IEMS 5722
Mobile Network Programming and Distributed Server Architecture
2015-2016 Semester 2

Assignment 2: Using HTTP for Data Communication

Due Date: 26th February, 2016

Notes:

i.) Read carefully the instructions and requirements to understand what you have to do

ii.) Follow the instructions in Section 4 to submit your files for marking

iii.) Late submissions will receive 30% mark penalty

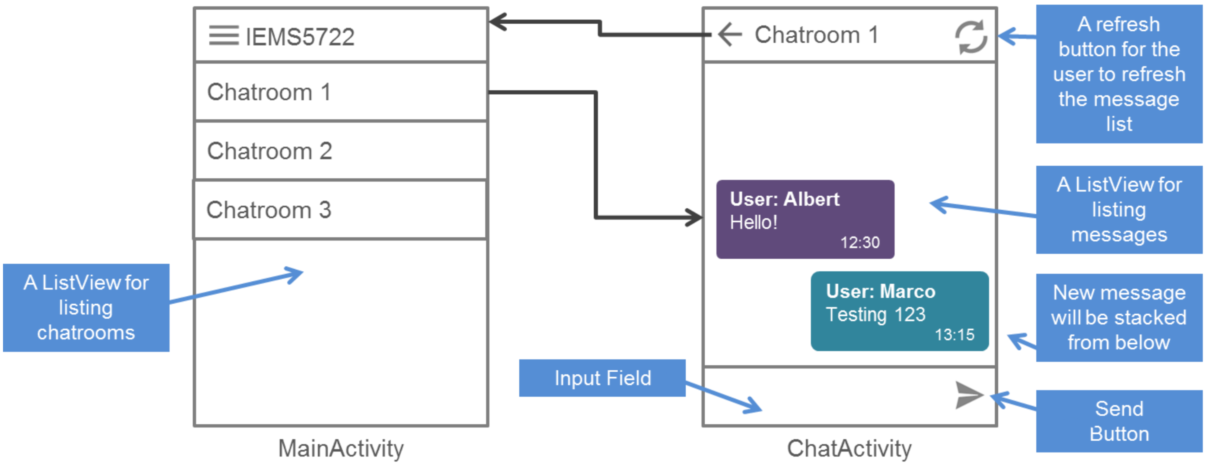
1. Objectives

- To learn how to use asynchronous tasks to handle network operations
- To learn how to send and receive HTTP messages in Android to and from APIs
- To learn how to handle data encoded in JSON format

2. Instructions

In this assignment, you are going to extend your previous assignment by adding networking functionality to the mobile app, such that it will become a working instant messaging app. We will provide you with several APIs (application programming interfaces), which will allow you to 1) get a list of chatrooms, 2) get a list of messages in a chatroom, and 3) send a message to a chatroom. The server will store any message sent to it through valid API requests, and therefore the user will be able to see the messages when he or she visits the chatroom again.

First of all, you will update the user interface of the messaging app as follows.



IEMS5722
Chatroom 1
Chatroom 2
Chatroom 3
A ListView for listing chatrooms
MainActivity
Chatroom 1
User: Albert
Hello!
12:30
User: Marco
Testing 123
13:15
Input Field
ChatActivity
A refresh button for the user to refresh the message list
A ListView for listing messages
New message will be stacked from below
Send Button

Figure 1: UI of the instant messaging app

As in the previous assignment, the app has two activities. In the first activity, you should now have a ListView for listing all available chat rooms. The list of chat rooms should be retrieved by sending a request to an API (details below). When the user clicks on one of the chat rooms, the app should transit to the Chat Activity, which will list the messages that have been sent to the selected chat room so far. The list of messages should also be retrieved by sending a request to an API.

In addition, you should also further develop the app such that the user can send messages to the chat room. This is done by submitting the message to the server via a specific API.

Finally, you will also add a ‘Refresh’ button, which when clicked by the user will cause the ListView of messages to refresh and fetch the latest messages from the server. (You will learn how to push any new message to the app in another assignment.)

3. APIs

You will need to use the following three APIs when implementing the app in this assignment.

API	**GET**: http://104.155.195.255/iems5722/get_chatrooms
Descriptions	For retrieving a list of chat rooms from the server
Input Parameters	No input parameters required
Example	http://104.155.195.255/iems5722/get_chatrooms

Sample Output	`{`
	`  "data": [`
	`    {`
	`      "id": 3,`
	`      "name": "Chatroom 002"`
	`    },`
	`    {`
	`      "id": 2,`
	`      "name": "General Chatroom"`
	`    }`
	`  ],`
	`  "status": "OK"`
	`}`

API	**GET**: http://104.155.195.255/iems5722/get_messages
Descriptions	For retrieving a list of messages in a specific chat room
Input Parameters	• chatroom_id (the ID of the chat room) • page (the page number of the list of messages)
Example	http://104.155.195.255/iems5722/get_messages?chatroom_id=2&page=1

