ASSIGNMENT 12

Data Service for World Bank Economic Indicators

In this assignment, you are asked to develop a Flask-Restplus data service that allows a client to read and store some publicly available economic indicator data for countries around the world, and allow the consumers to access the data through a REST API.

IMPORTANT *: For this assignment , you will be asked to access the given Web content programmatically. Some Web hosts do not allow their web pages to be accessed programmatically, some hosts may block your IP if you access their pages too many times. During the implementation, download a few test pages and access the content locally - try not to perform too many programmatically.*

- The source URL: http://api.worldbank.org/v2/
- Documentations on API Call Structure: https://datahelpdesk.worldbank.org/knowledgebase/articles/898581-api-basic-call-structure

The World Bank indicators API provides 50 year of data over more than 1500 indicators for countries around the world.

- List of indicators can be found at: http://api.w o rldbank.org/v2/indicators
- List of countries can be found at: http://api.worldbank.org/v2/countries

As the documentation shows , you can construct URLs specifying different parameters to acquire necessary data. Use a custom URL to get information about a specific indicator. For example, the data on the GDP of all countries from 2012 to 2017 can be acquired via this URL: http://api.worldbank.org/v2/countries/all/indicators/NY.GDP.MKTP.CD?date=2012:2017&format=json&per_page=1000

For this assignment we are interested in annual values of a user specified indicator from 2012 to 2017 for one or all countries. The content of the file is quite straightforward. The data service specification will ask you to 'import' this data into a 'data storage'. You should inspect the content of the file carefully and decide on a data model and storage method. You will find that a JSON format is suitable to accessing data and publishing it.

Assignment Specification

The data service should use JSON format to publish its data and implement following operations.

Question-1: Import a collection from the data service (2.5 marks)

This operation can be considered as an on-demand 'import' operation. The service will download the JSON data for all countries respective to the year 2012 to 2017 and identified by the indicator id given by the user and process the content into an internal data format and store it in the database; use **sqlite** for storing the data (the name of the database should be **YOURZID.db**) .
For example, the following link can be used to obtain data for the indicator called: **NY.GDP.MKTP.CD**
http://api.worldbank.org/v2/countries/all/indicators/ **NY.GDP.MKTP.CD** ?date=2012:2017&format=json&per_page=1000
To import data your require an indicator as a parameter given by the API user. The parameter should be a query parameter named:

- **indicator_id** : an indicator http://api.worldbank.org/v2/indicators

After importing the collection, the service should return a response containing at least the following information:

- **uri** : the URL with which the imported collection can be retrieved
- **id** : a unique integer identifier automatically generated
- **creation_time** : the time the collection stored in the database

Example:

```
HTTP operation: POST /collections?indicator_id=NY.GDP.MKTP.CD
```

Returns: 201 Created

```
{
    "uri" : "/collections/1",
    "id" : 1,
    "creation_time": "2019-04-08T12:06:11Z",
    "indicator_id" : "NY.GDP.MKTP.CD"
}
```

- You should return appropriate responses (JSON formatted) in case of already imported collections, invalid indicators or any invalid attempts to use the endpoint (e.g. If the input indicator id doesn't exist in the data source, return error 404)
- **UPDATE: Please remove those entries where the value is NONE**

Question 2- Deleting a collection with the data service (2.5 marks)

This operation deletes an existing collection from the database. The interface should look like as below:

```
HTTP operation: DELETE /collections/{id}
```

Returns: 200 OK

```
{
    "message" :"The collection 1 was removed from the
database!",
    "id": 1
}
```

Question 3 - Retrieve the list of available collections (2.5 marks)

This operation retrieves all available collections. The interface should look like as like below:

```
HTTP operation: GET /collections?order_by={+id,
+creation_time,+indicator,-id,-creation_time,-indicator}
```

"order_by" is a comma separated string value to sort the collection based on the given criteria. Each segment of this value indicates how the collection should be sorted, and it consists of two parts (+ or -, and the name of column e.g., id). In each segment, + indicates ascending order, and - indicates descending order. Here are some samples:

+id, +creation_time	order by "id ascending" and then "creation_time ascending" In this case sorting by "id" has priority over "creation_time "
-creation_time	order by "creation_time descending"

Returns: 200 OK

```
[
   {
    "uri" : "/collections/1",
    "id" : 1,
    "creation_time": "2019-04-08T12:06:11Z",
    "indicator" : "NY.GDP.MKTP.CD"
    },
   {
    "uri" : "/collections/2",
    "id" : 2,
    "creation_time": "2019-05-08T12:16:11Z",
    "indicator" : "2.0.cov.Math.pl_3.all"
   },
   ...
```

Question 4 - Retrieve a collection (2.5 marks)

This operation retrieves a collection by its ID . The response of this operation will show the imported content from world bank API for all 6 years. That is, the data model that you have designed is visible here inside a JSON entry's content part.
The interface should look like as like below:

```
HTTP operation: GET /collections/{id}
```

Returns: 200 OK

```
{
  "id" : 1,
  "indicator": "NY.GDP.MKTP.CD",
  "indicator_value": "GDP (current US$)",
  "creation_time" : "2019-04-08T12:06:11Z"
  "entries" : [
                {"country": "Arab World",  "date": 2016,
"value": 2500164034395.78 },
                {"country": "Australia",   "date": 2016,
"value": 780016444034.00 },
                ...
   ]
}
```

Question 5 - Retrieve economic indicator value for given country and a year (2.5 marks)

The interface should look like as like below:

```
HTTP operation: GET /collections/{id}/{year}/{country}
```

Returns: 200 OK

```
{
   "id": 1,
   "indicator" : "NY.GDP.MKTP.CD",
   "country": "Arab World",
   "year": 2016,
   "value": 780016444034.00
}
```

Question 6 - Retrieve top/bottom economic indicator values for a given year
(2.5 marks)

The interface should look like as like below:

```
HTTP operation: GET /collections/{id}/{year}?q=<query>
```

Returns: 200 OK

```
{
   "indicator": "NY.GDP.MKTP.CD",
   "indicator_value": "GDP (current US$)",
   "entries" : [
                  {
                     "country": "Arab World",
                     "value": 2500164034395.78
                  },
                  ...
               ]
}
```

The <query> is an optional integer parameter which can be either of following:

- **+N (or simply N)** : Returns top N countries sorted by indicator value (highest first)
- **-N** : Returns bottom N countries sorted by indicator value

where N can be an integer value between 1 and 100. For example, a request like " GET /collections/1 /2015?q=+10 " should returns top 10 countries according to the indicator value.

Notice:

- Your code must implemented in **flask-restplus** and automatically generate swagger doc for testing the endpoints.
- Your code must be executable in CSE machines
- Your code must not require installing any software (python libraries are fine)
- Your code must be written in Python 3.5 or newer versions.
- Your operations (API methods) must return appropriate responses in JSON format, and appropriate HTTP response code! e.g., **500 Internal Server Error** is inappropriate response!
- Make sure you are using right datatypes in the database and in you API methods (e.g., not string for years '2017')

- Install flask_restplus on cse machines by pip3 command, and make sure you install **pip3 install werkzeug~=0.16.1**

Submission:

- One and only one Python script file named " YOUR_ZID .py" which contains your code
- **How I can submit my assignment?**
- Go to the assignment page click on the "Make Submission" tab; pick your files which must be named "YOUR_ZID.py". Make sure that the files are not empty, and submit the files together.
- **Can I submit my file after deadline?**
- Yes, you can. But 25% of your assignment will be deducted as a late penalty per day. In other words, if you be late for more than 3 days, you will not be marked.