Web services – final project task

# RESTful Web service

# **Description**

In teams of 2-5 students create RESTful Web service and explore its functionality with web services testing tools, like SoapUI or Postman applications. Any platform could be used: Java, .NET, NodeJS, Python, Ruby, etc. Other types of API’s (like GraphQL) are also considerable.

# Steps

1. Assign different roles to each team member. The roles could be: developer, QA (Quality assurance specialist), analyst etc. Better if each member will have different roles (e.g. developer + QA).
2. Use online team board like Trello (see [11]) to define tasks for each member.
3. Create new project in GitHub. Use different branches to develop and merge final code to “master” branch. Each member should commit his own code.
4. Use any DBMS to create and manage your database. Create your data structure in database.
5. Create fully functional RESTful Web Service: cacheable, stateless, conforming to level 4 of Richardson Maturity Model (see [12]).
6. Using tools like APIMATIC (see [3]), transform the WADL to OpenAPI (see [4]) description in json format.
7. Using Swagger Editor (see [5]), explore your API, send the requests and observe the responses.
8. Using Postman tool (see [6]) send the requests to created and running RESTful application and observe the responses.
9. No any front-end is needed. The functionality should be presented via Postman and Swagger.

# Requirements

1. The entire code should be properly formatted.
2. The package/class/field/method names should conform to the naming conventions.
3. The Unit tests for all classes should present.
4. The entire code should be properly documented and JavaDoc generated.
5. The entire code should conform to S.O.L.I.D principles.

# References

1. [Building RESTful Web Services with JAX-RS](https://docs.oracle.com/javaee/6/tutorial/doc/giepu.html)
2. <https://www.soapui.org/>
3. <https://www.apimatic.io/transformer>
4. [OpenAPI Specification](https://swagger.io/specification/)
5. <https://editor.swagger.io/>
6. <https://www.getpostman.com/>
7. [S.O.L.I.D: The First 5 Principles of Object Oriented Design](https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design)
8. <https://trello.com/en>
9. https://martinfowler.com/articles/richardsonMaturityModel.html