

Mobile Development

Lecture 9: Android & RESTFUL Services

Mahmoud El-Gayyar

elgayyar@ci.suez.edu.eg

Elgayyar.weebly.com

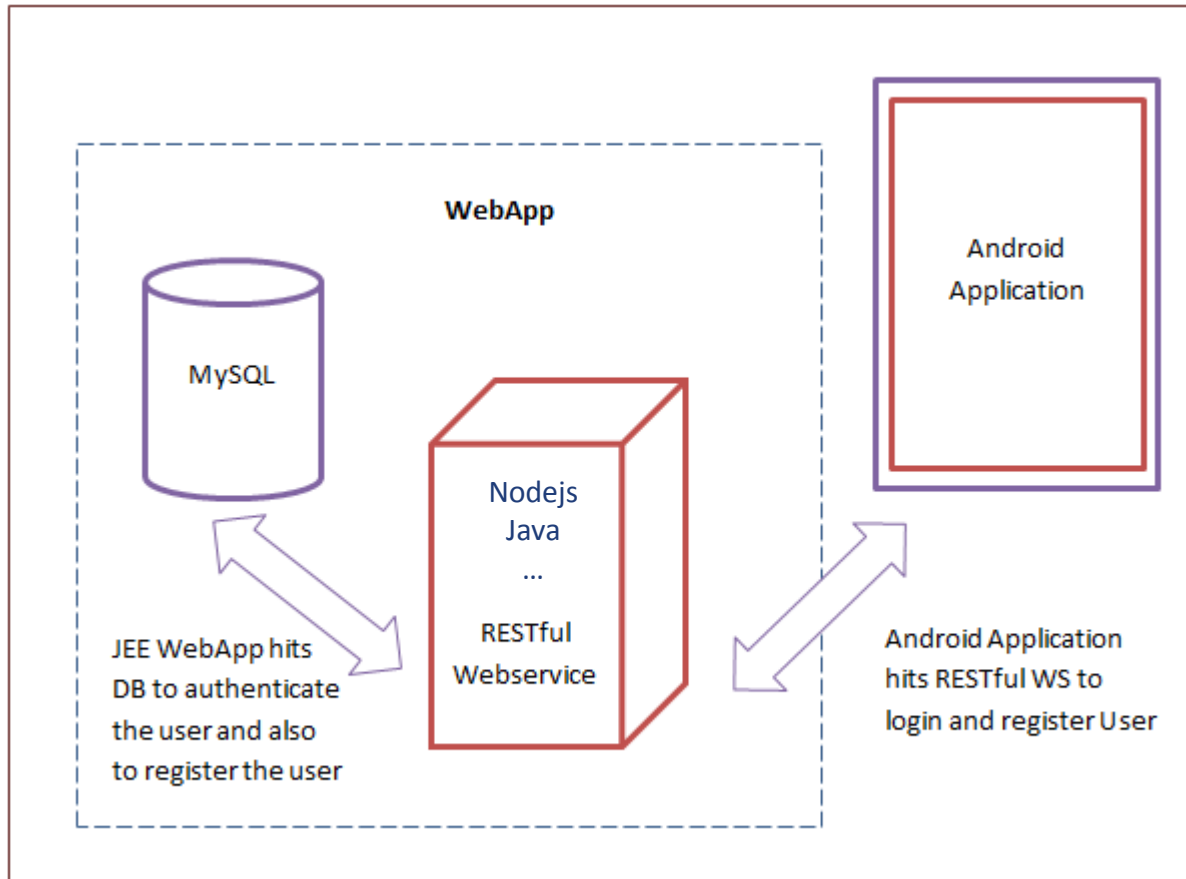


What is a RESTFUL Web Service

- *REST stands for **RE**presentational **S**tate **T**ransfer.*
- *In REST architecture:*
 - ◆ a REST Server simply provides access to resources and
 - ◆ REST client accesses and presents the resources.
 - ◆ JSON is the most popular format being used in web services.
- *HTTP Methods*

GET	Provides a read only access to a resource.
POST	Used to update a existing resource or create a new resource.
DELETE	Used to remove a resource.

What is a RESTFUL Web Service



Required Permissions

- *Well, your App will need to connect to the internet in order to request your web services. This requires the permission **android.permission.INTERNET**.*

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myusername.myapplication" >
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <application android:allowBackup="true"    android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"    android:theme="@style/AppTheme" >
        <activity android:name=".MainActivity"
            ...
        </activity>
        ...
    </application>
</manifest>
```

JavaScript Object Notation (JSON)

- *JSON is a specified format for the exchange of structured data.*
- *The syntax is the same like defining JavaScript objects*
- *JSON is more lightweight and easier to read than XML.*

```
{
  "customername": "Derp",
  "items": [
    {
      "id": 100,
      "name": "item X",
      "active": true,
      "categorieIds": [10,15,17]
    },
    {
      "id": 101,
      "name": "item Y",
      "active": false,
      "categorieIds": [11]
    }
  ]
}
```

REST Service

- *Now we need a Rest Service to test with 😊*
- *<http://hmkcode.appspot.com/rest/controller/get.json>*

