CIS 571: WEB SERVICES

Project - II

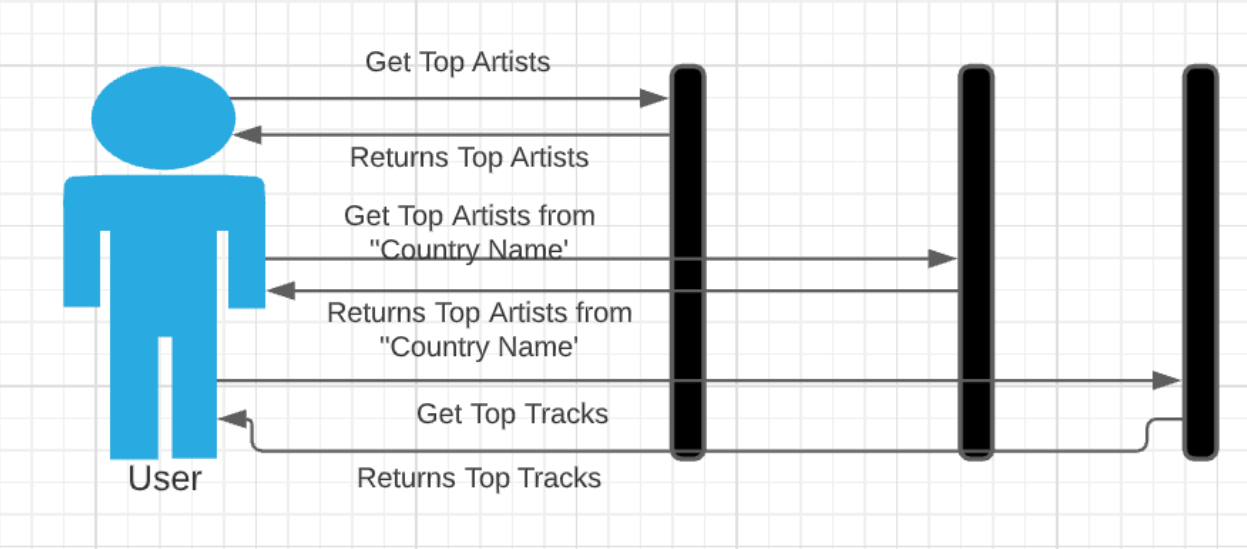
Demonstration of REST and SOAP API

Seema Sharanappa Kanaje

https://drive.google.com/file/d/1KAsT4xsYpFC3wqSWdtI5VaCaq2LHYdht/view?usp=sharing

REST API:

UML Diagram :



An application is built to demonstrate the REST API. I have built an application that would display top artists in the world or from a particular country and top tracks in the world. I have used Python as programming language. I am using REST API's from last.fm.

I have called 3 REST API's in my Web application

1) The "chart.gettopartists" is an API which returns top artists in the world.

   I ask the user to enter the number of artists they want to see. Number of artists is passed as an argument to "limit". API returns the those many numbers of top artists. Data is returned in JSON Format. I traverse through JSON format and extract the important information and put it in a table using PrettyTable library.

```
payload = {
        'method': 'chart.gettopartists',
        'limit': no_of_topartist,
        'page': 1,
        'api_key':'0520c721e4621fe75f7c29514462a669',
        'format':'json'
}

response = requests.get(url, headers=headers, params=payload)
```

Output:

```
Please chose
 1 to Top artists in the world
 2 to Top artists based on the country
 3 to get top tracks1
Please enter no of top artist you want to see10
The top artists are:
+----------------+-----------+-----------+-----------------------------------------+
| Artist's Name  | Playcount | Listeners |                   URL                   |
+----------------+-----------+-----------+-----------------------------------------+
|   The Weeknd   | 162141715 |  1879385  |   https://www.last.fm/music/The+Weeknd  |
| Ariana Grande  | 228442359 |  1622138  | https://www.last.fm/music/Ariana+Grande |
|  Taylor Swift  | 376255824 |  2720180  |  https://www.last.fm/music/Taylor+Swift |
|   Kanye West   | 342703011 |  4835266  |   https://www.last.fm/music/Kanye+West  |
| Billie Eilish  |  88022399 |  1091980  | https://www.last.fm/music/Billie+Eilish |
|    Dua Lipa    |  84237539 |  1016848  |    https://www.last.fm/music/Dua+Lipa   |
|     Drake      | 199127661 |  3823099  |     https://www.last.fm/music/Drake     |
|   Lady Gaga    | 372532786 |  4273757  |   https://www.last.fm/music/Lady+Gaga   |
|    Rihanna     | 231003194 |  4996669  |    https://www.last.fm/music/Rihanna    |
| Kendrick Lamar | 150324252 |  1885284  | https://www.last.fm/music/Kendrick+Lamar |
+----------------+-----------+-----------+-----------------------------------------+
```

2) The “geo.getTopArtists” is an API which returns top artists from the particular country.

