Department of Computing
Goldsmiths, University of London

Global and European Union Trade Visualisation

Introduction to Programming - Part II

by

Jonathan K. Y. Tang
Tristan Thomson

April 2018

Submitted in partial fulfillment for the degree of
Bachelor of Science in *Computer Science*

Word Count

1219

computed by `TeXcount`

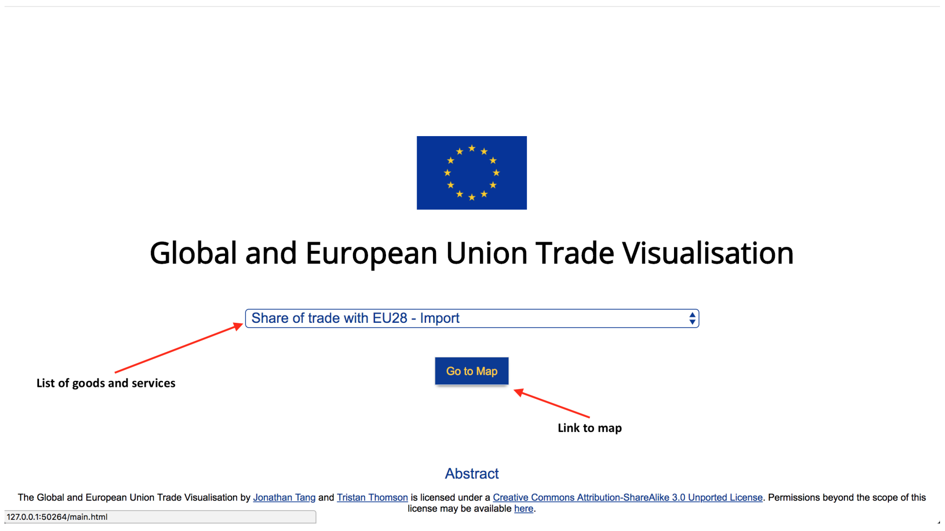
Contents

Chapter 1

Software Features

1.1 Map Visualization

One of the main features of this software is the visualisation of the data on the Google Maps API [2]. On loading of the software, users are directed to a landing page where they can choose the data set to view on the map.


Global and European Union Trade Visualisation
Share of trade with EU28 - Import
List of goods and services
Go to Map
Link to map
Abstract
127.0.0.1:50264/main.html

The Global and European Union Trade Visualisation by Jonathan Tang and Tristan Thomson is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. Permissions beyond the scope of this license may be available here.

Figure 1.1: Landing Page

Users are then directed to the main page where the map is displayed along with all the country's data. Only countries where data is available for, will show up on the map. Users can also see the change in other years by selecting from the drop-down menu, and the map will reload. This is so users can see the change in the trade area over different years.

