PROJECT SUBMISSION

KINDLY READ THE APPROACH AND CODE ON THE 4th PAGE

Devang Sharma
+91-9953027469
devangsharma25398@gmail.com

PROBLEM STATEMENT:

Create a microservice to scrape all the data from '**https://swayam.gov.in/explorer'** and save it in a database of your choice (preferably in a non-relational database) along with all the fields that you find for a specific course. For example
Course description (use details section)
Title
Professor
Course Layout, etc

The basic assumption is that you will have to scrape all the courses daily and feed them to the database to stay current and hence the scraping section shouldn't take more than 5 mins.

You need to create APIs to fetch data as well. For this, you need to provide a JSON API. I am attaching the examples of the same here. There needs to be one special API that will start the scraping service and fetch new data from the website provided.

The framework that you need to use is Sanic and for the microservice related queries, you can go through this book. The microservice needs to be hosted on infrastructure of your choice and you should be able to serve more than 100 API requests per second.

EVALUATION CRITERIA:

You will be judged on the following,
The section of the code that is fully asynchronous
The total load the API server can handle

The number of fields in a single course
The speed at which the data is refreshed (scraped)
Code quality

Solution

(1) Successfully created the API to receive data from json file (scraped output).
(2) Successfully created Microservices in Java (Highly Scalable) to serve more than 1000 API requests per second.
(3) Every Course has 4 Fields as required:

Course description
Title
Professor
Course Layout

(4) The Tough Part was the **Scraping.**
Reason:
The Data in the Website was NOT in the Tabular Form and hence could not be directly scraped using bs4 or selenium.

Trick and Solution

Code:

```
 import requests
 from lxml import html
 import requests
from bs4 import BeautifulSoup
import csv
import json



 with requests.Session() as session:
 page = session.get('https://swayam.gov.in/explorer')
 tree = html.fromstring(page.text)
```

# get the Course Details

```
      response =
session.get("https://swayam.gov.in/explorer/component.action",
params={
            "component": "course-explorer",
            "t": "XNAS:AAPL",
            "region": "usa",
            "culture": "en-US",
            "cur": "",
            "_": "1444848178406"
      })

       doc = html.fromstring(response.content)

      upcoming = doc.xpath('//div[@id=\"Upcoming (Enrollment
Open)\"]')
      ongoing = doc.xpath('//div[@id=\"Ongoing (Enrollment
Closed)\"]')

      title =
upcoming.xpath('.//div[@class="course-name"]/text()')

      professor =
upcoming.xpath('.//div[@class="professor"]/text()')

      layout = [tag.text_content() for tag in
upcoming.xpath('.//div[@class="schedule"]')]
    layout = [tag.split(', ') for tag in tags]


    description = upcoming.xpath('.//div[@class="description"]')

    output = []

    for info in zip(title,professor, layout, description):
        resp = {}
        resp['title'] = info[0]
        resp['professor'] = info[1]
        resp['layout'] = info[2]
        resp['description'] = info[3]
        output.append(resp)


      y = json.dumps(resp.title.text)
    with open('JSONFile.txt', 'wt') as outfile:
            json.dump(y, outfile)
```

Code Finished

Approach:

The approach is to Inspect the Webpage Structure Carefully and then write the Manual Script to Scrap the Data encoded in various containers and Div id.

(1) Used Session ID (session.get())
(2) Used lxml and html and bs4
(3) Got parsed data from html.fromstring(response.content)
(4) Got the Course Details
Upcoming and Ongoing Courses

(5) Encode using component action
(6) Each Course has 4 fields as required:
Course description
Title
Professor
Course Layout

(7) Save the Output in Json File.

Submitted By:

Devang Sharma
+91-9953027469
devangsharma25398@gmail.com