SESSION 02

export PATH=$PATH:/Users/sajib/Library/Android/sdk/platform-tools/

- AsyncTask - [DEPRECATED] - [https://developer.android.com/reference/android/os/AsyncTask]
    - AsyncTask<A, B, C> : android.os.AsyncTask<Params, Progress, Result>
        - protected C doInBackground(A)
        - protected void onPostExecute(C)
    - execute(A)
- okHttp - [https://square.github.io/okhttp/]
- rapidAPI - [https://rapidapi.com/montanaflynn/api/fifa-world-cup]
- JSON
    - JSONArray
        - length()
        - getJSONObject(index)
    - JSONObject
        - get()
        - getString()
        - getInt()
- ListView - [https://developer.android.com/reference/android/widget/ListView]
- ArrayAdapter - [https://developer.android.com/reference/android/widget/ArrayAdapter]
- ListAdapter - [https://developer.android.com/reference/android/widget/ListAdapter]. Extended Adapter that is the bridge between

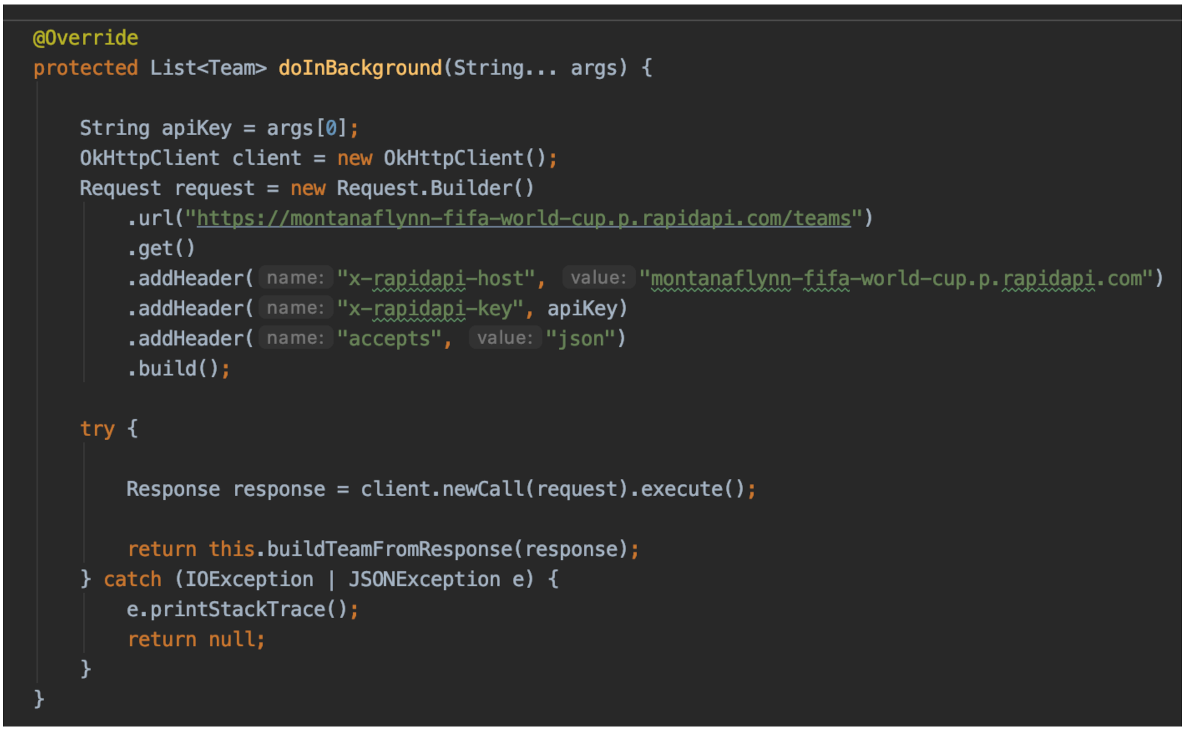
a ListView and the data that backs the list. Frequently that data comes from a Cursor, but that is not required. The ListView can display any data provided that it is wrapped in a ListAdapter.

