

```
In [1]: from datetime import datetime
start_time = datetime.now()
print("===== START =====")
print(start_time.strftime("%Y-%m-%d %H:%M:%S"))

===== START =====
2026-02-02 14:34:31
```

```
In [2]: from selenium import webdriver
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common.by import By
from bs4 import BeautifulSoup
import pandas as pd
import time
import re
import csv
import fnmatch
import pickle
import os
from tqdm import tqdm
import pikepdf
import openpyxl
from openpyxl import load_workbook
```

```
In [3]: df_full = pd.read_csv("/Users/qianqianli/Dropbox/BDL_code/1_data_processing/data/sample_selection/sdc/SDC_filen.csv")
print(df_full.shape)
df_full.head()
```

(31884, 2)

	Unnamed: 0	SEC File Form (Number)
0	1	148433
1	2	148440
2	3	148437
3	4	148439
4	5	148443

extract filings of each filen

```
In [4]: # creation of a new instance of Google Chrome
browser = webdriver.Chrome()
```

```
In [5]: # for i in range(319):
# here, we run the first 100 filen to generate sample log
for i in range(1):
    df = df_full.iloc[i*100:(i+1)*100].reset_index(drop = True)

    df["soup"] = ""
    df["cik"] = ""
    df["filings"] = ""

    for j in range(100):
        sec_number = df.iloc[j]["SEC File Form (Number)"]
        browser.get("https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&filenum=333-" + str(sec_number))
        time.sleep(2)

        soup = BeautifulSoup(browser.page_source, "lxml")

        try:

            cik = soup.find("span", {"class": "companyName"}).find("a").get_text().split()[0]

            filings = [tr.find("td").get_text() for tr in soup.find_all("tr")[5:]]
            format = [tr.find("a").get_text().strip() for tr in soup.find_all("tr")[5:]]
            description = [tr.find("td", {"class": "small"}).get_text(separator = " ").strip() for tr in soup.find_all("tr")[5:]]
            filing_date = [tr.find_all("td")[3].get_text() for tr in soup.find_all("tr")[5:]]
            file_number = [tr.find_all("td")[4].get_text(separator = " ").strip() for tr in soup.find_all("tr")[5:]]

            combine = [filings, format, description, filing_date, file_number]

            df.at[j, "soup"] = soup
            df.at[j, "cik"] = cik
            df.at[j, "filings"] = combine

            time.sleep(1)

        except:

            try:

                cik = soup.find("span", {"class": "companyName"}).find("a").get_text().split()[0]

                filing_list = []
                format_list = []
                description_list = []
                filing_date_list = []
                file_number_list = []

                filings = [tr.find("td").get_text() for tr in soup.find_all("tr")[5:-1]]
                format = [tr.find("a").get_text().strip() for tr in soup.find_all("tr")[5:-1]]
                description = [tr.find("td", {"class": "small"}).get_text(separator = " ").strip() for tr in soup.find_all("tr")[5:-1]]
                filing_date = [tr.find_all("td")[3].get_text() for tr in soup.find_all("tr")[5:-1]]
                file_number = [tr.find_all("td")[4].get_text(separator = " ").strip() for tr in soup.find_all("tr")[5:-1]]

                filing_list.append(filings)
                format_list.append(format)
                description_list.append(description)
                filing_date_list.append(filing_date)
                file_number_list.append(file_number)

                time.sleep(2)

            while soup.find("input", {"value": "Next40"}) is not None:

                browser.find_element(By.XPATH, "//input[@value='Next40']").click()

                time.sleep(2)

            try:

                soup = BeautifulSoup(browser.page_source, "lxml")

                filings = [tr.find("td").get_text() for tr in soup.find_all("tr")[5:-1]]
                format = [tr.find("a").get_text().strip() for tr in soup.find_all("tr")[5:-1]]
                description = [tr.find("td", {"class": "small"}).get_text(separator = " ").strip() for tr in soup.find_all("tr")[5:-1]]
                filing_date = [tr.find_all("td")[3].get_text() for tr in soup.find_all("tr")[5:-1]]
                file_number = [tr.find_all("td")[4].get_text(separator = " ").strip() for tr in soup.find_all("tr")[5:-1]]

                filing_list.append(filings)
                format_list.append(format)
                description_list.append(description)
                filing_date_list.append(filing_date)
                file_number_list.append(file_number)

            except:
                print("Done")

            filing_list = [x for xs in filing_list for x in xs]
            format_list = [x for xs in format_list for x in xs]
            description_list = [x for xs in description_list for x in xs]
            filing_date_list = [x for xs in filing_date_list for x in xs]
            file_number_list = [x for xs in file_number_list for x in xs]

            combine = [filing_list, format_list, description_list, filing_date_list, file_number_list]

            df.at[j, "soup"] = soup
            df.at[j, "cik"] = cik
            df.at[j, "filings"] = combine

            time.sleep(1)

        except:
            df.at[j, "soup"] = soup
            df.at[j, "cik"] = None
            df.at[j, "filings"] = None

    df.to_pickle("/Users/qianqianli/Dropbox/BDL_code/1_data_processing/data/sample_selection/sdc/edgar_" + str(i) + ".pkl")
```

