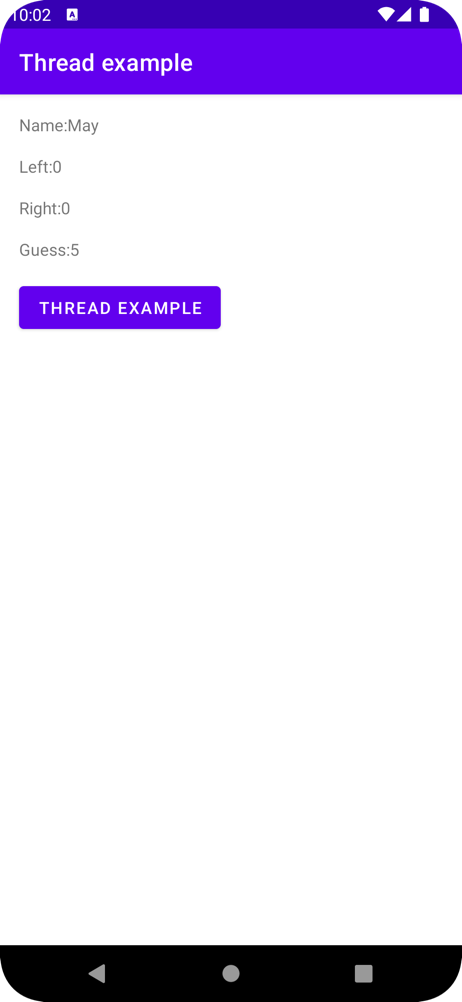
# Lab 6

In this part, students will develop a mobile application for parsing JSON from the Internet.

Here are the screen captures of the application:



The JSON Link:

<https://4qm49vppc3.execute-api.us-east-1.amazonaws.com/Prod/itp4501_api/opponent/2/>

Step 1:

In Android Studio, create a new project named Lab 6 with following project setting:

* Choose your project : **Empty Activity**
* Application Name : **Lab 6**
* Project location : use the default setting
* Language : **Java**
* Minimum API level: **API 21: Android 5.0 (Lollipop)**
* Click **Finish**

Step 2:  
Open strings.xml and insert the following code:

<resources>  
 <string name="app\_name">Thread example</string>  
 <string name="name">Name: </string>  
 <string name="left">Left: </string>  
 <string name="right">Right: </string>  
 <string name="guess">Guess: </string>  
</resources>

Step 3:  
Open activity\_main.xml and insert the following code:

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">

<TextView  
 android:id="@+id/name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:text="@string/name"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
<TextView  
 android:id="@+id/left"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:text="@string/left"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/name" />  
  
<TextView  
 android:id="@+id/right"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:text="@string/right"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/left" />

<TextView  
 android:id="@+id/guess"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:text="@string/guess"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/right" />  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp"  
 android:onClick="onClick"  
 android:text="@string/app\_name"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/guess" />  
</androidx.constraintlayout.widget.ConstraintLayout>

Step 4:

Right click com.example.lab6 -> New -> Java class, choose **Java** and type “MyThread”, then press Enter.

Step 5:

Open MyThread.java and insert the following code to define the object that we need to separate from JSON:

public class MyThread {  
 private String name = "name";  
 private String left = "left";  
 private String right = "right";  
 private String guess = "guess";  
 private String urlString ;String data = "";  
  
 final JSONObject[] maindata = {  
 new JSONObject()  
 };

}

Step 6:  
Add the volatile boolean in MyThread.java to check the status of thread, **volatile** guarantees happens-before relationship among threads sharing that variable.

public volatile boolean parsingComplete = true;

Step 7:

Add the constructor and GET method to return the values of JSON

public MyThread(String url){  
 this.urlString = url;  
}  
public String getName(){  
 return name;  
}  
public String getLeft(){  
 return left;  
}  
public String getRight(){  
 return right;  
}  
public String getGuess(){  
 return guess;  
}

Step 8:

Add the following code for separating the JSON

public void readAndParseJSON(String data) {  
 try {  
 maindata[0] = new JSONObject(data);  
 name = maindata[0].getString("name");  
 left = maindata[0].getString("left");  
 right = maindata[0].getString("right");  
 guess = maindata[0].getString("guess");

parsingComplete = false;  
} catch (Exception e) {  
 *// TODO Auto-generated catch block* e.printStackTrace();  
 }  
 }

Step 9:

Add the following code inside MyThread.java to create a thread to get the data from the Internet

public void fetchJSON(){  
 Thread thread = new Thread(new Runnable(){  
 @Override  
 public void run() {  
 try {  
 URL url = new URL(urlString);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 conn.setReadTimeout(10000 */\* milliseconds \*/*);  
 conn.setConnectTimeout(15000 */\* milliseconds \*/*);  
 conn.setRequestMethod("GET");  
 conn.setDoInput(true);  
 *// Starts the query* conn.connect();  
 InputStream stream = conn.getInputStream();  
 InputStreamReader inputStreamReader = new InputStreamReader(stream);  
  
 int inputStreamData = inputStreamReader.read();  
 while (inputStreamData != -1) {  
 char current = (char) inputStreamData;  
 inputStreamData = inputStreamReader.read();  
 data += current;  
 }  
 Log.*d*("GET RX =>", " " + data);  
 readAndParseJSON(data);stream.close();  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 });  
  
 thread.start();  
}

Step 10:

Back to MainActivity.java and replace with the following codes to define the layout:

public class MainActivity extends AppCompatActivity {  
  
 private TextView name,left,right,guess;  
 private MyThread myThread;  
 private String url = "https://4qm49vppc3.execute-api.us-east-1.amazonaws.com/Prod/itp4501\_api/opponent/2/";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 name=findViewById(R.id.*name*);  
 left=findViewById(R.id.*left*);  
 right=findViewById(R.id.*right*);  
 guess=findViewById(R.id.*guess*);  
  
 }

}

Step 11:

Add onClick() method

public void onClick(View v){  
  
 myThread=new MyThread(url);  
 myThread.fetchJSON();  
  
 while (myThread.parsingComplete);  
 name.setText(getResources().getString(R.string.*name*)+ myThread.getName());  
 left.setText(getResources().getString(R.string.*left*)+ myThread.getLeft());  
 right.setText(getResources().getString(R.string.*right*)+ myThread.getRight());  
 guess.setText(getResources().getString(R.string.*guess*)+ myThread.getGuess());  
  
}