- BaseAdapter - [https://developer.android.com/reference/android/widget/BaseAdapter]
  - attachToRoot - [https://stackoverflow.com/questions/12567578/what-does-the-layoutinflater-attachtoroot-parameter-mean]
- RecyclerView - [https://developer.android.com/reference/kotlin/androidx/recyclerview/widget/RecyclerView]
  - Adapter
    - onCreateViewHolder - called when the RecyclerView instantiates a new ViewHolder instance
    - onBindViewHolder - called when the RecyclerView wants to populate the view with data from our model
    - getItemCount - returns the number of items in our data source
  - DataSource
  - ViewHolder - determines how an individual entry is displayed
  - LayoutManager - determines how the collection of items is displayed. It determines when to recycle item views that are no longer visible
    - LinearLayoutManager
    - GridLayoutManager
    - StaggeredGridLayoutManager
  - implementation 'androidx.recyclerview:recyclerview:1.1.0'
  - RecyclerView asks the Adapter multiple times in the process of laying out the view about how

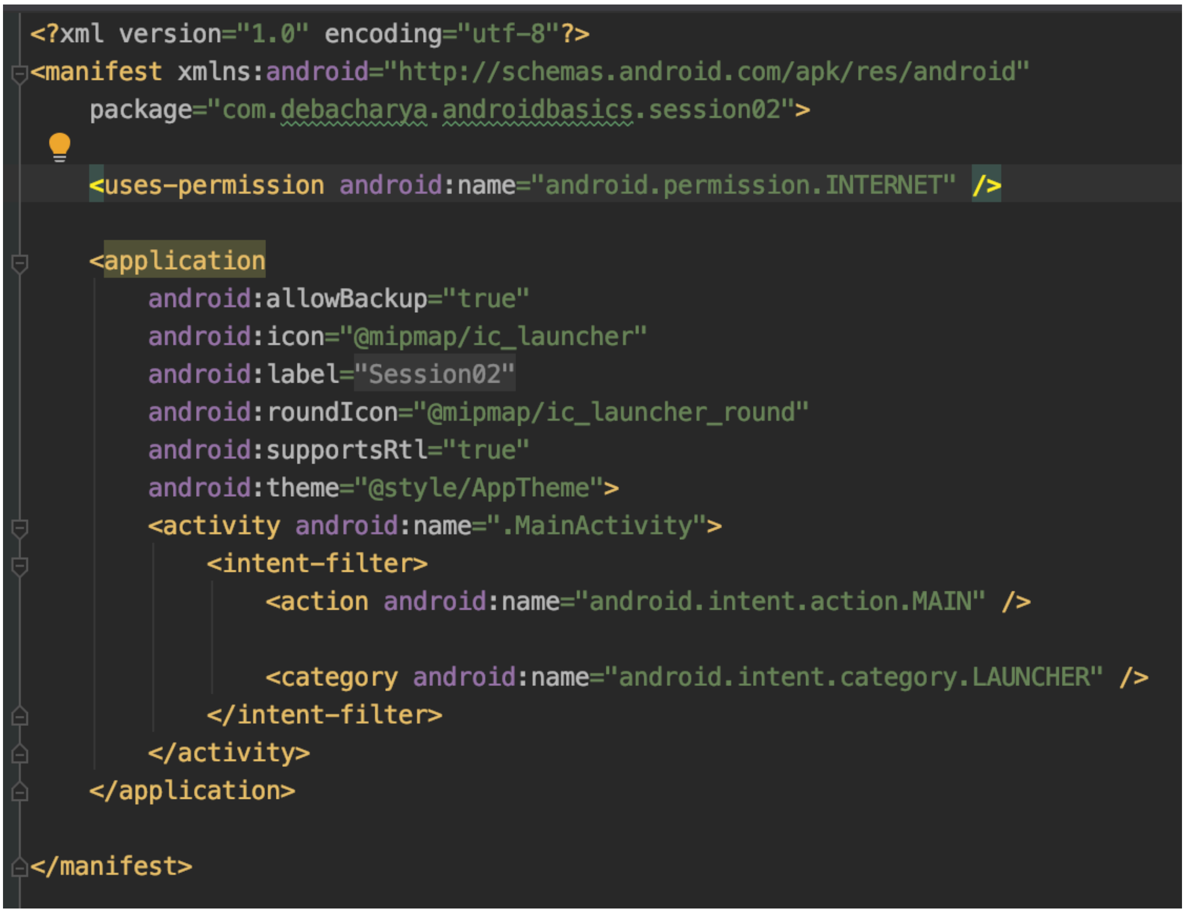
many data items need to be viewed

Add okHttp support: implementation 'com.squareup.okhttp3:okhttp:4.8.0'

