SEC CIK Mapper

Release 2.1.0

Jad Chaar

Sep 24, 2023

CONTENTS

1 Features **3**

2 Quick Start **5**
- 2.1 Installation . . . 5
- 2.2 Basic Usage . . . 5

3 Supported Mappings **9**

4 Pre-generated Mappings **11**
- 4.1 Example Usage . . . 11
- 4.2 Contributing . . . 13
- 4.3 Documentation . . . 13

5 API Guide **15**
- 5.1 StockMapper . . . 15
- 5.2 MutualFundMapper . . . 18

Python Module Index **25**

Index **27**

sec-cik-mapper is a Python package for generating mappings between stock and mutual fund identifier data provided by the SEC.

CHAPTER

ONE

FEATURES

- Generate up-to-date mappings from the SEC as native Python dictionaries
- Mappings for both stocks and mutual funds
- Mapping data exposed as a raw pandas dataframe for custom data processing and usage
- Full support for PEP 484-style type hints and the mypy type checker
- Pre-generated mappings, updated daily, available from GitHub and jsDelivr for use outside of Python
- Support for Python 3.6+

CHAPTER

TWO

QUICK START

2.1 Installation

Install and update this package using pip:

```
$ pip install -U sec-cik-mapper
```

2.2 Basic Usage

2.2.1 Stocks

```
>>> from sec_cik_mapper import StockMapper
>>> from pathlib import Path

# Initialize a stock mapper instance
>>> mapper = StockMapper()

# Get mapping from CIK to tickers
>>> mapper.cik_to_tickers
{'0000320193': {'AAPL'}, '0000789019': {'MSFT'}, '0001652044': {'GOOG', 'GOOGL'}, ...}

# Get mapping from ticker to CIK
>>> mapper.ticker_to_cik
{'AAPL': '0000320193', 'MSFT': '0000789019', 'GOOG': '0001652044', ...}

# Get mapping from CIK to company name
>>> mapper.cik_to_company_name
{'0000320193': 'Apple Inc.', '0000789019': 'Microsoft Corp', '0001652044': 'Alphabet Inc.
↪', ...}

# Get mapping from ticker to company name
>>> mapper.ticker_to_company_name
{'AAPL': 'Apple Inc.', 'MSFT': 'Microsoft Corp', 'GOOG': 'Alphabet Inc.', ...}

# Get mapping from ticker to exchange
>>> mapper.ticker_to_exchange
{'AAPL': 'Nasdaq', 'MSFT': 'Nasdaq', 'GOOG': 'Nasdaq', ...}
```

(continues on next page)

(continued from previous page)

```
# Get mapping from exchange to tickers
>>> mapper.exchange_to_tickers
{'Nasdaq': {'CYRN', 'OHPAW', 'SANW', ...}, 'NYSE': {'PLAG', 'TDW-WTB', 'RS', ...}, 'OTC
↪': {'ZICX', 'LTGJ', 'AVNI', ...}, ...}

# Get mapping from CIK to exchange
>>> mapper.cik_to_exchange
{'0000320193': 'Nasdaq', '0000789019': 'Nasdaq', '0001652044': 'Nasdaq', ...}

# Get mapping from exchange to CIKs
>>> mapper.exchange_to_ciks
{'Nasdaq': {'0000779544', '0001508171', '0001060955', ...}, 'NYSE': {'0000764478',
↪'0000008818', '0001725057', ...}, 'OTC': {'0001044676', '0001592411', '0001284452', ...
↪}, ...}

# Save CIK, ticker, exchange, and company name mappings to a CSV file
>>> csv_path = Path("example_mappings.csv")
>>> mapper.save_metadata_to_csv(csv_path)

# Get raw pandas dataframe
>>> mapper.raw_dataframe
              CIK Ticker                                Name Exchange
0      0000001750    AIR                           Aar Corp     NYSE
1      0000001800    ABT                Abbott Laboratories     NYSE
2      0000001961   WDDD                         Worlds Inc      OTC
3      0000002098    ACU                  Acme United Corp     NYSE
4      0000002178     AE     Adams Resources & Energy, Inc.     NYSE
...           ...    ...                                ...      ...
13167  0001894057   PPYA  Papaya Growth Opportunity Corp. I
13168  0001894370   HMAC   Hainan Manaslu Acquisition Corp.
13169  0001894630   GHIX            Gores Holdings Ix, Inc.
13170  0001897941   GTEN             Gores Holdings X, Inc.
13171  0001898416   ALVO       Alvotech Lux Holdings S.A.S.

[13172 rows x 4 columns]
```

