# Project Description

Pada proyek ini saya melakukan proses *scraping* pada website Bank Indonesia. Karena *website* tersebut adalah dinamik dan menggunakan JavaScript, tidak cukup menggunakan *library* Requests dan BeautifulSoup saja untuk melakukan *scraping*. Sehingga pada proyek ini saya menggunakan *library* Selenium untuk melakukan *scraping*. *Scraping* menggunakan *library* Selenium membutuhkan suatu *driver* yang sesuai dengan *browser* yang akan digunakan untuk *scraping*.

# Script Scraping Website BI

## BI 7-Day (Reverse) Repo Rate

# Import libraries

from selenium import webdriver

from selenium.webdriver.common.by import By

import pandas as pd

import time

import os

from selenium.common.exceptions import TimeoutException

# define the url

url= "https://www.bi.go.id/id/statistik/indikator/bi-7day-rr.aspx"

# Configure selenium

options = webdriver.ChromeOptions()

options.add\_argument("--headless=new")

driver = webdriver.Chrome(options=options)

# print status

print("Please wait, start scraping...")

# solution for timeout

driver.set\_page\_load\_timeout(10)

try:

    driver.get(url)

except TimeoutException:

    driver.execute\_script("window.stop();")

# wait for elements to show up

time.sleep(5)

# get the table

table = driver.find\_elements(By.CLASS\_NAME, 'text-center')

# print status again

print("Scraping finished!")

# get the href from the table elements

hrefs = []

for element in table:

    link\_elements = element.find\_elements(By.TAG\_NAME, 'a')

    for link\_element in link\_elements:

        href = link\_element.get\_attribute('href')

        hrefs.append(href)

# get the text from the table

texts = []

for element in table:

    texts.append(element.text)

# create df

header = texts[:4]

data = [texts[i:i+4] for i in range(4, len(texts), 4)]

df = pd.DataFrame(data, columns=header)

# Change the content of the column

df['Pranala Siaran Pers'] = hrefs

# save df to csv

df.to\_csv(os.getcwd() + '\data\scraped\_BI\_1.csv', index=False)

## JISDOR

# Import libraries

from selenium import webdriver

from selenium.webdriver.common.by import By

import pandas as pd

import time

import os

from selenium.common.exceptions import TimeoutException

# define the url

url= "https://www.bi.go.id/id/statistik/informasi-kurs/jisdor/Default.aspx"

# Configure selenium

options = webdriver.ChromeOptions()

options.add\_argument("--headless=new")

driver = webdriver.Chrome(options=options)

# print status

print("Please wait, start scraping...")

# solution for timeout

driver.set\_page\_load\_timeout(10)

try:

    driver.get(url)

except TimeoutException:

    driver.execute\_script("window.stop();")

# wait for elements to show up

time.sleep(5)

# get the table

table = driver.find\_elements(By.CLASS\_NAME, 'text-center')

# print status again

print("Scraping finished!")

# get the text from the table

texts = []

for element in table:

    texts.append(element.text)

# create df

header = texts[:2]

data = [texts[i:i+2] for i in range(2, len(texts), 2)]

df = pd.DataFrame(data, columns=header)

# save df to csv

# save df to csv

df.to\_csv(os.getcwd() + '\data\scraped\_BI\_2.csv', index=False)

## Data Kurs Transaksi Bank Indonesia

# Import libraries

from selenium import webdriver

from selenium.webdriver.common.by import By

import pandas as pd

import time

import os

from selenium.common.exceptions import TimeoutException

# define the url

url= "https://www.bi.go.id/id/statistik/informasi-kurs/transaksi-bi/Default.aspx"

# Configure selenium

options = webdriver.ChromeOptions()

options.add\_argument("--headless=new")

driver = webdriver.Chrome(options=options)

# print status

print("Please wait, start scraping...")

# solution for timeout

driver.set\_page\_load\_timeout(10)

try:

    driver.get(url)

except TimeoutException:

    driver.execute\_script("window.stop();")

# wait for elements to show up

time.sleep(5)

# get the table

table = driver.find\_elements(By.CLASS\_NAME, 'text-right')

# print status again

print("Scraping finished!")

# get the text from the table

lst\_elements = []

for element in table:

    lst\_elements.append(element.text)

# remove empty elements

lst\_elements = list(filter(bool, lst\_elements))

# create df

header = lst\_elements[:4]

data = [lst\_elements[i:i+4] for i in range(5, len(lst\_elements), 4)]

df = pd.DataFrame(data, columns=header)

# save df to csv

df.to\_csv(os.getcwd() + '\data\scraped\_BI\_3.csv', index=False)

# Output

## BI 7-Day (Reverse) Repo Rate

A screenshot of a computer

Description automatically generated

## JISDOR

A screenshot of a table

Description automatically generated

## Data Kurs Transaksi Bank Indonesia

A screenshot of a data

Description automatically generated

# Github Repo

<https://github.com/adrn-mm/web_scraping>

# References

* [Web scraping javascript using Python](https://towardsdatascience.com/data-science-skills-web-scraping-javascript-using-python-97a29738353f)
* [Web scraping with Python in Indonesian E-Commerce](https://medium.com/@yohan.ardiansyah90/web-scraping-with-python-in-indonesian-e-commerce-tokopedia-part-1-getting-the-data-a338ebd56306)