IN712 Web 3 Semester 1

# Assignment 4 – REST APIs and React

# Group Size: 1

# Value: 25% of course mark

# Due Date: Friday, 16th June, 5:00 pm

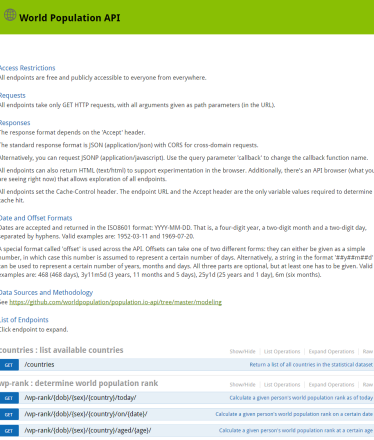
## Summary:

## In this assignment you are requested to carry out 3 separate tasks. The 1st task asks you to consume an existing REST API to retrieve data about the population of a country. The video demo countriesAPIConsumptionDemo.mp4 illustrates what you need to do. The 2nd task involves the creation of your own REST API, as delineated in the class Practicals for an online store of books. The 3rd tasks requires you to learn on your own how to use the popular React framework to create a front end view that is able to update/maintain state of components in reaction to user events. The video demo ReactDemo.avi illustrates what you need to do

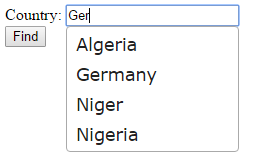
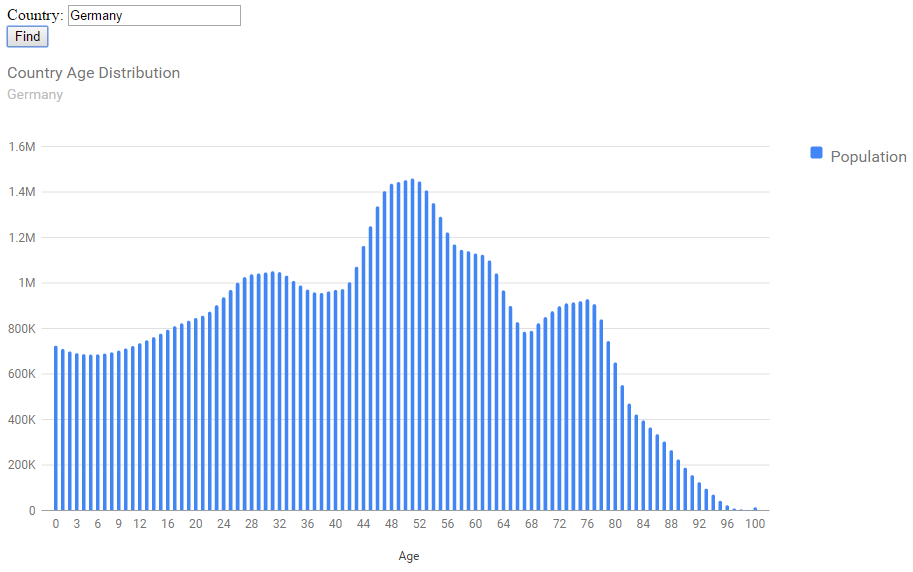
## Requirements (each of the bullet requirements is worth 1 mark):

CONSUME AND USE EXISTING APISs

* Create a simple HTML file that allows a user to input a country name and a button



* Using JavaScript download a list of all the countries available in the following world population API: <http://api.population.io/>
* Using jqueryUI create an autocomplete functionality in the input box created in requirement 1 using as source the object with all the country names downloaded in requirement 2



* When the user clicks on the button “Find”, using Ajax connect to the API <http://api.population.io/> to retrieve data about the population of the user entered country distributed by age in the year 2015.
* Using Google charts plot the extracted data, showing on the X axis the age and on the Y axis the number of people alive of that age bracket in the country in 2015.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

CREATE A REST API

Create a REST API as delineated in the in-class REST practicals (if you have completed the classroom Practicals, you just need to hand in your Django REST project). Specifically, you will need to:

* Create a model and a serializer for a book item.
* Create a book list APIView with methods to respond to GET and POST HTTP requests
* Create a book detail APIView with methods to respond to GET, PUT and DELETE HTTP requests
* Create proper authentication and permissions for your API. Create a superuser with user name: admin and password: password123. **This is important since these are the credentials I will use to test your application!** and make sure that only authenticated users are able to create, update and delete book items

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

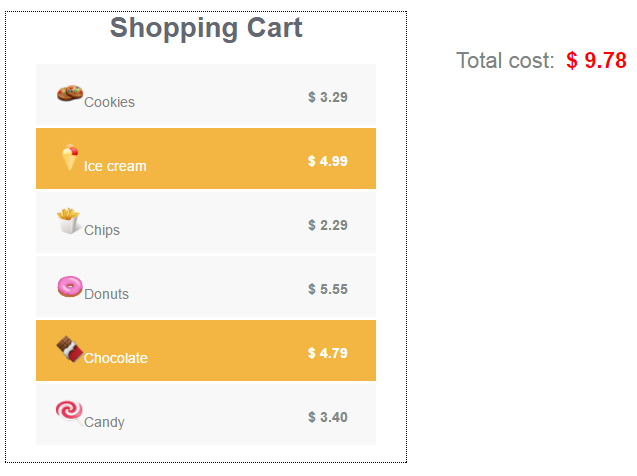
REACT FRAMEWORK

For this task, you will need to educate yourself about the React framework (<https://facebook.github.io/react/>) in order to create a shopping cart widget. Make sure to check the demo video ReactDemo.avi to understand the functionality you are required to implement. You don’t need to precompile your scripts for production. Just use the in-browser Babel “transpiler” functionality in the provided HTML. During development, start a local Python Web server (python -m http.server) on your development folder and test your application on localhost:8000 (this is how I will test your application, so make sure it works properly). You will need to:

* Create 2 React components

-a component to choose products (ProductChooser) that encapsulates a set of individual product components

-an individual product component (Product)

* the ProductChooser should maintain a state variable storing the total price of the products currently selected
* The Product component should maintain a state containing information about whether the given product is selected or not selected
* make sure you use the getInitialState method in the ProductChooser and the Product components
* make sure you use the setState method in the ProductChooser and the Product components to maintain state

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

* Create your own private repository for you to work on this assignment by using the following link: <https://classroom.github.com/assignment-invitations/138c04e0d2196e212ff4e37a858b37e8>

and **include a readme file in that repository with your complete name**. All code files and resources in a legible file structure should be available in this private GitHub repository with URL: “https://github.com/OPClasses/assignment4-YourGithubUserName”.

* Your code must be modular, readable, scalable robust and be properly documented

## Learning outcomes:

1. Understand the structure of a compiled web application or webservice and roles of elements in that structure
2. Understand web application, webservice, web pages and web control lifecycles and design items accordingly.
3. Understand and appropriately use caching, master pages, Asynchronous JavaScript and XML (AJAX), software service providers, datastore interaction, Cascading Style Sheets (CSS).
4. Understand and appreciate the impact of Web 2.0 and the semantic web.
5. Implement web software designs using an object-oriented language in a contemporary integrated development environment