ECE 448/528 – Application Software Design

HOMEWORK #4 SOLUTION

1. Consider the RESTful interface design of a service that allows clients to query power conumpstions of plugs during certain time period *(10 points)*

i) For a plug with name ‘plugName’, clients would like to know all available power consumption dasta of it, via a HTTP GET request to /api/power/plugName. How would you specify the path in @GetMapping and specify the parameters with annotations in Spring Boot?

```
@GetMapping(“/api/power/{plugName}”)
synchronized public Object GetPlugPower(@PathVariable(“plugName”) String plugName)
```

ii) For a plug with name “plugName”, clients would like to know all available power consumption data of it after a date time like “2022/03/05 11:34:55 PM” via a HTTP GET request to /api/power/plugName?begin=20220305113455AM. Show how the answers in i) should be updated while still supporting conditions from i).

```
@GetMapping(“/api/power/{plugName}”)
synchronized public Object GetPlugPower(@PathVariable(“plugName”) String plugName,
        @RequestParam(value=”begin”, required=false) String begin)
```

iii) Based on ii), clients would like to specify an ending data time. Show how the answers in ii) should be updated while still supporting conditions from i) and ii).

```
@GetMapping(“/api/power/{plugName}”)
synchronized public Object GetPlugPower(@PathVariable(“plugName”) String plugName,
        @RequestParam(value=”begin”, required=false) String begin,
        @RequestParam(value”end”, required=false) String end))
```

iv) Provide an example of a response body in JSON for the above RESTful requests should return.

```
{
        “PlugName”: “abc”,
        “Power”: {
                        “20230330093653”: 89.023,
                        “20230330093654”: 100.435,
                        “20230330093655”: 120.345,
                        “20230330093656”: 102.23,
                        “20230330093657”: 90.43,
                        “20230330093658”: 110.45
        }
}
```

2. For Project 4, we mentioned in the lecture that you’ll need a data model for plug states and that plug states should come from the MQTT broker. Since clients will also be able to control those plugs via RESTful services, should you update the state of a plug to “on” if a RESTful request to switch that plug on is received? Explain your answer with a short discussion of pros and cons. *(5 points)*

If we put the status of plugName in PlugModel to ON once we received the RESTful request to switch it on, there will be potentially a problem since that the MQTT action message posted to the PlugSim object could be lost. In this scenario, the RESTful service will show that the plug is ON, while the plug is actually OFF. It is a better practice to wait for the MQTT update message from the PlugSim object instead of updating the status in PlugModel directly. Nevertheless, doing so would incur delays on when the status in PlugModel actually get updated.