2.2.2 Mutual Funds

```
>>> from sec_cik_mapper import MutualFundMapper
>>> from pathlib import Path

# Initialize a mutual fund mapper instance
>>> mapper = MutualFundMapper()

# Get mapping from CIK to tickers
>>> mapper.cik_to_tickers
{'0000002110': {'CRBYX', 'CEFZX', 'CSSRX', ...}, '0000002646': {'IIBPX', 'IPISX', 'IIBTX
↪', ...}, '0000002663': {'IMSXX', 'VMTXX', 'IVMXX', ...}, ...}

# Get mapping from ticker to CIK
```

(continues on next page)

(continued from previous page)

```
>>> mapper.ticker_to_cik
{'LACAX': '0000002110', 'LIACX': '0000002110', 'ACRNX': '0000002110', ...}

# Get mapping from CIK to series ID
>>> mapper.cik_to_series_ids
{'0000002110': {'S000009184', 'S000033622', 'S000009185', ...}, '0000002646': {
↪'S000008760'}, '0000002663': {'S000008702'}, ...}

# Get mapping from ticker to series ID
>>> mapper.ticker_to_series_id
{'LACAX': 'S000009184', 'LIACX': 'S000009184', 'ACRNX': 'S000009184', ...}

# Get mapping from series ID to CIK
>>> mapper.series_id_to_cik
{'S000009184': '0000002110', 'S000009185': '0000002110', 'S000009186': '0000002110', ...}

# Get mapping from series ID to tickers
>>> mapper.series_id_to_tickers
{'S000009184': {'CEARX', 'CRBYX', 'ACRNX', ...}, 'S000009185': {'ACINX', 'CACRX', 'CAIRX
↪', ...}, 'S000009186': {'LAUCX', 'LAUAX', 'CUSAX', ...}, ...}

# Get mapping from series ID to class IDs
>>> mapper.series_id_to_class_ids
{'S000009184': {'C000024956', 'C000122737', 'C000024957', ...}, 'S000009185': {
↪'C000024958', 'C000122739', 'C000097733', ...}, 'S000009186': {'C000024962',
↪'C000024964', 'C000122740', ...}, ...}

# Get mapping from ticker to class ID
>>> mapper.ticker_to_class_id
{'LACAX': 'C000024954', 'LIACX': 'C000024956', 'ACRNX': 'C000024957', ...}

# Get mapping from CIK to class IDs
>>> mapper.cik_to_class_ids
{'0000002110': {'C000024958', 'C000024969', 'C000024957', ...}, '0000002646': {
↪'C000023849', 'C000074893', 'C000028785', ...}, '0000002663': {'C000023718',
↪'C000028786', 'C000076529', ...}, ...}

# Get mapping from class ID to CIK
>>> mapper.class_id_to_cik
{'C000024954': '0000002110', 'C000024956': '0000002110', 'C000024957': '0000002110', ...}

# Get mapping from class ID to ticker
>>> mapper.class_id_to_ticker
{'C000024954': 'LACAX', 'C000024956': 'LIACX', 'C000024957': 'ACRNX', ...}

# Save CIK, ticker, series ID, and class ID mappings to a CSV file
>>> csv_path = Path("mutual_fund_mappings.csv")
>>> mapper.save_metadata_to_csv(csv_path)

# Get raw pandas dataframe
>>> mapper.raw_dataframe
             CIK Ticker   Series ID    Class ID
```

(continues on next page)

