

Tecnologie per IoT

LAB Software Part 1

SUGGESTION: Use **Postman**, a chrome plugin for testing REST web services by managing HTTP requests

Exercise 1. Develop in Object Oriented Programming (OOP) a RESTful-style program that exposes web services to convert a temperature value to a different unit of measurement. The web services will exploit the **HTTP GET** method that will receive three input parameters: i) the numerical value to convert, ii) the original unit of measurement (either Celsius, Kelvin or Fahrenheit) and iii) the final unit of measurement (either Celsius, Kelvin or Fahrenheit).

Handle possible errors in invoking the web services (e.g. wrong command or wrong number of parameters).

The output should be a JSON reporting i) the original numerical value before conversion, ii) the original unit of measurement, iii) the numerical value after conversion and iv) the final unit of measurement (validate with <http://jsonlint.com/>)

Example:

- `http://localhost:8080/converter?value=10&originalUnit=C&targetUnit=K`
where **convert** is the web service end-point, the parameters **value**, **originalUnit** and **targetUnit** are the numerical value to convert, the original and the final units of measurement, respectively.

Exercise 2. *Exercise_1 follow-up:* redesign RESTful-style calculator for exposing full URL fashion web services where parameters must be provided slash-separated.

Example:

- `http://localhost:8080/converter/10/C/K`

Exercise 3. Develop in Object Oriented Programming (OOP) a RESTful-style program that exposes web services to convert a list of temperature values to a different unit of measurement. The web services will exploit the **HTTP POST** method that will receive three input parameters: i) the list of numerical values to convert, ii) the original unit of measurement (either Celsius, Kelvin or Fahrenheit) and iii) the final unit of measurement (either Celsius, Kelvin or Fahrenheit). The **HTTP POST** method will receive in the body-message the following JSON:

Example:

```
{
  "values": [10, 9, 8, 7, 6, 5, 3, 2, 1],
  "originalUnit": "C",
  "targetUnit": "K"
}
```

Finally, the **POST** method should return a JSON reporting the input parameters and a list of converted values (validate with <http://jsonlint.com/>)

i) the list of numerical values to convert, ii) the original unit of measurement (either Celsius, Kelvin or Fahrenheit) iii) the numerical value after conversion, iv) the final unit of measurement (either Celsius, Kelvin or Fahrenheit), and v) the timestamp of the conversion in EPOCH time (use Python time library)(validate with <http://jsonlint.com/>).

Example:

```
{
  "values": [10, 9, 8, 7, 6, 5, 3, 2, 1],
  "targetValues" : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
  "originalUnit": "C",
  "targetUnit": "K",
  "timestamp": 1652097843
}
```

The returned JSON must be stored in a list by the webservice.