I take the input from the user for the country name. I have set limit to 15, so API returns top 15 artists of that country. Data is returned in JSON Format. I traverse through JSON format and extract the important information and put it in a table using PrettyTable library.

```
geodetails = {
        'method': 'geo.getTopArtists',
        'country':geo_topartists ,
        'limit':15,
        'page': 1,
        'api_key':'0520c721e4621fe75f7c29514462a669',
        'format':'json'
}

geo_response = requests.get(url_geo, headers=headers_geo, params=geodetails)
#print(geo_response.text)
return geo_response
```

Output: Here Top 15 artists from Spain are returned.

```
Please chose
 1 to Top artists in the world
 2 to Top artists based on the country
 3 to get top tracks2
Please enter country name to see top artists over thereSpain
+----------------------+-----------+-------------------------------------------------+
|     Artists' Name    | Listeners |                       URL                       |
+----------------------+-----------+-------------------------------------------------+
|      David Bowie     |  3649080  |      https://www.last.fm/music/David+Bowie      |
|       Radiohead      |  5050146  |       https://www.last.fm/music/Radiohead       |
|         Queen        |  4410122  |         https://www.last.fm/music/Queen         |
|        Coldplay      |  5764412  |        https://www.last.fm/music/Coldplay       |
|   The Rolling Stones |  4066735  |   https://www.last.fm/music/The+Rolling+Stones  |
|      The Beatles     |  4037257  |      https://www.last.fm/music/The+Beatles      |
|          Muse        |  4301928  |          https://www.last.fm/music/Muse         |
|       Daft Punk      |  4097186  |       https://www.last.fm/music/Daft+Punk       |
|        The Cure      |  3200703  |        https://www.last.fm/music/The+Cure       |
|     Arctic Monkeys   |  3879005  |     https://www.last.fm/music/Arctic+Monkeys    |
|          Blur        |  3077919  |          https://www.last.fm/music/Blur         |
| Red Hot Chili Peppers |  4898921  | https://www.last.fm/music/Red+Hot+Chili+Peppers |
|        Nirvana       |  4596950  |        https://www.last.fm/music/Nirvana        |
|       The Weeknd     |  1879385  |       https://www.last.fm/music/The+Weeknd      |
|       Ed Sheeran     |  2028642  |       https://www.last.fm/music/Ed+Sheeran      |
+----------------------+-----------+-------------------------------------------------+
```

3)The “chart.getTopTracks” is an API which returns top tracks in the world.

I take the input from the user for the number of tracks. API returns number of tracks specified by the user. Data is returned in JSON Format. I traverse through JSON format and extract the important information and put it in a table using PrettyTable library.

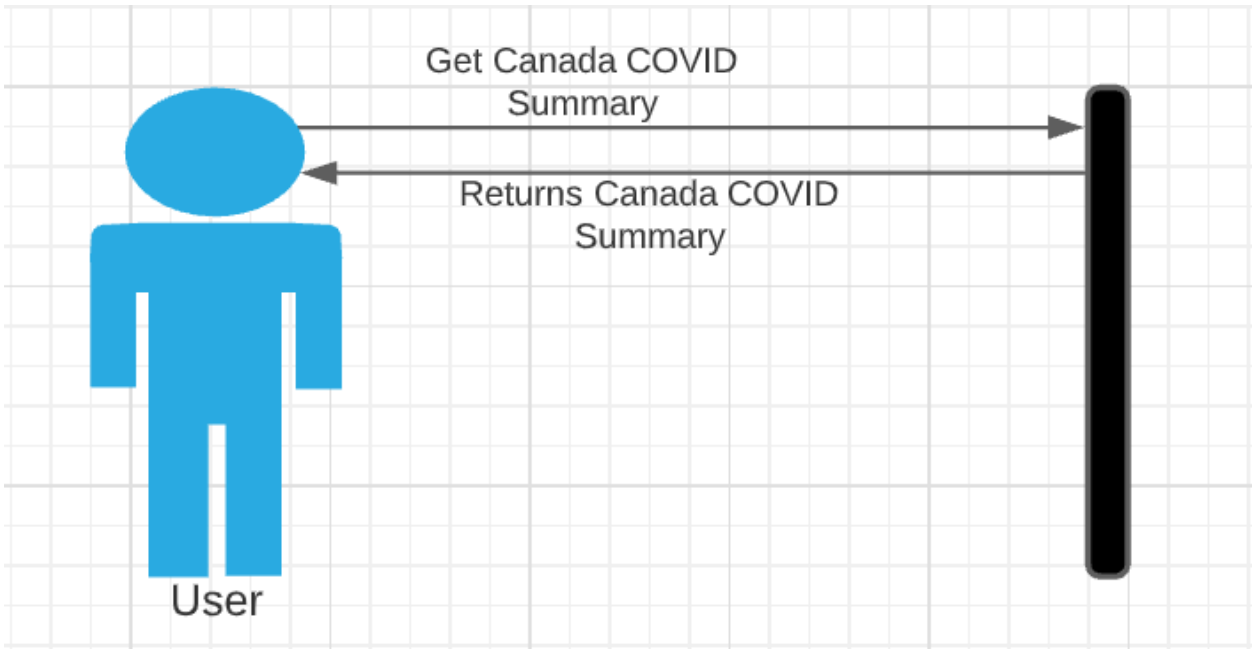
```
get_toptrack():
headers_toptrack = {'user-agent': 'skanaje'}
url_toptrack = 'http://ws.audioscrobbler.com/2.0/'
no_of_toptrack= int(input("Please enter no of top tracks you want to see"))
toptrack = {
    'method': 'chart.getTopTracks',
    'limit': no_of_toptrack,
    'page': 1,
    'api_key': '0520c721e4621fe75f7c29514462a669',
    'format': 'json'
}

toptrack_response = requests.get(url_toptrack, headers=headers_toptrack, params=toptrack)
#print(geo_response.text)
return toptrack_response
```

