Webscraping Project_GitHub (FL_Supreme)

June 25, 2021

1 Github Scraping Project

1.1 FL_Supreme

[4]:
```
!pip install jovian --upgrade --quiet
```

[5]:
```
import jovian
```

[6]:
```
# Execute this to save new versions of the notebook
jovian.commit(project="github-project")
```

```
<IPython.core.display.Javascript object>
```

```
[jovian] Updating notebook "fl-supreme/github-project" on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/fl-supreme/github-project
```

[6]:
```
'https://jovian.ai/fl-supreme/github-project'
```

2 Pick a website and describe your objective

Outline: - Aim is to scrape the topics page on Github (https://github.com/topics) - Get a list of topics. For each topic, obtain 'topic title', 'topic page URL', and 'topic description' - For each topic, obtain the top 25 repositories from topic page - For each repository, obtain 'repo name', 'username', 'stars', and 'repo URL' - For each topic, create a CSV file in the following format:

3 Use the requests library to download web pages

[7]:
```
!pip install requests --upgrade --quiet
import requests
```

[8]:
```
topics_url = 'https://github.com/topics'
```

[9]:
```
response = requests.get(topics_url)
```

[10]:
```
response.status_code # If status_code = 200, its all good. Check HTTP codes for␣
↪more information
```

[10]:
```
200
```

[11]:
```
page_contents = response.text
```

[12]:
```
page_contents[:1000]
```

[12]:
```
'\n\n<!DOCTYPE html>\n<html lang="en" data-color-mode="auto" data-light-
theme="light" data-dark-theme="dark">\n  <head>\n    <meta charset="utf-8">\n
<link rel="dns-prefetch" href="https://github.githubassets.com">\n  <link
rel="dns-prefetch" href="https://avatars.githubusercontent.com">\n  <link
rel="dns-prefetch" href="https://github-cloud.s3.amazonaws.com">\n  <link
rel="dns-prefetch" href="https://user-images.githubusercontent.com/">\n\n\n\n
<link crossorigin="anonymous" media="all" integrity="sha512-esN1/6aDl0Gvs0VpTsQl
gSFdM9A4iTeMmOmXnpAg1dy/FpI38lc+2tsMbWNz29y7yYSr7FiJt4EyTKfBU7ZsZQ=="
rel="stylesheet" href="https://github.githubassets.com/assets/frameworks-7ac375f
fa6839741afb345694ec42581.css" />\n  \n    <link crossorigin="anonymous"
media="all" integrity="sha512-JD7XwMfOqfTPKro6hELWTUp8kPg2kxLmSGKmr/9lCzva5wqdN1
n0AVkJid3/oyd+QJ0LsjQq2h+tLL4mqxdfnw==" rel="stylesheet" href="https://github.gi
thubassets.com/assets/behaviors-243ed7c0c7cea9f4cf2aba3a8442d64d.css" />\n    \n
\n    \n    <link cro'
```

[13]:
```
with open('webpage.html', 'w') as file:
    file.write(page_contents)
```

4 Use Beautiful Soup to parse and extract information

[14]:
```
!pip install beautifulsoup4 --upgrade --quiet
```

[36]:
```
from bs4 import BeautifulSoup as bs
```

[16]:
```
doc = bs(page_contents, 'html.parser')
```

[17]:
```
selection_class = 'f3 lh-condensed mb-0 mt-1 Link--primary'
topic_title_tags = doc.find_all('p', {'class': selection_class})
```

[18]:
```
desc_selector = 'f5 color-text-secondary mb-0 mt-1'
topic_desc_tags = doc.find_all('p', {'class': desc_selector})
topic_desc_tags[0].text
```

[18]:
```
'\n              3D modeling is the process of virtually developing the surface
and structure of a 3D object.\n            '
```

[19]:
```
# topic_title_tag0 = topic_title_tags[0]
# div_tag = topic_title_tag0.parent
```