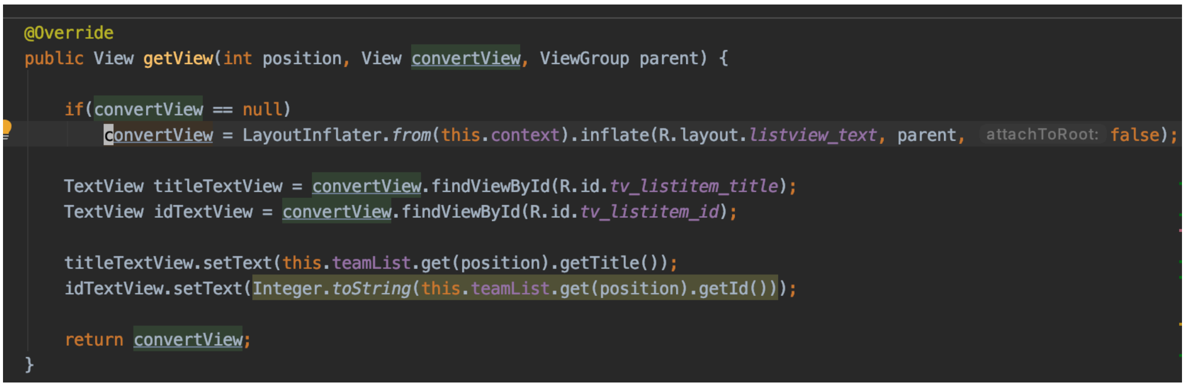
FIFA API URL : https://rapidapi.com/montanaflynn/api/fifa-world-cup



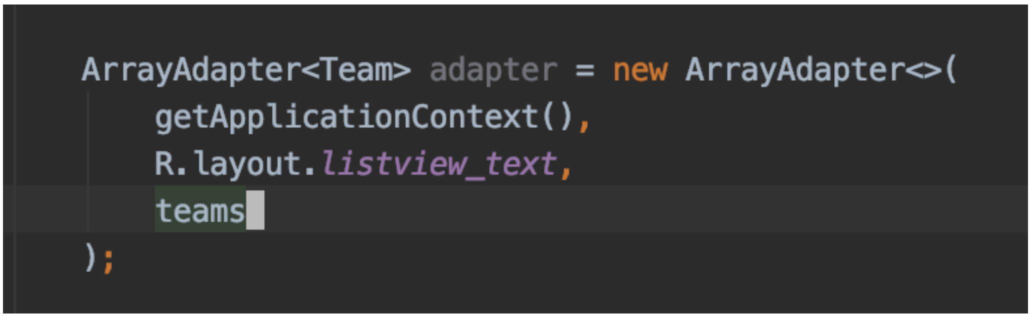
```java
@Override
protected List<Team> doInBackground(String... args) {

    String apiKey = args[0];
    OkHttpClient client = new OkHttpClient();
    Request request = new Request.Builder()
        .url("https://montanaflynn-fifa-world-cup.p.rapidapi.com/teams")
        .get()
        .addHeader( name: "x-rapidapi-host",  value: "montanaflynn-fifa-world-cup.p.rapidapi.com")
        .addHeader( name: "x-rapidapi-key", apiKey)
        .addHeader( name: "accepts",  value: "json")
        .build();

    try {

        Response response = client.newCall(request).execute();

        return this.buildTeamFromResponse(response);
    } catch (IOException | JSONException e) {
        e.printStackTrace();
        return null;
    }
}
```



