

REST and REST APIs



General part

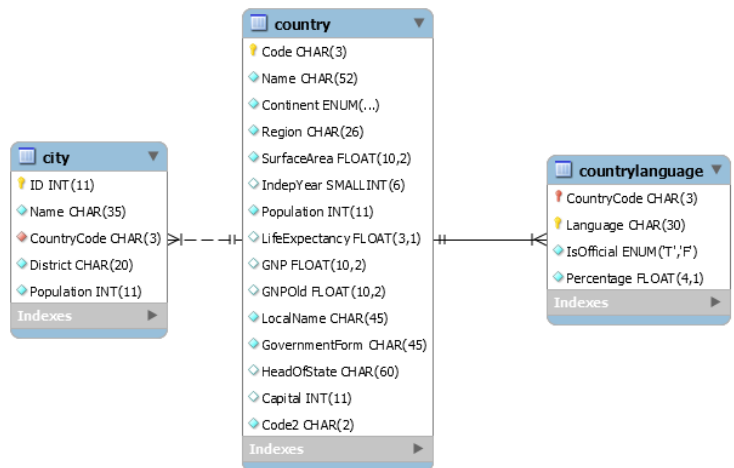
- Elaborate on some of the characteristics of REST, like: Stateless, Cacheable, Layered System, Uniform Interface etc.
- Elaborate on how JSON or XML supports communication between subsystems, even when the subsystems are implemented on different platforms.

Practical part

The starting point for this exercise is the world database provided by MySQL.

<http://dev.mysql.com/doc/index-other.html>.

The script setups the database with the three tables sketched in this diagram, and data for 239 different countries, and more than 4000 cities



Getting Started:

- Execute the script (use the file *world.sql* on Fronter, I have added a small change to simplify Entity-modelling), from MySQL Workbench, to setup the world database
- Use the NetBeans Wizard to create a set of matching Entity-classes
Hint. Do not assume that all data exists, not even that a country have a capital, so make sure to do null checks on your data.

Tasks

1) The main task for this exercise is to design a REST API given the specifications below¹:

It must be possible to:

1. Get a list of all countries, with *code*, *name*, *continent* and the *name* of the capital
2. Get a list of all countries with a population greater than an number provided (return data as above)
3. Get a list of all cities in a country (provided the Country code) including name and population
4. Create a new city for a Country

For each service you should: Define the API call (URI) and the (JSON) format used to transfer data back and forth between Client and Server².

2) Test the REST API using a browser for the GET methods and Postman for the POST method (or using Rest Assured if you prefer).

3) If you have time create an HTML page that could fetch and show data (using JQuery or Angular) for some of the methods created above.

¹ All transfer of data between the API and Clients must be encapsulated in JSON

² Ideally this should be done in a way so Client and Server development could take place in parallel