Data Processing

# Christof du Toit

24/03/2022

## Goal of project:

The goal of this project was to make an API that could access a database and visualize the data received in both xml and json formats.

## Data Sets:

The datasets that were used were chosen by me and were approved. The data was then imported into a database made using MYSQL on PHPMyAdmin using Xammp, The API was made using Nodejs.

## Reasoning behind designs:

**PUT VS UPDATE:**

I have chosen to use the method of UPDATE in this application as UPDATE provides more control over the data I want to adjust, it also allows for updating of a full object and not only a single field.

**Documentation of the API:**

The documentation provided on the API is all concentrated in the worldGPD\_ file endpoints as all other endpoints are very similar in structure, and thus the comments found on one file can be applied to the others.

## Prerequisites:

To run this project:

1. Please extract files located within this DataProccesing.zip to directory of: C:\xampp\htdocs, this will allow the connection from the application to the local database.
2. Import database by following these steps:
   1. Please start Apache and MySQL in the xampp control panel.
   2. Create a new database called “worlddata” in the local database manager.
   3. Import the worlddata.sql file into your localhost using xampp. (Data in database can now be accessed)

Graphical user interface, application

Description automatically generated

1. Download Nodejs for the desired system: <https://nodejs.org/en/download/> and install.
2. After installation open the command prompt and navigate to the project location using:

cd C:\xampp\htdocs\DataProcessing

1. Run the following commands:
   1. npm install mysql
   2. npm install express –save
   3. npm install nodemon -g
   4. npm install --save is-my-json-valid
   5. npm install express-xml-bodyparser --save
   6. npm install --save xsd-schema-validator
2. Install the latest version of java (jdk): <https://www.oracle.com/java/technologies/downloads/#java18> and save the file location.
3. After this jdk is installed press the windows key and enter “variable” into the serachbar and select the ‘Edit the system environment variables’ option.
4. Click the Environment Variables button.
5. Click New in System variables

Graphical user interface, text, application, email

Description automatically generated

1. Create a variable with the name JAVA\_HOME and as the value select the installed jdk:

Graphical user interface, text, application

Description automatically generated

1. After command npm npm install --save xsd-schema-validator has been run, a JAVA\_HOME variable must be declared. To

## Running API:

After all modules have been installed on the C:\xampp\htdocs\DataProcessing directory, and the JAVA\_HOME variable made, run the command “nodemon app.js”, this will start the API and allow endpoints to be tested in Postman.

## Testing the API:

For ease of testing, I will include the http references of each endpoint and the data so each test can be performed with in xml and json. To test the endpoint please follow these steps:

1. Set the desired Header (application/json or application/xml value) including the Key to Content type.

Graphical user interface, text, application, email

Description automatically generated

1. Paste the correlating data objects inside the body.

Graphical user interface, text, application, email

Description automatically generated

1. Set the body of the request to json or xml (depending on the object).
2. Set the method to be executed. (POST, SELECT, UPDATE, DELETE)
3. Press the Send button.

## Endpoints:

### World GDP:

**Insert:**

<http://localhost:3000/worldGDP_insert/countrygdp>

Data objects:

|  |  |
| --- | --- |
| JSON object | XML object |
| {      "Country" : "Test",      "Country\_Code": "Test",      "GDP\_2000": 7708.1,      "Suicide\_Rate\_2000": 9.5,      "GDP\_2005": 5109.85,      "Suicide\_Rate\_2005": 9,      "GDP\_2010": 10386,      "Suicide\_Rate\_2010": 8.7,      "GDP\_2015": 13789.1,      "Suicide\_Rate\_2015": 8.8,      "GDP\_2016": 12790.2,      "Suicide\_Rate\_2016": 9.2  } | <Country>      <Country>Test</Country>      <Country\_Code>Test</Country\_Code>      <GDP\_2000>39921833.0</GDP\_2000>      <Suicide\_Rate\_2000>2766890</Suicide\_Rate\_2000>      <GDP\_2005>14.4</GDP\_2005>      <Suicide\_Rate\_2005>0.18</Suicide\_Rate\_2005>      <GDP\_2010>0.61</GDP\_2010>      <Suicide\_Rate\_2010>15.18</Suicide\_Rate\_2010>      <GDP\_2015>11200</GDP\_2015>      <Suicide\_Rate\_2015>97.1</Suicide\_Rate\_2015>      <GDP\_2016>220.4</GDP\_2016>      <Suicide\_Rate\_2016>12.31</Suicide\_Rate\_2016>  </Country> |

**Select:**

<http://localhost:3000/worldGDP_select/countrygdp>

This endpoint returns all countries withing the database.

**Update:**

<http://localhost:3000/worldGDP_update/updateCountrygdp>

Data objects:

|  |  |
| --- | --- |
| JSON object | XML object |
| {      "Country" : "Test",      "Country\_Code": "Test2",      "GDP\_2000": 7708.1,      "Suicide\_Rate\_2000": 9.5,      "GDP\_2005": 5109.85,      "Suicide\_Rate\_2005": 9,      "GDP\_2010": 10386,      "Suicide\_Rate\_2010": 8.7,      "GDP\_2015": 13789.1,      "Suicide\_Rate\_2015": 8.8,      "GDP\_2016": 12790.2,      "Suicide\_Rate\_2016": 9.2  } | <Country>      <Country>Test</Country>      <Country\_Code>Test2</Country\_Code>      <GDP\_2000>39921833.0</GDP\_2000>      <Suicide\_Rate\_2000>2766890.0</Suicide\_Rate\_2000>      <GDP\_2005>14.4</GDP\_2005>      <Suicide\_Rate\_2005>0.18</Suicide\_Rate\_2005>      <GDP\_2010>0.61</GDP\_2010>      <Suicide\_Rate\_2010>15.18</Suicide\_Rate\_2010>      <GDP\_2015>11200</GDP\_2015>      <Suicide\_Rate\_2015>97.1</Suicide\_Rate\_2015>      <GDP\_2016>220.4</GDP\_2016>      <Suicide\_Rate\_2016>12.31</Suicide\_Rate\_2016>  </Country> |