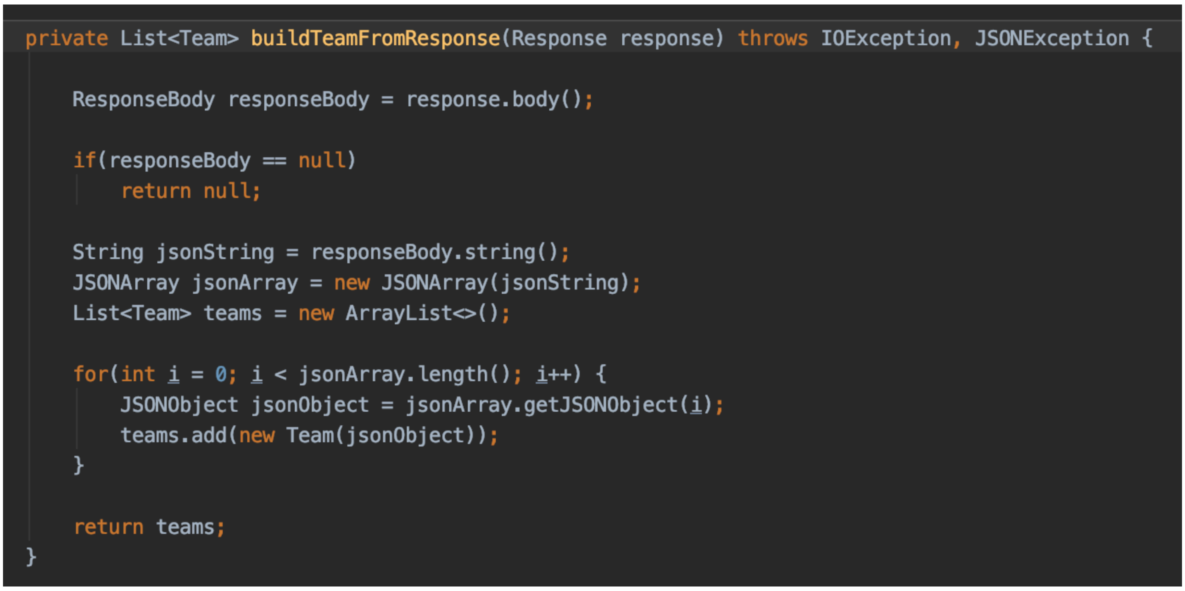
```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.debacharya.androidbasics.session02">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Session02"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```



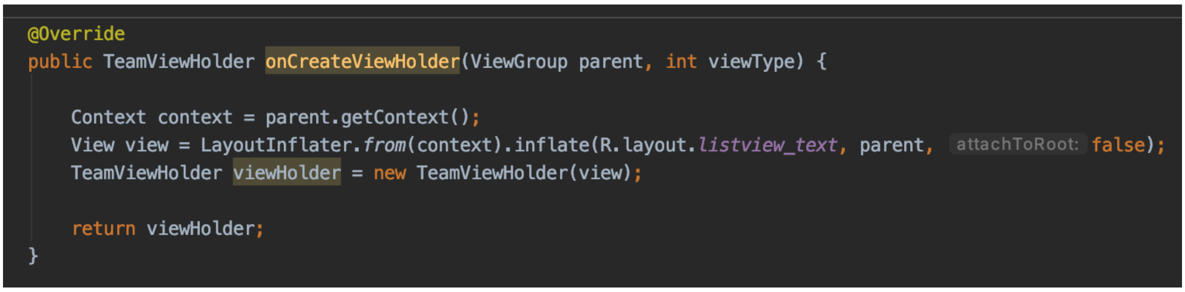
```java
@Override
public View getView(int position, View convertView, ViewGroup parent) {

    if(convertView == null)
        convertView = LayoutInflater.from(this.context).inflate(R.layout.listview_text, parent,  attachToRoot: false);

    TextView titleTextView = convertView.findViewById(R.id.tv_listitem_title);
    TextView idTextView = convertView.findViewById(R.id.tv_listitem_id);

    titleTextView.setText(this.teamList.get(position).getTitle());
    idTextView.setText(Integer.toString(this.teamList.get(position).getId()));

