Android Developer Fundamentals V2

Background Tasks

Lesson 7
Internet connection

Steps to connect to the Internet

1. Add permissions to Android Manifest
2. Check Network Connection
3. Create Worker Thread
4. Implement background task
    a. Create URI
    b. Make HTTP Connection
    c. Connect and GET Data
5. Process results
    a. Parse Results
    b. Display results

Permissions in AndroidManifest

Internet

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

Check Network State

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

Getting Network information

- ConnectivityManager
  - Answers queries about the state of network connectivity
  - Notifies applications when network connectivity changes
- NetworkInfo
  - Describes status of a network interface of a given type
  - Mobile or Wi-Fi

Check if network is available

```
ConnectivityManager connMgr = (ConnectivityManager)
            getSystemService(Context.CONNECTIVITY_SERVICE);

NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();

if (networkInfo != null && networkInfo.isConnected()) {
    // Create background thread to connect and get data
    new DownloadWebpageTask().execute(stringUrl);
} else {
    textView.setText("No network connection available.");
}
```

Check for WiFi & Mobile

```
NetworkInfo networkInfo =
    connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
boolean isWifiConn = networkInfo.isConnected();

networkInfo =
    connMgr.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);
boolean isMobileConn = networkInfo.isConnected();
```

Worker Thread

Use Worker Thread

- AsyncTask—very short task, or no result returned to UI
- AsyncTaskLoader—for longer tasks, returns result to UI
- Background Service—later chapter

Background work

In the background task (for example in `doInBackground()`)

1. Create URI
2. Make HTTP Connection
3. Download Data

URI = Uniform Resource Identifier

String that names or locates a particular resource

- file://
- http:// and https://
- content://

```
String url = “https://www.googleapis.com/books/v1/volumes?
    q=pride+prejudice&maxResults=5&printType=books”;
URL requestURL = new URL(url);
```

Create a HttpURLConnection

```
HttpURLConnection conn =
   (HttpURLConnection) requestURL.openConnection();
conn.setReadTimeout(10000 /* milliseconds */);
conn.setConnectTimeout(15000 /* milliseconds */);
conn.setRequestMethod("GET");
conn.setDoInput(true);
```

Connect and get response

```
conn.connect();
int response = conn.getResponseCode();

InputStream is = conn.getInputStream();
String contentAsString = convertIsToString(is, len);
return contentAsString;
```

Close connection and stream

```
try {}
catch {}
finally {
        conn.disconnect();
         if (is != null) {
             is.close();
         }
}
```

Convert input stream into a string

```
public String convertIsToString(InputStream stream, int len)
    throws IOException, UnsupportedEncodingException {

    Reader reader = null;
    reader = new InputStreamReader(stream, "UTF-8");
    char[] buffer = new char[len];
    reader.read(buffer);
    return new String(buffer);
}
```

BufferedReader is more efficient

```
StringBuilder builder = new StringBuilder();
BufferedReader reader =
    new BufferedReader(new InputStreamReader(inputStream));
String line;
while ((line = reader.readLine()) != null) {
   builder.append(line + "\n");
}
if (builder.length() == 0) {
   return null;
}
resultString = builder.toString();
```

HTTP Client Connection Libraries

Make a connection using libraries

- Use a third party library like OkHttp or Volley
- Makes networking requests for Android apps easier and faster
- Much less code

Make a connection using libraries

Add the library to your project in app's build.gradle file:

```
dependencies {
    implementation 'com.android.volley:volley:1.2.1’
    implementation 'com.squareup.okhttp3:okhttp:4.9.2'
}
```

Add the INTERNET permission to AndroidManifest.xml file:

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

OkHttp

```
OkHttpClient client = new OkHttpClient();
Request request = new Request.Builder()
    .url("http://...").build();
client.newCall(request).enqueue(new Callback() {
    @Override
    public void onResponse(Call call, final Response response)
        throws IOException {
        try {
            String responseData = response.body().string();
            // Do something with response
            ...
        } catch (JSONException e) {}
    }
});
```

Volley

```
RequestQueue queue = Volley.newRequestQueue(this);
String url ="http://...";

StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
            new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        // Do something with response
        ...
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {}
});
queue.add(stringRequest);
```

Parse Results

Parsing the results

- Implement method to receive and handle results (`onPostExecute()`)
- Response is often JSON or XML

Parse results using helper classes

- JSONObject, JSONArray
- XMLPullParser—parses XML

JSON basics

```
{
 "population":1252000000,
 "country":"India",
 "cities":["New Delhi","Mumbai","Kolkata","Chennai"]
}
```