Output:

```
Please chose
 1 to Top artists in the world
 2 to Top artists based on the country
 3 to get top tracks3
Please enter no of top tracks you want to see10
+------------------------------+-----------+-----------+-----------------------------------------------------------------------------------+
|          Track's Name        | Playcount | Listeners |                                        URL                                        |
+------------------------------+-----------+-----------+-----------------------------------------------------------------------------------+
|       Blinding Lights        |  11570049 |   672742  |           https://www.last.fm/music/The+Weeknd/_/Blinding+Lights                  |
|       drivers license        |  4479374  |   271926  |         https://www.last.fm/music/Olivia+Rodrigo/_/drivers+license                |
| Love Story (Taylor's Version) |  1684308  |   154291  | https://www.last.fm/music/Taylor+Swift/_/Love+Story+(Taylor%E2%80%99s+Version)     |
|       Save Your Tears        |  3563753  |   320026  |           https://www.last.fm/music/The+Weeknd/_/Save+Your+Tears                  |
|          Positions           |  7075640  |   314472  |            https://www.last.fm/music/Ariana+Grande/_/Positions                    |
|          Good Days           |  2306871  |   199703  |                 https://www.last.fm/music/SZA/_/Good+Days                         |
|          telepatía           |  1567362  |   134845  |           https://www.last.fm/music/Kali+Uchis/_/telepat%C3%ADa                   |
|            34+35             |  3460905  |   250574  |            https://www.last.fm/music/Ariana+Grande/_/34%252B35                    |
|           Streets            |  2234871  |   184789  |               https://www.last.fm/music/Doja+Cat/_/Streets                        |
|          We're Good          |   743895  |   105291  |             https://www.last.fm/music/Dua+Lipa/_/We%27re+Good                     |
+------------------------------+-----------+-----------+-----------------------------------------------------------------------------------+
```

SOAP API:



An application is built to demonstrate the SOAP API. This is the API service for the COVID-19 Tracker Canada. I am displaying the results from this API which was returned in JSON format. I have used Python as programming language. I am using SOAP API's from POSTMAN.

```
url = "https://api.covid19tracker.ca/summary"

payload = {'signUp.email': 'skanaje@umich.edu'}
headers = {
'content-type': "text/xml"
}

response = requests.request("GET", url, data=payload, headers=headers)
```

Output:

Canandian COVID Data	Results
latest_date	2021-03-10
change_cases	3155
change_fatalities	26
change_tests	113260
change_hospitalizations	-20
change_criticals	-8
change_recoveries	2930
change_vaccinations	78036
change_vaccinated	4451
change_vaccines_distributed	586410
total_cases	901939
total_fatalities	22346
total_tests	25819304
total_hospitalizations	2066
total_criticals	550
total_recoveries	849365
total_vaccinations	2621289
total_vaccinated	583483
total_vaccines_distributed	3684390

