

1- Import a collection from the data service

This operation can be considered as an on-demand 'import' operation. The service will download the JSON data for all countries (http://api.worldbank.org/v2/countries/all/indicators/NY.GDP.MKTP.CD?date=2012:2017&format=json) 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 (mLab).

Parameters should be given to the endpoint (in the payload) by the user:

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

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

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

Example:

HTTP operation: POST /<collections>

Input Payload:

{ "indicator_id" : "NY.GDP.MKTP.CD" }

Returns: 201 Created

{
 "location" : "/<collections>/<collection_id>",
 "collection_id" : "<collection_id>",
 "creation_time": "2018-04-08T12:06:11Z",
 "indicator" : "<indicator>"
}

- The return response contains the location of the data entry created as the result of the processing the import. What is shown above is just a suggestion. You can decide what else is appropriate. You do not have to return the actual imported content with POST. Just return the location.
- You should return appropriate responses in case of 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)
- If an input contains a n indicator that already has been imported before, you should still return the location of the data entry - but with status code 200 OK (instead of 20 1 Created).
- A POINT TO PONDER ABOUT: An `asynchronous POST'?? If a POST takes too long, you may not want the client to wait. What you would do? You do not need to address this in the assignment.
- The source API has pagination; in order to get all of data you need to send many request to import a single collection; however, you are required to get only first **two pages** instead of all:
http://api.worldbank.org/v2/countries/all/indicators/NY.GDP.MKTP.CD?date=2012:2017&format=json&page=2
(http://api.worldbank.org/v2/countries/all/indicators/NY.GDP.MKTP.CD?date=2012:2017&format=json&page=2)
- The data entries inside the collection must be converted as described below:

Data entry conversion:

Here is an example of source data entry as it is in the source API
(http://api.worldbank.org/v2/countries/all/indicators/NY.GDP.MKTP.CD?date=2012:2017&format=json) :

{
 "indicator": {
 "id": "NY.GDP.MKTP.CD",
 "value": "GDP (current US\$)"
 },
 "country": {
 "id": "1A",
 "value": "Arab World"
 },
 "countryiso3code": "",
 "date": "2016",
 "value": 2500164034395.78,
 "unit": "",
 "obs_status": "",
 "decimal": 0
}

However, you do not need to store all of its attributes; instead convert it to a JSON format as below:

{
 "country": "Arab World",
 "date": "2016",
 "value": 2500164034395.78
}

And as a result a collection should be formatted and stored in the database (mLab) as follow:

{
 "collection_id" : "<collection_id>",
 "indicator": "NY.GDP.MKTP.CD",
 "indicator_value": "GDP (current US\$)",
 "creation_time" : "<creation_time>"
 "entries" : [
 { "country": "Arab World", "date": "2016", "value": 2500164034395.78 },
 { "country": "Australia", "date": "2013", "value": 780016444034.00 },
 ...
]
}

2- Deleting a collection with the data service

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

HTTP operation: DELETE /<collections>/{collection_id}

Returns: 200 OK

{
 "message" : "Collection = <collection_id> is removed from the database!"
}

3 - Retrieve the list of available collections

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

HTTP operation: GET /<collections>

Returns: 200 OK

[
 {
 "location" : "/<collections>/<collection_id_1>",
 "collection_id" : "collection_id_1",
 "creation_time": "<time>",
 "indicator" : "<indicator>"
 },
 {
 "location" : "/<collections>/<collection_id_2>",
 "collection_id" : "collection_id_2",
 "creation_time": "<time>",
 "indicator" : "<indicator>"
 },
 ...
]

4 - Retrieve a collection

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>/{collection_id}

Returns: 200 OK

{
 "collection_id" : "<collection_id>",
 "indicator": "NY.GDP.MKTP.CD",
 "indicator_value": "GDP (current US\$)",
 "creation_time" : "<creation_time>"
 "entries" : [
 { "country": "Arab World", "date": "2016", "value": 2500164034395.78 },
 { "country": "Australia", "date": "2016", "value": 780016444034.00 },
 ...
]
}

5 - Retrieve economic indicator value for given country and a year

The interface should look like as like below:

HTTP operation: GET /<collections>/{collection_id}/{year}/{country}

Returns: 200 OK

{
 "collection_id": <collection_id>,
 "indicator" : "<indicator_id>",
 "country": "<country>,"
 "year": "<year>,"
 "value": <indicator_value_for_the_country>
}

6 - Retrieve top/bottom economic indicator values for a given year

The interface should look like as like below:

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

Returns: 200 OK

{
 "indicator": "NY.GDP.MKTP.CD",
 "indicator_value": "GDP (current US\$)",
 "entries" : [
 {
 "country": "Arab World",
 "date": "2016",
 "value": 2500164034395.78
 },
 ...
]
}

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

- **top<N>** : Return top N countries sorted by indicator value
- **bottom<N>** : Return 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>/< id>/2015?query=top10 " should returns top 10 countries according to the collection_id.

Updates: