CSE575-Section4: Calendar activity

1. This activity will apply the “GET” request which can use to parse the JSON objects into our App.
   1. Firstly, I will create a simple calendar list in JSON on the server,
   2. We will use PHP to output JSON. I have talked about this implementation on Section2

Calendar.php

<?php

$content = [ "Calendar" => [

[

"Date" =>"Sep. 13. 2017",

"Note" =>"Academic Year Begins",

],

[

"Date" =>"Sep. 21. 2017",

"Note" =>"Fall 2017 First Day of Classes",

],

[

"Date" =>"Sep. 23. 2017",

"Note" =>"Fall 2017 First Day of Saturday Classes",

],

[

"Date" =>"Sep. 28. 2017",

"Note" =>"Fall 2017 Last Day to Add Open Classes",

],

[

"Date" =>"Oct. 11. 2017",

"Note" =>"Fall 2017 Census",

],

[

"Date" =>"Oct. 16-27. 2017",

"Note" =>"Winter 2018 Graduate Advising",

],

[

"Date" =>"Oct. 20. 2017",

"Note" =>"Winter 2018 Registration Access e-mailed",

],

[

"Date" =>"Nov. 10. 2017",

"Note" =>"Veterans Day Holiday—Campus Closed",

],

[

"Date" =>"Nov. 24-26. 2017",

"Note" =>"Thanksgiving Day Holiday—Campus Closed",

],

[

"Date" =>"Nov. 10. 2017",

"Note" =>"Fall 2017 Last Day of Monday classes",

],

[

"Date" =>"Dec. 02. 2017",

"Note" =>"Fall 2017 Last Day of Saturday Classes",

],

[

"Date" =>"Dec. 04. 2017",

"Note" =>"Fall 2017 Last Day of Classes",

],

[

"Date" =>"Dec. 05-09. 2017",

"Note" =>"Fall 2017 Final Exams",

],

[

"Date" =>"Dec. 09. 2017",

"Note" =>"Fall Commencement",

],

[

"Date" =>"Dec. 23-31. 2017",

"Note" =>"Winter Break—Campus Closed",

],

[

"Date" =>"Jan. 01. 2018",

"Note" =>"New Year’s Day—Campus Closed",

],

[

"Date" =>"Jan. 05. 2018",

"Note" =>"Winter 2018 Full Refund Deadline",

],

[

"Date" =>"Jan. 06. 2018",

"Note" =>"Winter 2018 First Day of Saturday Classes",

],

[

"Date" =>"Jan. 08. 2018",

"Note" =>"Winter 2018 First Day of Classes",

],

[

"Date" =>"Jan. 12. 2018",

"Note" =>"Winter 2018 Last Day to Add Open Classes",

],

[

"Date" =>"Jan. 13-15. 2018",

"Note" =>"Martin Luther King Holiday—Campus Closed

\nNo Saturday and Monday classes",

],

[

"Date" =>"Jan. 29. 2018",

"Note" =>"Winter 2018 Census",

],

[

"Date" =>"Jan. 29-31. 2018",

"Note" =>"Spring 2018 Graduate Advising begins",

],

[

"Date" =>"Feb. 01-02. 2018",

"Note" =>"Spring 2018 Graduate Advising",

],

[

"Date" =>"Feb. 02. 2018",

"Note" =>"Spring 2018 Registration Access e-mailed",

],

[

"Date" =>"Feb. 05. 2018",

"Note" =>"Spring 2018 Advising begins",

],

[

"Date" =>"March. 17. 2018",

"Note" =>"Winter 2018 Last Day of Saturday Classes",

],

[

"Date" =>"March. 19. 2018",

"Note" =>"Winter 2018 Last Day of Classes",

],

[

"Date" =>"March. 20-24. 2018",

"Note" =>"Winter 2018 Final Exams",

],

[

"Date" =>"March. 26-29. 2018",

"Note" =>"Spring 2018 Break",

],

[

"Date" =>"March. 30. 2018",

"Note" =>"Cesar Chavez Holiday—Campus Closed",

],

[

"Date" =>"March. 31. 2018",

"Note" =>"Spring 2018 First Day of Saturday Classes",

],

[

"Date" =>"April. 02. 2018",

"Note" =>"Spring 2018 First Day of Classes",

],

[

"Date" =>"April. 06. 2018",

"Note" =>"Spring 2018 Last Day to Add Open Classes",

],

[

"Date" =>"April. 16-27. 2018",

"Note" =>"Summer 2018 Graduate Advising",

],