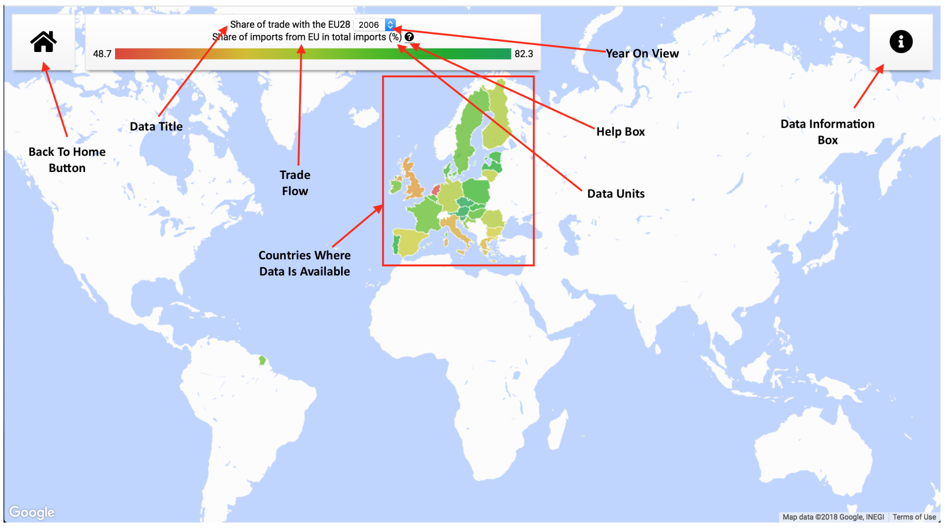

Share of trade with the EU28 2006
Share of imports from EU in total imports (%)
48.7
82.3
Year On View
Help Box
Data Information Box
Data Title
Back To Home Button
Trade Flow
Data Units
Countries Where Data Is Available
Google
Map data ©2018 Google, INEGI Terms of Use

Figure 1.2: Main map view

Each country is filled with a colour depending on where the country's value lies on the percentile.

$$percentile = \frac{value - minimum}{maximum - minimum}$$

The value is then placed on a range of red and green, depending where the percentile is, the country will turn that colour `(line 94-99, maps.js)`. This is so users can see which countries have the greatest or least effect on that area of trade.

When users hover over the country, they can see the country's name along with the true value from the dataset. The percentile value is never shown to the user, only used to plot the point on the colour bar.

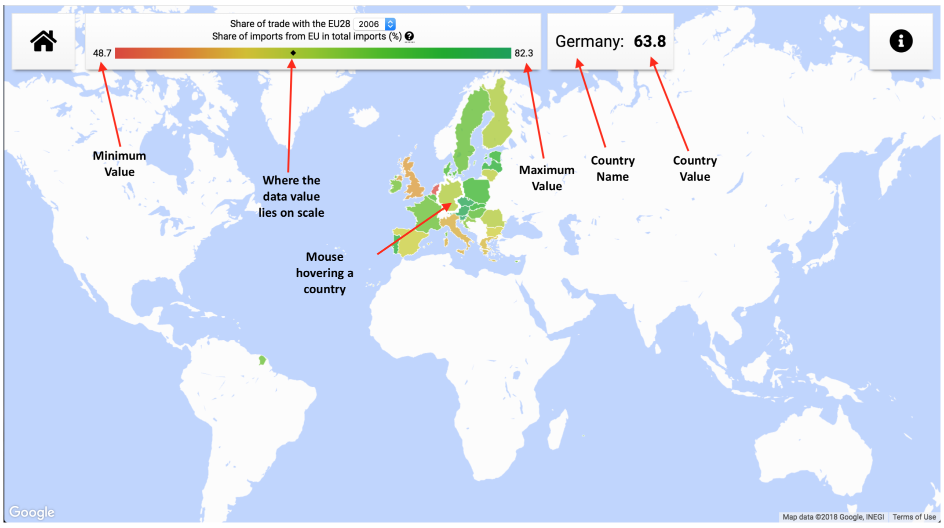
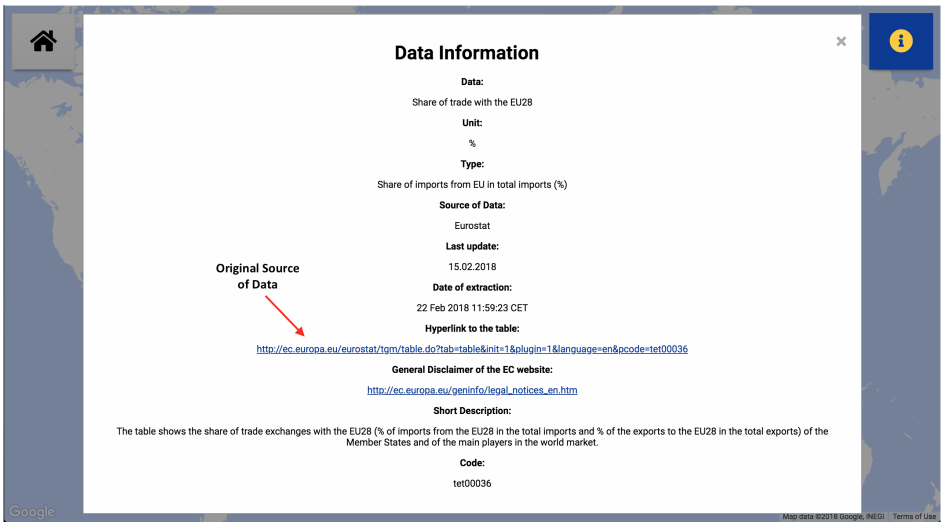

Share of trade with the EU28 2006
Share of imports from EU in total imports (%)
48.7
82.3
Germany: 63.8
Minimum Value
Where the data value lies on scale
Maximum Value
Country Name
Country Value
Mouse hovering a country
Google
Map data ©2018 Google, INEGI Terms of Use

Figure 1.3: Mouse over country

If the user wanted to find out more about the context of the data, original source and when the data was sourced, they can click on the information box in the top-right hand corner, and a modal will appear. Equally, if the user wanted to know how to use the map, then they can hover over the question mark and a tooltip [11] will appear on the screen with instructions.



Data Information

Data:

Share of trade with the EU28

Unit:

%

Type:

Share of imports from EU in total imports (%)

Source of Data:

Eurostat

Last update:

15.02.2018

Date of extraction:

22 Feb 2018 11:59:23 CET

Hyperlink to the table:

http://ec.europa.eu/eurostat/tgm/table.do?tab=table&init=1&plugin=1&language=en&pcode=tet00036

General Disclaimer of the EC website:

http://ec.europa.eu/geninfo/legal_notices_en.htm

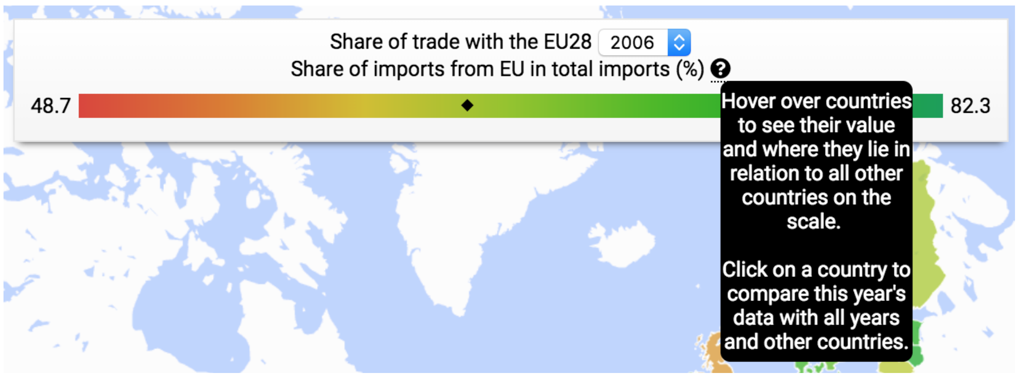
Short Description:

The table shows the share of trade exchanges with the EU28 (% of imports from the EU28 in the total imports and % of the exports to the EU28 in the total exports) of the Member States and of the main players in the world market.

Code:

tet00036

Figure 1.4: Information modal box


Share of trade with the EU28 2006
Share of imports from EU in total imports (%)
48.7
82.3