**Delete:**

<http://localhost:3000/worldGDP_delete/deleteCountrygdp>

Data objects:

|  |  |
| --- | --- |
| JSON object | XML object |
| {      "Country" : "Test"  } | <Country>      <Country>Test</Country>  </Country> |

### World Alcohol:

**Insert:**

<http://localhost:3000/worldHappyness_insert/countryhappyness>

Data objects:

|  |  |
| --- | --- |
| JSON object | XML object |
| {      "Country" : "Test",      "Happyness\_Rank": 7708.1,      "Happyness\_Score": 7708.1,      "Whisker\_high": 9.5,      "Whisker\_low": 5109.85,      "Economy": 9,      "Family": 10386,      "Life\_Expectancy": 8.7,      "Freedom": 13789.1,      "Generosity": 8.8,      "Gov\_Trust": 12790.2,      "Dystopia\_Residual": 9.2  } | <Country>      <Country>Test</Country>      <Happyness\_Rank>1.1</Happyness\_Rank>      <Happyness\_Score>39921833.0</Happyness\_Score>      <Whisker\_high>2766890.0</Whisker\_high>      <Whisker\_low>14.4</Whisker\_low>      <Economy>0.18</Economy>      <Family>0.61</Family>      <Life\_Expectancy>15.18</Life\_Expectancy>      <Freedom>11200</Freedom>      <Generosity>97.1</Generosity>      <Gov\_Trust>220.4</Gov\_Trust>      <Dystopia\_Residual>12.31</Dystopia\_Residual>  </Country> |

**Select:**

<http://localhost:3000/worldAlcohol_select/countryalcohol>

This endpoint returns all countries withing the database.

**Update:**

<http://localhost:3000/worldGDP_update/updateCountrygdp>

Data objects:

|  |  |
| --- | --- |
| JSON object | XML object |
| {      "Country" : "Test",      "Happyness\_Rank": 10000,      "Happyness\_Score": 7708.1,      "Whisker\_high": 9.5,      "Whisker\_low": 5109.85,      "Economy": 9,      "Family": 10386,      "Life\_Expectancy": 8.7,      "Freedom": 13789.1,      "Generosity": 8.8,      "Gov\_Trust": 12790.2,      "Dystopia\_Residual": 9.2  } | <Country>      <Country>Test</Country>      <Happyness\_Rank>10000</Happyness\_Rank>      <Happyness\_Score>39921833.0</Happyness\_Score>      <Whisker\_high>2766890.0</Whisker\_high>      <Whisker\_low>14.4</Whisker\_low>      <Economy>0.18</Economy>      <Family>0.61</Family>      <Life\_Expectancy>15.18</Life\_Expectancy>      <Freedom>11200</Freedom>      <Generosity>97.1</Generosity>      <Gov\_Trust>220.4</Gov\_Trust>      <Dystopia\_Residual>12.31</Dystopia\_Residual>  </Country> |

**Delete:**

<http://localhost:3000/worldAlcohol_delete/deleteCountryalcohol>

Data objects:

|  |  |
| --- | --- |
| JSON object | XML object |
| {      "Country" : "Test"  } | <Country>      <Country>Test</Country>  </Country> |

### World happiness:

**Insert:**

<http://localhost:3000/worldGDP_insert/countrygdp>

Data objects:

|  |  |
| --- | --- |
| JSON object | XML object |
| {      "Country" : "Test",      "beer\_servings": 1456,      "spirit\_servings": 7708.1,      "wine\_servings": 9.5,      "total\_liters\_of\_alcohol": 5109.85  } | <Country>      <Country>Test</Country>      <beer\_servings>1456</beer\_servings>      <spirit\_servings>7708.1</spirit\_servings>      <wine\_servings>9.5</wine\_servings>      <total\_liters\_of\_alcohol>5109.85</total\_liters\_of\_alcohol>  </Country> |

**Select:**

<http://localhost:3000/worldAlcohol_select/countryalcohol>

This endpoint returns all countries withing the database.

**Update:**

<http://localhost:3000/worldAlcohol_update/updateCountryAlcohol>

Data objects:

|  |  |
| --- | --- |
| JSON object | XML object |
| {      "Country" : "Test",      "beer\_servings": 1456,      "spirit\_servings": 7708.1,      "wine\_servings": 9.5,      "total\_liters\_of\_alcohol": 5109.85  } | <Country>      <Country>Test</Country>      <beer\_servings>1456</beer\_servings>      <spirit\_servings>7708.1</spirit\_servings>      <wine\_servings>9.5</wine\_servings>      <total\_liters\_of\_alcohol>5109.85</total\_liters\_of\_alcohol>  </Country> |

**Delete:**

<http://localhost:3000/worldAlcohol_delete/deleteCountryalcohol>

Data objects:

|  |  |
| --- | --- |
| JSON object | XML object |
| {      "Country" : "Test"  } | <Country>      <Country>Test</Country>  </Country> |