```python
In [1]:   %load_ext jupyter_ai
```

```python
In [2]:   # 1. Import the necessary functions from edgartools
          from edgar import *
          import requests
          import pandas as pd
          from concurrent.futures import ThreadPoolExecutor, as_completed
          import time
          from tqdm.auto import tqdm
          from edgar import Company
          from collections import defaultdict

          import multiprocessing as mp
          import swifter
          import time
          from typing import List, Dict, Any, Tuple
          from datetime import datetime
          import re
          import pdb
```

```python
In [3]:   # Show more rows
          pd.set_option('display.max_rows', 10)  # Change 100 to your desired number

          # Show more columns
          pd.set_option('display.max_columns', 50)  # Change 50 to your desired number
```

```python
In [4]:   # 2. Tell the SEC who you are
          set_identity("yishaibz@gmail.com")
          headers = {'User-Agent': "yishaibz@gmail.com"}
```

base data

```python
In [5]:   industries = pd.read_csv(r'sic_industry_code.csv')
          industries
```

Out[5]:

	SIC Code	Office	Industry Title
0	100	Industrial Applications and Services	AGRICULTURAL PRODUCTION-CROPS
1	200	Industrial Applications and Services	AGRICULTURAL PROD-LIVESTOCK & ANIMAL SPECIALTIES
2	700	Industrial Applications and Services	AGRICULTURAL SERVICES
3	800	Industrial Applications and Services	FORESTRY
4	900	Industrial Applications and Services	FISHING, HUNTING AND TRAPPING
...	...	...	...
439	8880	Office of International Corp Fin	AMERICAN DEPOSITARY RECEIPTS
440	8888	Office of International Corp Fin	FOREIGN GOVERNMENTS
441	8900	Office of Trade & Services	SERVICES-SERVICES, NEC
442	9721	Office of International Corp Fin	INTERNATIONAL AFFAIRS
443	9995	Office of Real Estate & Construction	NON-OPERATING ESTABLISHMENTS

444 rows × 3 columns

In [6]:
```python
industries[industries['Industry Title'].str.contains("agri", case=False, na=False)]
```

Out[6]:

	SIC Code	Office	Industry Title
0	100	Industrial Applications and Services	AGRICULTURAL PRODUCTION-CROPS
1	200	Industrial Applications and Services	AGRICULTURAL PROD-LIVESTOCK & ANIMAL SPECIALTIES
2	700	Industrial Applications and Services	AGRICULTURAL SERVICES
95	2870	Industrial Applications and Services	AGRICULTURAL CHEMICALS

In [7]:
```python
companies_df = pd.read_csv(r'companies_info.csv')
companies_df
```

Out[7]:

	cik_str	ticker	title	industry
0	320193	AAPL	Apple Inc.	3571.0
1	789019	MSFT	MICROSOFT CORP	7372.0
2	1045810	NVDA	NVIDIA CORP	3674.0
3	1018724	AMZN	AMAZON COM INC	5961.0
4	1652044	GOOGL	Alphabet Inc.	7370.0
...	...	...	...	...
9703	1884046	SPKLU	Spark I Acquisition Corp	6770.0
9704	1884046	SPKLW	Spark I Acquisition Corp	6770.0
9705	2019804	HONDW	HCM II Acquisition Corp.	6770.0
9706	2019804	HONDU	HCM II Acquisition Corp.	6770.0
9707	797468	OXY-WT	OCCIDENTAL PETROLEUM CORP /DE/	1311.0

9708 rows × 4 columns

In [8]:
```python
ag_companies = companies_df[companies_df['industry']==2870]
ag_companies
```

Out[8]:

	cik_str	ticker	title	industry
515	1725964	NTR	Nutrien Ltd.	2870.0
768	1324404	CF	CF Industries Holdings, Inc.	2870.0
1076	941221	ICL	ICL Group Ltd.	2870.0
1097	1285785	MOS	MOSAIC CO	2870.0
1654	825542	SMG	SCOTTS MIRACLE-GRO CO	2870.0
...	...	...	...	...
6011	875729	BNET	BION ENVIRONMENTAL TECHNOLOGIES INC	2870.0
6030	1574565	EVGN	Evogene Ltd.	2870.0
6324	1789192	NITO	N2OFF, Inc.	2870.0
6517	1680378	SNES	SenesTech, Inc.	2870.0
6545	1410708	NUVI	Emo Capital Corp.	2870.0

17 rows × 4 columns

by industry

understand sec

sec has a few api endpoints:

not that cik's need to be 10 digits, with pre filled 0.

submissions, which gets the metadata of all submissions. usually browsed by cik:

```
filingMetadata = requests.get(
    f'https://data.sec.gov/submissions/CIK{cik}.json',
    headers=headers
    )
```

which returns:

```
dict_keys(['cik', 'entityType', 'sic', 'sicDescription',
'ownerOrg', 'insiderTransactionForOwnerExists',
'insiderTransactionForIssuerExists', 'name', 'tickers',
'exchanges', 'ein', 'description', 'website', 'investorWebsite',
'category', 'fiscalYearEnd', 'stateOfIncorporation',
'stateOfIncorporationDescription', 'addresses', 'phone', 'flags',
'formerNames', 'filings'])
```

the interesting dict is the fllings dict which contains a entry 'files' which is a dict of all accessions, and a 'recent' entry which is a summary dict of number of fillings.

that can be converted to a data frame with the following collumns:

''' files = pd.DataFrame.from_dict(filingMetadata.json()['filings']['recent']) Index(['accessionNumber', 'filingDate', 'reportDate', 'acceptanceDateTime', 'act', 'form', 'fileNumber', 'filmNumber', 'items', 'core_type', 'size', 'isXBRL', 'isInlineXBRL', 'primaryDocument', 'primaryDocDescription'], dtype='object') '''

the facts api:

```
companyFacts = requests.get(
    f'https://data.sec.gov/api/xbrl/companyfacts/CIK{cik}.json',
    headers=headers
    )
```

`companyFacts.json().keys() = dict_keys(['cik', 'entityName', 'facts'])`
where the all the data is in the facts:
`companyFacts.json()['facts'].keys() = dict_keys(['dei', 'us-gaap'])`
the **dei dict holds metadata**, exmaple: `companyFacts.json()['facts']`
`['dei'].keys() = dict_keys(['EntityCommonStockSharesOutstanding',`
`'EntityPublicFloat'])` but could have other fields. and each of these is a further dict:

```
companyFacts.json()['facts']['dei'][
    'EntityCommonStockSharesOutstanding'].keys()

dict_keys(['label', 'description', 'units'])
```

where **units** is the key dict: `companyFacts.json()['facts']['dei']`
`['EntityCommonStockSharesOutstanding']['units']['shares']` holds a list of dicts,
each holding the actual data for a period:

```
{'end': '2025-01-17',
 'val': 15022073000,
 'accn': '0000320193-25-000008',
 'fy': 2025,
 'fp': 'Q1',
 'form': '10-Q',
 'filed': '2025-01-31',
 'frame': 'CY2024Q4I'}
```

and the **gap dict holds specific reporting items**:

`companyFacts.json()['facts']['us-gaap'].keys()` is a massive dict of various line
items, eg: 'DeferredTaxAssetsLiabilitiesNet' and 'AvailableForSaleSecuritiesCurrent'.

eg, to get the income for operations: `len(companyFacts.json()['facts']['us-gaap']`
`['OperatingIncomeLoss']['units']['USD'])=213` .

note that the actual data is in the dict 'USD' in the dict 'units'

the helper function `def get_fact_names(company_facts,name):` can be used to
quickly get the list of fields with some key word

the **concept** endpoint is similar to the facts api:

```
companyConcept = requests.get(
    (
    f'https://data.sec.gov/api/xbrl/companyconcept/CIK{cik}'
     f'/us-gaap/Assets.json'
    ),
    headers=headers
    )
```

but it holds more meta data on the field: `dict_keys(['cik', 'taxonomy', 'tag',`
`'label', 'description', 'entityName', 'units'])`

but the actual values in the units:usd, are the same:
`len(companyConcept.json()['units']['USD']) = len(companyFacts.json()`
`['facts']['us-gaap']['Assets']['units']['USD']) = 134`

submissions

```python
In [ ]:   # create request header

          # get all companies data
          companyTickers = requests.get(
              "https://www.sec.gov/files/company_tickers.json",
              headers=headers
              )

          # review response / keys
          print(companyTickers.json().keys())

          # format response to dictionary and get first key/value
          firstEntry = companyTickers.json()['0']

          # parse CIK // without leading zeros
          directCik = companyTickers.json()['0']['cik_str']

          # dictionary to dataframe
          companyData = pd.DataFrame.from_dict(companyTickers.json(),
                                               orient='index')

          # add leading zeros to CIK to 10 digits
          companyData['cik_str'] = companyData['cik_str'].astype(
                                   str).str.zfill(10)
```

```python
In [53]:  cik = companyData.iloc[0,0]
          cik
```

```
Out[53]:  '0000320193'
```

```python
In [50]:  firstEntry
```

```
Out[50]:  {'cik_str': 320193, 'ticker': 'AAPL', 'title': 'Apple Inc.'}
```

```python
In [54]:  # get company specific filing metadata
          filingMetadata = requests.get(
              f'https://data.sec.gov/submissions/CIK{cik}.json',
              headers=headers
              )
```

```python
In [55]:  # review json
          print(filingMetadata.json().keys())
```

```
dict_keys(['cik', 'entityType', 'sic', 'sicDescription', 'ownerOrg', 'insiderTransac
tionForOwnerExists', 'insiderTransactionForIssuerExists', 'name', 'tickers', 'exchan
ges', 'ein', 'description', 'website', 'investorWebsite', 'category', 'fiscalYearEn
d', 'stateOfIncorporation', 'stateOfIncorporationDescription', 'addresses', 'phone',
'flags', 'formerNames', 'filings'])
```

```python
In [ ]:   filingMetadata.json()['filings']['files']
```

In [74]: `filingMetadata.json()['insiderTransactionForOwnerExists']`

Out[74]: 0

In [65]: `pd.DataFrame.from_dict(filingMetadata.json()['filings']['files'])`

Out[65]:

	name	filingCount	filingFrom	filingTo
0	CIK0000320193-submissions-001.json	1133	1994-01-26	2014-07-16

In [75]: `files = pd.DataFrame.from_dict(filingMetadata.json()['filings']['recent'])`
`files`

Out[75]:

	accessionNumber	filingDate	reportDate	acceptanceDateTime	act	form	fileNumber
0	0000320193-25-000036	2025-02-27	2025-02-25	2025-02-27T18:35:41.000Z		4	
1	0000320193-25-000035	2025-02-27	2025-02-25	2025-02-27T18:34:58.000Z		4	
2	0000320193-25-000034	2025-02-27	2025-02-25	2025-02-27T18:34:19.000Z		4	
3	0000320193-25-000033	2025-02-27	2025-02-25	2025-02-27T18:33:20.000Z		4	
4	0000320193-25-000032	2025-02-27	2025-02-25	2025-02-27T18:32:35.000Z		4	
...	...	...	...	...	...	...	..
995	0001193125-14-277160	2014-07-23	2014-06-28	2014-07-23T16:32:48.000Z	34	10-Q	000-10030
996	0001193125-14-275598	2014-07-22	2014-07-22	2014-07-22T16:31:08.000Z	34	8-K	000-10030
997	0001181431-14-026992	2014-07-17	2014-07-15	2014-07-17T18:32:50.000Z		4	
998	0001181431-14-026991	2014-07-17	2014-07-15	2014-07-17T18:32:08.000Z		3	
999	0001193125-14-271698	2014-07-17	2014-07-15	2014-07-17T16:39:12.000Z	34	8-K	000-10030