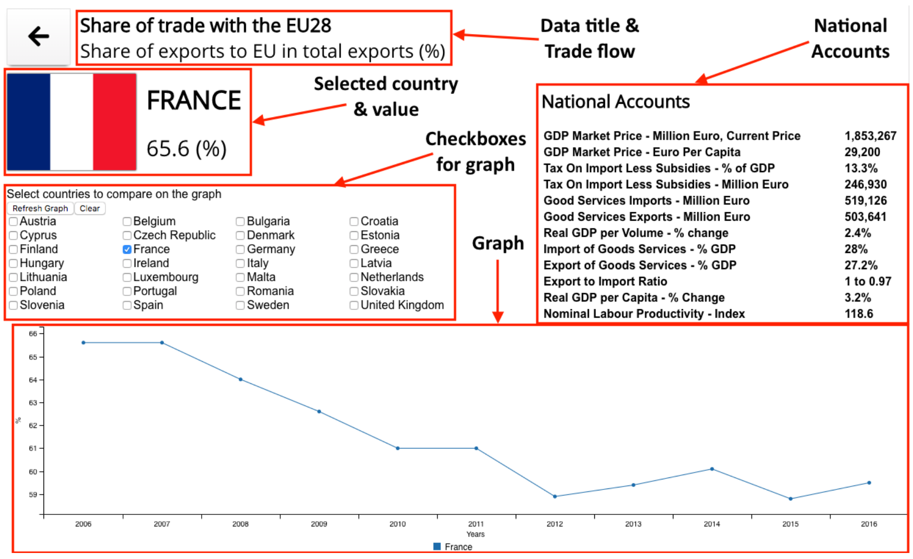
Hover over countries to see their value and where they lie in relation to all other countries on the scale.

Click on a country to compare this year's data with all years and other countries.

Figure 1.5: Help tooltip

1.2 Country Profile

When a country is clicked, users can see a graph of the area of trade across all available years for that country. Users can also compare different countries data with the selected country, and refresh the graph. This is so users can see trends in data, and key trade of certain commodities. National accounts for the selected country, and year are displayed in order to give more context to the data, and how it affects the country's economy.



Share of trade with the EU28
Share of exports to EU in total exports (%)

FRANCE
65.6 (%)

Select countries to compare on the graph

☐ Austria	☐ Belgium	☐ Bulgaria	☐ Croatia
☐ Cyprus	☐ Czech Republic	☐ Denmark	☐ Estonia
☐ Finland	✓ France	☐ Germany	☐ Greece
☐ Hungary	☐ Ireland	☐ Italy	☐ Latvia
☐ Lithuania	☐ Luxembourg	☐ Malta	☐ Netherlands
☐ Poland	☐ Portugal	☐ Romania	☐ Slovakia
☐ Slovenia	☐ Spain	☐ Sweden	☐ United Kingdom

National Accounts

GDP Market Price - Million Euro, Current Price	1,853,267
GDP Market Price - Euro Per Capita	29,200
Tax On Import Less Subsidies - % of GDP	13.3%
Tax On Import Less Subsidies - Million Euro	246,930
Good Services Imports - Million Euro	519,126
Good Services Exports - Million Euro	503,641
Real GDP per Volume - % change	2.4%
Import of Goods Services - % GDP	28%
Export of Goods Services - % GDP	27.2%
Export to Import Ratio	1 to 0.97
Real GDP per Capita - % Change	3.2%
Nominal Labour Productivity - Index	118.6

Figure 1.6: Country Profile

Chapter 2

Implementation and Development

2.1 Data Sourcing and Collection

Most of the initial part of the project was to source all the necessary trade data, and to find the coordinates of each country in order to style the shape of it.

First, we tried to reformat the country coordinates ourselves in order to generate outlines of each country to display on the map. Later, a website was found in order to compile all the coordinates of the world for the project [8]. The only area that was not readily available was the whole of the EU28, and a manual process of combining all member states to generate coordinates for it.

All the trade data and national accounts are from Eurostat [6] [7], and refined the data to make sure it was all consistent for use in implementing the program. So that each data set has a unique reference, `data-overview.csv` lists all data sets as:

Table 2.1: data-overview.csv

Directory	Group No.	File Ref.	Data/Info	Group	Flow
A	01	a	i	Trade in...	Import
A	01	a	ii	Trade in...	Import
...	...	...	...	...	...

Each data set had the data itself followed by footnotes. Using python, each sheet had to be split so that data (Axxy**i**) was on one, and the data information (Axxy**ii**) on another.

