10/18/2024

Nama Mahasiswa : Kevin Ramses Stifan

NPM : 2111010043

Kelas : 7TI1

Prodi : Teknik Informatika

Laporan tugas lab 1

Daftar Isi

[Laporan Coding web scraper 3](#_Toc179535233)

[Laporan hasil dari coding web scraper 7](#_Toc179535234)

# Laporan Coding web scraper

1. Install selenium, beautifulsoup4, dan webdriver manager.

!pip install selenium

!pip install beautifulsoup4

!pip install webdriver-manager

1. Import module.

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.firefox.options import Options

from bs4 import BeautifulSoup

1. Konfigurasi web driver.

# Function to scrape data from Dicoding

def scraper(url):

    try:

        # Configure WebDriver to use headless Firefox

        options = Options()

        options.add\_argument('-headless')

        driver = webdriver.Firefox(options=options)

1. Akses website.

       # Get the URL given

        driver.get(url)

        # Selenium will wait for a maximum of 5 seconds for an element matching the given criteria to be found.

        # If no element is found in that time, Selenium will raise an error.

        # try:

        #     wait = WebDriverWait(driver, timeout=5)

        #     wait.until(EC.presence\_of\_element\_located((By.ID, 'course-list')))

        # except:

        #     raise LookupError("There is no element specified")

1. Parse HTML.

        # BeautifulSoup will parse the URL

        content = driver.page\_source

        soup = BeautifulSoup(content, 'html.parser')

        print(content)

        print(soup)

        # Prepare the variable for JSON data

        courses = []

1. Ekstrak informasi.

        # BeautifulSoup will find the CSS class that contain product container

        for course in soup.find\_all('div', class\_='col-md-6 mb-3'):

            # Get the text from the specified element and assign them to the variables

            course\_name = course.find('h5', class\_='course-card\_\_name').text

            course\_hour = course.find\_all('span', {'class':'mr-2'})[0].text

            course\_summary = course.select('div.course-card\_\_summary p')[0].text

            course\_total\_module = course.find\_all('div', class\_= 'course-card\_\_info-item')[0].find\_all('span')[0].contents[0]

            course\_level = course.find('span', attrs={'class': None}).text

            # Not all courses in the list has rating, so we should manage it.

            # If it has rating, get the text. If none, set it to empty string.

            try:

                course\_rating = course.find\_all('span', {'class':'mr-2'})[1].text

            except IndexError:

                # Handle the case when no span elements with the specified class are found

                course\_rating = ''

            # Not all courses in the list has total students, so we should manage it.

            # If it has total students, get the text. If none, set it to empty string.

            try:

                course\_total\_students = course.find\_all('span', {'class':'mr-3'})[1].get\_text()

            except:

                course\_total\_students = ''

1. Penambahan data.

            # Append the scraped data into courses variable for JSON data

            courses.append(

                {

                    'Course Name': course\_name,

                    'Learning Hour': course\_hour,

                    'Rating': course\_rating,

                    'Level': course\_level,

                    'Summary': course\_summary,

                    'Total Modules': course\_total\_module,

                    'Total Students': course\_total\_students

                }

            )

1. Penutupan sesi.

        # Close the WebDriver

        driver.quit()

        return courses

1. Penanganan eror.

    except Exception as e:

        # Print the error message

        print('An error occurred: ', e)

        # Close the WebDriver

        driver.quit()

1. Pemanggilan fungsi scraper.

# import the modules to support the scraping process

import json

if \_\_name\_\_ == '\_\_main\_\_':

    # Define the URL

    url = 'https://www.dicoding.com/academies/list'

    data = scraper(url)

    # Save data to JSON file

    with open('/content/sample\_datadicoding\_data.json', 'w') as json\_file:

        json.dump(data, json\_file, indent=4)

# Laporan hasil dari coding web scraper