[

"Date" =>"April. 20. 2018",

"Note" =>"Spring 2018 Census",

],

[

"Date" =>"April. 30. 2018",

"Note" =>"Fall 2018 Graduate Advising begins",

],

[

"Date" =>"May. 01-11. 2018",

"Note" =>"Fall 2018 Graduate Advising",

],

[

"Date" =>"May. 04. 2018",

"Note" =>"Fall 2018 Registration Access e-mailed",

],

[

"Date" =>"May. 07. 2018",

"Note" =>"Fall 2018 Advising begins",

],

[

"Date" =>"May. 26-28. 2018",

"Note" =>"Memorial Day Holiday

\nNo Saturday and Monday classes",

],

[

"Date" =>"June. 09. 2018",

"Note" =>"Spring 2018 Last Day of Saturday Classes",

],

[

"Date" =>"June. 11. 2018",

"Note" =>"Spring 2018 Last Day of Classes",

],

[

"Date" =>"June. 12-16. 2018",

"Note" =>"Spring 2018 Final Exams",

],

[

"Date" =>"June. 16. 2018",

"Note" =>"Spring Commencement",

],

]

];

// Convert Array to JSON String

$objectJSON = json\_encode($content);

echo $objectJSON;

?>

The output should be somethings like this:



1. Next, create and style the UI for Calendar activity. This activity will use the ListView again. I have discussed about ListView on Section 2.

Activity\_calendar.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout 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="com.tando.school.Calendar"  
 android:orientation="vertical"  
 android:background="#FFF"**>  
 *<!--Logo -->* <**ImageView  
 android:id="@+id/logo"  
 android:layout\_width="150dp"  
 android:layout\_height="100dp"  
 android:background="@drawable/school\_logo"  
 android:layout\_gravity="center"**/>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Calendar"  
 android:textColor="#000"  
 android:textSize="30dp"  
 android:textStyle="bold"  
 android:gravity="center"**/>  
 <**ListView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:id="@+id/calendarList"**>  
 </**ListView**>  
  
</**LinearLayout**>

Note\*: we will use the id of the ListView later on, so it has to have an ID.

Next, create an XML layout file to style the ListView. Check section 2 for reference. I named this layout calendar\_listview.xml

calendar\_listview.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="10dp"**>  
 <**TextView  
 android:id="@+id/aDate"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingBottom="2dp"  
 android:paddingTop="6dp"  
 android:textColor=" #0080ff"  
 android:textSize="20dp"  
 android:textStyle="bold"  
 android:text="Sep. 23. 2017"**/>  
 <**TextView  
 android:id="@+id/aEvent"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingBottom="2dip"  
 android:textColor="#000"  
 android:textSize="15dp"  
 android:text="Fall 2017 last Day to Add Open Classes"  
 android:textStyle="bold"** />  
</**LinearLayout**>

1. Next, go to Calendar.java to make the activity functioning. Details of the codes are explained in the comments

Calendar.java

**package** com.tando.school;  
  
**import** android.app.ProgressDialog;  
**import** android.os.AsyncTask;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.widget.ListAdapter;  
**import** android.widget.ListView;  
**import** android.widget.SimpleAdapter;  
**import** android.widget.Toast;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.util.ArrayList;  
**import** java.util.HashMap;  
  