2.2 Map View

Constructors were created to get the correct column of the sheet referring to the year in the drop down menu `(lines 51-82, tableData.js)`. There were issues when we tried to pass the data through the API; it was resolved by loading the API to the HTML file synchronously [4]. The customisation of the map is in maps.js where the mapping of the value to the colour scale is done `(lines 92-99, maps.js)`.

Mouse events [3] where the user hovers or clicks on a country is updated, `(lines 126-162, maps.js)` and the country's name, value, and where it lies on the colour scale is shown.

Despite having the data shown on the map, there were details such as the title, units, and others that had to be displayed. Key information was displayed on the data caret. If the user wanted to see a description or data source, a modal box [12] was made, and the user can access them with ease `(lines 36-55, tableInfo.js, lines 77-120, sketchMain.js)`.

2.3 Landing Page

Once the map was displaying one sheet, and so that all 81 sets are not loaded into maps, a landing page was created so users can choose which sheet to load to the map. The drop box is generated from `data-overview.csv`, and when the button is clicked, the unique reference is stored locally [5] on the browser. This was an issue as we were unsure how to pass variables between HTMLs. Once the reference has been passed, the paths are generated in `tableData.js` and `tableInfo.js`.

2.4 Country Profile

The selected country, year, and file path were collected from local storage to pass through the `clickedView.js`, and `countryProfile.js` constructors. The flag code is generated to get the correct ISO 3166-1 alpha-2 code to show the correct flag [9]. All national accounts are searched in `countryProfile.js` where all B-directory (see table 2.1) files are loaded to the constructor, as this was the most efficient way of finding the value according to the selected year, returned as an array. Not all national account files are the same, so implementing a loop to repeat the same process over would produce inconsistencies in the data output.

On research, we came across the c3.js module, and had problems implementing the graph since we could not dynamically change the graph with different inputs, This was solved by loading the callback function `(lines 142-144, sketchClicked.js)` synchronously. The graph data is generated in `graphData.js` where all the ticked countries are passed through the constructor, and returns the data as a row of the sheet. It is then passed through the `lineplotRender.js` with the units (y-axis label), and the available years (x-axis points) [10].

Finally, the check-boxes [1] of all countries from that data set are displayed so users can compare other country, and see any trends in the data. Initially, the program would detect if there were any changes to the division and reload the graph. This severely decreased the performance of the program, and a *refresh graph* button was added, to only reload the graph on the click of the button. Further, trying to uniquely identify each check-box with each country was more complex. Regular expressions were used to catch countries with two or more words, since HTML id tags cannot contain spaces. This issue was to resolve clearing the graph to the selected country, but would leave any two or more worded country checked.

