Exercise 3

Instructions:

- The exercises are to be done individually (direct copying of code from friends is not allowed), but you can of course ask your friends for advice. Good place for discussing the assignments and the course in general is channel #momib at IRCNet.
- Some individuals will be asked to present their work to the group at the exercise sessions.
- Support for Android development is provided by the course staff.
- You can also use IOS, Windows phone, Meego etc., but with less support.
- Testing on real mobile devices is recommended, but not mandatory.

- Be prepared to demo and explain your solution at the exercise session 9.4.2013 at 14:15 in D123.
- Return the source code of your solution in a .zip, .tar.gz or tar.bz archive to the Moodle page of the course at https://moodle.helsinki.fi/course/view.php?id=9438 before 9.4.2013 14:15. Note that Moodle uses the University of Helsinki AD login and password, not the ones of the department of computer science. Please enclose the files you intend to return into a single folder that is named according to the format firstnamefamilyname, and include that folder into the archive. The maximum size of the archive is 20 MB.
- Getting the point from the exercise requires that you both return your solution to moodle and are present at the exercise session. If you cannot be present at the exercise session for some good reason, you can compensate for your absence by demoing your solution personally to the course assistant.

Assignment

1. Modify the application that you created in exercise 2 to send the list of sensor names to the CoAP server using the POST operation to the URI “/devices”, in the JSON format:

   ```json
   
   "deviceId": "your_device_id",
   "sensors": [
   "some_sensor_name",
   ...
   ]
   
   ```

2. Modify the CoAP server to reply to the message described in (1) by sending a CoAP PUT message to the URI “/subscriptions” on the mobile device. The payload of this message should be in the JSON format:

   ```json
   
   "subscriptions": [
   
   "sensor": "some_sensor_name",
   "sampling_interval": some_interval,
   "sending_interval": some_other_interval,
   "duration": some_duration
   },
   ...
   ]
   ```
Where “sampling_interval”, “sending_interval” and “duration” are given as milliseconds. Passing value of -1 as the value of “sampling_interval” means that the sensor should be sampled whenever its value changes. Passing value of -1 as the value of “sending_interval” means that the sensor reading should be sent whenever its value changes. Passing value of -1 as the value of “duration” means that the sampling and sending should continue as long as the application is running.

For the demo, make the server subscribe to at least two different sensors with a different “sampling_interval”, “sending_interval”, and “duration”.

(3) Modify the application on the mobile device to react to the message specified in (2) by starting the reading of each subscribed sensor every “sampling_interval” milliseconds and storing the readings into a data structure. The readings stored in this data structure should be sent to the CoAP server every “sending_interval” milliseconds with a CoAP PUT message to the URI “/devices/your_device_id/sensors” with the payload in the JSON format:

```json
{
    "sensorReadings": [{
        "sensor": "some_sensor_name",
        "readings": [
            some_reading_time,
            reading_value,
            other_reading_time,
            other_reading_value,
            ...
        ],
    },
    ...
}
```

Where reading times are expressed as milliseconds passed since midnight January 1, 1970 UTC. Continue the sampling and sending of the readings of each sensor for the duration of “duration” milliseconds. You do not need to take the potentially needed splitting of readings into several messages into account.

Note that if two or more subscriptions are started in the same time with an identical sending interval, you can use one message for posting readings from all the involved sensors.

**Hints for Android developers:**

You can find discussion about Android device IDs in:
http://stackoverflow.com/questions/2785485/is-there-a-unique-android-device-id