Table of Contents

[Knowledge Check Test 2](#_Toc153979819)

[Web Service and UI 2](#_Toc153979820)

[Data structure 3](#_Toc153979821)

[Task (IMF.WEO) – Solution structure 3](#_Toc153979822)

[Development.API 3](#_Toc153979823)

[Development.UI 4](#_Toc153979824)

[Run the task 5](#_Toc153979825)

[Backend 5](#_Toc153979826)

[Frontend 5](#_Toc153979827)

[To be implemented 5](#_Toc153979828)

[References 6](#_Toc153979829)

# Knowledge Check Test

The multidimensional data model is an integral part of On-Line Analytical Processing, or OLAP.

Cube

Dimensions

Measures

Dimension Attributes

Levels

Hierarchies

Figure 1Diagram of the Logical Multidimensional Model

## Web Service and UI

Based on macroeconomics data IMF WEO (the example is attached) create an OLAP

● The data should be stored within the database (ex. MS SQL)

● Create a web service on C# .NET Core or .NET Framework

○ The service should read the data out of the database

○ The service should process the data into the multidimensional cube

○ The service should serve API requests and return data in a JSON format

● Create a UI using TypeScript (you can also use any frameworks like React, Angular,

Vue, etc.)

○ UI should be able to fetch the data via API from the Web Service

○ UI should visualize the data as a table

○ UI should have controls for dimensions so the user can change the selected

elements (ex. change the country)

■ Once the selection is changed UI should get new data from the API based

on the user’s selection

○ (Optional) UI to have an ability to swap dimensions across rows and columns and

move some of the dimensions to filters

## Data structure

The data has two dimensions:

● Countries

○ Countries dimensions has two attributes (the name and the code)

● Indicators

○ Indicators dimensions has two attributes (the name and the code)

The data itself are numeric values by date, the frequency of the data is annual data (A) and

quarterly data (Q)

## Task (IMF.WEO) – Solution structure

**Documentation**

**Database Scripts**

**Development.UI**

**Development.API**

**Development.sln**

Figure 2 Solution folder structure

### Development.API

.NET Core Web swagger API provides information about the service, using the generated OpenAPI specification. Available methods:

GET /api/v1/Development/items

GET /api/v1/Development/itemslist

Currently Development.UI uses itemslist method.