1000 rows × 15 columns

In [78]: `files.columns`

```
Out[78]: Index(['accessionNumber', 'filingDate', 'reportDate', 'acceptanceDateTime',
                'act', 'form', 'fileNumber', 'filmNumber', 'items', 'core_type', 'size',
                'isXBRL', 'isInlineXBRL', 'primaryDocument', 'primaryDocDescription'],
               dtype='object')
```

```
In [ ]: filingMetadata.json()['filings']['recent'].keys()
```

company facts

```
In [12]: # get company facts data
         cik = '0000857949'
         companyFacts = requests.get(
             f'https://data.sec.gov/api/xbrl/companyfacts/CIK{cik}.json',
             headers=headers
             )
```

```
In [102… def get_fact_names(company_facts,name):
          return[key for key in company_facts.keys() if re.search(name, key, re.IGNORECASE)]
```

```
In [38]: companyFacts.json()['entityName']
```

```
Out[38]: 'ENLIGHTIFY, INC.'
```

```
In [156… companyFacts.json()['facts']['dei'].keys()
```

```
Out[156… dict_keys(['EntityCommonStockSharesOutstanding', 'EntityPublicFloat'])
```

```
In [157… companyFacts.json()['facts'].keys()
```

```
Out[157… dict_keys(['dei', 'us-gaap'])
```

```
In [94]: companyFacts.json()['facts']['dei'][
             'EntityCommonStockSharesOutstanding']['units']['shares'][-1]
```

```
Out[94]: {'end': '2025-01-17',
          'val': 15022073000,
          'accn': '0000320193-25-000008',
          'fy': 2025,
          'fp': 'Q1',
          'form': '10-Q',
          'filed': '2025-01-31',
          'frame': 'CY2024Q4I'}
```

```
In [97]: comp_facts = companyFacts.json()['facts']['us-gaap']
```

```
In [159… get_fact_names(comp_facts,'')
```

```
Out[159… []
```

```
In [ ]: def get_fact_values(comp_facts,name,form = '10-K'):
            dict_list = comp_facts['us-gaap'][name]['units']['USD']
            dict_list = [{'value':item['val'],'year':item['fy'],'period':item['fp'] for ite
```

```
In [195…   cga_fil.filter(accession_number='0001144204-13-003152').current()
```

Out[195…

```
 ┌─────────────────────────────── Filings for Enlightify Inc. [857949] ───
 |
 |
 |    #   Form        Description                                          Filing Dat
e   Accession Number   |
 |   ───────────────────────────────────────────────────────────────────────────────
────────────────────   |
 |    0   10-K/A       Annual report for public companies                   2013-01-18
0001144204-13-003152   |
 |
 |
 └─────────────────────────────── Company filings between 2013-01-18 and 2013-01-1
8 ───────────────────────┘
```

example: `fact_list = companyFacts.json()['facts']['us-gaap']`
`['OperatingIncomeLoss']['units']['USD']` yields a list of items. and we see that there
are several items with the following fields:

```
    'start': '2009-07-01',
      'end': '2010-06-30',
      'val': 24926621,
```

but with different accessionds, since the same fact (income for the year 2010 = 24,926,621)
may appear in several reports.

		As of June 30,			
	2012	2011	2010	2009	2008
Revenue	$ 217,524,205	$ 179,717,966	$ 52,090,752	$ 35,207,997	$ 22,604,719
Cost of goods sold	138,248,972	116,097,931	21,138,552	14,712,066	9,792,856
Gross profit	79,275,233	63,620,035	30,952,200	20,495,931	12,811,863
Operating expenses	25,350,223	21,508,604	6,025,579	3,405,918	3,494,946
Income from operations	53,925,010	42,111,431	24,926,621	17,090,013	9,316,917
Non-operating income (expense)	(1,165,872)	(160,186)	157,653	(294,043)	(845,916)
Provision for income taxes	10,801,313	9,037,144	3,794,516	2,331,548	692,474
Net income	$ 41,957,825	$ 32,914,101	$ 21,289,758	$ 14,464,422	$ 7,778,527

eg,

https://www.sec.gov/Archives/edgar/data/857949/000114420412051214/v322271_10k.htm

but a single table may have several frames.

```
In [28]:   # different amounts of data available per concept
           fact_list = companyFacts.json()['facts']['us-gaap']['OperatingIncomeLoss']['units']
           fact_list[:2]
```

```
Out[28]:  [{'start': '2009-07-01',
            'end': '2010-06-30',
            'val': 24926621,
            'accn': '0001144204-12-052602',
            'fy': 2012,
            'fp': 'FY',
            'form': '10-K/A',
            'filed': '2012-09-24'},
           {'start': '2009-07-01',
            'end': '2010-06-30',
            'val': 24926621,
            'accn': '0001144204-13-003152',
            'fy': 2012,
            'fp': 'FY',
            'form': '10-K/A',
            'filed': '2013-01-18',
            'frame': 'CY2010'}]
```

In [21]:

```
In [24]:  form = '10-K'
          filtered = [{'value':item['val'],'start':item['start'],'end':item['end'],'year':ite
          filtered[:2]
```

```
Out[24]:  [{'value': 24926621,
            'start': '2009-07-01',
            'end': '2010-06-30',
            'year': 2012,
            'period': 'FY'},
           {'value': 24926621,
            'start': '2009-07-01',
            'end': '2010-06-30',
            'year': 2012,
            'period': 'FY'}]
```

```
In [26]:  data  = pd.DataFrame.from_dict(filtered)
```

```
In [121…  companyFacts.json()['facts']['us-gaap']['Assets'].keys()
```

```
Out[121…  dict_keys(['label', 'description', 'units'])
```

```
In [31]:  parse_facts_smart(company_facts)
```

Out[31]:

Concept	PeriodLabel	OperatingIncome
0	CY2010	24926621
1	CY2010Q3	9625155
2	CY2010Q4	7872756
3	CY2011	42111431
4	CY2011Q1	12124854
...	...	...
39	FY2019	18804748
40	FY2020	-134172358
41	FY2021	-113747463
42	FY2022	-83798104
43	FY2023	-13613360

44 rows × 2 columns

company parser class

doesn't work for ntr. need to understand what they reported and how

In [156...

```python
# mapping terms
#map each term to its irs or gap terms
#cash from operations
#'CashFlowsFromUsedInOperatingActivities'
TERMS = {
    'operating profits': {
        'us-gaap': 'OperatingIncomeLoss',
        'ifrs-full': 'ProfitLossFromOperatingActivities'
    },
    'gross profit': {
        'us-gaap': 'GrossProfit',
        'ifrs-full': 'GrossProfit'
    },
    'revenue': {
        'us-gaap': 'Revenues',
        'ifrs-full': 'Revenues'
    },
    'costs of goods': {
        'us-gaap': 'CostOfGoodsSold',
        'ifrs-full': 'CostOfSales'
    },
    'sg&a expenses': {
        'us-gaap': 'SellingGeneralAndAdministrativeExpense',
        'ifrs-full': 'SellingGeneralAndAdministrativeExpense'
    },
    'operating expenses': {
```

```python
        'us-gaap': 'OperatingExpenses',
        'ifrs-full': 'OperatingExpenses'
    },
    'interest expenses': {
        'us-gaap': 'InterestExpense',
        'ifrs-full': 'FinanceCosts'
    },
    'tax expenses': {
        'us-gaap': 'IncomeTaxExpenseBenefit',
        'ifrs-full': 'IncomeTaxExpense'
    }
}


SYSTEMS = {'us-gaap':'us-gaap','ifrs-full':'ifrs-full'}
```

```python
In [115…   import re
           from typing import Dict, Any, Optional, Tuple
           import pandas as pd
           from datetime import datetime

           def compute_actual_period(data: Dict[str, Any], period_type: str = "annual") -> Opt
               """
               Compute the actual reporting period for a fact.

               If a 'frame' exists, use it after stripping leading 'CY' and trailing 'I'.
               For annual data, regardless of what quarter the frame indicates, we force a uni
               annual label based on the year (e.g. "FY2010Q04"). For quarterly data, the fram
               normalized normally.

               If no frame is provided, derive the period from the 'start' and 'end' dates,
               but only if the period spans approximately 365±10 days (annual) or 90±5 days (q

               Returns a string in the format 'FY{year}Q{quarter:02d}', or None if the period
               """
               # Use the authoritative frame if available.
               if data.get('frame'):
                   frame = data.get('frame')
                   # Remove leading "CY" and trailing "I"
                   frame = frame.lstrip("CY").rstrip("I")
                   m = re.fullmatch(r"(\d{4})(?:Q(\d{1,2}))?", frame)
                   if m:
                       year = m.group(1)
                       quarter = m.group(2)
                       if period_type == "annual":
                           # Force an annual label regardless of any quarter in the frame.
                           return f"FY{year}"
                       else:
                           # For quarterly mode, use the provided quarter (pad if needed) or d
                           if quarter:
                               return f"FY{year}Q{int(quarter):02d}"
                           else:
                               return f"FY{year}Q05"
                   else:
                       return f"FY{frame}"
```

```python
        # If no frame, derive period from the start and end dates.
        try:
            start = pd.to_datetime(data.get('start'))
            end = pd.to_datetime(data.get('end'))
        except Exception:
            return None

        # Calculate the total number of days in the period (inclusive)
        days = (end - start).days + 1

        if period_type == "annual":
            # Accept only if the period is roughly 365 days ± 10 days.
            if not (365 - 10 <= days <= 365 + 10):
                return None  # Incomplete annual period; discard.
            # For annual, force the label to be based on the end date's year and defaul
            return f"FY{end.year}"
        elif period_type == "monthly":
            # Accept only if the period spans roughly 90 days ± 5 days.
            if not (90 - 5 <= days <= 90 + 5):
                return None  # Incomplete quarterly period; discard.
            # For quarterly data, derive quarter from the end date.
            year = end.year
            month = end.month
            if 1 <= month <= 3:
                quarter = 1
            elif 4 <= month <= 6:
                quarter = 2
            elif 7 <= month <= 9:
                quarter = 3
            elif 10 <= month <= 12:
                quarter = 4
            return f"FY{year}Q{quarter:02d}"

        return f"{data.get('start')}_{data.get('end')}"

def parse_normalized_period(period: str) -> Tuple[int, int]:
    """
    Parse a normalized period (e.g., 'FY2010Q03') into a tuple (year, quarter) for
    """
    m = re.fullmatch(r"FY(\d{4})Q(\d{2})", period)
    if not m:
        raise ValueError(f"Period {period} is not in expected normalized format")
    return int(m.group(1)), int(m.group(2))
```

```python
class FactRecord:
    def __init__(self, data: Dict[str, Any]):
        self.data = data
        self.start = data.get('start')
        self.end = data.get('end')
        self.value = data.get('val') / 1000000
        self.accn = data.get('accn')
        self.fy = data.get('fy')
        self.fp = data.get('fp')
        self.form = data.get('form')
        self.filed = pd.to_datetime(data.get('filed')) if data.get('filed') else No
        self.frame = data.get('frame')
```

```
                self.period: str = ""


        def set_period(self, period_type: str = "annual") -> None:
            self.period = compute_actual_period(self.data, period_type)



        def __repr__(self) -> str:
            return (f"FactRecord(period={self.period}, value={self.value}, "
                    f"start={self.start}, end={self.end}, form={self.form}, "
                    f"fy={self.fy}, fp={self.fp}, accn={self.accn}, filed={self.filed},
                    f"frame={self.frame})")
```