Sample Output	`{`
	`  "data": {`
	`    "current_page": 1,`
	`    "messages": [`
	`      {`
	`        "message": "18",`
	`        "name": "Albert",`
	`        "timestamp": "2016-01-23 19:36",`
	`        "user_id": 1`
	`      },`
	`      {`
	`        "message": "17",`

```
        "name": "Albert",
        "timestamp": "2016-01-23 19:36",
        "user_id": 1
      },
      {
        "message": "16",
        "name": "Albert",
        "timestamp": "2016-01-23 19:36",
        "user_id": 1
      },
      {
        "message": "15",
        "name": "Albert",
        "timestamp": "2016-01-23 19:36",
        "user_id": 1
      },
      {
        "message": "14",
        "name": "Albert",
        "timestamp": "2016-01-23 19:36",
        "user_id": 1
      }
    ],
    "total_pages": 4
  },
  "status": "OK"
}
```

API	**POST**: http://104.155.195.255/iems5722/send_message
Descriptions	For retrieving a list of messages in a specific chat room
Input Parameters	• chatroom_id (the ID of the chat room) • user_id (the unique ID of the user, use your student ID for now) • name (the name you want to be displayed in the chat room) • message (the message input by the user)
Example	POST the following to the API: chatroom_id=2&user_id=11223344&name=Alan&message=1234

Sample Output	`{`
	`  "status": "OK"`
	`}`

In case any of your input parameters is invalid, the API will return “status”=”ERROR”. Therefore, you should check whether “status”=”OK” before you continue to use the data returned by the APIs.

4. Requirements

The app you submit in this assignment should have the following features:

- **Two activities** (Main Activity and Chat Activity)
- When the Main Activity is created, you should fetch **a list of chat rooms** from the server using the **/iems5722/get_chatrooms API**, and then **populate** the ListView in the activity with the names of the chat rooms
- When the user clicks on one of the chat rooms, he or she **should be shown the Chat Activity**, in which the following functions should be realized

- There should be a **refresh button** on the Action Bar
- The Chat Activity should fetch **the first page of messages** in the selected chat room using the **/iems5722/get_messages API** when the user **enters the chat room**, of when the user **clicks on the refresh button**, and the messages should be shown in the ListView
- When the user inputs some text and clicks the Send button, the app should clear the input area, and **submit the message** to the server using the **/iems5722/send_message API**, and then **add the message** to the bottom of the ListView
- When the user **scrolls to the topmost message** in the ListView, the app should automatically **fetch the next page** of the messages, unless the current page is the last page. (Pay attention to the 'total_pages' field in the JSON returned by the /iems5722/get_messages API.
- Each message should display the **name of the user** who wrote the message, **the message**, as well as the **timestamp** of the message.

5. Guidelines

AsyncTask

When sending an HTTP request to the server, you should not do any network operation on the UI thread as discussed in lecture. Use an AsynTask for this purpose. Refer to the lecture notes and the following Website for more information about how to use AsyncTask:
http://developer.android.com/guide/components/processes-and-threads.html.

ListView

The app should load the next page of messages if the user has scrolled to the topmost message. You can implement an OnScrollListener and attach it to the ListView for this purpose (http://developer.android.com/reference/android/widget/AbsListView.OnScrollListener.html). You can start with the following sample code

```
listview.setOnScrollListener(new AbsListView.OnScrollListener() {
    @Override
    public void onScroll(AbsListView v, int first, int visible, int total) {
        // Put your code here ...
    }
    @Override
    public void onScrollStateChanged(AbsListView arg0, int arg1) {
    }
});
```

6. Submission

To facilitate marking of the assignments, you should strictly follow the instructions below:

- The **package name** of your app should be in the following format:
  hk.edu.cuhk.ie.iems5722.a2_<your_student_id>
- The **name** of your app should be in the following format:
  A2_<your_student_id>

To submit your assignment, create a folder name <your_student_id>_assgn2. In the folder, you should include the following items:

- The **APK file** generated using Android Studio (a debug version is enough)
- The whole **project folder** of the app

Compress this folder using ZIP, you should now have a file named **<your_student_id>_assgn2.zip**. Submit it in the CUHK eLearning System online: https://elearn.cuhk.edu.hk/

7. Marking Criteria (Total 100 Marks)

- **(20 marks)** Able to retrieve and list Chatrooms automatically
- **(20 marks)** Clicking on a chatroom in MainActivity goes to the correct Chatroom
- **(20 marks)** Able to retrieve messages and refresh messages in the Chatroom
- **(20 marks)** ListView is able to page to retrieve older messages and handles the last page correctly
- **(20 marks)** Able to send messages to a specific Chatroom and UI updates correctly