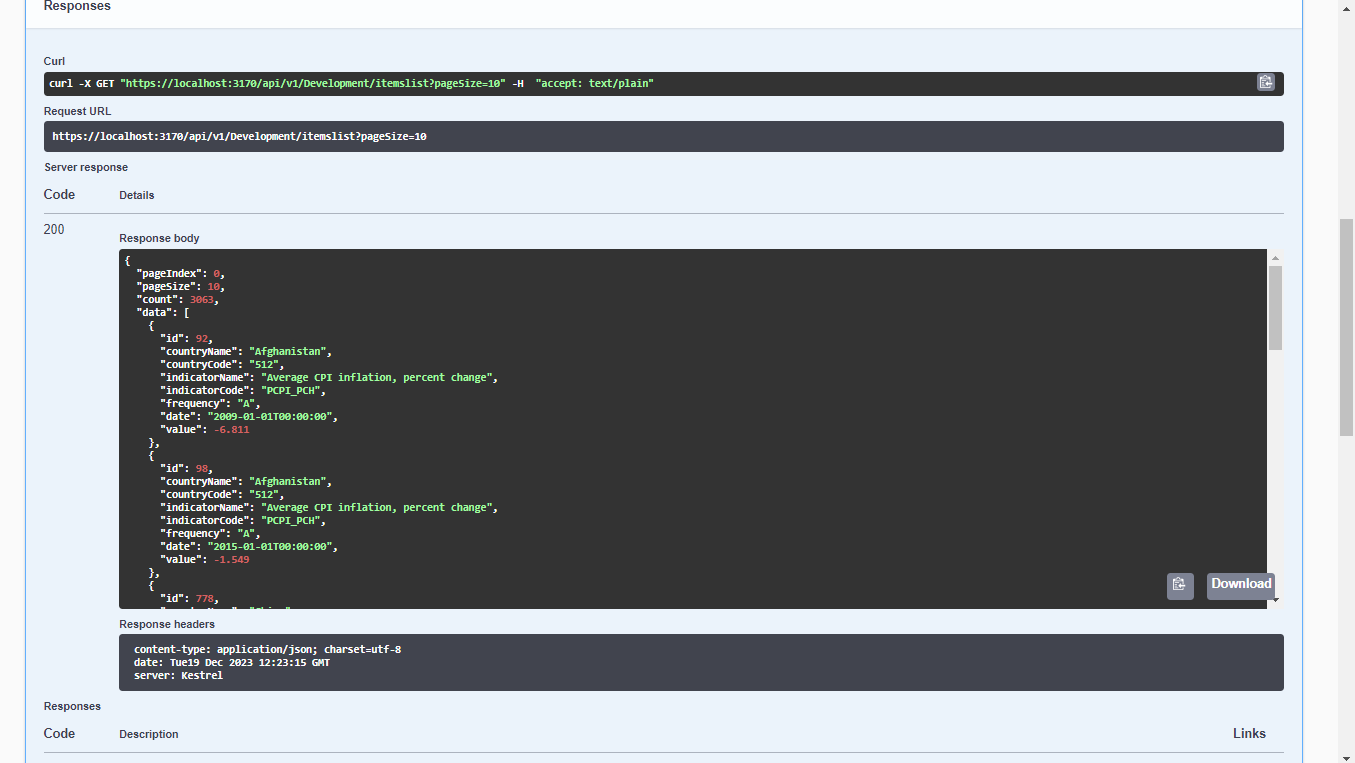


Figure 3 Swagger API

### Development.UI

UI is uses JavaScript and taking an advantage of cube.js lib implemented by Orlov Leonid. <https://github.com/feonit/olap-cube-js>

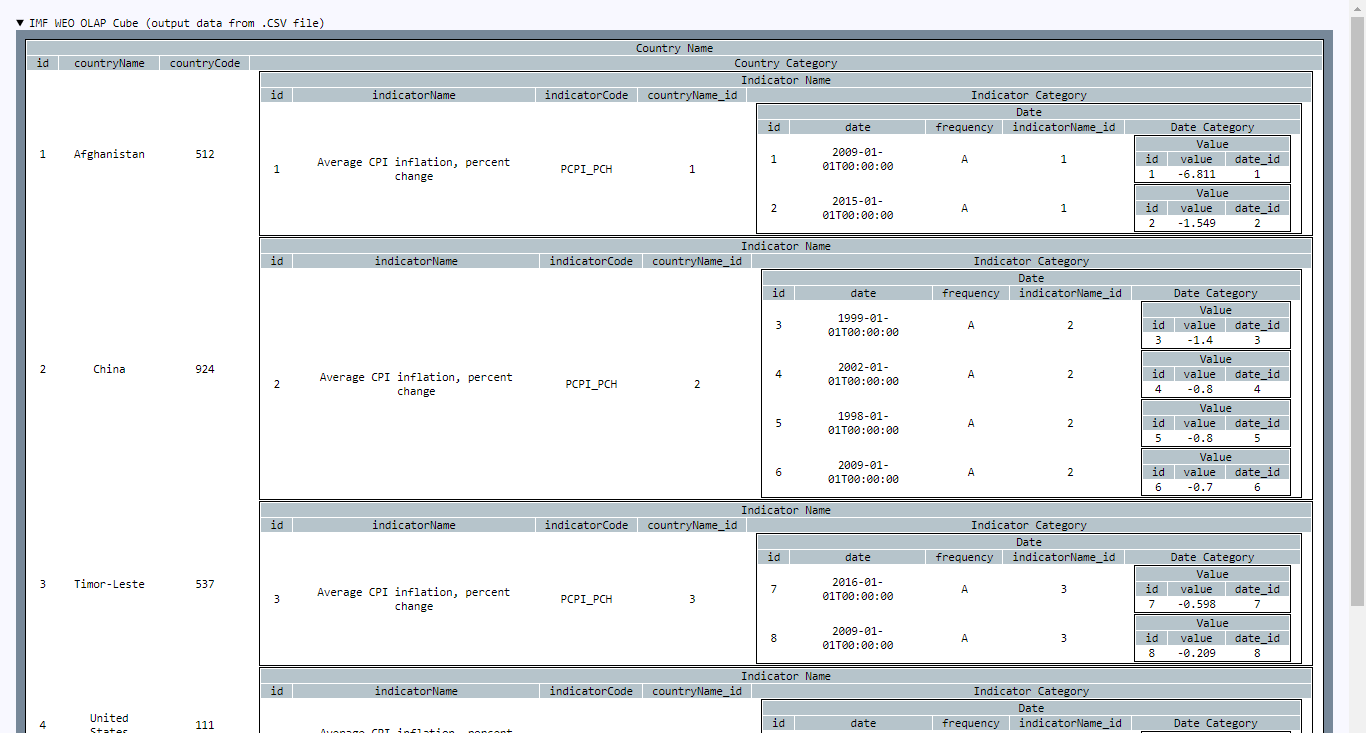


Figure 4 OLAP Table UI

## Run the task

### Backend

Navigate to IMF\Development.API root folder and open your preferred command prompt. Type:

donet run --profile “IMF.WEO”