In [107…
```
def resolve_duplicates(records: list) -> Dict[str, FactRecord]:
    """
    Group FactRecord instances by their normalized period and resolve duplicates.
    Resolution strategy:
      - Group by the computed actual period.
      - If multiple records exist for the same period and their values differ,
        log a warning.
      - Within each group, if any record had a frame, choose the one with the lates
        otherwise, choose the record with the latest filing date.
    Returns a dictionary mapping normalized period to the chosen FactRecord.
    """
    grouped = {}
    for rec in records:
        # Only include records with a valid period.
        if rec.period is None:
            continue
        grouped.setdefault(rec.period, []).append(rec)

    resolved = {}
    for period_label, recs in grouped.items():
        values = {r.value for r in recs}
        if len(values) > 1:
            print(f"Warning: Different values found for period {period_label}: {val
        with_frame = [r for r in recs if r.frame]
        if with_frame:
            chosen = max(with_frame, key=lambda r: r.filed or datetime.min)
        else:
            chosen = max(recs, key=lambda r: r.filed or datetime.min)
        resolved[period_label] = chosen
    return resolved
```

In [147…
```
class CompanyParser:
    def __init__(self, company_facts: Dict[str, Any] = None, cik: str = None, perio
        """
        Initializes the parser.
        Either a company_facts dict (from the SEC API) or a cik must be provided.
        The period parameter should be 'annual' (for 10-K data) or 'monthly' (for 1
        """
        self.period = period.lower()
        if company_facts is not None:
            self.company_facts = company_facts
        elif cik is not None:
            cik = str(cik).zfill(10)
```

```python
            url = f"https://data.sec.gov/api/xbrl/companyfacts/CIK{cik}.json"
            headers = {'User-Agent': 'your-email@example.com'}
            response = requests.get(url, headers=headers)
            response.raise_for_status()
            self.company_facts = response.json()
        else:
            raise ValueError("Either a company facts dictionary or a CIK must be pr

    def get_fact(self, fact_name: str, period: str = None) -> pd.Series:
        """
        Extracts and deduplicates a fact from the company facts.
        Returns a pandas Series indexed by the normalized period (format: FYYYYYQQ)
        """
        period_type = period.lower() if period is not None else self.period
        try:
            system =  [key for key in self.company_facts["facts"].keys() if key in
            if fact_name in TERMS.keys():
                fact_name = TERMS[fact_name][system]
            #fact_list = self.company_facts["facts"]["us-gaap"][fact_name]["units"]
            fact_list = self.company_facts["facts"][system][fact_name]["units"]["US
        except KeyError:
            raise KeyError(f"Fact {fact_name} not found in the company facts.")

        records = []
        for item in fact_list:
            form = item.get("form", "")
            # For annual data, accept forms like "10-K" or "10-K/A"
            # For annual data, accept forms like "10-K", "10-K/A", "20-F", and "40-
            if period_type == "annual" and not re.match(r"(10-K|10-K/A|20-F|40-F)$"
                continue
            # For quarterly data (here using "monthly" mode), accept "10-Q" or "10-
            if period_type == "monthly" and not re.match(r"10-Q(/A)?$", form):
                continue
            rec = FactRecord(item)
            rec.set_period(period_type)
            if rec.period is not None:
                records.append(rec)

        if not records:
            raise ValueError(f"No valid records found for fact {fact_name} with per

        resolved = resolve_duplicates(records)
        fact_values = {period_label: rec.value for period_label, rec in resolved.it
        try:
            sorted_keys = sorted(fact_values.keys(), key=lambda x: parse_normalized
        except Exception:
            sorted_keys = list(fact_values.keys())

        series = pd.Series({k: fact_values[k] for k in sorted_keys}, name=fact_name
        return series
```

In [148...   `enlightify_parser = CompanyParser(facts, period='annual')`

In [133...   `facts  =enlightify_parser.company_facts`

```
In [159…   enlightify_parser.get_fact('sg&a expenses')
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[147], line 32, in CompanyParser.get_fact(self, fact_name, period)
     31     #fact_list = self.company_facts["facts"]["us-gaap"][fact_name]["units"]
["USD"]
---> 32     fact_list = self.company_facts["facts"][system][fact_name]["units"]["US
D"]
     33 except KeyError:

KeyError: 'SellingGeneralAndAdministrativeExpense'

During handling of the above exception, another exception occurred:

KeyError                                  Traceback (most recent call last)
Cell In[159], line 1
----> 1 enlightify_parser.get_fact('sg&a expenses')

Cell In[147], line 34, in CompanyParser.get_fact(self, fact_name, period)
     32     fact_list = self.company_facts["facts"][system][fact_name]["units"]["US
D"]
     33 except KeyError:
---> 34     raise KeyError(f"Fact {fact_name} not found in the company facts.")
     36 records = []
     37 for item in fact_list:

KeyError: 'Fact SellingGeneralAndAdministrativeExpense not found in the company fact
s.'
```

```
In [158…   TERMS.keys()
```

```
Out[158…   dict_keys(['operating profits', 'gross profit', 'revenue', 'costs of goods', 'sg&a
           expenses', 'operating expenses', 'interest expenses', 'tax expenses'])
```

```
In [137…   facts["facts"]["ifrs-full"]["CashFlowsFromUsedInOperatingActivities"]["units"]["USD
```

```
Out[137…    [{'start': '2017-01-01',
             'end': '2017-12-31',
             'val': 1225000000,
             'accn': '0001193125-19-053855',
             'fy': 2018,
             'fp': 'FY',
             'form': '40-F',
             'filed': '2019-02-27',
             'frame': 'CY2017'},
            {'start': '2018-01-01',
             'end': '2018-12-31',
             'val': 2052000000,
             'accn': '0001193125-19-053855',
             'fy': 2018,
             'fp': 'FY',
             'form': '40-F',
             'filed': '2019-02-27'},
            {'start': '2018-01-01',
             'end': '2018-12-31',
             'val': 2052000000,
             'accn': '0001193125-20-055383',
             'fy': 2019,
             'fp': 'FY',
             'form': '40-F',
             'filed': '2020-02-28',
             'frame': 'CY2018'},
            {'start': '2019-01-01',
             'end': '2019-12-31',
             'val': 3665000000,
             'accn': '0001193125-20-055383',
             'fy': 2019,
             'fp': 'FY',
             'form': '40-F',
             'filed': '2020-02-28'},
            {'start': '2019-01-01',
             'end': '2019-12-31',
             'val': 3665000000,
             'accn': '0001193125-21-059301',
             'fy': 2020,
             'fp': 'FY',
             'form': '40-F',
             'filed': '2021-02-26',
             'frame': 'CY2019'},
            {'start': '2020-01-01',
             'end': '2020-12-31',
             'val': 3323000000,
             'accn': '0001193125-21-059301',
             'fy': 2020,
             'fp': 'FY',
             'form': '40-F',
             'filed': '2021-02-26'},
            {'start': '2020-01-01',
             'end': '2020-12-31',
             'val': 3323000000,
             'accn': '0001193125-22-053904',
             'fy': 2021,
```

        'fp': 'FY',
        'form': '40-F',
        'filed': '2022-02-25',
        'frame': 'CY2020'},
       {'start': '2021-01-01',
        'end': '2021-12-31',
        'val': 3886000000,
        'accn': '0001193125-22-053904',
        'fy': 2021,
        'fp': 'FY',
        'form': '40-F',
        'filed': '2022-02-25'},
       {'start': '2021-01-01',
        'end': '2021-12-31',
        'val': 3886000000,
        'accn': '0001193125-23-048055',
        'fy': 2022,
        'fp': 'FY',
        'form': '40-F',
        'filed': '2023-02-24',
        'frame': 'CY2021'},
       {'start': '2022-01-01',
        'end': '2022-12-31',
        'val': 8110000000,
        'accn': '0001193125-23-048055',
        'fy': 2022,
        'fp': 'FY',
        'form': '40-F',
        'filed': '2023-02-24'},
       {'start': '2022-01-01',
        'end': '2022-12-31',
        'val': 8110000000,
        'accn': '0001193125-24-055096',
        'fy': 2023,
        'fp': 'FY',
        'form': '40-F',
        'filed': '2024-03-01',
        'frame': 'CY2022'},
       {'start': '2023-01-01',
        'end': '2023-12-31',
        'val': 5066000000,
        'accn': '0001193125-24-055096',
        'fy': 2023,
        'fp': 'FY',
        'form': '40-F',
        'filed': '2024-03-01'},
       {'start': '2023-01-01',
        'end': '2023-12-31',
        'val': 5066000000,
        'accn': '0001193125-25-041502',
        'fy': 2024,
        'fp': 'FY',
        'form': '40-F',
        'filed': '2025-02-28',
        'frame': 'CY2023'},
       {'start': '2024-01-01',

```
                'end': '2024-12-31',
                'val': 3535000000,
                'accn': '0001193125-25-041502',
                'fy': 2024,
                'fp': 'FY',
                'form': '40-F',
                'filed': '2025-02-28',
                'frame': 'CY2024'}]
```

In [132...

```python
def print_company_facts_summary(company_facts: Dict[str, Any]) -> None:
    """
    Iterates through the company facts dictionary and prints all available keys
    along with the number of reported items under each key.

    Parameters:
      - company_facts: Dictionary containing SEC filing facts.
    """
    facts = company_facts.get("facts", {})

    for taxonomy, taxonomy_data in facts.items():
        print(f"\n### Taxonomy: {taxonomy} ###")
        for key, key_data in taxonomy_data.items():
            units = key_data.get("units", {})
            total_items = sum(len(unit_data) for unit_data in units.values())
            print(f"  - {key}: {total_items} reported values across {len(units)} un
```

In [134...

```python
print_company_facts_summary(facts)
```

Taxonomy: dei
  - EntityCommonStockSharesOutstanding: 3 reported values across 1 units.