(continued from previous page)

```
0      0000002110  ACFFX  S000009188  C000024973
1      0000002110  ACINX  S000009185  C000024961
2      0000002110  ACRNX  S000009184  C000024957
3      0000002110  ACTWX  S000009187  C000024969
4      0000002110  AUSAX  S000009186  C000024965
...           ...    ...         ...         ...
29236  0001860434   INNO  S000073580  C000230585
29237  0001860434   SIFI  S000072556  C000228889
29238  0001860434   SIHY  S000072555  C000228888
29239  0001877493    BTF  S000074058  C000231452
29240  0001877493    VBB  S000075054  C000233857

[29241 rows x 4 columns]
```

CHAPTER

THREE

SUPPORTED MAPPINGS

Mappings can be formed between the following SEC identifiers and metadata:

Key	Value	StockMapper	MutualFundMapper
CIK	Set(Tickers)		
CIK	Company Name		
CIK	Exchange		
Exchange	Set(CIKs)		
Exchange	Set(Tickers)		
Ticker	CIK		
Ticker	Company Name		
Ticker	Exchange		
CIK	Set(Series IDs)		
CIK	Set(Class IDs)		
Class ID	CIK		
Class ID	Ticker		
Series ID	CIK		
Series ID	Set(Class IDs)		
Series ID	Set(Tickers)		
Ticker	Class ID		
Ticker	Series ID		

CHAPTER

FOUR

PRE-GENERATED MAPPINGS

Pre-generated mappings are also available for download and use outside of Python (e.g. manually or via automated scripts/curl requests). These mappings are updated daily via a fully automated daily CRON job, which fetches, transforms, validates, and uploads the latest mapping data from the SEC to GitHub (save location: sec-cik-mapper/mappings · GitHub). These mappings are available for download and usage from both GitHub and the jsDelivr CDN.

4.1 Example Usage

Example curl commands, which download the specified mapping files and saves them to the current working directory:

4.1.1 Stocks

Hosted via GitHub:

```
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/stocks/
↪mappings.csv -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/stocks/
↪cik_to_exchange.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/stocks/
↪cik_to_tickers.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/stocks/
↪ticker_to_exchange.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/stocks/
↪cik_to_company_name.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/stocks/
↪ticker_to_cik.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/stocks/
↪ticker_to_company_name.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/stocks/
↪exchange_to_tickers.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/stocks/
↪exchange_to_ciks.json -O
```

Hosted via jsDelivr CDN:

```
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/stocks/mappings.
↪csv -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/stocks/cik_to_
↪exchange.json -O
```

(continues on next page)

(continued from previous page)

```
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/stocks/cik_to_
↪tickers.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/stocks/ticker_
↪to_exchange.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/stocks/cik_to_
↪company_name.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/stocks/ticker_
↪to_cik.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/stocks/ticker_
↪to_company_name.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/stocks/exchange_
↪to_tickers.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/stocks/exchange_
↪to_ciks.json -O
```

4.1.2 Mutual Funds

Hosted via GitHub:

```
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/ticker_to_class_id.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/series_id_to_class_ids.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/mappings.csv -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/cik_to_class_ids.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/cik_to_series_ids.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/series_id_to_cik.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/ticker_to_series_id.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/cik_to_tickers.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/class_id_to_cik.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/series_id_to_tickers.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/class_id_to_ticker.json -O
$ curl https://raw.githubusercontent.com/jadchaar/sec-cik-mapper/main/mappings/mutual_
↪funds/ticker_to_cik.json -O
```

Hosted via jsDelivr CDN:

```
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪ticker_to_class_id.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪series_id_to_class_ids.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪mappings.csv -O
```

(continues on next page)

(continued from previous page)

```
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪cik_to_class_ids.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪cik_to_series_ids.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪series_id_to_cik.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪ticker_to_series_id.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪cik_to_tickers.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪class_id_to_cik.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪series_id_to_tickers.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪class_id_to_ticker.json -O
$ curl https://cdn.jsdelivr.net/gh/jadchaar/sec-cik-mapper@main/mappings/mutual_funds/
↪ticker_to_cik.json -O
```

4.2 Contributing

If you encounter a bug or would like to see a new company filing or feature added to **sec-cik-mapper**, please file an issue or submit a pull request.

4.3 Documentation

For full documentation, please visit sec-cik-mapper.readthedocs.io.