As a result API will be available at url <https://localhost:3170/swagger/index.html>

### Frontend

W:\Projects\IMF\Development.UI root folder and open your preferred command prompt. Type:

npm install

Once the node modules are in place we can run ui using:

node index.js

As a result UI will be available at url <http://localhost:3175/>

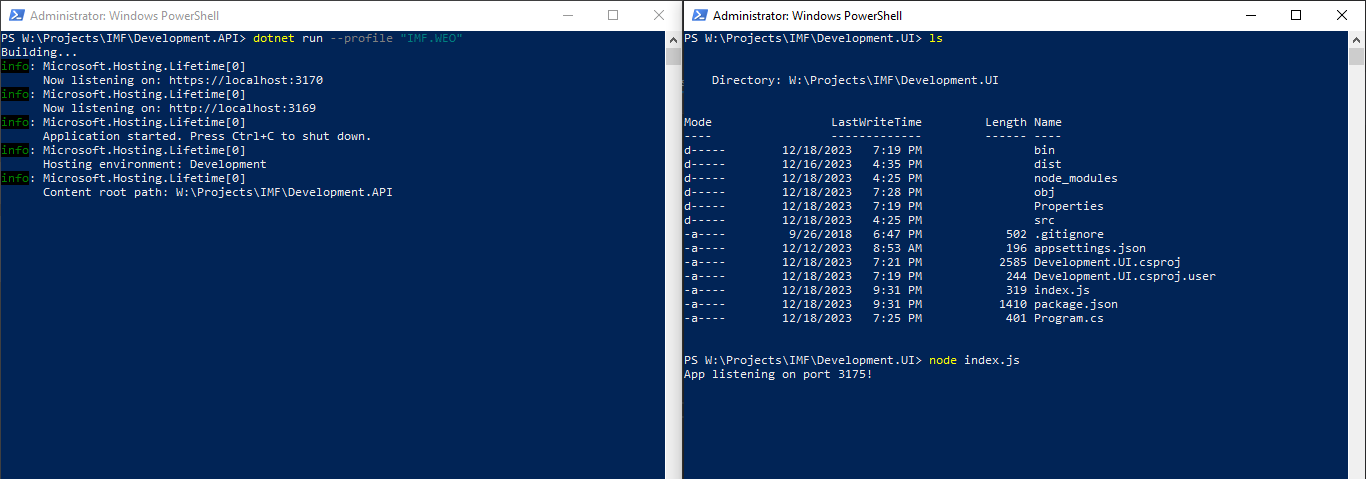
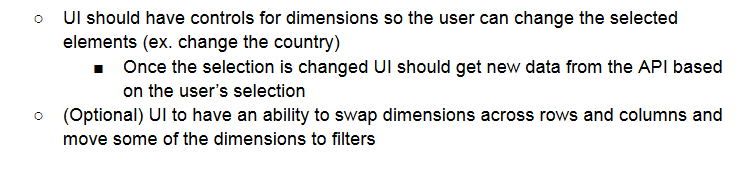


Figure 5 Power shell screenshots

### To be implemented

Still missing certain parts from the requirements.



## References

* <https://en.wikipedia.org/wiki/Online_analytical_processing>
* <https://ru.wikipedia.org/wiki/OLAP-%D0%BA%D1%83%D0%B1>
* <https://help.sap.com/docs/SAP_BUSINESSOBJECTS_ANALYSIS,_EDITION_FOR_OLAP/4ae5a34085c44f4ea2b4cab4d7c43e19/47146f866e041014910aba7db0e91070.html?locale=ru-RU>
* <http://www.olap.ru/basic/multi_dim_DWH.asp>
* <https://github.com/feonit/olap-cube-js>
* <https://docs.oracle.com/cd/B13789_01/olap.101/b10333/multimodel.htm>