Taxonomy: ifrs-full
  - AccumulatedOtherComprehensiveIncome: 14 reported values across 1 units.
  - AcquisitionrelatedCostsRecognisedAsExpenseForTransactionRecognisedSeparatelyFrom
AcquisitionOfAssetsAndAssumptionOfLiabilitiesInBusinessCombination: 6 reported value
s across 1 units.
  - AcquisitionsThroughBusinessCombinationsOtherProvisions: 1 reported values across
1 units.
  - AcquisitionsThroughBusinessCombinationsPropertyPlantAndEquipment: 11 reported va
lues across 1 units.
  - ActuarialGainsLossesArisingFromChangesInDemographicAssumptionsNetDefinedBenefitL
iabilityAsset: 13 reported values across 1 units.
  - ActuarialGainsLossesArisingFromChangesInFinancialAssumptionsNetDefinedBenefitLia
bilityAsset: 13 reported values across 1 units.
  - AdditionalPaidinCapital: 14 reported values across 1 units.
  - AdditionsOtherThanThroughBusinessCombinationsPropertyPlantAndEquipment: 14 repor
ted values across 1 units.
  - AdditionsToRightofuseAssets: 11 reported values across 1 units.
  - AdjustedWeightedAverageShares: 14 reported values across 1 units.
  - AdjustmentsForCurrentTaxOfPriorPeriod: 14 reported values across 1 units.
  - AdjustmentsForDecreaseIncreaseInInventories: 10 reported values across 1 units.
  - AdjustmentsForDecreaseIncreaseInTradeAccountReceivable: 14 reported values acros
s 1 units.
  - AdjustmentsForDeferredTaxExpense: 14 reported values across 1 units.
  - AdjustmentsForDeferredTaxOfPriorPeriods: 14 reported values across 1 units.
  - AdjustmentsForDepreciationAndAmortisationExpense: 14 reported values across 1 un
its.
  - AdjustmentsForGainLossOnDisposalOfInvestmentsInSubsidiariesJointVenturesAndAssoc
iates: 3 reported values across 1 units.
  - AdjustmentsForImpairmentLossReversalOfImpairmentLossRecognisedInProfitOrLossProp
ertyPlantAndEquipment: 10 reported values across 1 units.
  - AdjustmentsForIncomeTaxExpense: 3 reported values across 1 units.
  - AdjustmentsForIncreaseDecreaseInOtherLiabilities: 14 reported values across 1 un
its.
  - AdjustmentsForIncreaseDecreaseInTradeAndOtherPayables: 14 reported values across
1 units.
  - AdjustmentsForSharebasedPayments: 14 reported values across 1 units.
  - AdjustmentsToReconcileProfitLossOtherThanChangesInWorkingCapital: 2 reported val
ues across 1 units.
  - AmountsReceivableRelatedPartyTransactions: 2 reported values across 1 units.
  - ApplicableTaxRate: 14 reported values across 1 units.
  - Assets: 14 reported values across 1 units.
  - BasicEarningsLossPerShare: 14 reported values across 1 units.
  - BasicEarningsLossPerShareFromContinuingOperations: 4 reported values across 1 un
its.
  - BasicEarningsLossPerShareFromDiscontinuedOperations: 4 reported values across 1
units.
  - BiologicalAssets: 4 reported values across 1 units.
  - BorrowingCostsCapitalised: 14 reported values across 1 units.
  - Borrowings: 6 reported values across 1 units.
  - CapitalCommitments: 7 reported values across 1 units.
  - CapitalisationRateOfBorrowingCostsEligibleForCapitalisation: 14 reported values
across 1 units.
  - Cash: 14 reported values across 1 units.

- CashAndCashEquivalents: 21 reported values across 1 units.
- CashAndCashEquivalentsAmountContributedToFairValueOfPlanAssets: 14 reported values across 1 units.
- CashAndCashEquivalentsInSubsidiaryOrBusinessesAcquiredOrDisposed2013: 3 reported values across 1 units.
- CashFlowsFromUsedInFinancingActivities: 14 reported values across 1 units.
- CashFlowsFromUsedInIncreaseDecreaseInCurrentBorrowings: 14 reported values across 1 units.
- CashFlowsFromUsedInInvestingActivities: 14 reported values across 1 units.
- CashFlowsFromUsedInInvestingActivitiesDiscontinuedOperations: 5 reported values across 1 units.
- CashFlowsFromUsedInOperatingActivities: 14 reported values across 1 units.
- CashFlowsFromUsedInOperationsBeforeChangesInWorkingCapital: 12 reported values across 1 units.
- CashFlowsUsedInObtainingControlOfSubsidiariesOrOtherBusinessesClassifiedAsInvestingActivities: 13 reported values across 1 units.
- CashOutflowForLeases: 12 reported values across 1 units.
- CommercialPapersIssued: 10 reported values across 1 units.
- ComprehensiveIncome: 14 reported values across 1 units.
- ContractualAmountsToBeExchangedInDerivativeFinancialInstrumentForWhichGrossCashFlowsAreExchanged: 1 reported values across 1 units.
- ContributionsToPlanByEmployerNetDefinedBenefitLiabilityAsset: 13 reported values across 1 units.
- ContributionsToPlanByPlanParticipantsNetDefinedBenefitLiabilityAsset: 12 reported values across 1 units.
- ContributionsToPlanNetDefinedBenefitLiabilityAsset: 14 reported values across 1 units.
- CostOfSales: 14 reported values across 1 units.
- CurrentAssets: 14 reported values across 1 units.
- CurrentAssetsOtherThanAssetsOrDisposalGroupsClassifiedAsHeldForSaleOrAsHeldForDistributionToOwners: 2 reported values across 1 units.
- CurrentDepositsFromCustomers: 13 reported values across 1 units.
- CurrentDerivativeFinancialLiabilities: 14 reported values across 1 units.
- CurrentDividendPayables: 14 reported values across 1 units.
- CurrentInventoriesHeldForSale: 13 reported values across 1 units.
- CurrentLeaseLiabilities: 12 reported values across 1 units.
- CurrentLiabilities: 14 reported values across 1 units.
- CurrentLiabilitiesOtherThanLiabilitiesIncludedInDisposalGroupsClassifiedAsHeldForSale: 2 reported values across 1 units.
- CurrentMaterialsAndSuppliesToBeConsumedInProductionProcessOrRenderingServices: 14 reported values across 1 units.
- CurrentPortionOfLongtermBorrowings: 13 reported values across 1 units.
- CurrentPrepaymentsAndOtherCurrentAssets: 14 reported values across 1 units.
- CurrentProvisionsForEmployeeBenefits: 14 reported values across 1 units.
- CurrentRecognisedLiabilitiesDefinedBenefitPlan: 14 reported values across 1 units.
- CurrentServiceCostNetDefinedBenefitLiabilityAsset: 13 reported values across 1 units.
- CurrentTaxAssetsCurrent: 14 reported values across 1 units.
- CurrentTaxAssetsNoncurrent: 14 reported values across 1 units.
- CurrentTaxExpenseIncome: 14 reported values across 1 units.
- CurrentTaxExpenseIncomeAndAdjustmentsForCurrentTaxOfPriorPeriods: 14 reported values across 1 units.
- CurrentTaxLiabilitiesCurrent: 14 reported values across 1 units.
- CurrentTradeReceivables: 14 reported values across 1 units.
- DebtInstrumentsAmountContributedToFairValueOfPlanAssets: 14 reported values acro

ss 1 units.
  - DeductibleTemporaryDifferencesForWhichNoDeferredTaxAssetIsRecognised: 7 reported
values across 1 units.
  - DeferredTaxAssets: 14 reported values across 1 units.
  - DeferredTaxExpenseIncomeRecognisedInProfitOrLoss: 14 reported values across 1 un
its.
  - DeferredTaxExpenseIncomeRelatingToOriginationAndReversalOfTemporaryDifferences:
14 reported values across 1 units.
  - DeferredTaxLiabilities: 14 reported values across 1 units.
  - DeferredTaxLiabilityAsset: 15 reported values across 1 units.
  - DefinedBenefitObligationAtPresentValue: 2 reported values across 1 units.
  - DepreciationAndAmortisationExpense: 12 reported values across 1 units.
  - DepreciationPropertyPlantAndEquipment: 14 reported values across 1 units.
  - DepreciationRightofuseAssets: 11 reported values across 1 units.
  - DerivativeFinancialAssets: 1 reported values across 1 units.
  - DilutedEarningsLossPerShare: 14 reported values across 1 units.
  - DilutedEarningsLossPerShareFromContinuingOperations: 4 reported values across 1
units.
  - DilutedEarningsLossPerShareFromDiscontinuedOperations: 4 reported values across
1 units.
  - DilutiveEffectOfShareOptionsOnNumberOfOrdinaryShares: 13 reported values across
1 units.
  - DisposalsPropertyPlantAndEquipment: 13 reported values across 1 units.
  - DividendsPaidClassifiedAsFinancingActivities: 14 reported values across 1 units.
  - DividendsPaidOrdinaryShares: 18 reported values across 1 units.
  - DividendsProposedOrDeclaredBeforeFinancialStatementsAuthorisedForIssueButNotReco
gnisedAsDistributionToOwnersPerShare: 1 reported values across 1 units.
  - EffectOfExchangeRateChangesOnCashAndCashEquivalents: 13 reported values across 1
units.
  - Equity: 21 reported values across 1 units.
  - EquityAndLiabilities: 14 reported values across 1 units.
  - ExercisePriceOfOutstandingShareOptions2019: 4 reported values across 1 units.
  - ExpenseByNature: 14 reported values across 1 units.
  - ExpenseFromSharebasedPaymentTransactionsWithEmployees: 14 reported values across
1 units.
  - FairValueOfInvestmentsInEquityInstrumentsDesignatedAsMeasuredAtFairValueThroughO
therComprehensiveIncome: 14 reported values across 1 units.
  - FinanceCosts: 14 reported values across 1 units.
  - FinishedGoods: 14 reported values across 1 units.
  - GainLossOnChangesInEffectOfLimitingNetDefinedBenefitAssetToAssetCeiling: 13 repo
rted values across 1 units.
  - GainLossOnRemeasurementOfNetDefinedBenefitLiabilityAsset: 14 reported values acr
oss 1 units.
  - GainsLossesOnDisposalsOfInvestments: 2 reported values across 1 units.
  - GainsLossesOnExchangeDifferencesOnTranslationNetOfTax: 13 reported values across
1 units.
  - GainsLossesOnExchangeDifferencesOnTranslationRecognisedInProfitOrLoss: 14 report
ed values across 1 units.
  - GainsLossesOnFinancialAssetsMeasuredAtFairValueThroughOtherComprehensiveIncomeNe
tOfTax: 14 reported values across 1 units.
  - GainsOnDisposalsOfInvestments: 7 reported values across 1 units.
  - GeneralAndAdministrativeExpense: 14 reported values across 1 units.
  - Goodwill: 14 reported values across 1 units.
  - GrossProfit: 14 reported values across 1 units.
  - ImpairmentLossRecognisedInProfitOrLossPropertyPlantAndEquipment: 10 reported val
ues across 1 units.