**public class** Calendar **extends** AppCompatActivity {  
 *//returns the simple name of the underlying class, easier to track in the Android monitor* **private** String **TAG** = Calendar.**class**.getSimpleName();  
 *//Prpgress bar while retrieving data* **private** ProgressDialog **pDialog**;  
 *//Declare ListView* **private** ListView **calendarList**;  
  
 *// URL to get calendar JSON* **private static** String *url* =**"https://schoolserver-tand089.c9users.io/Calendar.php"**;  
 *//Declare an array to store the list of items* ArrayList<HashMap<String, String>> **cList**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_calendar***);  
  
 **cList** = **new** ArrayList<>();  
  
 **calendarList** = (ListView) findViewById(R.id.***calendarList***);  
  
 *////execute the GetEvents class* **new** GetEvents().execute();  
 }  
 */\*\*  
 \* Async task class to get json by making HTTP call  
 \*/  
 //Create a GetEvents class to make http calls on background thread* **private class** GetEvents **extends** AsyncTask<Void, Void, Void> {  
 @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
 *// Showing progress dialog* **pDialog** = **new** ProgressDialog(Calendar.**this**);  
 **pDialog**.setMessage(**"Loading..."**);  
 **pDialog**.setCancelable(**false**);  
 **pDialog**.show();  
  
 }  
 @Override  
 **protected** Void doInBackground(Void... arg0) {  
 HttpHandler httpHandler = **new** HttpHandler();  
 *// Making a request to url and getting response* String jsonStr = httpHandler.makeServiceCall(*url*);  
  
 Log.*e*(**TAG**, **"Response from url: "** + jsonStr);  
 **if** (jsonStr != **null**) {  
 **try** {  
 JSONObject jsonObj = **new** JSONObject(jsonStr);  
  
 *// Getting JSON Array node* JSONArray calendarList = jsonObj.getJSONArray(**"Calendar"**);  
  
 *// looping through All Events* **for** (**int** i = 0; i < calendarList.length(); i++) {  
 JSONObject c = calendarList.getJSONObject(i);  
 *//get string from the json file* String Date = c.getString(**"Date"**);  
 String Note = c.getString(**"Note"**);  
  
  
 *// tmp hash map for single event* HashMap<String, String> calendars = **new** HashMap<>();  
  
 *// adding each child node to HashMap key => value* calendars.put(**"Date"**, Date);  
 calendars.put(**"Note"**, Note);  
  
  
 *// adding contact to cList* **cList**.add(calendars);  
 }  
 *//check if server error* } **catch** (**final** JSONException e) {  
 Log.*e*(**TAG**, **"Json parsing error: "** + e.getMessage());  
 runOnUiThread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 Toast.*makeText*(getApplicationContext(),  
 **"Json parsing error: "** + e.getMessage(),  
 Toast.***LENGTH\_LONG***)  
 .show();  
 }  
 });  
 }  
 } *//End if* **else** {  
 Log.*e*(**TAG**, **"Couldn't get json objects from server."**);  
 runOnUiThread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 Toast.*makeText*(getApplicationContext(),  
 **"errors!"**,  
 Toast.***LENGTH\_LONG***)  
 .show();  
 }  
 });  
  
 }  
 **return null**;  
 } *//End doing background* @Override  
 **protected void** onPostExecute(Void result) {  
 **super**.onPostExecute(result);  
 *// Dismiss the progress dialog* **if** (**pDialog**.isShowing())  
 **pDialog**.dismiss();  
 */\*\*  
 \* Updating parsed JSON data into ListView  
 \* \*/* ListAdapter adapter = **new** SimpleAdapter(  
 Calendar.**this**, **cList**,  
 R.layout.***calendar\_listview***, **new** String[]{**"Date"**, **"Note"**}, **new int**[]{R.id.***aDate***,  
 R.id.***aEvent***});  
  
 **calendarList**.setAdapter(adapter);  
  
 }  
  
  
 }*//End AsyncTask*}

1. Run the app

 