Mobile_ICP10 Report

ICP 10

. ICP Group 3

. Name : Ashwini Reddy Konidala

. Email : akdrw@mail.umkc.edu

My Partner Details

. Partner Name : Rishmitha Chennupati

. Partner Email : rchxc@umsystem.edu

. Partner Repository : https://github.com/UMKC-APL-WebMobileProgramming/ICP9-RishmithaChennupati

My Video and source code links:

. ICP10 video：https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=6515d4c5-ed80-4c37-bbe1-add801881d5a

. ICP10 task：https://github.com/UMKC-APL-WebMobileProgramming/ICP9-akonidala19/tree/main/ICP9/Source

GitHub Repository

https://github.com/UMKC-APL-WebMobileProgramming/ICP10-akonidala19

Overview

Android development with JSON data from APIs, parsing JSON data, error handling, Async Task Class and ListView

Programming elements:

- RESTful Services

- ListView
- Adapter
- Recycling
- Multi-Threading
- Async Task
- Install Android Studio https://developer.android.com/studio/index.html
- Install Java Development Kit (JDK) http://java.oracle.com/

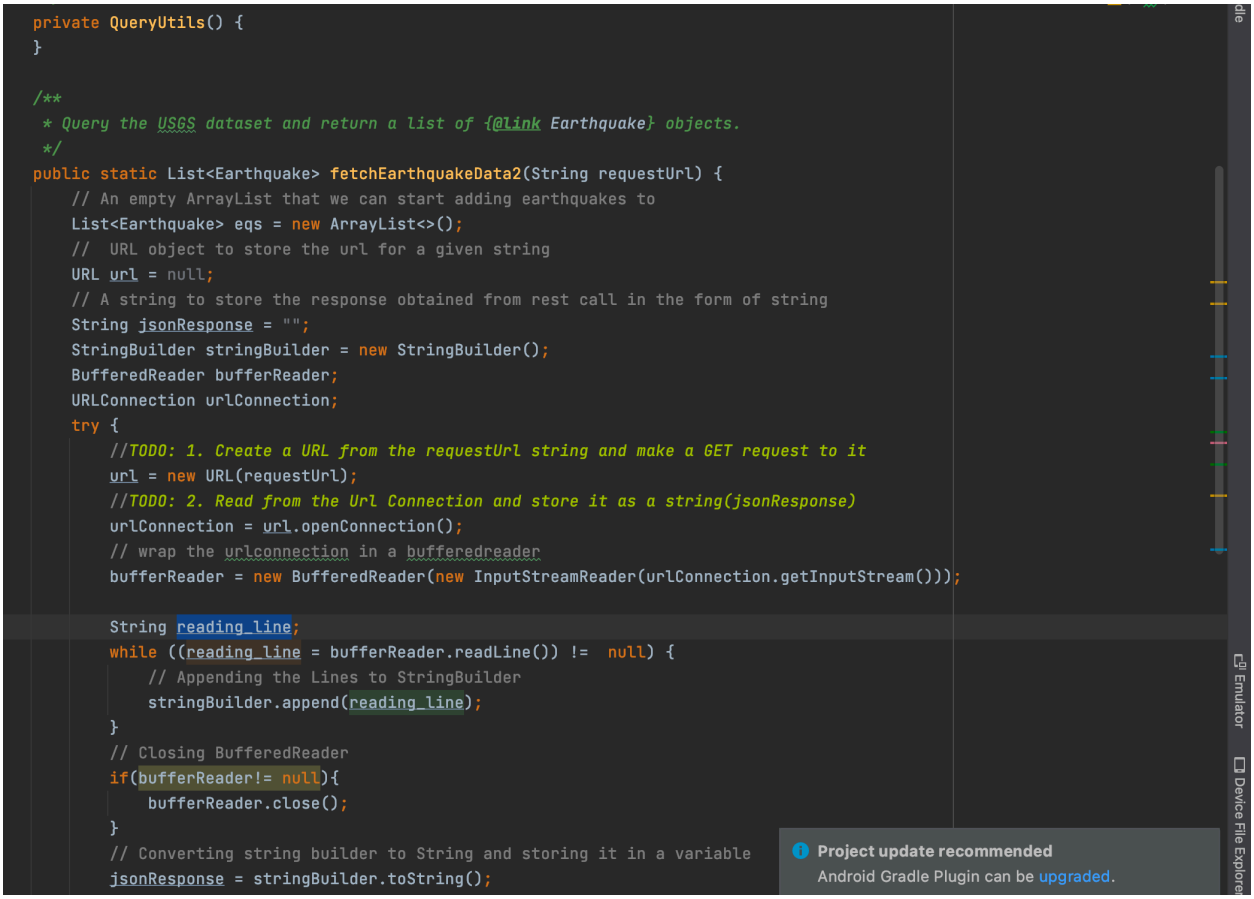
Task 1

- Create an app on Earthquake by displaying recent earthquake information
    - Main activity should display a list of information about earthquakes
    - User will be directed to a webpage of USGS, with more information about the specific earthquake
- API - https://earthquake.usgs.gov/fdsnws/event/1/
- Follow all the TODO instructions in the file given by course/professor

Screenshots

To-Do 1: Making a GET request by creating a URL from the requestUrl string.

To-Do 2: Reading from the Url Connection and then store it as a string named as jsonResponse by using Buffered Reader to get the input.