CHAPTER

FIVE

API GUIDE

5.1 StockMapper

Provides a *StockMapper* class for mapping CIKs, tickers, exchanges, and company names.

class sec_cik_mapper.StockMapper.**StockMapper**

A *StockMapper* object.

Usage:

```
>>> from sec_cik_mapper import StockMapper
>>> stock_mapper = StockMapper()
```

save_metadata_to_csv(*path*)

Save stock mapping metadata (CIK, ticker, exchange, and company name) or mutual fund mapping metadata (CIK, ticker, series ID, class ID) from SEC to CSV.

Usage:

```
>>> from sec_cik_mapper import StockMapper, MutualFundMapper
>>> from pathlib import Path
>>> stock_mapper = StockMapper()
>>> mutual_fund_mapper = MutualFundMapper()
>>> csv_path = Path("cik_mapping.csv")
# Save full CIK, ticker, exchange, and company name mapping to a CSV file
>>> stock_mapper.save_metadata_to_csv(csv_path)
# Save full CIK, ticker, series ID, and class ID mapping to a CSV file
>>> mutual_fund_mapper.save_metadata_to_csv(csv_path)
```

Return type
None

property cik_to_company_name: Dict[str, str]

Get CIK to company name mapping.

Usage:

```
>>> from sec_cik_mapper import StockMapper
>>> stock_mapper = StockMapper()
>>> stock_mapper.cik_to_company_name
{'0000320193': 'Apple Inc.', '0000789019': 'Microsoft Corp', ...}
```

Return type
Dict[str, str]

property cik_to_exchange: Dict[str, str]

Get CIK to exchange mapping.

Usage:

```
>>> from sec_cik_mapper import StockMapper
>>> stock_mapper = StockMapper()
>>> stock_mapper.cik_to_exchange
{'0000320193': 'Nasdaq', '0000789019': 'Nasdaq', '0001652044': 'Nasdaq', ...}
```

Return type
Dict[str, str]

property cik_to_tickers: Dict[str, Set[str]]

Get CIK to tickers mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper, StockMapper
>>> stock_mapper = StockMapper()
>>> mutual_fund_mapper = MutualFundMapper()
>>> stock_mapper.cik_to_tickers
{'0000320193': {'AAPL'}, '0001652044': {'GOOG', 'GOOGL'}, ...}
>>> mutual_fund_mapper.cik_to_tickers
{'0000002110': {'CRBYX', 'CEFZX', ...}, '0000002646': {'IIBPX', 'IPISX', ...}, .
↪..}
```

Return type
Dict[str, Set[str]]

property exchange_to_ciks: Dict[str, Set[str]]

Get exchange to CIKs mapping.

Usage:

```
>>> from sec_cik_mapper import StockMapper
>>> stock_mapper = StockMapper()
>>> stock_mapper.exchange_to_ciks
{'Nasdaq': {'0000779544', ...}, 'NYSE': {'0000764478', ...}, ...}
```

Return type
Dict[str, Set[str]]

property exchange_to_tickers: Dict[str, Set[str]]

Get exchange to tickers mapping.

Usage:

```
>>> from sec_cik_mapper import StockMapper
>>> stock_mapper = StockMapper()
```

(continues on next page)

(continued from previous page)

