

# Package ‘edgarWebR’

May 25, 2018

**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://sec.gov>) including company and fund filings along with all associated metadata.

**Version** 1.0.0

**Maintainer** Micah J Waldstein <[micah@waldste.in](mailto:micah@waldste.in)>

**Date** 2018-05-24

**Depends** R (>= 3.4.0)

**Imports** xml2, methods, httr

**Suggests** covr, ggplot2, knitr, purrr, rmarkdown, httpptest, tokenizers, devtools, dplyr, tidyr, roxygen2

**VignetteBuilder** knitr

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**URL** <https://mwaldstein.github.io/edgarWebR>,  
<https://github.com/mwaldstein/edgarWebR>

**BugReports** <https://github.com/mwaldstein/edgarWebR/issues>

**NeedsCompilation** no

**Author** Micah J Waldstein [aut, cre]

**Repository** CRAN

**Date/Publication** 2018-05-25 03:45:57 UTC

## R topics documented:

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

company_href . . . . .	4
company_information . . . . .	5
company_search . . . . .	6
current_events . . . . .	7
effectiveness . . . . .	8
filing_details . . . . .	9
filing_documents . . . . .	10
filing_filers . . . . .	11
filing_funds . . . . .	11
filing_information . . . . .	12
full_text . . . . .	13
fund_search . . . . .	14
header_search . . . . .	15
latest_filings . . . . .	16
parse_filing . . . . .	17
parse_submission . . . . .	18
parse_text_filing . . . . .	19
sic_codes . . . . .	20
submission_index_href . . . . .	20
variable_insurance_search . . . . .	21

## **Index** **23**

---

cik_search	<i>SEC CIK Search</i>
------------	-----------------------

---

### **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

### **Examples**

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

---

company_details	<i>SEC Company Details</i>
-----------------	----------------------------

---

## 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	yyyymmdd 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

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

---

company_filings	<i>SEC Company Filings</i>
-----------------	----------------------------

---

**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 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	yyyymmdd 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**

```
company_filings("AAPL", before = "20170810")
```

---

company_href	<i>Company URL for a CIK</i>
--------------	------------------------------

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

A string with URL requested

**Examples**

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

---

`company_information`    *SEC Company Info*

---

**Description**

Fetches basic information on a given company from the SEC site

**Usage**

```
company_information(x)
```

**Arguments**

<code>x</code>	Either a stock symbol (for the 10,000 largest companies) or CIK code
----------------	--

**Value**

a dataframe with all SEC company information

**Examples**

```
company_information("INTC")
```

---

company_search	<i>SEC Company Search</i>
----------------	---------------------------

---

### Description

Provides access to the SEC Company Name Search from [here](#) using a company's formal name rather than its common name.

### Usage

```
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: <b>X1</b> The United States <b>A0</b> Alberta, Canada <b>A1</b> British Columbia, Canada <b>A2</b> Manitoba, Canada <b>A3</b> New Brunswick, Canada <b>A4</b> Newfoundland, Canada <b>A5</b> Nova Scotia, Canada <b>A6</b> Ontario, Canada <b>A7</b> Prince Edward Island, Canada <b>A8</b> Quebec, Canada <b>A9</b> Saskatchewan, Canada <b>B0</b> 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.

**Value**

A dataframe of companies

- cik
- company\_href
- name
- location
- location\_href
- formerly
- sic
- sic\_description
- sic\_href

**Examples**

```
company_search("Intel")
```

---

current\_events

*SEC Current Events*

---

**Description**

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

**Usage**

```
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 Includes the following columns -

- cik
- type
- href
- company\_name
- company\_href
- filing\_date

## Examples

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

---

effectiveness	<i>SEC Notice of Effectiveness</i>
---------------	------------------------------------

---

## Description

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

## Usage

```
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

```
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")
filing_details(x)
```

---

filing_documents	<i>SEC Filing Documents</i>
------------------	-----------------------------

---

**Description**

If you know you're going to want all the details of a filing, including documents funds and filers, look at 'filing\_details'

**Usage**

```
filing_documents(x)

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

## S3 method for class 'xml_node'
filing_documents(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")
filing_documents(x)
```

---

filing_filers	<i>SEC Filing Included Filers</i>
---------------	-----------------------------------

---

**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")
filing_filers(x)
```

---

filing_funds	<i>SEC Filing Funds</i>
--------------	-------------------------

---

**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")
filing_funds(x)
```

---

filing_information	SEC Filing Information
--------------------	------------------------

---

**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)

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

## 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 and filers, 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")
filing_information(x)
```

---

full_text	<i>SEC Full-Text Search</i>
-----------	-----------------------------

---

### Description

Provides access to the SEC filings [full-text search tool](#).

