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, ii) 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 PUT** 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 PUT** method will receive in the body-message the following JSON:

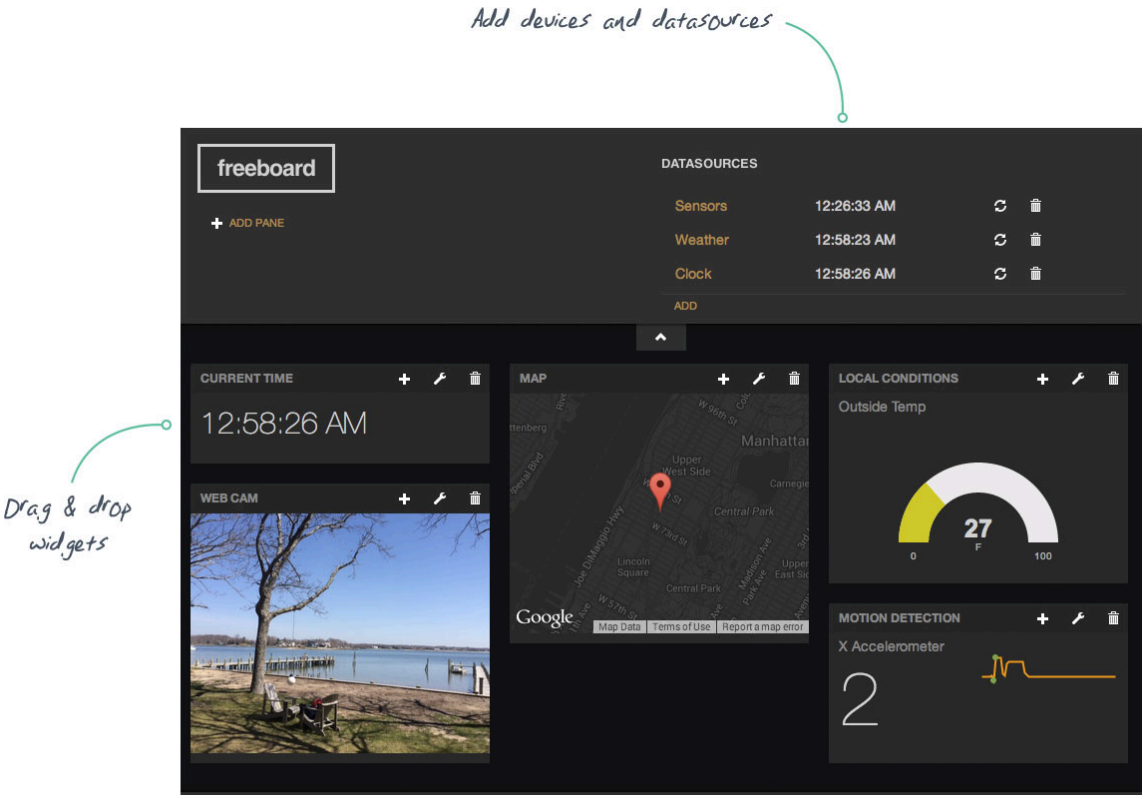
```
{
   "values": [10, 9, 8, 7, 6, 5, 3, 2, 1],
   "originalUnit ": "C",
   "targetUnit ": "K"
}
```

Finally, the **PUT** 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 and iv) the final unit of measurement (either Celsius, Kelvin or Fahrenheit) (validate with http://jsonlint.com/)

Exercise 4. Develop REST web services for deploying freeboard[1] (use the version provided as additional material) with cherrypy.

Develop the **HTTP GET** method for providing the index.html
Develop the **HTTP POST** for saving the new dashboard configuration (the resulting *dashboard.json* file must be saved in the *"freeboard/dashboard"* folder)

[1] Freeboard (http://freeboard.io/) is an open source software useful for building real-time, interactive dashboards and visualizations in minutes using the intuitive drag & drop interface.


Add devices and datasources
freeboard
+ ADD PANE
DATASOURCES
Sensors 12:26:33 AM
Weather 12:58:23 AM
Clock 12:58:26 AM
ADD
CURRENT TIME
12:58:26 AM
WEB CAM
MAP
Google
Map Data
Terms of Use
Report a map error
LOCAL CONDITIONS
Outside Temp
27
F
0
100
MOTION DETECTION
X Accelerometer
2
Drag & drop widgets