And also using the while loop checking whether the Buffered Reader is empty or not.
If it is not empty we are appending it to the string builder if not we are closing it.

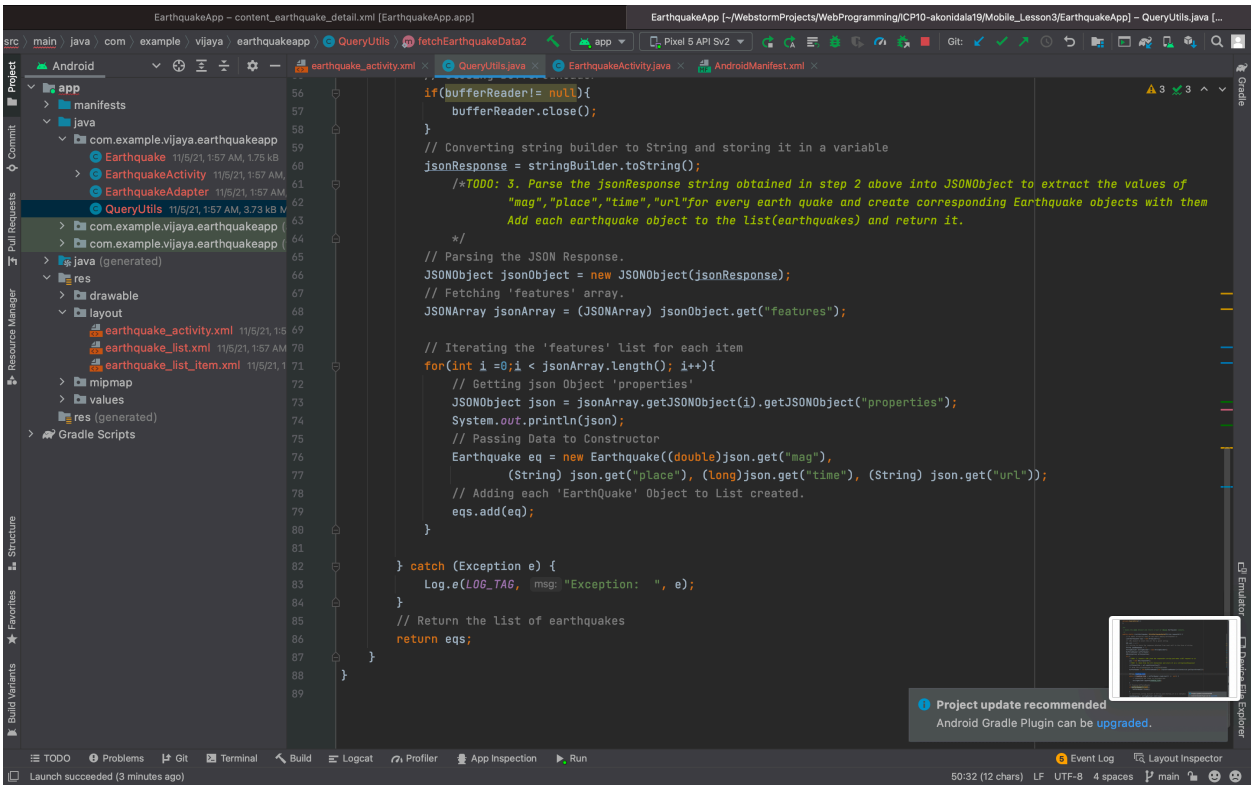


```
private QueryUtils() {
}

/**
 * Query the USGS dataset and return a list of {@link Earthquake} objects.
 */
public static List<Earthquake> fetchEarthquakeData2(String requestUrl) {
    // An empty ArrayList that we can start adding earthquakes to
    List<Earthquake> eqs = new ArrayList<>();
    //  URL object to store the url for a given string
    URL url = null;
    // A string to store the response obtained from rest call in the form of string
    String jsonResponse = "";
    StringBuilder stringBuilder = new StringBuilder();
    BufferedReader bufferReader;
    URLConnection urlConnection;
    try {
        //TODO: 1. Create a URL from the requestUrl string and make a GET request to it
        url = new URL(requestUrl);
        //TODO: 2. Read from the Url Connection and store it as a string(jsonResponse)
        urlConnection = url.openConnection();
        // wrap the urlconnection in a bufferedreader
        bufferReader = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()));

        String reading_line;
        while ((reading_line = bufferReader.readLine()) !=  null) {
            // Appending the Lines to StringBuilder
            stringBuilder.append(reading_line);
        }
        // Closing BufferedReader
        if(bufferReader!= null){
            bufferReader.close();
        }
        // Converting string builder to String and storing it in a variable
        jsonResponse = stringBuilder.toString();
```

