420-616-AB Oracle Technologies
Fall 2016

October 13th, 2015
Worth 5%

Assignment IV

(Due date is on Léa)

In this assignment, you will develop two client-side classes, `User` and `Note` to represent the information stored on the server. Since the server uses the JSON serialization format in its communication, these two classes should be able to parse and format their JSON representation.

Instead of using these classes right away in an app, we will first test them using unit testing. Testing will run in two phases:

- First the JSON parsing and formatting will be tested.
- Second the CRUD operations on the server will be tested.

1 Classes `Note` and `User`

Create a sub-package inside your app package called `model.server`. Create two classes `Note` and `User` to communicate with the `user` and `note` repositories on the server. For a description about what is being stored in the server and the JSON format used in communication see “Note Server Functional Requirements” and “Note Server User Manual”.

These two classes should pass all the unit tests in the file `TestJson.java`. Use the test driven development techniques as we did in class.

1.1 Using The `Gson` Library

Use the Gson library is used to parse and format JSON. You can find their API and user guide at `https://github.com/google/gson`. To use Gson in your app, include the following in the dependencies section of `build.gradle`:

```
compile 'com.google.code.gson:gson:2.7'
```

2 Server

Read through the three documents included with this assignment to understand the structure of the server and how to use it: “Note Server Functional Requirements”, “Note Server Installation” and “Note Server User Manual”.

Your `User` and `Note` class must pass all the unit tests in `TestServer.java`.

2.1 HTTP classes

Communication with the server will be done using HTTP using the classes `HttpRequest` and `HttpResponse` provided. Add them in your project in the `model.server` package.

2.2 Extra Unit Tests

Write three unit tests in `TestServer.java`:

Test Name	Description
`testNoteTitleUnique`	Test that the server will reject non-unique note titles.
`testUserNameUnique`	Test that the server will reject non-unique user names.
`testNoteReminderAfterCreated`	Test that the server will reject notes whose creation date is before the reminder date.

3 Requirements

- Your program should be clear and well commented. It must follow the “420-616 Style Guidelines” (on Léa).
- Work from your previous assignment, or create a new Android project with minimum SDK 21 or later.
- Create the classes `Note` and `User` in a project sub-package called `model.server`.
- Use the Gson library to implement `parse` and `format` operations.
- The above classes pass the unit tests in `TestJson.java` and `TestServer.java`. Note: individual tests might count for different amounts of the final grade.
- Write the unit tests indicated in Section 2.2.
- Submit your project using git. Follow the Git Submission instructions on Léa.