Chapter 3

Software Structure

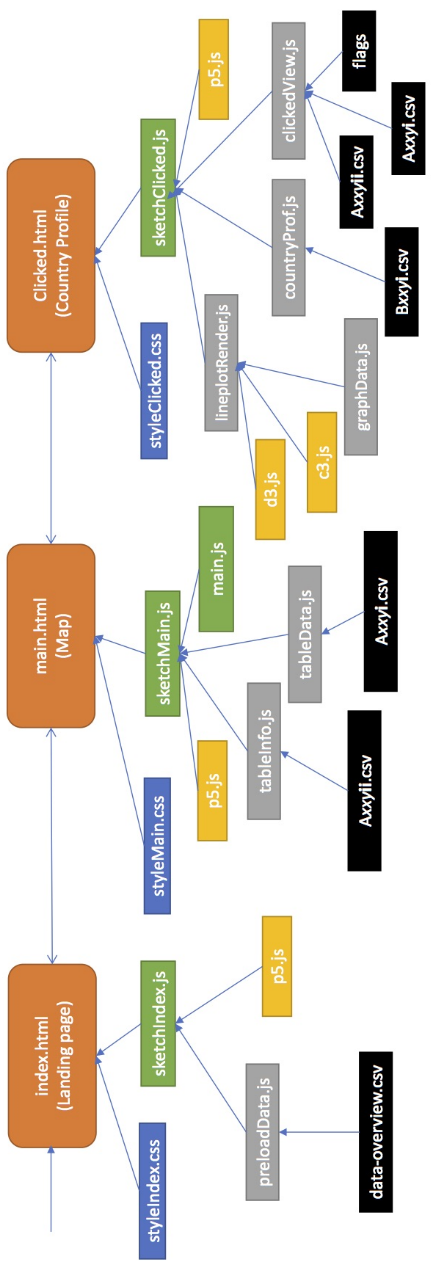


Figure 3.1: Software Structure Overview

Chapter 4

Evaluation

Within the time constraints, we believe the project has been completed to a high standard, and nearly all of the original plan has been achieved; most notably the map being able to visualise all of the datasets consistently. Instead of attempting to implement machine learning, refinement of the different visualisations, and building up the country profile was completed instead.

At first, we underestimated how long it would take to process all the data to ensure it was all consistent, leading to a re-plan of the project in week 4/5.

Further improvements to the project include graphical visualisations to national accounts, and a ”per capita” view, so developed nations do not always show up green on the map.

Overall, the collaborative software development was smooth with a pair programming approach taken on reducing numerous mistakes, though there were times where it was more suitable to program separately. Development was hugely aided with GitLab being used to track issues and store files.

Appendix A

Initial Plan and Timeline

A.1 Project Plan

A.1.1 Abstract

The project aims to visualise global trade data from Eurostat on imports and exports of goods and services within the EU28 member states, along with other key international trade nations. Using Google Maps API and the data from Eurostat, a graphical representation of the data will show as a range of colours determining which countries contribute to certain key areas of EU trade. The trade data is accompanied with national accounts (GDP, tax of goods etc.) along with graphical representation of various other trade.

A.1.2 Map Visualisation

- Map all values of one year for each country
- Display outlines of countries and show correct colours
- Display country's value when mouse is hovered over it
- See more information about the data

A.1.3 Country Profiles

- Show national accounts for the selected country for that year
- Display graph of data for the viewed data set for one country
- Compare different countries to the selected one and visualise trends

- Visualise trends in national accounts

A.1.4 Stretch Goals

- Forecast trade data using linear regression for trade data

A.2 Timeline

1 19 Feb - 25 Feb

- Gathering assets
- General analysis of datasets

2 26 Feb - 4 Mar

- Refinement of datasets

3 5 Mar - 11 Mar

- Build program structure

4 12 Mar - 18 Mar

- Building global trade data on program

5 19 Mar - 25 Mar

- Designing global trade data on program

6 26 Mar - 1 Apr

- Machine learning implementation

7 2 Apr - 8 Apr

- Machine learning implementation

8 9 Apr - 15 Apr

- Report

9 16 Apr - 22 Apr

- Tidy up/Fall back week

Appendix B

Project Wiki

1 19 Feb - 25 Feb

Tasks Completed

- Gathering Eurostat Datasets
- Converted .xls datasets to .csv

Tasks Not Achieved/Problems Encountered

- N/A

Tasks To Be Completed

- Refine datasets
- Find how to import them into p5.js

2 26 Feb - 4 Mar

Tasks Completed

- Gathering Eurostat Datasets
- Started importing datasets for map display
- Started dataset refinement
- CSV Country coordinates collection

Tasks Not Achieved/Problems Encountered

- N/A

Tasks To Be Completed

- Implement Maps API
- Format country coordinates for map use

3 5 Mar - 11 Mar

Tasks Completed

- Started implementing Google Maps API
- Dataset refinement
- Started country coordinates formatting

Tasks Not Achieved/Problems Encountered

- Coordinates not processing in CSV

Tasks To Be Completed

- Finish coordinates formatting

4 12 Mar - 18 Mar

Tasks Completed

- Attempting CSV country coordinates formatting in JavaScript and Python
- Preparing for peer-review

Tasks Not Achieved/Problems Encountered

- Coordinates not processing in CSV

Tasks To Be Completed

- Find a more usable way to overlay country polygons on the map

5 19 Mar - 25 Mar

Tasks Completed

- Understanding the Google Maps API
- Formatted Eurostat datasets using python
- Started using KML country polygons

