API Routes for Smarthepia project
Distributed sytems

Axel Fahy & Rudolf Höhn

December 16, 2015

1 Overview

This document talks about the REST routes used in Smarthepia project. Now, there are several Rasberry PI deployed over 2 floors at hepia which provide data from sensors that retrieve information on the room temperature, humidity, etc..

Our goal is to develop a web based application that displays data from these sensors in a cool way.

2 Technologies used

2.1 Database

We are going to use MongoDB. The model is :

```
{
    "battery": "Number",
    "controller": "String",
    "humidity": "Number",
    "location": "String",
    "luminance": "Number",
    "motion": "Boolean",
    "sensor": "Number",
    "temperature": "Number",
    "updateTime": "Number"
}
```

2.2 Client REST

Its job is to collect data from the Rasberry PI and insert them into the database. The client is going to be implemented using Scala.

2.3 Server REST

The server has two roles. The first one is to provide routes to get the information from the database with filters like between dates and things like this. And the second one is to serve the web pages. The server is also implemented using Scala

2.4 Front END

The web application is implemetend with AngularJS. Its goal is to provide a clean and userfriendly interface in order to analyze the data from the sensors.

3 Routes

Method	Request URI	Arguments
GET	/sensor/:ID/:PI_ID/temperature/	start=DD&end=DD
GET	/sensor/:ID/:PI_ID/humidity/	start=DD&end=DD
GET	/sensor/:ID/:PI_ID/luminance/	start=DD&end=DD
GET	/sensor/:ID/:PI_ID/all/	start=DD&end=DD
GET	/sensors/:PI_ID	start=DD&end=DD
GET	/room/:NAME/temperature/	start=DD&end=DD
GET	/room/:NAME/humidity/	start=DD&end=DD
GET	/room/:NAME/luminance/	start=DD&end=DD
GET	/room/:NAME/all/	start=DD&end=DD
GET	/battery/:PI_ID/:POURCENTAGE/	/

- :ID = Id of a sensor
- :PI_ID = Id of a Rasberry PI
- :NAME = Name of the room
- DD = A date