- IncomeTaxExpenseContinuingOperations: 14 reported values across 1 units.
- IncomeTaxesPaidClassifiedAsOperatingActivities: 14 reported values across 1 units.
- IncomeTaxRelatingToComponentsOfOtherComprehensiveIncome: 10 reported values across 1 units.
- IncreaseDecreaseInCashAndCashEquivalents: 14 reported values across 1 units.
- IncreaseDecreaseInNetDefinedBenefitLiabilityAsset: 14 reported values across 1 units.
- IncreaseDecreaseInWorkingCapital: 2 reported values across 1 units.
- IncreaseDecreaseThroughAcquisitionOfSubsidiary: 2 reported values across 1 units.
- IncreaseDecreaseThroughBusinessCombinationsAndDisposalsNetDefinedBenefitLiabilityAsset: 3 reported values across 1 units.
- IncreaseDecreaseThroughBusinessCombinationsDeferredTaxLiabilityAsset: 6 reported values across 1 units.
- IncreaseDecreaseThroughChangesInForeignExchangeRatesNetDefinedBenefitLiabilityAsset: 13 reported values across 1 units.
- IncreaseDecreaseThroughEffectOfChangesInForeignExchangeRatesLiabilitiesArisingFromFinancingActivities: 3 reported values across 1 units.
- IncreaseDecreaseThroughFinancingCashFlowsLiabilitiesArisingFromFinancingActivities: 14 reported values across 1 units.
- IncreaseDecreaseThroughNetExchangeDifferencesOtherProvisions: 7 reported values across 1 units.
- IncreaseDecreaseThroughNetExchangeDifferencesPropertyPlantAndEquipment: 1 reported values across 1 units.
- IncreaseDecreaseThroughObtainingOrLosingControlOfSubsidiariesOrOtherBusinessesLiabilitiesArisingFromFinancingActivities: 5 reported values across 1 units.
- IncreaseDecreaseThroughOtherChangesLiabilitiesArisingFromFinancingActivities: 10 reported values across 1 units.
- IncreaseDecreaseThroughOtherChangesPropertyPlantAndEquipment: 14 reported values across 1 units.
- IncreaseDecreaseThroughSharebasedPaymentTransactions: 14 reported values across 1 units.
- IncreaseDecreaseThroughTransfersPropertyPlantAndEquipment: 12 reported values across 1 units.
- IncreaseThroughNewLeasesLiabilitiesArisingFromFinancingActivities: 10 reported values across 1 units.
- IntangibleAssetsOtherThanGoodwill: 14 reported values across 1 units.
- InterestExpenseIncomeNetDefinedBenefitLiabilityAsset: 13 reported values across 1 units.
- InterestPaidClassifiedAsOperatingActivities: 14 reported values across 1 units.
- Inventories: 14 reported values across 1 units.
- InvestmentAccountedForUsingEquityMethod: 14 reported values across 1 units.
- InvestmentFundsAmountContributedToFairValueOfPlanAssets: 6 reported values across 1 units.
- InvestmentsInSubsidiariesJointVenturesAndAssociates: 12 reported values across 1 units.
- IssuedCapital: 14 reported values across 1 units.
- KeyManagementPersonnelCompensation: 14 reported values across 1 units.
- KeyManagementPersonnelCompensationPostemploymentBenefits: 14 reported values across 1 units.
- KeyManagementPersonnelCompensationSharebasedPayment: 14 reported values across 1 units.
- KeyManagementPersonnelCompensationShorttermEmployeeBenefits: 14 reported values across 1 units.
- KeyManagementPersonnelCompensationTerminationBenefits: 13 reported values across

1 units.
   - LeaseCommitmentsForShorttermLeasesForWhichRecognitionExemptionHasBeenUsed: 1 rep
orted values across 1 units.
   - LeaseLiabilities: 12 reported values across 1 units.
   - Liabilities: 14 reported values across 1 units.
   - LiabilitiesArisingFromFinancingActivities: 21 reported values across 1 units.
   - LiabilitiesIncludedInDisposalGroupsClassifiedAsHeldForSale: 1 reported values ac
ross 1 units.
   - LiabilityAssetOfDefinedBenefitPlans: 18 reported values across 1 units.
   - LongtermBorrowings: 14 reported values across 1 units.
   - LossesOnDisposalsOfNoncurrentAssets: 3 reported values across 1 units.
   - MaximumExposureToCreditRisk: 14 reported values across 1 units.
   - MinimumLeasePaymentsPayableUnderNoncancellableOperatingLease: 1 reported values
across 1 units.
   - MiscellaneousOtherOperatingExpense: 14 reported values across 1 units.
   - NetDebt: 10 reported values across 1 units.
   - NoncurrentAssetsOtherThanFinancialInstrumentsDeferredTaxAssetsPostemploymentBene
fitAssetsAndRightsArisingUnderInsuranceContracts: 14 reported values across 1 units.
   - NoncurrentLeaseLiabilities: 12 reported values across 1 units.
   - NoncurrentRecognisedAssetsDefinedBenefitPlan: 14 reported values across 1 units.
   - NoncurrentRecognisedLiabilitiesDefinedBenefitPlan: 14 reported values across 1 u
nits.
   - NumberOfOutstandingShareOptions: 15 reported values across 2 units.
   - NumberOfShareOptionsExercisableInSharebasedPaymentArrangement: 5 reported values
across 1 units.
   - NumberOfShareOptionsExercisedInSharebasedPaymentArrangement: 13 reported values
across 1 units.
   - NumberOfShareOptionsExpiredInSharebasedPaymentArrangement: 9 reported values acr
oss 1 units.
   - NumberOfShareOptionsForfeitedInSharebasedPaymentArrangement: 9 reported values a
cross 1 units.
   - NumberOfShareOptionsGrantedInSharebasedPaymentArrangement: 9 reported values acr
oss 1 units.
   - NumberOfSharesOutstanding: 15 reported values across 1 units.
   - OtherAssetsAmountContributedToFairValueOfPlanAssets: 14 reported values across 1
units.
   - OtherCashPaymentsToAcquireEquityOrDebtInstrumentsOfOtherEntitiesClassifiedAsInve
stingActivities: 5 reported values across 1 units.
   - OtherComponentsOfDeferredTaxExpenseIncome: 4 reported values across 1 units.
   - OtherComprehensiveIncome: 14 reported values across 1 units.
   - OtherComprehensiveIncomeNetOfTaxGainsLossesOnRemeasurementsOfDefinedBenefitPlan
s: 14 reported values across 1 units.
   - OtherCurrentPayables: 14 reported values across 1 units.
   - OtherCurrentReceivables: 14 reported values across 1 units.
   - OtherExpenseByNature: 2 reported values across 1 units.
   - OtherInflowsOutflowsOfCashClassifiedAsInvestingActivities: 13 reported values ac
ross 1 units.
   - OtherLongtermProvisions: 7 reported values across 1 units.
   - OtherNoncurrentAssets: 14 reported values across 1 units.
   - OtherNoncurrentFinancialLiabilities: 14 reported values across 1 units.
   - OtherOperatingIncomeExpense: 14 reported values across 1 units.
   - OtherProvisions: 14 reported values across 1 units.
   - OtherShorttermProvisions: 9 reported values across 1 units.
   - OtherTaxEffectsForReconciliationBetweenAccountingProfitAndTaxExpenseIncome: 14 r
eported values across 1 units.
   - PastServiceCostAndGainsLossesArisingFromSettlementsNetDefinedBenefitLiabilityAss

et: 13 reported values across 1 units.
  - PaymentsForDebtIssueCosts: 10 reported values across 1 units.
  - PaymentsFromPlanNetDefinedBenefitLiabilityAsset: 12 reported values across 1 units.
  - PaymentsOfLeaseLiabilitiesClassifiedAsFinancingActivities: 12 reported values across 1 units.
  - PaymentsToAcquireOrRedeemEntitysShares: 13 reported values across 1 units.
  - PlanAssetsAtFairValue: 14 reported values across 1 units.
  - PostemploymentBenefitExpenseDefinedContributionPlans: 18 reported values across 1 units.
  - ProceedsFromBorrowingsClassifiedAsFinancingActivities: 12 reported values across 1 units.
  - ProceedsFromDisposalOfNoncurrentAssetsOrDisposalGroupsClassifiedAsHeldForSaleAndDiscontinuedOperations: 2 reported values across 1 units.
  - ProceedsFromIssuingShares: 14 reported values across 1 units.
  - ProfitLoss: 14 reported values across 1 units.
  - ProfitLossBeforeTax: 14 reported values across 1 units.
  - ProfitLossFromContinuingOperations: 4 reported values across 1 units.
  - ProfitLossFromDiscontinuedOperations: 4 reported values across 1 units.
  - ProfitLossFromOperatingActivities: 14 reported values across 1 units.
  - PropertyPlantAndEquipment: 13 reported values across 1 units.
  - PurchaseOfIntangibleAssetsClassifiedAsInvestingActivities: 10 reported values across 1 units.
  - PurchaseOfPropertyPlantAndEquipmentClassifiedAsInvestingActivities: 12 reported values across 1 units.
  - PurchaseOfTreasuryShares: 14 reported values across 1 units.
  - RawMaterials: 14 reported values across 1 units.
  - RentalExpense: 2 reported values across 1 units.
  - RepaymentsOfNoncurrentBorrowings: 14 reported values across 1 units.
  - RetainedEarnings: 14 reported values across 1 units.
  - RevenueFromContractsWithCustomers: 14 reported values across 1 units.
  - RevenueFromInterest: 14 reported values across 1 units.
  - RightofuseAssets: 13 reported values across 1 units.
  - SellingExpense: 14 reported values across 1 units.
  - ShareOfProfitLossOfAssociatesAndJointVenturesAccountedForUsingEquityMethod: 12 reported values across 1 units.
  - ShorttermBorrowings: 14 reported values across 1 units.
  - ShorttermInvestmentsClassifiedAsCashEquivalents: 14 reported values across 1 units.
  - TaxEffectFromChangeInTaxRate: 13 reported values across 1 units.
  - TaxEffectOfExpenseNotDeductibleInDeterminingTaxableProfitTaxLoss: 10 reported values across 1 units.
  - TaxEffectOfForeignTaxRates: 14 reported values across 1 units.
  - TaxEffectOfRevenuesExemptFromTaxation2011: 10 reported values across 1 units.
  - TaxExpenseIncomeAtApplicableTaxRate: 14 reported values across 1 units.
  - TemporaryDifferencesAssociatedWithInvestmentsInSubsidiariesBranchesAndAssociatesAndInterestsInJointVentures: 14 reported values across 1 units.
  - TradeAndOtherCurrentPayables: 14 reported values across 1 units.
  - TradeAndOtherCurrentReceivables: 14 reported values across 1 units.
  - TradeAndOtherPayables: 7 reported values across 1 units.
  - TradeAndOtherPayablesToTradeSuppliers: 14 reported values across 1 units.
  - TradeAndOtherPayablesUndiscountedCashFlows: 1 reported values across 1 units.
  - TransfersOutOfLevel1IntoLevel2OfFairValueHierarchyAssets: 5 reported values across 1 units.
  - TransfersOutOfLevel1IntoLevel2OfFairValueHierarchyLiabilities: 5 reported values across 1 units.