Tasks Not Achieved/Problems Encountered

- Dataset combining one sheet
- Organising all datasets

Tasks To Be Completed

- Find a more versatile way to overlay country polygons on the map
- Import data to the map

6 26 Mar - 1 Apr

Tasks Completed

- Found high-resolution GeoJson country polygons
- Started importing table data to map
- Work on mapping data to country colours

Tasks Not Achieved/Problems Encountered

- Loading one sheet into map
- Encountered a problem because of asynchronous Google maps loading

Tasks To Be Completed

- Map trade data to map colours
- Improve UI on map

7 2 Apr - 8 Apr

Tasks Completed

- Finished data to colour mapping
- Added year-selection drop-down
- Shortened GeoJson polygon file
- Added landing page
- Styled landing page
- Added features to map page(info modal and home button)
- Styled map page
- Started work on clickedView.html layout (for displaying more info)
- Started clickedView.html graph work

Tasks Not Achieved/Problems Encountered

- Passing data between HTML pages
- Loading the graph in clickedView.html

Tasks To Be Completed

- Improve clickedView UI
- Make the map work for the main partners data
- Add control to the graph

8 9 Apr - 15 Apr

Tasks Completed

- Added country flags to clickedView.html
- Added National Accounts display to EU countries
- More graph work and added checkboxes for graph control
- Added help button to map page

- Finished national accounts
- Finished graph checkboxes
- European union GeoJson file
- Styling and code cleanup

Tasks Not Achieved/Problems Encountered

- Checkboxes with multiple-word country names in HTML id creation
- National account values not working right
- Had to make an EU28 GeoJson multipolygon

Tasks To Be Completed

- Finish the project
- Prepare project for submission
- Comments

9 16 Apr - 22 Apr

Tasks Completed

- Comments
- Final polishing and details
- Report
- Bibliography
- Code cleanup

Tasks Not Achieved/Problems Encountered

- N/A

Tasks To Be Completed

- Per-capita view to compare country data more objectively

Bibliography

[1] bytutorial.com Andy. *How to get selected checkboxes value using JQuery and Javascript*. 2013. URL: `http://bytutorial.com/blogs/jquery/jquery-get-selected-checkboxes`.

[2] Google Developers. *Maps JavaScript API: Combining and Visualizing Multiple Data Sources*. 2018. URL: `https://developers.google.com/maps/documentation/javascript/combining-data#loading-the-state-boundary-polygons`.

[3] Google Developers. *Maps JavaScript API: Data Layer- Event Handling*. 2018. URL: `https://developers.google.com/maps/documentation/javascript/examples/layer-data-event`.

[4] MDN Web Docs. *Synchronous and asynchronous requests*. 2018. URL: `https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Synchronous_and_Asynchronous_Requests`.

[5] MDN Web Docs. *Web Storage API*. 2018. URL: `https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API`.

[6] Eurostat. *International Trade in Goods*. 2018. URL: `http://ec.europa.eu/eurostat/web/international-trade-in-goods/data/main-tables`.

[7] Eurostat. *National Accounts (Including GDP)*. 2018. URL: `http://ec.europa.eu/eurostat/web/national-accounts/data/main-tables`.

[8] Ash Kyd. *GeoJson Editor*. 2018. URL: `https://geojson-maps.ash.ms/`.

[9] Lipis. *Flag Icon CSS*. 2016. URL: `http://flag-icon-css.lip.is/`.

[10] Masayuki Tanaka. *Simple XY Line Chart*. 2014. URL: `http://c3js.org/samples/simple_xy.html`.

[11] W3Schools. *CSS Tooltip*. 2018. URL: `https://www.w3schools.com/css/css_tooltip.asp`.

[12] W3schools. *How To Create a Modal Box*. 2018. URL: `https://www.w3schools.com/howto/howto_css_modals.asp`.