Exercise 4

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 23.4.2013 at 16:15 in D122.
• 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

You can use either your own solution to exercise 3, or the example solution as the basis for completing this assignment.

(1) Modify the application on the mobile device to send the sensor readings in the JSON format:

```json
{
    "sensorReadings": [
        {
            "sensor": "some_sensor_name",
            "readings": [
                {"time": some_reading_time,
                 "values": [value1, value2, ...]},
                ...
            ],
        },
        ...
    ]
}
```

(2) Expose the location of the mobile device as a pseudo-sensor, and integrate this pseudo-sensor into the application.
(3) Create a web page or mobile application that shows/tracks the location of the mobile device and the sensor readings on Google maps, openstreetmap.org, or similar service. Fetch this information from the CoAP server using HTTP protocol.

**Hints:**

Jcoap library itself contains a simple in-memory-database, and a http server.