 - TransportationExpense: 14 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsExercisableInSharebasedPaymentArrangem
ent: 1 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsExercisableInSharebasedPaymentArrangem
ent2019: 4 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsExercisedInSharebasedPaymentArrangemen
t: 1 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsExercisedInSharebasedPaymentArrangemen
t2019: 8 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsExpiredInSharebasedPaymentArrangement:
1 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsExpiredInSharebasedPaymentArrangement2
019: 8 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsForfeitedInSharebasedPaymentArrangemen
t: 1 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsForfeitedInSharebasedPaymentArrangemen
t2019: 8 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsGrantedInSharebasedPaymentArrangement:
1 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsGrantedInSharebasedPaymentArrangement2
019: 8 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsOutstandingInSharebasedPaymentArrangem
ent: 3 reported values across 1 units.
 - WeightedAverageExercisePriceOfShareOptionsOutstandingInSharebasedPaymentArrangem
ent2019: 12 reported values across 1 units.
 - WeightedAverageFairValueAtMeasurementDateShareOptionsGranted: 9 reported values
across 1 units.
 - WeightedAverageLesseesIncrementalBorrowingRateAppliedToLeaseLiabilitiesRecognise
dAtDateOfInitialApplicationOfIFRS16: 1 reported values across 1 units.
 - WeightedAverageRemainingContractualLifeOfOutstandingShareOptions: 1 reported val
ues across 1 units.
 - WeightedAverageSharePriceShareOptionsGranted: 1 reported values across 1 units.
 - WeightedAverageSharePriceShareOptionsGranted2019: 8 reported values across 1 uni
ts.
 - WeightedAverageShares: 14 reported values across 1 units.
 - WorkInProgress: 14 reported values across 1 units.
 - AcquisitiondateFairValueOfTotalConsiderationTransferred: 2 reported values acros
s 1 units.
 - AdditionalProvisionsOtherProvisions: 1 reported values across 1 units.
 - AmortisationIntangibleAssetsOtherThanGoodwill: 8 reported values across 1 units.
 - AmountByWhichUnitsRecoverableAmountExceedsItsCarryingAmount: 1 reported values a
cross 1 units.
 - CancellationOfTreasuryShares: 2 reported values across 1 units.
 - ComprehensiveIncomeAttributableToOwnersOfParent: 8 reported values across 1 unit
s.
 - DecreaseThroughTransferToLiabilitiesIncludedInDisposalGroupsClassifiedAsHeldForS
aleOtherProvisions: 4 reported values across 1 units.
 - EquityAttributableToOwnersOfParent: 8 reported values across 1 units.
 - IncreaseDecreaseThroughChangeInDiscountRateOtherProvisions: 4 reported values ac
ross 1 units.
 - IncreaseDecreaseThroughTimeValueOfMoneyAdjustmentOtherProvisions: 6 reported val
ues across 1 units.
 - NoncontrollingInterests: 8 reported values across 1 units.
 - OtherComprehensiveIncomeAttributableToNoncontrollingInterests: 8 reported values
across 1 units.
 - OtherInflowsOutflowsOfCashClassifiedAsFinancingActivities: 8 reported values acr

oss 1 units.
   - ProfitLossAttributableToNoncontrollingInterests: 8 reported values across 1 units.
   - ProfitLossAttributableToOwnersOfParent: 8 reported values across 1 units.
   - AdjustmentsForUndistributedProfitsOfAssociates: 2 reported values across 1 units.
   - BestEstimateAtAcquisitionDateOfContractualCashFlowsNotExpectedToBeCollectedForAcquiredReceivables: 1 reported values across 1 units.
   - DescriptionOfDiscountRatesUsedInPreviousEstimateOfValueInUse: 1 reported values across 1 units.
   - GrossContractualAmountsReceivableForAcquiredReceivables: 2 reported values across 1 units.
   - ImpairmentLossReversalOfImpairmentLossRecognisedInProfitOrLoss: 6 reported values across 1 units.
   - InterestExpense: 6 reported values across 1 units.
   - InterestPayable: 6 reported values across 1 units.
   - NoncurrentPortionOfOtherNoncurrentBorrowings: 4 reported values across 1 units.
   - OtherFinanceCost: 6 reported values across 1 units.
   - PercentageOfVotingEquityInterestsAcquired: 2 reported values across 1 units.
   - ReceivablesFromTaxesOtherThanIncomeTax: 2 reported values across 1 units.
   - ReversalOfImpairmentLossRecognisedInProfitOrLossPropertyPlantAndEquipment: 3 reported values across 1 units.
   - AdjustmentsForImpairmentLossReversalOfImpairmentLossRecognisedInProfitOrLoss: 4 reported values across 1 units.
   - BondsIssuedUndiscountedCashFlows: 2 reported values across 1 units.
   - CostOfInventoriesRecognisedAsExpenseDuringPeriod: 4 reported values across 1 units.
   - CurrentPayablesOnSocialSecurityAndTaxesOtherThanIncomeTax: 4 reported values across 1 units.
   - DividendsRecognisedAsDistributionsToOwnersPerShare: 4 reported values across 1 units.
   - GrossLeaseLiabilities: 2 reported values across 1 units.
   - ImpairmentLossRecognisedInProfitOrLoss: 2 reported values across 1 units.
   - ImpairmentLossRecognisedInProfitOrLossIntangibleAssetsAndGoodwill: 1 reported values across 1 units.
   - IncreaseDecreaseInNumberOfSharesOutstanding: 5 reported values across 1 units.
   - LiabilitiesFromSharebasedPaymentTransactions2011: 4 reported values across 1 units.
   - NoncurrentFinancialAssetsAtAmortisedCost: 4 reported values across 1 units.
   - NoncurrentFinancialAssetsAtFairValueThroughProfitOrLossMandatorilyMeasuredAtFairValue: 4 reported values across 1 units.
   - NumberOfPotentialOrdinarySharesThatAreAntidilutiveInPeriodPresented: 4 reported values across 1 units.
   - OtherAssets: 4 reported values across 1 units.
   - OtherCurrentBorrowingsAndCurrentPortionOfOtherNoncurrentBorrowings: 4 reported values across 1 units.
   - PropertyPlantAndEquipmentIncludingRightofuseAssets: 4 reported values across 1 units.
   - ReserveOfCashFlowHedges: 4 reported values across 1 units.
   - ReserveOfRemeasurementsOfDefinedBenefitPlans: 4 reported values across 1 units.
   - ReversalOfImpairmentLossRecognisedInProfitOrLoss: 2 reported values across 1 units.
   - TaxBenefitArisingFromPreviouslyUnrecognisedTaxLossTaxCreditOrTemporaryDifferenceOfPriorPeriodUsedToReduceDeferredTaxExpense: 4 reported values across 1 units.
   - UndrawnBorrowingFacilities: 2 reported values across 1 units.
   - UnsecuredBankLoansReceived: 4 reported values across 1 units.

- BorrowingsInterestRate: 1 reported values across 1 units.
- DiscountRateUsedInPreviousMeasurementOfFairValueLessCostsOfDisposal: 1 reported values across 1 units.
- GainsLossesOnChangeInFairValueOfDerivatives: 2 reported values across 1 units.
- IncomeFromReimbursementsUnderInsurancePolicies: 2 reported values across 1 units.
- LossesOnChangeInFairValueOfDerivatives: 2 reported values across 1 units.

In [129…   **%debug**

```
> c:\users\jesse\appdata\local\temp\ipykernel_6656\4278471653.py(48)get_fact()

     43                  rec.set_period(period_type)
     44                  if rec.period is not None:
     45                      records.append(rec)
     46
     47          if not records:
---> 48              raise ValueError(f"No valid records found for fact {fact_name} with period type '{period_type}'.")
     49
     50          resolved = resolve_duplicates(records)
     51          fact_values = {period_label: rec.value for period_label, rec in resolved.items()}
     52          try:
     53              sorted_keys = sorted(fact_values.keys(), key=lambda x: parse_normalized_period(x))
```

for ENFY, operating income, loss: the parser returns

```
FY2010Q03      9.625155
FY2010Q04      7.872756
FY2011Q01     12.124854
FY2011Q03     13.279563
FY2011Q04     10.195755
                 ...
FY2023Q03     -1.785907
FY2023Q04     -5.390183
FY2024Q01    -10.971286
FY2024Q03     -1.765024
FY2024Q04     -4.792605
```

this lines up with https://www.macrotrends.net/stocks/charts/ENFY/enlightify/income-statement?freq=Q although it's missing a quarter

Quarterly Data | Millions of US $ except per share data		2024-12-31	2024-09-30	2024-06-30	2024-03-31	2023-12-31	2023-09-30	2023-06-30
Revenue		$14.348	$17.365	$22.71	$32.938	$17.8	$22.398	$26.742
Cost Of Goods Sold		$10.881	$13.378	$17.307	$26.597	$14.472	$17.748	$21.899
Gross Profit		$3.467	$3.987	$5.403	$6.341	$3.329	$4.65	$4.843
Research And Development Expenses		-	-	-	-	-	-	-
SG&A Expenses		$8.26	$5.751	$16.104	$17.312	$8.719	$6.436	$14.423
Other Operating Income Or Expenses		$-0	$0	-	-	-	-	-
Operating Expenses		$8.26	$5.751	$16.107	$17.312	$8.719	$6.436	$14.423
Operating Income		$-4.793	$-1.765	$-10.704	$-10.971	$-5.39	$-1.786	$-9.579

Quarterly Data | Millions of US $ except per share data		2012-03-31	2011-12-31	2011-09-30	2011-06-30	2011-03-31	2010-12-31	2010-09-30
Revenue	📊	$60	$47.1	$53.1	$60.3	$44.7	$35.312	$39.5
Cost Of Goods Sold	📊	$37.54	$30.7	$34.19	$39.17	$27.61	$22.97	$26.34
Gross Profit	📊	$22.46	$16.4	$18.91	$21.13	$17.09	$12.342	$13.16
Research And Development Expenses		-	-	-	-	-	-	-
SG&A Expenses	📊	$6.61	$6.19	$5.62	$8.62	$4.91	$4.46	$3.51
Other Operating Income Or Expenses	📊	-	-	-	-	-	-	-
Operating Expenses	📊	$6.61	$6.19	$5.62	$8.62	$4.91	$4.46	$3.51
Operating Income	📊	$15.85	$10.21	$13.29	$12.51	$12.18	$7.882	$9.65

concept

```
In [123…   companyConcept = requests.get(
               (
                   f'https://data.sec.gov/api/xbrl/companyconcept/CIK{cik}'
                    f'/us-gaap/Assets.json'
               ),
               headers=headers
               )

           # review data
           companyConcept.json().keys()
```

```
Out[123…   dict_keys(['cik', 'taxonomy', 'tag', 'label', 'description', 'entityName', 'units'])
```

```
In [124…   len(companyConcept.json()['units']['USD'])
```

```
Out[124…   134
```

```
In [117…   companyConcept.json()['units'].keys()
```

```
Out[117…   dict_keys(['USD'])
```

```
In [ ]:    companyConcept.json()['units']

           companyConcept.json()['units']['USD']
           companyConcept.json()['units']['USD'][0]

           # parse assets from single filing
           companyConcept.json()['units']['USD'][0]['val']

           # get all filings data
           assetsData = pd.DataFrame.from_dict((
                       companyConcept.json()['units']['USD']))
```

edgar tools

https://github.com/dgunning/edgartools/blob/main/docs/quick-guide.md

the base classes are fillings and companies

get fillings

In [177…   `?get_filings`

Signature:
```
get_filings(
    year: Union[int, List[int], range, NoneType] = None,
    quarter: Union[int, List[int], range, NoneType] = None,
    form: Union[str, List[Union[str, int]], NoneType] = None,
    amendments: bool = True,
    filing_date: Optional[str] = None,
    index='form',
    priority_forms: Optional[List[str]] = None,
) -> Optional[edgar._filings.Filings]
```
Docstring:
Downloads the filing index for a given year or list of years, and a quarter or list of quarters.