```
{
  "articleList": [
    {
      "title": "Android Internet Connection Using HTTP GET (HttpClient)",
      "url": "http://hmkcode.com/android-internet-connection-using-http-get-httpclient/",
      "categories": ["Android"],
      "tags": ["android", "httpClient", "internet"]
    },
    {
      "title": " Android | Taking Photos with Android Camera ",
      "url": "http://hmkcode.com/android-camera-taking-photos-camera/",
      "categories": ["Android"],
      "tags": ["android", "camera"]
    }
  ]
}
```

Check Network Connectivity

```
// check network connection  
public boolean isConnected(){  
    ConnectivityManager connMgr = (ConnectivityManager)  
        getSystemService(this.CONNECTIVITY_SERVICE);  
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();  
    if (networkInfo != null && networkInfo.isConnected())  
        return true;  
    else  
        return false;  
}
```

Get Data from URL

```
public String requestContent(String url) {
    HttpClient httpClient = new DefaultHttpClient\(\);
    String result = null; HttpResponse response =
null;
    InputStream instream = null;
    HttpGet httpget = new HttpGet\(url\);
    try {
        response = httpClient.execute(httpget);
        HttpEntity entity = response.getEntity();
        if (entity != null) {
            instream = entity.getContent();
            result = convertStreamToString\(instream\);
        }
    } catch (Exception e) {
        // manage exceptions
    }
}
```

```
finally {
    if (instream != null) {
        try {
            instream.close();
        }
        catch (Exception ex){

        }
    }
}

return result;
}
```


Convert Stream to String

```
// convert inputStream to String
```

```
private static String convertStreamToString(InputStream inputStream)
    throws IOException{
    BufferedReader bufferedReader = new BufferedReader( new
        InputStreamReader(inputStream));

    String line = "";
    String result = "";
    while((line = bufferedReader.readLine()) != null)
        result += line;

    inputStream.close();
    return result;
}
```

The Previous Code will not Work !!!



Sending Back a Result

- *we need to run network operations on a **Separate Thread** from Android 3.0 , Otherwise it will never work (Using **AsyncTask**)*
 - ◆ In the same class MainActivity.java add the following private class

```
private class HttpAsyncTask extends AsyncTask<String, Void, String> {
    @Override
    protected String doInBackground(String... urls) {
        return requestContent(urls[0]);
    }
    // onPostExecute displays the results of the AsyncTask.
    @Override
    protected void onPostExecute(String result) {
        //now you have JSON response in result
        // setListAdapter must not be called at doInBackground()
        // since it would be executed in separate Thread
    }
}
```

Use the AsyncTask

- *Now to make the HTTP GET request use the below code*

```
HttpAsyncTask httpTask= new HttpAsyncTask();
```

```
httpTask.execute("http://hmkcode.appspot.com/rest/controller/get.json");
```

- *Notice that `doInBackground()` & `onPostExecute()` will be called implicitly i.e. we are not calling them directly AsyncTask will call them for us, we just call the `execute()`.*

Working with JSON Data - I

```
{
  "articleList": [
    {
      "title": "Android Internet Connection Using HTTP GET (HttpClient)",
      "url": "http://hmkcode.com/android-internet-connection-using-http-get-httpclient/",
      "categories": ["Android"],
      "tags": ["android", "httpClient", "internet"]
    },
    {
      "title": " Android | Taking Photos with Android Camera ",
      "url": "http://hmkcode.com/android-camera-taking-photos-camera/",
      "categories": ["Android"],
      "tags": ["android", "camera"]
    }
  ]
}
```

Working with JSON Data - II

- Remember received data is passed to result argument of *AsyncTask.onPostExecute()*.
- Convert JSON data from String to *JSONObject*
`JSONObject json = new JSONObject(result);`
- *JSONObject Vs. JSONArray*
 - ◆ [...] is an array e.g. ["android" , "httpclient" , "internet"]
 - ◆ {...} is an object e.g. { tags:[...] , categories:[...] , url:"..." , title:"..." }
 - ◆ *JSONObject* can contain *JSONArray*
 - ◆ *JSONArray* can contain *JSONObject*

Working with JSON Data - III

- *Get the array of articles “articleList“, its length, get first article keys “names” & first article url.*

```
JSONObject json = new JSONObject(result); // convert String to JSONObject
JSONArray articles = json.getJSONArray("articleList"); // get articles array
articles.length(); // --> 2
articles.getJSONObject(0); // get first article in the array
articles.getJSONObject(0).names(); // get first article keys
//[title,url,categories,tags]
articles.getJSONObject(0).getString("url"); // return an article url
```

Showing Progress Dialog - I

- *Now we improve our App by displaying a progress dialog while the service is processing in the background.*

```
private class RequestItemsServiceTask
    extends AsyncTask<Void, Void, Void> {
    private ProgressDialog dialog =
        new ProgressDialog(ItemsListActivity.this);
    private List<MyItem> itemsList;

    @Override
    protected void onPreExecute() {
        dialog.setMessage("Please wait..");
        dialog.show();
    }
}
```


Showing Progress Dialog - II

- *Now we improve our App by displaying a progress dialog while the service is processing in the background.*

```
@Override
protected void onPostExecute(Void unused) {

    if (dialog.isShowing()) {
        dialog.dismiss();
    }
}
```

Assignment

- *Use the RESTful API from this link to get weather data for Cairo and show it in a ListView*

Open Weather MAP