Project update recommended
Android Gradle Plugin can be upgraded.

Emulator
Device File Explorer

To-Do 3: Now Converting the obtained String into a Json Object by parsing the jsonResponse string into JSONObject, to extract the values of "mag" (magnitude), "place"(location), "time", "url" for every earthquake. Correspondingly creating the Earthquake objects and adding each object of the earthquake to the list(i.e., earthquakes) and finally returning them.



To-Do 4: To view the earthquake URI by creating a new intent (Action View) and then sending the intent to launch a new activity.

```
        @Override
        public void onItemClick(AdapterView<?> adapterView, View view, int position, long l) {
            // Find the current earthquake that was clicked on
            Earthquake currentEarthquake = mAdapter.getItem(position);

            // Convert the String URL into a URI object (to pass into the Intent constructor)
            Uri earthquakeUri = Uri.parse(currentEarthquake.getUrl());

            //TODO: 4. Create a new intent to view the earthquake URI.Send the intent to launch a new activity
            Intent eqIntent = new Intent(Intent.ACTION_VIEW, earthquakeUri);
            startActivity(eqIntent);
        }
    });

    // Start the AsyncTask to fetch the earthquake data
    EarthquakeAsyncTask task = new EarthquakeAsyncTask();
    task.execute(USGS_REQUEST_URL);
}
```

To-Do 5: Finally adding the internet permissions in AndroidManifest.xml file.

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.vijaya.earthquakeapp">

    <!--TODO 5: Add internet permission-->
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:fullBackupContent="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name="com.example.vijaya.earthquakeapp.EarthquakeActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

OUTPUT:

In the above screenshots those are the output screens for the Earthquake Info Application where all the recent reported incidents are listed and each item in the list, when clicked, redirects to the USGS website (of that incident) in a browser.

After clicking or selecting any item from the earthquake list, it will navigate and display the information about earthquake in detail.