So you can download for 2020, [2020,2021,2022] or range(2020, 2023)

Examples

>>> from edgar import get_filings

>>> filings_ = get_filings(2021) # Get filings for 2021

>>> filings_ = get_filings(2021, 4) # Get filings for 2021 Q4

>>> filings_ = get_filings(2021, [3,4]) # Get filings for 2021 Q3 and Q4

>>> filings_ = get_filings([2020, 2021]) # Get filings for 2020 and 2021

>>> filings_ = get_filings([2020, 2021], 4) # Get filings for Q4 of 2020 and 2021

>>> filings_ = get_filings(range(2010, 2021)) # Get filings between 2010 and 2021 - does not include 2021

>>> filings_ = get_filings(2021, 4, form="D") # Get filings for 2021 Q4 for form D

>>> filings_ = get_filings(2021, 4, filing_date="2021-10-01") # Get filings for 2021 Q4 on "2021-10-01"

>>> filings_ = get_filings(2021, 4, filing_date="2021-10-01:2021-10-10") # Get filings for 2021 Q4 between
                                                                          # "2021-10-0
1" and "2021-10-10"

:param year The year of the filing
:param quarter The quarter of the filing
:param form The form or forms as a string e.g. "10-K" or a List ["10-K", "8-K"]
:param amendments If True will expand the list of forms to include amendments e.g. "10-K/A"
:param filing_date The filing date to filter by in YYYY-MM-DD format
           e.g. filing_date="2022-01-17" or filing_date="2022-01-17:2022-02-28"
:param index The index type - "form" or "company" or "xbrl"
:return:
File: c:\users\jesse\miniforge3\lib\site-packages\edgar_filings.py
Type: function

In [23]:
```python
# 3. Start using the library
filings = get_filings()
filings = filings.filter(date="2015-01-01:")
filings
```

Out[23]:
```
                                      ──────── SEC Filings ──────────────────
  |
  |
  | Form              CIK    Ticker   Company                            Fili
ng Date    Accession Number  |
  |───────────────────────────────────────────────────────────────────────────
──────────────────────  |
  |    10-Q           715446   ANIX    Anixa Biosciences Inc              2025
-03-11    0001493152-25-0098…  |
  |    10-Q          1059689           BrightRock Gold Corp               2025
-03-11    0001683168-25-0014…  |
  |    10-Q          1059689           BrightRock Gold Corp               2025
-03-11    0001683168-25-0014…  |
  |    10-Q           726958   CASY    CASEYS GENERAL STORES INC          2025
-03-11    0000726958-25-0000…  |
  |    10-Q          1703956   BBCP    Concrete Pumping Holdings, Inc.    2025
-03-11    0001437749-25-0070…  |
  |    10-Q           886128   FCEL    FUELCELL ENERGY INC                2025
-03-11    0001558370-25-0026…  |
  |    10-Q          2011641   FERG    Ferguson Enterprises Inc. /DE/     2025
-03-11    0002011641-25-0000…  |
  |    10-Q            56679   KFY     KORN FERRY                         2025
-03-11    0001628280-25-0119…  |
  |    10-Q          1341439   ORCL    ORACLE CORP                        2025
-03-11    0000950170-25-0371…  |
  |    10-Q            74046   ODC     Oil-Dri Corp of America            2025
-03-11    0000074046-25-0000…  |
  |    10-Q          1423221   NX      Quanex Building Products CORP      2025
-03-11    0001423221-25-0000…  |
  |    10-Q          1020859   UNFI    UNITED NATURAL FOODS INC           2025
-03-11    0001020859-25-0000…  |
  |    10-Q          1746119   MUGH    Vanguard Green Investment Ltd      2025
-03-11    0001493152-25-0097…  |
  |    10-K          2012964   AAM     AA Mission Acquisition Corp.       2025
-03-11    0001213900-25-0227…  |
  |    10-K           949858   ACHV    ACHIEVE LIFE SCIENCES, INC.        2025
-03-11    0000950170-25-0368…  |
  |    10-K           874761   AES     AES CORP                           2025
-03-11    0000874761-25-0000…  |
  |    10-K           750574   AUBN    AUBURN NATIONAL BANCORPORATION, INC 2025
-03-11    0001193125-25-0515…  |
  |    10-K          1754170   ASLE    AerSale Corp                       2025
-03-11    0001558370-25-0026…  |
  |    10-K          1454789   ATXS    Astria Therapeutics, Inc.          2025
-03-11    0001410578-25-0003…  |
  |    10-K          1728085           BANK 2018-BNK10                    2025
-03-11    0001888524-25-0042…  |
  |    10-K          1744707           BANK 2018-BNK13                    2025
-03-11    0001888524-25-0042…  |
  |    10-K          1763333           BANK 2019-BNK16                    2025
-03-11    0001888524-25-0042…  |
  |    10-K          1774962           BANK 2019-BNK18                    2025
-03-11    0001888524-25-0042…  |
  |    10-K          1789307           BANK 2019-BNK22                    2025
-03-11    0001888524-25-0042…  |
```

	10-K	1832874		BANK 2020-BNK30	2025
	-03-11	0001888524-25-0042…			
	10-K	**1863688**		**BANK 2021-BNK34**	**2025**
	-03-11	**0001888524-25-0043…**			
	10-K	1880921		BANK 2021-BNK36	2025
	-03-11	0001888524-25-0042…			
	10-K	**1907569**		**BANK 2022-BNK40**	**2025**
	-03-11	**0001888524-25-0042…**			
	10-K	743367	BHB	BAR HARBOR BANKSHARES	2025
	-03-11	0001558370-25-0026…			
	10-K	**1864196**		**BBCMS Mortgage Trust 2021-C10**	**2025**
	-03-11	**0001888524-25-0043…**			
	10-K	1462056	BLZE	Backblaze, Inc.	2025
	-03-11	0001462056-25-0000…			
	10-K	**1665988**	**BVS**	**Bioventus Inc.**	**2025**
	-03-11	**0001628280-25-0118…**			
	10-K	1817640	BREZ	Breeze Holdings Acquisition Corp.	2025
	-03-11	0001213900-25-0228…			
	10-K	**1059689**		**BrightRock Gold Corp**	**2025**
	-03-11	**0001683168-25-0014…**			
	10-K	726601	CCBG	CAPITAL CITY BANK GROUP INC	2025
	-03-11	0000726601-25-0000…			
	10-K	**1685854**		**CFCRE 2016-C6 Mortgage Trust**	**2025**
	-03-11	**0001888524-25-0043…**			
	10-K	1690110		CFCRE 2016-C7 Mortgage Trust	2025
	-03-11	0001888524-25-0042…			
	10-K	**1703075**		**CFCRE 2017-C8 Mortgage Trust**	**2025**
	-03-11	**0001888524-25-0043…**			
	10-K	803164	COFS	CHOICEONE FINANCIAL SERVICES INC	2025
	-03-11	0000950170-25-0368…			
	10-K	**1995413**	**CLBR**	**COLOMBIER ACQUISITION CORP. II**	**2025**
	-03-11	**0001213900-25-0223…**			
	10-K	1617959		COMM 2014-UBS5 Mortgage Trust	2025
	-03-11	0001888524-25-0043…			
	10-K	**1640052**		**COMM 2015-CCRE23 Mortgage Trust**	**2025**
	-03-11	**0001888524-25-0043…**			
	10-K	1653323		COMM 2015-CCRE27 Mortgage Trust	2025
	-03-11	0001888524-25-0042…			
	10-K	**1645384**		**COMM 2015-PC1 Mortgage Trust**	**2025**
	-03-11	**0001888524-25-0043…**			
	10-K	1663239		COMM 2016-CCRE28 Mortgage Trust	2025
	-03-11	0001888524-25-0043…			
	10-K	**1685212**		**COMM 2016-COR1 Mortgage Trust**	**2025**
	-03-11	**0001888524-25-0043…**			
	10-K	1792157		COMM 2019-GC44 Mortgage Trust	2025
	-03-11	0001888524-25-0043…			
	10-K	**1026655**	**CMT**	**CORE MOLDING TECHNOLOGIES INC**	**2025**
	-03-11	**0001026655-25-0000…**			
	10-K	1639353		CSAIL 2015-C2 Commercial Mortgage Tru…	2025
	-03-11	0001888524-25-0043…			
	10-K	**1647980**		**CSAIL 2015-C3 Commercial Mortgage Tru…**	**2025**
	-03-11	**0001888524-25-0042…**			

```
|
|
|  Showing 1 to 50 of 266,514 filings. Page using ← prev() and next() →
|
|                                              ─── SEC Filings between 2025-01-02 and 2025-03-11
└──────────────────────────────────────┘
```

In [26]: ```python
fillings = get_filings(year=range(2015,2026),form='10-Q')
```

In [31]: ```python
#by cik
ag_filling = fillings.filter(cik=list(ag_companies))
ag_filling
```