[20]:
```
topic_link_tags = doc.find_all('a', {'class': 'd-flex no-underline'})
topic_link_tags[0]['href']
```

[20]:
```
'/topics/3d'
```

[21]:
```
topic0url = "https://github.com" + topic_link_tags[0]['href']
```

[22]:
```
topic_titles = []

for tag in topic_title_tags:
    topic_titles.append(tag.text)

topic_titles[:5]
```

[22]:
```
['3D', 'Ajax', 'Algorithm', 'Amp', 'Android']
```

[23]:
```
topic_descriptions = []

for desc in topic_desc_tags:
    topic_descriptions.append(desc.text.strip())

topic_descriptions[:5]
```

[23]:
```
['3D modeling is the process of virtually developing the surface and structure
of a 3D object.',
 'Ajax is a technique for creating interactive web applications.',
 'Algorithms are self-contained sequences that carry out a variety of tasks.',
 'Amp is a non-blocking concurrency framework for PHP.',
 'Android is an operating system built by Google designed for mobile devices.']
```

[142]:
```
topic_urls = []

for url in topic_link_tags:
    topic_urls.append("https://github.com" + url['href'])

topic_urls[:5]
```

[142]:
```
['https://github.com/topics/3d',
 'https://github.com/topics/ajax',
 'https://github.com/topics/algorithm',
 'https://github.com/topics/amphp',
 'https://github.com/topics/android']
```

[143]:
```
!pip install pandas --upgrade --quiet
import pandas as pd
```

[144]:
```
topics_dict = {
    'Title': topic_titles,
    'Description': topic_descriptions,
    'URL': topic_urls
}
```

[145]:
```
topics_df = pd.DataFrame(topics_dict)
```

[28]:
```
topics_df.to_csv('topics.csv')
```

5 Getting information out of a topic page

[162]:
```
# topic_page_url = topic_urls[1]
```

[163]:
```
# topic_page_url
```

[164]:
```
# response = requests.get(topic_page_url)
# len(response.text)
```

[165]:
```
# topic_doc = bs(response.text, 'html.parser')
```

[167]:
```
len(topic_urls) - 1
```

[167]: 29

[171]:
```
for x in range(len(topic_urls)):
    topic_page_url = topic_urls[x]
    response = requests.get(topic_page_url)
    topic_doc = bs(response.text, 'html.parser')

    repo_tags = topic_doc.find_all('h1', {'class': 'f3 color-text-secondary␣
 ↪text-normal lh-condensed'})
    # repo_tags[1].find_all('a')[0].text.strip()

    username_list = []
    for i in range(len(repo_tags) - 1):
        username_list.append(repo_tags[i].find_all('a')[0].text.strip())

    project_name_list = []
    for i in range(len(repo_tags) - 1):
        project_name_list.append(repo_tags[i].find_all('a')[1].text.strip())

    star_tags = topic_doc.find_all('a', {'class': 'social-count float-none'})
    stars_list = []
    for i in range(len(star_tags) - 1):
        stars_list.append(int(float(star_tags[i].text.strip()[:-1]) * 1000))
```

```
    # GETTING URLs OF PROJECTS FROM TOPICS_PAGE
    project_url_list = []
    project_url_tags = topic_doc.find_all('a', {'class': 'text-bold'})

    for i in range(len(project_url_tags) - 1):
        project_url_list.append("https://github.com" +␣
↪project_url_tags[i]['href'])

    # CREATING DATAFRAME AND CSV FILE

    topics_dict = {
    'Username': username_list,
    'Project Name': project_name_list,
    'Stars': stars_list,
    'Project URL': project_url_list
    }

    dataFrame = pd.DataFrame(topics_dict)
    dataFrame.to_csv('{}.csv'.format(topic_titles[x]))

# LIST OF VARIABLES OBTAINED
# project_name_list
# username_list
# project_url_list
```

[ ]:
```
import jovian
jovian.commit()
```

```
<IPython.core.display.Javascript object>
```