School Computing, Faculty of Engineering
Universiti Teknologi Malaysia

SCSJ3623 Mobile Application Programming
Semester 2, 2019/2020

Exercise 3
HTTP and JSON

Exercise Material

- Program template is provided for this exercise. Please download from the elearning and extract the ZIP file to your local drive.
- The tasks to be completed are stated in the Problem section below.

Pair Programming

- You may be adopting the “Pair Programming” strategy to do this exercise.
- Arrange yourself with you partner for the pair programming session.
- Each pair must ensure to switch role (between Navigator and Driver) periodically.

Duration

- You may need 60 to 90 minutes to complete this exercise.
- Should you need more time, you can grant it yourself provided you do not exceed the submission deadline

Plagiarism Warning

- You are allowed to refer to any resources.
- You are allowed to have discussions with others during the exercise session.
- However, you and your partner must do the exercise on your own.
- Any kind of plagiarism (e.g., copying and pasting code by any means) would lead your submission dismissed. This applies to both parties, students that copy others’ code and students that give their code to others.

Late Submission Penalty

- **10% deduction for every 10 minutes late**.
- For example: if the duration of the exercise is 60 minutes, and your submission is received on the 61st minute, you are only eligible to earn 90% at most of the total marks.

Problem

In this exercise you will be implementing fetching data over the web using the Dart's package http. You will be developing a simple Todo List application that accesses data from an API server. You will be using JSON server software to setup a fake API server.

The app will show the list of todo items along with their id number. The app allows the user to perform the following operations:

- Toggle the status of a todo from "not-completed" to "completed" or vice-versa by tapping on the todo. A completed task is shown in a line-through text, whereas a not-completed todo in normal text.
- Remove a todo item by long-pressing on the todo.
- Add a new todo item by tapping on the plus (+) button. Note that, the id for the todo item will be given by the server.

Note: all the above operations will update the database on the fake server permanently.

Watch the attached video for example run of the app

Tasks

You are provided with the files that you are going to work on to accomplish the following tasks:

- Define some HTTP Request examples that the app will be using. See the questions in the file **rest_client/rest_request.http**
- Add fromJson() and toJson() methods to the Todo class, in **models/todo.dart**
- Complete the class DataService class in **services/data_service.dart**
- Modify the TodoListScreen so that the todo list is fetched from the server. Note: do not use global variables to store the data. See the file **screens/todo_list.dart** for the other tasks to be done for this class.

Preparing the Exercise Material

- Download the zip file (exercise3.zip) and extract it to a folder with the same name (exercise3).
- You should get the following materials:
    - A base flutter project for the exercise
    - A video of the expected result (expected_result.mp4)
    - Question script (question.pdf)
    - Sample of JSON database to be run on JSON server (db.json)

Initializing Git

- Go inside the exercise folder (using Git Bash)
- Type the following command to initialize git

```
git init
```

- Identify yourself to git so that everytime you perform git commit, it will also log your name. Note: use your short name and your partner's.

```
git config user.name  "you and yourpartner"
```

Note: you can only use the above command once. Running it again will overwrite the first one. Thus, combine your name and your partner's to be used to the command.

- Add files to be tracked by git

```
git add .
```

Committing Changes

- Type the following command to commit (or, to record) any changes you have made.

```
Git commit -a -m "write a concise message for each commit"
```

- Do perform **a commit for each task above** (see the Task section).

Creating A Git Bundle File

- Once you have done with your exercise, prepare a git bundle for submission.

```
git bundle create exercise3.git HEAD master
```

- If you want to test out the git bundle file (exercise3.git) before you submit, clone the repo to another folder. First, copy the git file (using Window explorer) to the other folder. Then clone the repo (using Git bash)

```
git clone exercise3.git exercise3
```

Setting Up A Fake API Server

- Watch the lecture video (Part 2- Demo)

Submissions

- Submit your work in **a git bundle** file (i.e. **exercise3.git**)
- Submit on elearning.
- Deadline: **refer on elearning**
- Only one member from each pair needs to do the submission.