Code- REST API:

```
import json
import requests
from prettytable import PrettyTable

x = PrettyTable()
y=PrettyTable()
z=PrettyTable()
x.field_names = ["Artist's Name", "Playcount", "Listeners", "URL"]
y.field_names=["Artists' Name","Listeners","URL"]
z.field_names = ["Track's Name", "Playcount", "Listeners", "URL"]


def get_topartists():
    headers = {'user-agent': 'skanaje'}
    url = 'http://ws.audioscrobbler.com/2.0/'
    try:
        no_of_topartist=int(input("Please enter no of top artist you want to
see"))
    except:
        print("Wrong Input, Please enter correct input")
        exit()

    # Add API key and format to the payload

    payload = {
            'method': 'chart.gettopartists',
            'limit': no_of_topartist,
            'page': 1,
            'api_key':'0520c721e4621fe75f7c29514462a669',
            'format':'json'
    }

    response = requests.get(url, headers=headers, params=payload)
    #print(response)
    return response

def get_location_topartists():
    headers_geo= {'user-agent': 'skanaje'}
    url_geo= 'http://ws.audioscrobbler.com/2.0/'

    geo_topartists=input("Please enter country name to see top artists over
there")

    geodetails = {
            'method': 'geo.getTopArtists',
            'country':geo_topartists ,
            'limit':15,
            'page': 1,
            'api_key':'0520c721e4621fe75f7c29514462a669',
            'format':'json'
    }

    geo_response = requests.get(url_geo, headers=headers_geo,
params=geodetails)
```

```
    #print(geo_response.text)
    return geo_response

def get_toptrack():
    headers_toptrack = {'user-agent': 'skanaje'}
    url_toptrack = 'http://ws.audioscrobbler.com/2.0/'
    no_of_toptrack= int(input("Please enter no of top tracks you want to
see"))
    toptrack = {
        'method': 'chart.getTopTracks',
        'limit': no_of_toptrack,
        'page': 1,
        'api_key': '0520c721e4621fe75f7c29514462a669',
        'format': 'json'
    }

    toptrack_response = requests.get(url_toptrack, headers=headers_toptrack,
params=toptrack)
    #print(geo_response.text)
    return toptrack_response

def read_json(response,choice):
    json_string = response.text
    #print(response.text)
    json_format = json.loads(json_string)
    if(choice==1):

        print("The top artists are: ")
        for i in json_format:
         for j in json_format[i]:
            #print(j)
            for k in json_format[i][j]:
                try:
                    #print(k['name']+" "+k['playcount']+" "+k["listeners"]+"
"+k["url"])
                    x.add_row([k['name'], k['playcount'], k["listeners"],
k["url"]])
                except:
                    pass
        print(x)
    elif choice==2:
        for i in json_format:
            for j in json_format[i]:
                #print(json_format[i][j])
                for k in json_format[i][j]:
                    try:
                        #print(k['name']+" "+k["listeners"]+" "+k["url"])
                        y.add_row([k['name'], k["listeners"], k["url"]])
                    except:
                        pass
        print(y)
    elif choice==3:
        for i in json_format:
            for j in json_format[i]:
                for k in json_format[i][j]:
                    try:
                        #print(k['name']+" "+k["playcount"]+"
```

```
"+k["listeners"]+" "+k["url"])
                        z.add_row([k['name'], k['playcount'], k["listeners"],
k["url"]])
                    except:
                        pass
        print(z)

responses = []
# API call
choice=int(input("Please chose \n 1 to Top artists in the world \n 2 to Top
artists based on the country \n 3 to get top tracks"))
if (choice==1):
    response = get_topartists()
    if response.status_code != 200:
        print(response.text)
    read_json(response,choice)
elif(choice==2):
    geo_response = get_location_topartists()
    if geo_response.status_code != 200:
        print(geo_response.text)
    read_json(geo_response,choice)
elif(choice==3):
    toptrack_response=get_toptrack()
    if toptrack_response.status_code != 200:
        print(toptrack_response.text)
    read_json(toptrack_response,choice)
else:
    print("Invalid Choice, Please enter correct Input ")
```

CODE- SOAP API:

```
import json
import requests
from prettytable import PrettyTable

x = PrettyTable()
x.field_names = ["Canandian COVID Data", "Results"]

url = "https://api.covid19tracker.ca/summary"

payload = { 'signUp.email': 'skanaje@umich.edu'}
headers = {
'content-type': "text/xml"
}

response = requests.request("GET", url, data=payload, headers=headers)

#print(response.text)
json_format = json.loads(response.text)
#print(json_format)

for i in json_format:
   #print(i)
   if(i=='data'):
    #print(json_format[i][0])
    for j in json_format[i][0]:
        #print(j+" "+json_format[i][0][j])
        x.add_row([j,json_format[i][0][j]])
print(x)
```