    return convertView;
}
```



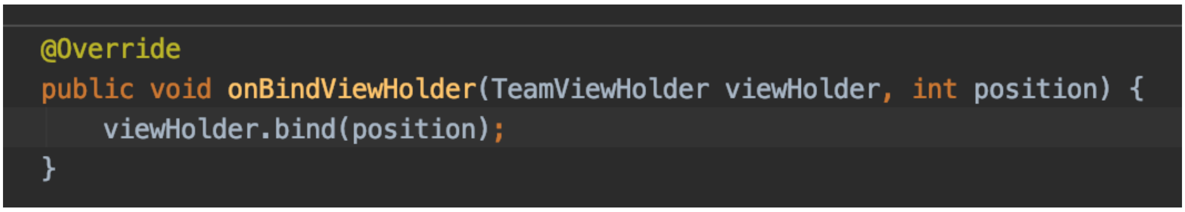
```java
ArrayAdapter<Team> adapter = new ArrayAdapter<>(
    getApplicationContext(),
    R.layout.listview_text,
    teams
);
```



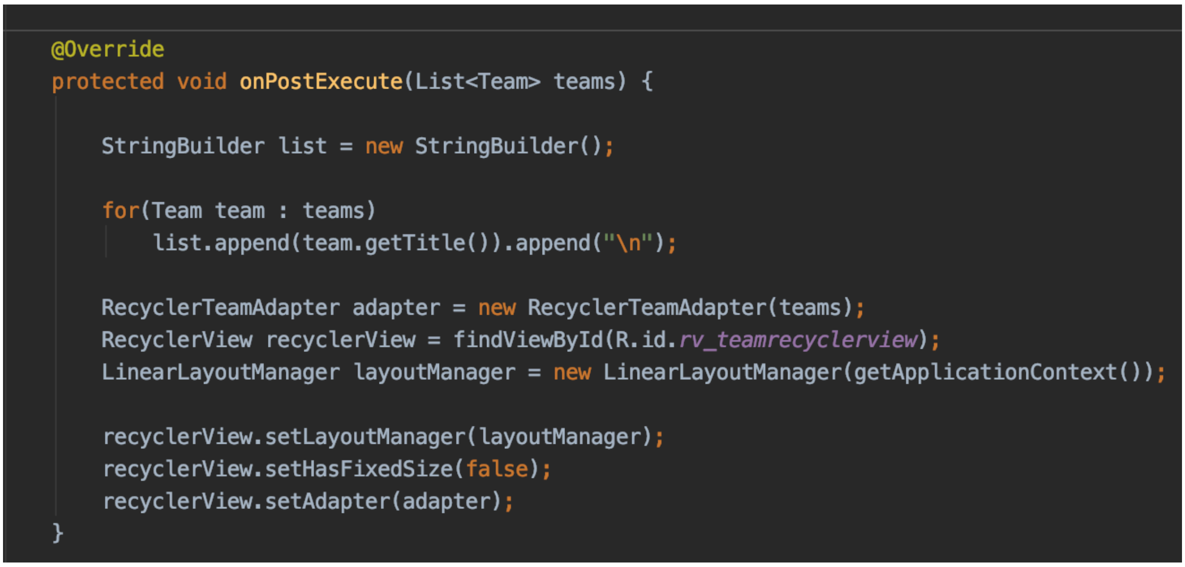
```
private List<Team> buildTeamFromResponse(Response response) throws IOException, JSONException {

    ResponseBody responseBody = response.body();

    if(responseBody == null)
        return null;

    String jsonString = responseBody.string();
    JSONArray jsonArray = new JSONArray(jsonString);
    List<Team> teams = new ArrayList<>();

    for(int i = 0; i < jsonArray.length(); i++) {
        JSONObject jsonObject = jsonArray.getJSONObject(i);
        teams.add(new Team(jsonObject));
    }

    return teams;
}
```



```
@Override
public TeamViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

    Context context = parent.getContext();
    View view = LayoutInflater.from(context).inflate(R.layout.listview_text, parent,  attachToRoot: false);
    TeamViewHolder viewHolder = new TeamViewHolder(view);

    return viewHolder;
}
```



```
@Override
public void onBindViewHolder(TeamViewHolder viewHolder, int position) {
    viewHolder.bind(position);
}
```



```
@Override
protected void onPostExecute(List<Team> teams) {

    StringBuilder list = new StringBuilder();

    for(Team team : teams)
        list.append(team.getTitle()).append("\n");

    RecyclerTeamAdapter adapter = new RecyclerTeamAdapter(teams);
    RecyclerView recyclerView = findViewById(R.id.rv_teamrecyclerview);
    LinearLayoutManager layoutManager = new LinearLayoutManager(getApplicationContext());

    recyclerView.setLayoutManager(layoutManager);
    recyclerView.setHasFixedSize(false);
    recyclerView.setAdapter(adapter);
}
```