```
>>> stock_mapper.exchange_to_tickers
{'Nasdaq': {'CYRN', 'OHPAW', ...}, 'NYSE': {'PLAG', 'TDW-WTB', ...}, ...}
```

Return type
Dict[str, Set[str]]

property raw_dataframe: DataFrame

Get raw pandas dataframe.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper, StockMapper
>>> stock_mapper = StockMapper()
>>> mutual_fund_mapper = MutualFundMapper()
>>> stock_mapper.raw_dataframe
           CIK  Ticker                                  Name Exchange
0      0000320193    AAPL                         Apple Inc.   Nasdaq
1      0000789019    MSFT                     Microsoft Corp   Nasdaq
2      0001652044    GOOG                      Alphabet Inc.   Nasdaq
3      0001018724    AMZN                     Amazon Com Inc   Nasdaq
4      0001318605    TSLA                        Tesla, Inc.   Nasdaq
...           ...     ...                                ...      ...
13184  0001866816   OLITU          Omnilit Acquisition Corp.   Nasdaq
13185  0001870778   OHAAU            Opy Acquisition Corp. I   Nasdaq
13186  0001873324   PEPLW  Pepperlime Health Acquisition Corp   Nasdaq
13187  0001877557  WEL-UN  Integrated Wellness Acquisition Corp     NYSE
13188  0001877787  ZGN-WT   Ermenegildo Zegna Holditalia S.P.A.     NYSE

[13189 rows x 4 columns]
>>> mutual_fund_mapper.raw_dataframe
               CIK Ticker   Series ID    Class ID
0      0000002110  LACAX  S000009184  C000024954
1      0000002110  LIACX  S000009184  C000024956
2      0000002110  ACRNX  S000009184  C000024957
3      0000002110  CEARX  S000009184  C000122735
4      0000002110  CRBRX  S000009184  C000122736
...           ...    ...         ...         ...
29237  0001860434   SIHY  S000072555  C000228888
29238  0001860434   SIFI  S000072556  C000228889
29239  0001860434   INNO  S000073580  C000230585
29240  0001877493    BTF  S000074058  C000231452
29241  0001877493    VBB  S000075054  C000233857

[29242 rows x 4 columns]
```

Return type
DataFrame

property ticker_to_cik: Dict[str, str]

Get ticker to CIK mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper, StockMapper
>>> stock_mapper = StockMapper()
>>> mutual_fund_mapper = MutualFundMapper()
>>> stock_mapper.ticker_to_cik
{'AAPL': '0000320193', 'MSFT': '0000789019', 'GOOG': '0001652044', ...}
>>> mutual_fund_mapper.ticker_to_cik
{'LACAX': '0000002110', 'LIACX': '0000002110', 'ACRNX': '0000002110', ...}
```

Return type
Dict[str, str]

property ticker_to_company_name: Dict[str, str]

Get ticker to company name mapping.

Usage:

```
>>> from sec_cik_mapper import StockMapper
>>> stock_mapper = StockMapper()
>>> stock_mapper.ticker_to_company_name
{'AAPL': 'Apple Inc.', 'MSFT': 'Microsoft Corp', 'GOOG': 'Alphabet Inc.', ...}
```

Return type
Dict[str, str]

property ticker_to_exchange: Dict[str, str]

Get ticker to exchange mapping.

Usage:

```
>>> from sec_cik_mapper import StockMapper
>>> stock_mapper = StockMapper()
>>> stock_mapper.ticker_to_exchange
{'AAPL': 'Nasdaq', 'MSFT': 'Nasdaq', 'GOOG': 'Nasdaq', ...}
```

Return type
Dict[str, str]

5.2 MutualFundMapper

Provides a *MutualFundMapper* class for mapping CIKs, tickers, series IDs, and class IDs.

class sec_cik_mapper.MutualFundMapper.**MutualFundMapper**

A *MutualFundMapper* object.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper
>>> mutual_fund_mapper = MutualFundMapper()
```

save_metadata_to_csv(*path*)

Save stock mapping metadata (CIK, ticker, exchange, and company name) or mutual fund mapping metadata (CIK, ticker, series ID, class ID) from SEC to CSV.

Usage:

```
>>> from sec_cik_mapper import StockMapper, MutualFundMapper
>>> from pathlib import Path
>>> stock_mapper = StockMapper()
>>> mutual_fund_mapper = MutualFundMapper()
>>> csv_path = Path("cik_mapping.csv")
# Save full CIK, ticker, exchange, and company name mapping to a CSV file
>>> stock_mapper.save_metadata_to_csv(csv_path)
# Save full CIK, ticker, series ID, and class ID mapping to a CSV file
>>> mutual_fund_mapper.save_metadata_to_csv(csv_path)
```

Return type
None

property cik_to_class_ids: Dict[str, Set[str]]

Get CIK to class IDs mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper
>>> mutual_fund_mapper = MutualFundMapper()
>>> mutual_fund_mapper.cik_to_class_ids
{'0000002110': {'C000024958', ...}, '0000002646': {'C000023849', ...}, ...}
```

Return type
Dict[str, Set[str]]

property cik_to_series_ids: Dict[str, Set[str]]

Get CIK to series ID mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper
>>> mutual_fund_mapper = MutualFundMapper()
>>> mutual_fund_mapper.cik_to_series_ids
{'0000002110': {'S000009184', 'S000033622', ...}, '0000002646': {'S000008760'},␣
↪...}
```