combine to dataframe

```
In [6]: df = pd.DataFrame()
# for i in range(319):
for i in range(1):
    a = pd.read_pickle("/Users/qianqianli/Dropbox/BDL_code/1_data_processing/data/sample_selection/sdc/edgar_" + str(i) + ".pkl")
    df = pd.concat([df, a])
```

```
In [7]: df_1 = df[~df["cik"].isna()].reset_index(drop = True)
df_1["number_of_filings"] = df_1["filings"].map(lambda x: len(x[0]))
```

S1

```
In [8]: df_1["S1"] = df_1["filings"].map(lambda x: 1 if "S-1" in x[0] else 0)
df_1["S1_index"] = df_1["filings"].map(lambda x: x[0].index("S-1") if "S-1" in x[0] else None)
df_1["S1_filing_date"] = df_1.apply(lambda x: x["filings"][3][int(x["S1_index"])] if pd.notnull(x["S1_index"]) else None, axis = 1)
df_1["S1_filing_name"] = df_1.apply(lambda x: re.search(r"Acc-no:(.*?)\[([S])", x["filings"][2][int(x["S1_index"])]).group(1).strip() if pd.notnull(x["S1_index"]) else None, axis = 1)
```

RW

```
In [9]: df_1["RW"] = df_1["filings"].map(lambda x: 1 if "RW" in x[0] else 0)
df_1["RW_index"] = df_1["filings"].map(lambda x: x[0].index("RW") if "RW" in x[0] else None)
df_1["RW_filing_date"] = df_1.apply(lambda x: x["filings"][3][int(x["RW_index"])] if pd.notnull(x["RW_index"]) else None, axis = 1)
df_1["RW_filing_name"] = df_1.apply(lambda x: re.search(r"Acc-no:(.*?)\[([S])", x["filings"][2][int(x["RW_index"])]).group(1).strip() if pd.notnull(x["RW_index"]) else None, axis = 1)
```

EFFECT

```
In [10]: df_1["EFFECT"] = df_1["filings"].map(lambda x: 1 if "EFFECT" in x[0] else 0)
df_1["EFFECT_index"] = df_1["filings"].map(lambda x: x[0].index("EFFECT") if "EFFECT" in x[0] else None)
df_1["EFFECT_filing_date"] = df_1.apply(lambda x: x["filings"][3][int(x["EFFECT_index"])] if pd.notnull(x["EFFECT_index"]) else None, axis = 1)
df_1["EFFECT_filing_name"] = df_1.apply(lambda x: re.search(r"Acc-no:(.*?)\[([S])", x["filings"][2][int(x["EFFECT_index"])]).group(1).strip() if pd.notnull(x["EFFECT_index"]) else None, axis = 1)
```

save

```
In [11]: df_to_save = df_1[['SEC File Form (Number)', 'cik',
                           'number_of_filings', 'S1', 'S1_filing_date',
                           'S1_filing_name', 'RW', 'RW_filing_date', 'RW_filing_name',
                           'EFFECT', 'EFFECT_filing_date']]
print(df_to_save.shape)
df_to_save.head()
```

(93, 11)

	SEC File Form (Number)	cik	number_of_filings	S1	S1_filing_date	S1_filing_name	RW	RW_filing_date	RW_filing_name	EFFECT	EFFECT_filing_date
0	148433	0001328511	5	0	None	None	0	None	None	1	2008-04-09
1	148440	0001421636	3	0	None	None	0	None	None	1	2008-01-17 15:00:00
2	148437	0001421603	3	0	None	None	0	None	None	1	2008-01-17 15:00:00
3	148439	0001421602	3	0	None	None	0	None	None	1	2008-01-17 15:00:00
4	148443	0000894490	4	0	None	None	0	None	None	1	2008-02-13 13:00:00

```
In [12]: df_to_save.to_csv("/Users/qianqianli/Dropbox/BDL_code/1_data_processing/data/sample_selection/sdc/SDC_filen_EDGAR_sample.csv")
```

```
In [13]: end_time = datetime.now()
print("===== END =====")
print(end_time.strftime("%Y-%m-%d %H:%M:%S"))
print("Elapsed seconds:", (end_time - start_time).total_seconds())

===== END =====
2026-02-02 14:40:58
Elapsed seconds: 387.424698
```