### Usage

```
full_text(q = "*", type = "", reverse_order = FALSE, count = 10,
          page = 1, stemming = TRUE, name = "", cik = "", sic = "",
          from = "", to = "")
```

### Arguments

q	Search query. For details on special formatting, see the <a href="#">FAQ</a> .
type	Type of forms to search - e.g. '10-K'
reverse_order	If true, order by oldest first instead of newest first
count	Number of results to return
page	Which page of results to return
stemming	Search by base words(default) or exactly as entered
name	Company name. Cannot be combined with 'cik' or 'sic'.
cik	Company code to search. Cannot be combined with 'name' or 'sic'
sic	Standard Industrial Classification of filer to search for. Cannot be combined with 'cik' or 'name'. Special options - 1: all, 0: Unspecified.
from	Start date. Must be in the form of 'mm/dd/yyyy'. Must also specify 'to'
to	End date. Must be in the form of 'mm/dd/yyyy'. Must also specify 'from'

**Value**

A dataframe list of results including the following columns -

- filing\_date
- name
- href
- company\_name
- cik
- sic
- content
- parent\_href
- index\_href

**Examples**

```
## This can be very slow running  
full_text('intel')
```

---

fund\_search

*SEC Mutual Fund Search*

---

**Description**

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

**Usage**

```
fund_search(term)
```

```
fund_fast_search(identifier)
```

**Arguments**

term                    Search term to search for in a fund name

identifier            A Series, Class/Contract ID, Ticker Symbol or CIK

**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**

```
fund_search("precious metals")
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 <https://www.sec.gov/cgi-bin/srch-edgar>.

**Usage**

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

**Arguments**

q	The search string. Documentation <a href="https://www.sec.gov/edgar/searchedgar/search_help.htm">https://www.sec.gov/edgar/searchedgar/search_help.htm</a>
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**

```
## This can be very slow running
header_search("company-name = Apple")
```

---

latest_filings	<i>SEC Latest Filings</i>
----------------	---------------------------

---

**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 <ul style="list-style-type: none"> <li>• include (default)</li> <li>• exclude</li> <li>• only</li> </ul>
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**

```
latest_filings()
```

---

parse\_filing

*Parse Filing*

---

**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**

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

**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**

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

---

parse_submission	<i>Parse Submission</i>
------------------	-------------------------

---

**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**

```
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.

**Examples**

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

---

parse\_text\_filing      *Parse Text Filing*

---

**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**

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

---

 sic\_codes

*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/pls/imis/sic\\_manual.html](https://www.osha.gov/pls/imis/sic_manual.html)

<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)

```

**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**

```
variable_insurance_search("precious metals")  
variable_insurance_fast_search("VMFVX")
```

# Index

## \*Topic **datasets**

[sic\\_codes](#), [20](#)

[cik\\_search](#), [2](#)

[company\\_details](#), [3](#)

[company\\_filings](#), [3](#), [4](#)

[company\\_href](#), [4](#)

[company\\_information](#), [3](#), [5](#)

[company\\_search](#), [6](#)

[current\\_events](#), [7](#)

[effectiveness](#), [8](#)

[filing\\_details](#), [9](#)

[filing\\_documents](#), [9](#), [10](#), [18](#)

[filing\\_filers](#), [9](#), [11](#)

[filing\\_funds](#), [9](#), [11](#)

[filing\\_information](#), [9](#), [12](#)

[full\\_text](#), [13](#)

[fund\\_fast\\_search](#) ([fund\\_search](#)), [14](#)

[fund\\_search](#), [14](#)

[header\\_search](#), [15](#)

[latest\\_filings](#), [16](#)

[parse\\_filing](#), [17](#)

[parse\\_submission](#), [18](#)

[parse\\_text\\_filing](#), [19](#)

[sic\\_codes](#), [20](#)

[submission\\_file\\_href](#)  
([submission\\_index\\_href](#)), [20](#)

[submission\\_href](#)  
([submission\\_index\\_href](#)), [20](#)

[submission\\_index\\_href](#), [20](#)

[variable\\_insurance\\_fast\\_search](#)  
([variable\\_insurance\\_search](#)), [21](#)

[variable\\_insurance\\_search](#), [21](#)