Return type
Dict[str, Set[str]]

property cik_to_tickers: Dict[str, Set[str]]

Get CIK to tickers mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper, StockMapper
>>> stock_mapper = StockMapper()
```

(continues on next page)

(continued from previous page)

```
>>> mutual_fund_mapper = MutualFundMapper()
>>> stock_mapper.cik_to_tickers
{'0000320193': {'AAPL'}, '0001652044': {'GOOG', 'GOOGL'}, ...}
>>> mutual_fund_mapper.cik_to_tickers
{'0000002110': {'CRBYX', 'CEFZX', ...}, '0000002646': {'IIBPX', 'IPISX', ...}, .
↪..}
```

Return type
Dict[str, Set[str]]

property class_id_to_cik: Dict[str, str]

Get class ID to CIK mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper
>>> mutual_fund_mapper = MutualFundMapper()
>>> mutual_fund_mapper.class_id_to_cik
{'C000024954': '0000002110', 'C000024956': '0000002110', ...}
```

Return type
Dict[str, str]

property class_id_to_ticker: Dict[str, str]

Get class ID to ticker mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper
>>> mutual_fund_mapper = MutualFundMapper()
>>> mutual_fund_mapper.class_id_to_ticker
{'C000024954': 'LACAX', 'C000024956': 'LIACX', 'C000024957': 'ACRNX', ...}
```

Return type
Dict[str, str]

property raw_dataframe: DataFrame

Get raw pandas dataframe.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper, StockMapper
>>> stock_mapper = StockMapper()
>>> mutual_fund_mapper = MutualFundMapper()
>>> stock_mapper.raw_dataframe
            CIK  Ticker                                Name Exchange
0      0000320193    AAPL                          Apple Inc.   Nasdaq
1      0000789019    MSFT                      Microsoft Corp   Nasdaq
2      0001652044    GOOG                       Alphabet Inc.   Nasdaq
3      0001018724    AMZN                      Amazon Com Inc   Nasdaq
4      0001318605    TSLA                         Tesla, Inc.   Nasdaq
...           ...     ...                                ...      ...
```

(continues on next page)

(continued from previous page)

```
13184  0001866816   OLITU             Omnilit Acquisition Corp.   Nasdaq
13185  0001870778   OHAAU               Opy Acquisition Corp. I   Nasdaq
13186  0001873324   PEPLW    Pepperlime Health Acquisition Corp   Nasdaq
13187  0001877557  WEL-UN  Integrated Wellness Acquisition Corp     NYSE
13188  0001877787  ZGN-WT   Ermenegildo Zegna Holditalia S.P.A.     NYSE

[13189 rows x 4 columns]
>>> mutual_fund_mapper.raw_dataframe
                CIK Ticker   Series ID    Class ID
0      0000002110  LACAX  S000009184  C000024954
1      0000002110  LIACX  S000009184  C000024956
2      0000002110  ACRNX  S000009184  C000024957
3      0000002110  CEARX  S000009184  C000122735
4      0000002110  CRBRX  S000009184  C000122736
...           ...    ...         ...         ...
29237  0001860434   SIHY  S000072555  C000228888
29238  0001860434   SIFI  S000072556  C000228889
29239  0001860434   INNO  S000073580  C000230585
29240  0001877493    BTF  S000074058  C000231452
29241  0001877493    VBB  S000075054  C000233857

[29242 rows x 4 columns]
```

Return type
DataFrame

property series_id_to_cik: Dict[str, str]

Get series ID to CIK mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper
>>> mutual_fund_mapper = MutualFundMapper()
>>> mutual_fund_mapper.series_id_to_cik
{'S000009184': '0000002110', 'S000009185': '0000002110', ...}
```

Return type
Dict[str, str]

property series_id_to_class_ids: Dict[str, Set[str]]

Get series ID to class IDs mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper
>>> mutual_fund_mapper = MutualFundMapper()
>>> mutual_fund_mapper.series_id_to_class_ids
{'S000009184': {'C000024956', ...}, 'S000009185': {'C000024958', ...}, ...}
```