Out[31]:

```
┌───────────────────────────── SEC Filings ──────────────────────┐
│                                                                 │
│                                                                 │
│      Form           CIK     Ticker   Company                    Fili
└─ng Date    Accession Number │
│   ─────────────────────────────────────────────────────────────────
│   ──────────────────────────│
│     10-Q         857949    CGA      Enlightify Inc.                2025
-02-24    0001213900-25-0166… │
│     10-Q         875729    BNET     BION ENVIRONMENTAL TECHNOLOGIES INC  2025
-02-14    0001079973-25-0002… │
│     10-Q         825542    SMG      SCOTTS MIRACLE-GRO CO          2025
-02-05    0001546380-25-0000… │
│     10-Q        1794276    YCQH     YCQH Agricultural Technology Co. Ltd  2024
-11-19    0001493152-24-0466… │
│     10-Q         875729    BNET     BION ENVIRONMENTAL TECHNOLOGIES INC  2024
-11-14    0001079973-24-0015… │
│     10-Q         857949    CGA      China Green Agriculture, Inc.  2024
-11-14    0001213900-24-0984… │
│     10-Q        1789192    SVFD     N2OFF, Inc.                    2024
-11-14    0001493152-24-0458… │
│     10-Q           5981    AVD      AMERICAN VANGUARD CORP         2024
-11-12    0000950170-24-1256… │
│     10-Q        1285785    MOS      MOSAIC CO                      2024
-11-12    0001618034-24-0000… │
│     10-Q        1680378    SNES     SenesTech, Inc.                2024
-11-12    0001628280-24-0473… │
│     10-Q        1705843    CLXT     Cibus, Inc.                    2024
-11-07    0001628280-24-0463… │
│     10-Q        1324404    CF       CF Industries Holdings, Inc.   2024
-10-31    0001324404-24-0000… │
│     10-Q        1425292    UAN      CVR PARTNERS, LP               2024
-10-29    0001425292-24-0000… │
│     10-Q        1575858    PUBC     PureBase Corp                  2024
-10-15    0001493152-24-0410… │
│     10-Q        1794276    YCQH     YCQH Agricultural Technology Co. Ltd  2024
-08-23    0001493152-24-0337… │
│     10-Q        1789192    SVFD     N2OFF, Inc.                    2024
-08-14    0001493152-24-0321… │
│     10-Q           5981    AVD      AMERICAN VANGUARD CORP         2024
-08-09    0000950170-24-0948… │
│     10-Q        1324404    CF       CF Industries Holdings, Inc.   2024
-08-08    0001324404-24-0000… │
│     10-Q        1705843    CLXT     Cibus, Inc.                    2024
-08-08    0001628280-24-0361… │
│     10-Q        1680378    SNES     SenesTech, Inc.                2024
-08-08    0001628280-24-0362… │
│     10-Q        1285785    MOS      MOSAIC CO                      2024
-08-07    0001618034-24-0000… │
│     10-Q         825542    SMG      SCOTTS MIRACLE-GRO CO          2024
-08-07    0001546380-24-0000… │
│     10-Q        1425292    UAN      CVR PARTNERS, LP               2024
-07-30    0001425292-24-0000… │
│     10-Q        1575858    PUBC     PureBase Corp                  2024
-07-15    0001493152-24-0277… │
```

	10-Q	1794276	YCQH	YCQH Agricultural Technology Co. Ltd	2024
-05-20	0001493152-24-0206…				
	10-Q	857949	CGA	China Green Agriculture, Inc.	2024
-05-16	0001213900-24-0443…				
	10-Q	875729	BNET	BION ENVIRONMENTAL TECHNOLOGIES INC	2024
-05-15	0001079973-24-0007…				
	10-Q	1789192	SVFD	N2OFF, Inc.	2024
-05-15	0001493152-24-0198…				
	10-Q	5981	AVD	AMERICAN VANGUARD CORP	2024
-05-09	0000950170-24-0571…				
	10-Q	1705843	CLXT	Cibus, Inc.	2024
-05-09	0001628280-24-0222…				
	10-Q	1680378	SNES	SenesTech, Inc.	2024
-05-09	0001628280-24-0223…				
	10-Q	825542	SMG	SCOTTS MIRACLE-GRO CO	2024
-05-08	0001546380-24-0000…				
	10-Q	1324404	CF	CF Industries Holdings, Inc.	2024
-05-02	0001324404-24-0000…				
	10-Q	1285785	MOS	MOSAIC CO	2024
-05-02	0001618034-24-0000…				
	10-Q	1425292	UAN	CVR PARTNERS, LP	2024
-04-30	0001425292-24-0000…				
	10-Q	1575858	PUBC	PureBase Corp	2024
-04-15	0001493152-24-0146…				
	10-Q	875729	BNET	BION ENVIRONMENTAL TECHNOLOGIES INC	2024
-02-14	0001079973-24-0002…				
	10-Q	857949	CGA	China Green Agriculture, Inc.	2024
-02-09	0001213900-24-0121…				
	10-Q	825542	SMG	SCOTTS MIRACLE-GRO CO	2024
-02-07	0001546380-24-0000…				
	10-Q	857949	CGA	China Green Agriculture, Inc.	2023
-11-20	0001213900-23-0887…				
	10-Q	1789192	SVFD	Save Foods, Inc.	2023
-11-14	0001493152-23-0410…				
	10-Q	1794276	YCQH	YCQH Agricultural Technology Co. Ltd	2023
-11-14	0001493152-23-0407…				
	10-Q	875729	BNET	BION ENVIRONMENTAL TECHNOLOGIES INC	2023
-11-13	0001079973-23-0015…				
	10-Q	5981	AVD	AMERICAN VANGUARD CORP	2023
-11-09	0000950170-23-0613…				
	10-Q	1705843	CLXT	Cibus, Inc.	2023
-11-09	0001628280-23-0382…				
	10-Q	1680378	SNES	SenesTech, Inc.	2023
-11-09	0001628280-23-0383…				
	10-Q	1285785	MOS	MOSAIC CO	2023
-11-08	0001618034-23-0000…				
	10-Q	1324404	CF	CF Industries Holdings, Inc.	2023
-11-02	0001324404-23-0000…				
	10-Q	1425292	UAN	CVR PARTNERS, LP	2023
-10-31	0001425292-23-0000…				
	10-Q	1575858	PUBC	PureBase Corp	2023
-10-13	0001493152-23-0370…				

```
 │
 │
 │   Showing 1 to 50 of 335 filings. Page using ← prev() and next() →
 │
 └──────────────────────────────────────────┐
                                             ── SEC Filings between 2015-02-05 and 2025-02-24
```

read xbrl

some fillings are xbrl, and there fore can be read: `cga_q_xbrl = ag_filling[0].xbrl()`
and `xbrl.facts` returns the df of facts. this can also be used to access the facts or
conepts: using the gap names. can access specific facts:
cga_q_xbrl.instance.query_facts(concept="us-gaap:Assets")

to see specific tables use the statements api: `cga_q_xbrl.statements` to see their names
and `cga_q_xbrl.statements['ConsolidatedIncomeStatement']` to print it

In [145…
```python
cga_q_xbrl = ag_filling[0].xbrl()
cga_q_xbrl
```

Out[145…
```
┌─────────────────────────────────────────────────── XBRL ───────────────────────────────
└─────────────────────────────────┐
 │                      XBRL Instance
 │
 │
 │   Company          Form    Period              Facts
 │  ────────────────────────────────────────────────────────
 │
 │   ENLIGHTIFY, INC.  10-Q    December 31, 2024   781
 │
 │
 │
 │                      Financial Statements
 │
 │
 │
 │   Name                              Accessor
 │  ──────────────────────────────────────────────────────
 │
 │   Balance Sheet                     financials.balance_sheet
 │
 │   Income Statement                  financials.income
 │
 │   Cash Flow Statement               financials.cashflow
 │
 │
 │                                                              ────
└─────────────────────────────────┐
```

In [ ]: `ge`

In [153… `cga_q_xbrl.statements['ConsolidatedIncomeStatement']`

Out[153…

ENLIGHTIFY, INC.
Income Statement
3 Months

	Dec 31, 2024	Dec 31, 2023
_		
Revenue from Contract with Customer, Including Assessed Tax	14,348,138	17,800,114
Cost of Goods and Services Sold	(10,881,359)	(14,471,627)
Gross Profit	**3,466,779**	**3,328,487**
Selling Expense	1,817,206	1,770,860
General and Administrative Expense	6,442,440	6,947,810
Fair Value Adjustment Of Bitcoin	(262)	
Operating Expenses	**(8,259,384)**	**(8,718,670)**
Operating Income (Loss)	**(4,792,605)**	**(5,390,183)**
Other Nonoperating Income (Expense)	(268,594)	30,926
Investment Income, Interest	39,346	51,125
Investment Income, Investment Expense	(121,189)	(73,813)
Nonoperating Income (Expense)	**(350,437)**	**8,238**
Income (Loss) from Continuing Operations before Income Taxes, Noncontrolling Interest	(5,143,043)	(5,381,945)
Income Tax Expense (Benefit)	(10,393)	(11,942)
Net Income Loss	**(5,132,650)**	**(5,370,003)**
Other Comprehensive Income (Loss), Foreign Currency Transaction and Translation Adjustment, Net of Tax	(5,314,813)	4,705,400
Comprehensive Income (Loss), Net of Tax, Attributable to Parent	**(10,447,463)**	**(664,603)**
Weighted Average Number of Shares Outstanding, Basic	14,793,538	13,380,914
Earnings Per Share, Basic	(0.35)	(0.4)
Weighted Average Number of Shares Outstanding, Diluted	14,793,538	13,380,914
Earnings Per Share, Diluted	(0.35)	(0.4)

Available durations for this statement:
3 Months   6 Months

In [162… `get_fact_names(comp_facts,'OperatingIncomeLoss')`

Out[162… `['OperatingIncomeLoss', 'SegmentReportingInformationOperatingIncomeLoss']`

In [ ]:

edgar tool comapny

```
In [178…   company = Company("CGA")
```

```
In [182…   cga_fil = company.get_filings()
```

```
In [183…   cga_fil.filter('accession_number'=)
```

Signature:
```
cga_fil.filter(
    form: Union[str, List[str]] = None,
    amendments: bool = None,
    filing_date: str = None,
    date: str = None,
    cik: Union[str, int, List[Union[str, int]]] = None,
    ticker: Union[str, List[str]] = None,
    accession_number: Union[str, List[str]] = None,
)
```
Docstring:
```
Get some filings

>>> filings = get_filings()

Filter the filings

On a date
>>> filings.filter(date="2020-01-01")

Up to a date
>>> filings.filter(date=":2020-03-01")

From a date
>>> filings.filter(date="2020-01-01:")

# Between dates
>>> filings.filter(date="2020-01-01:2020-03-01")

:param form: The form or list of forms to filter by
:param amendments: Whether to include amendments to the forms e.g. include "10-K/A"
if filtering for "10-K"
:param filing_date: The filing date
:param date: An alias for the filing date
:param cik: The CIK or list of CIKs to filter by
:param exchange: The exchange or list of exchanges to filter by
:param ticker: The ticker or list of tickers to filter by
:param accession_number: The accession number or list of accession numbers to filter
by
:return: The filtered filings
```
File: c:\users\jesse\miniforge3\lib\site-packages\edgar\entities.py
Type: method

The financials property returns a Financials instance. This instance has methods that return the balance sheet, income statement and cash flow statement.

```
from edgar import MultiFinancials
```

```
        filings = company.latest("10-K", 5)
        financials = MultiFinancials(filings)
```

In [ ]:
```
ntr_filings = company.latest("10-K", 5)
ntr_filings  = MultiFinancials(ntr_filings)
ntr_filings
```

In [19]: `df_tickers`

Out[19]:

	cik_str	ticker	title
0	320193	AAPL	Apple Inc.
1	789019	MSFT	MICROSOFT CORP
2	1045810	NVDA	NVIDIA CORP
3	1018724	AMZN	AMAZON COM INC
4	1652044	GOOGL	Alphabet Inc.
...	...	...	...
9703	1884046	SPKLU	Spark I Acquisition Corp
9704	1884046	SPKLW	Spark I Acquisition Corp
9705	2019804	HONDW	HCM II Acquisition Corp.
9706	2019804	HONDU	HCM II Acquisition Corp.
9707	797468	OXY-WT	OCCIDENTAL PETROLEUM CORP /DE/

9708 rows × 3 columns