

# Package ‘tidyedgar’

May 8, 2026

**Title** Tidy Fundamental Financial Data from 'SEC's 'EDGAR' 'API'

**Version** 1.0.1

## Description

Streamline the process of accessing fundamental financial data from the United States Securities and Exchange Commission's ('SEC') Electronic Data Gathering, Analysis, and Retrieval system ('EDGAR') 'API' <<https://www.sec.gov/edgar/sec-api-documentation>>, transforming it into a tidy, analysis-ready format.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** dplyr, jsonlite, parallel, tidyr, httr

**URL** <https://gerardgimenezadsuar.github.io/tidyedgar/>

**NeedsCompilation** no

**Author** Gerard Gimenez-Adsuar [aut, cre]

**Maintainer** Gerard Gimenez-Adsuar <gerard@solucionsdedades.cat>

**Repository** CRAN

**Date/Publication** 2024-02-09 15:20:02 UTC

## Contents

get_qdata . . . . .	2
get_ydata . . . . .	2
prepare_data . . . . .	3
retrieve_data . . . . .	4
safe_max . . . . .	5
yearly_data . . . . .	5
<b>Index</b>	<b>6</b>

---

`get_qdata`*Getting quarterly data from all public companies from EDGAR*

---

**Description**

Getting quarterly data from all public companies from EDGAR

**Usage**

```
get_qdata(  
  account = "Revenues",  
  years = 2020:2023,  
  quarters = c("Q3"),  
  max_cores = TRUE  
)
```

**Arguments**

<code>account</code>	A string representing the account (eg NetIncomeLoss, Revenues, OperatingIncomeLoss, ...)
<code>years</code>	A sequence of numeric values representing the years.
<code>quarters</code>	A string representing the quarter.
<code>max_cores</code>	Boolean limiting the number of cores to 1.

**Value**

A dataframe

**Examples**

```
get_qdata(account = "NetIncomeLoss", years = 2022:2023, quarters = c("Q4"))
```

---

`get_ydata`*Getting yearly data from all public companies from EDGAR*

---

**Description**

Getting yearly data from all public companies from EDGAR

**Usage**

```
get_ydata(account = "Revenues", years = 2020:2023)
```

**Arguments**

account	A string representing the account (eg NetIncomeLoss, Revenues, OperatingIncomeLoss, ...)
years	A sequence of numeric values representing the years.

**Value**

A dataframe

**Examples**

```
get_ydata(account = "NetIncomeLoss", years = 2022:2023)
```

---

prepare_data	<i>Data wrangling for tidy fundamental data from EDGAR</i>
--------------	--

---

**Description**

Data wrangling for tidy fundamental data from EDGAR

**Usage**

```
prepare_data(df = NULL, quarterly = TRUE, ...)
```

**Arguments**

df	A dataframe, output from get_qdata() or get_ydata().
quarterly	Boolean indicating if quarterly data is present.
...	Additional dataframes to be combined from other accounts (NetIncomeLoss, OperatingIncomeLoss, etc).

**Value**

A dataframe

**Examples**

```
revenue <- data.frame(
  taxonomy = rep("us-gaap", 3),
  tag = rep("Revenues", 3),
  ccp = rep("CY2020", 3),
  uom = rep("USD", 3),
  label = rep("Revenues", 3),
  description = rep("Amount of revenue recognized from goods sold, services rendered, ...", 3),
  pts = rep(2762, 3),
  data.accn = c("0001564590-22-012597", "0000002178-23-000038", "0001654954-22-005679"),
  data.cik = c(2098, 2178, 2186),
  data.entityName = c("ACME CORP", "ADAMS RESOURCES, INC.", "BK TECHNOLOGIES"),
```

```

data.loc = c("US-CT", "US-TX", "US-FL"),
data.start = rep("2020-01-01", 3),
data.end = rep("2020-12-31", 3),
data.val = c(164003040, 1022422000, 44139000),
year = rep(2020, 3)
netincome <- data.frame(
  taxonomy = rep("us-gaap", 3),
  tag = rep("NetIncomeLoss", 3),
  ccp = rep("CY2020", 3),
  uom = rep("USD", 3),
  label = rep("NetIncomeLoss", 3),
  description = rep("Net Income from operating activities", 3),
  pts = rep(2762, 3),
  data.accn = c("0001564590-22-012597", "0000002178-23-000038", "0001654954-22-005679"),
  data.cik = c(2098, 2178, 2186),
  data.entityName = c("ACME CORP", "ADAMS RESOURCES, INC.", "BK TECHNOLOGIES"),
  data.loc = c("US-CT", "US-TX", "US-FL"),
  data.start = rep("2020-01-01", 3),
  data.end = rep("2020-12-31", 3),
  data.val = c(100000, 200000, 4000000),
  year = rep(2020, 3))
prepare_data(revenue,netincome, quarterly = FALSE)

```

---

retrieve\_data

*Helper function for quarterly financial data retrieval*

---

## Description

Helper function for quarterly financial data retrieval

## Usage

```
retrieve_data(account, year, quarter)
```

## Arguments

account	A string representing the account.
year	A numeric value representing the year.
quarter	A string representing the quarter.

## Value

A dataframe

---

safe_max	<i>Safely calculating the max.</i>
----------	------------------------------------

---

**Description**

Safely calculating the max.

**Usage**

```
safe_max(x, na.rm = FALSE)
```

**Arguments**

x	A number.
na.rm	Boolean.

**Value**

A number.

---

yearly_data	<i>Getting a summary with the basic financials for all companies</i>
-------------	--

---

**Description**

Getting a summary with the basic financials for all companies

**Usage**

```
yearly_data(years = 2020:2023)
```

**Arguments**

years	A sequence of numeric values representing the years.
-------	--

**Value**

A dataframe

**Examples**

```
yearly_data(years = 2022:2023)
```

# Index

`get_qdata`, 2

`get_ydata`, 2

`prepare_data`, 3

`retrieve_data`, 4

`safe_max`, 5

`yearly_data`, 5