Return type
Dict[str, Set[str]]

property series_id_to_tickers: Dict[str, Set[str]]

Get series ID to tickers mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper
>>> mutual_fund_mapper = MutualFundMapper()
>>> mutual_fund_mapper.series_id_to_tickers
{'S000009184': {'CEARX', 'CRBYX', ...}, 'S000009185': {'ACINX', 'CACRX', ...}, .
↪..}
```

Return type
Dict[str, Set[str]]

property ticker_to_cik: Dict[str, str]

Get ticker to CIK mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper, StockMapper
>>> stock_mapper = StockMapper()
>>> mutual_fund_mapper = MutualFundMapper()
>>> stock_mapper.ticker_to_cik
{'AAPL': '0000320193', 'MSFT': '0000789019', 'GOOG': '0001652044', ...}
>>> mutual_fund_mapper.ticker_to_cik
{'LACAX': '0000002110', 'LIACX': '0000002110', 'ACRNX': '0000002110', ...}
```

Return type
Dict[str, str]

property ticker_to_class_id: Dict[str, str]

Get ticker to class ID mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper
>>> mutual_fund_mapper = MutualFundMapper()
>>> mutual_fund_mapper.ticker_to_class_id
{'LACAX': 'C000024954', 'LIACX': 'C000024956', 'ACRNX': 'C000024957', ...}
```

Return type
Dict[str, str]

property ticker_to_series_id: Dict[str, str]

Get ticker to series ID mapping.

Usage:

```
>>> from sec_cik_mapper import MutualFundMapper
>>> mutual_fund_mapper = MutualFundMapper()
>>> mutual_fund_mapper.ticker_to_series_id
{'LACAX': 'S000009184', 'LIACX': 'S000009184', 'ACRNX': 'S000009184', ...}
```

Return type
Dict[str, str]

PYTHON MODULE INDEX

S

sec_cik_mapper.MutualFundMapper, 18
sec_cik_mapper.StockMapper, 15

INDEX

C

cik_to_class_ids (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 19
cik_to_company_name (*sec_cik_mapper.StockMapper.StockMapper property*), 15
cik_to_exchange (*sec_cik_mapper.StockMapper.StockMapper property*), 16
cik_to_series_ids (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 19
cik_to_tickers (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 19
cik_to_tickers (*sec_cik_mapper.StockMapper.StockMapper property*), 16
class_id_to_cik (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 20
class_id_to_ticker (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 20

E

exchange_to_ciks (*sec_cik_mapper.StockMapper.StockMapper property*), 16
exchange_to_tickers (*sec_cik_mapper.StockMapper.StockMapper property*), 16

M

module
    sec_cik_mapper.MutualFundMapper, 18
    sec_cik_mapper.StockMapper, 15
MutualFundMapper (*class in sec_cik_mapper.MutualFundMapper*), 18

R

raw_dataframe (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 20
raw_dataframe (*sec_cik_mapper.StockMapper.StockMapper property*), 17

S

save_metadata_to_csv() (*sec_cik_mapper.MutualFundMapper.MutualFundMapper method*), 18
save_metadata_to_csv() (*sec_cik_mapper.StockMapper.StockMapper method*), 15
sec_cik_mapper.MutualFundMapper module, 18
sec_cik_mapper.StockMapper module, 15
series_id_to_cik (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 21
series_id_to_class_ids (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 21
series_id_to_tickers (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 21
StockMapper (*class in sec_cik_mapper.StockMapper*), 15

T

ticker_to_cik (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 22
ticker_to_cik (*sec_cik_mapper.StockMapper.StockMapper property*), 17
ticker_to_class_id (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 22
ticker_to_company_name (*sec_cik_mapper.StockMapper.StockMapper property*), 18
ticker_to_exchange (*sec_cik_mapper.StockMapper.StockMapper property*), 18
ticker_to_series_id (*sec_cik_mapper.MutualFundMapper.MutualFundMapper property*), 22