is important to note that these are provided by the filer and have little standardization or enforcement.

- **SEQUENCE** Sequence number of the file
- **TYPE** The type of document, e.g. 10-K, EX-99, GRAPHIC
- **DESCRIPTION** The type of document, e.g. 10-K, EX-99, GRAPHIC
- **FILENAME** The document’s filename
- **TEXT** The text representation of the document. For text-based documents (txt, html) this is the actual file contents. For binary files (graphics, pdfs) this contains the uuencoded contents.

### Examples

```r
try(
  parse_submission(paste0('https://www.sec.gov/Archives/edgar/data/37996/000003799617000084/0000037996-17-000084.txt'),
                  c('SEQUENCE', 'TYPE', 'DESCRIPTION', 'FILENAME'))
)
```
Description

Given a link to a filing document (e.g. the 10-K, 8-K) in TXT, process the file into parts and items. This enables follow-up processing of a desired section - e.g. just the Risk Factors. 'item.name' and 'part.name' are taken directly from the document without any attempt to normalize.

Usage

parse_text_filing(x, strip = TRUE, include.raw = FALSE, fix.errors = TRUE)

Arguments

- **x**: URL to a filing text document or actual text
- **strip**: Should non-text elements be removed? Default: true
- **include.raw**: Include unprocessed nodes in result? Default: false
- **fix.errors**: Try to fix document errors (e.g. missing part labels). WIP. Default: true

Details

**NOTE:** This has been tested on a range of documents, but formatting differences could cause failures. Please report an issue for any document that isn’t parsed correctly.

**FURTHER NOTE:** Not all filings are well formed - missing headings, bad spacing, etc. These can all throw the parsing off!

Value

- a dataframe with one row per paragraph
  - **part.name**: Detected name of the Part
  - **item.name**: Detected name of the Item
  - **text**: Text of the paragraph / node
  - **raw**: Raw HTML of the node if include.raw = TRUE

Examples

```
try(head(parse_text_filing(
  "https://www.sec.gov/Archives/edgar/data/37996/000003799602000015/v7.txt"
)))
```
**sic_codes**

*Description*

SIC code table with structure.

*Usage*

sic_codes

*Format*

A data frame with 1005 rows and 6 variables:

- **sic**: Standard Industrial Classification
- **industry**: Name of industry
- **division_id**: Letter code for the division
- **division**: Name of the division
- **major**: Name of the major group, identified by 1st 2 digits of the sic
- **group**: Name of the group, identified by the 1st 3 digits of the sic

*Source*

https://www.osha.gov/data/sic-manual
https://www.sec.gov/info/edgar/siccodes.htm

---

**submission_index_href**

*Submission URL Tools*

*Description*

EDGAR submissions are organized fairly regularly. These functions help to find the URL to submission components.

*Usage*

submission_index_href(cik, accession)

submission_href(cik, accession)

submission_file_href(cik, accession, filename)
variable_insurance_search

Arguments

cik  Company code
accession  accession number for a filing
filename  filename provided in a submission

Value
A string with URL requested

Functions

- submission_href: Creates a link to the master submission sgml submission file
- submission_file_href: provides the link to a given file within a particular submission.

Examples

```
submission_index_href("0000712515", "0000712515-17-000090")
submission_href("0000712515", "0000712515-17-000090")
submission_file_href("0000712515", "0000712515-17-000090", "pressrelease-ueberroth.htm")
```

---

variable_insurance_search  

SEC Variable Insurance Search

Description

Provides access to the results of the SEC’s Variable Insurance Product search tool available here

Usage

```
variable_insurance_search(term)
variable_insurance_fast_search(identifier)
```

Arguments

term  Search term to search for in a company, fund or contract name
identifier  A Series, Class/Contract ID, Ticker Symbol or CIK
Value

A dataframe of products found including the following columns -

- class_id
- class_filings_href
- class_name
- class_ticker
- series_id
- series_filings_href
- series_name
- series_funds_href
- cik
- cik_name
- cik_filings_href
- cik_funds_href

Functions

- `variable_insurance_fast_search`: Performs a 'Fast Search' based on an identifier

Examples

```r
try(variable_insurance_search("precious metals"))
try(variable_insurance_fast_search("VMFVX"))
```
Index

* datasets
  - sic_codes, 22

cik_search, 2
cik_company_details, 3, 7
cik_company_filings, 3, 4, 7
cik_company_href, 5
cik_company_information, 3, 5, 7
cik_company_search, 6
cik_current_events, 8
effectiveness, 8
filing_details, 9
filing_documents, 10, 10, 20
filing_fillers, 10, 11
filing_funds, 10, 12
filing_information, 10, 13
full_text, 14
fund_fast_search (fund_search), 15
fund_search, 15

header_search, 16

latest_filings, 17

parse_filing, 18
parse_submission, 19
parse_text_filing, 21

sic_codes, 22
submission_file_hhref
  - (submission_index_hhref), 22
submission_hhref
  - (submission_index_hhref), 22
submission_index_hhref, 22

variable_insurance_fast_search
  - (variable_insurance_search), 23
variable_insurance_search, 23