JSONObject basics

```
JSONObject jsonObject = new JSONObject(response);

String nameOfCountry = (String) jsonObject.get("country");
long population = (Long) jsonObject.get("population");
JSONArray listOfCities = (JSONArray) jsonObject.get("cities");

Iterator<String> iterator = listOfCities.iterator();
while (iterator.hasNext()) {
    // do something
}
```

Another JSON example

```
{"bookInfo": {
  "id": "...",
  "description": "...",
  "title": "...",
  "authors": [
      {"id": "6046989264", "name": "Jane Austen"},
      {"id": "8126905492", "name": "Conan Doyle"},
      {"id": "1631590766", "name": "Cris Silvestri"}
    ]
  }
}
```

Another JSON example

Get "name" value of the 3rd item in the "authors" array

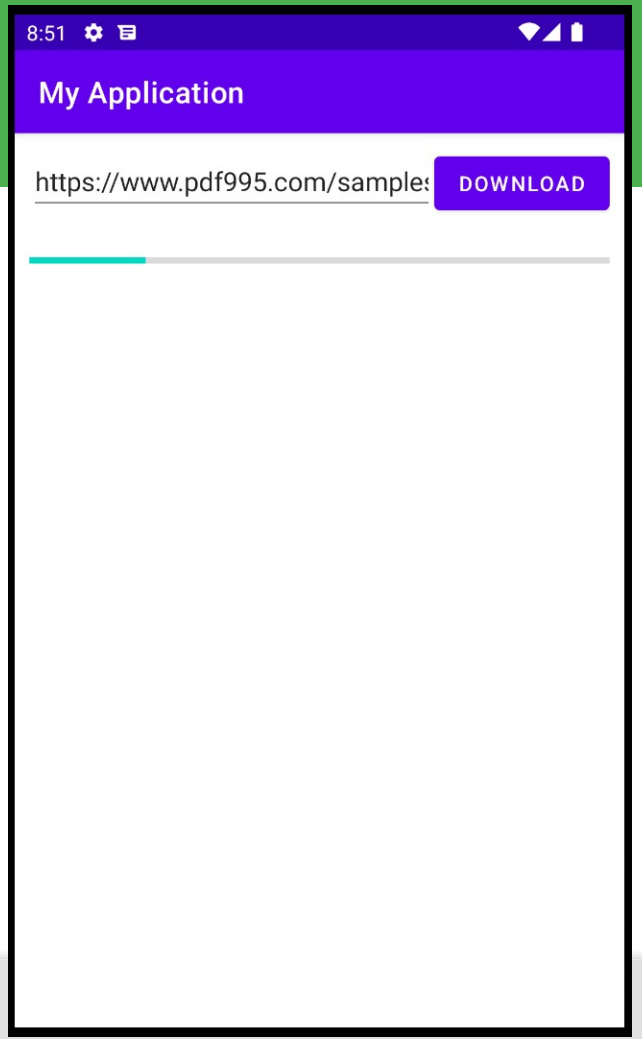
```
JSONObject data = new JSONObject(responseString);
JSONObject bookInfo = data.getJSONObject("bookInfo");
JSONArray authors = bookInfo.getJSONArray("authors");
JSONObject thirdItem = authors.getJSONObject(2);
String name = thirdItem.getString("name");
```

Demo OkHttp

```
final int CHUNK_SIZE = 1024;

OkHttpClient client = new OkHttpClient();
Request request = new Request.Builder()
        .url(downloadUrl)
        .build();
Response response = client.newCall(request).execute();
long totalSize = response.body().contentLength();

byte[] buffer = new byte[CHUNK_SIZE];
int bytesRead;
long bytesDownloaded = 0;
```



Sample file: https://www.pdf995.com/samples/pdf.pdf

Demo OkHttp

`do … while (bytesRead != -1)`

```
int nextSize = (int) Math.min(CHUNK_SIZE, totalSize - bytesDownloaded);
long nextOffset = bytesDownloaded + nextSize - 1;
request = new Request.Builder()
        .url(downloadUrl)
        .header("Range", "bytes=" + bytesDownloaded + "-" + nextOffset)
        .build();

response = client.newCall(request).execute();
bytesRead = response.body().byteStream().read(buffer);
bytesDownloaded += bytesRead;

int progress = (int) (bytesDownloaded * 100/totalSize);
```

Demo OkHttp

```
...

request = new Request.Builder()
        .url(downloadUrl)
        .header("Range", "bytes=" + bytesDownloaded + "-" + next)
        .build();

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

if (!response.isSuccessful()) {
    throw new IOException("Unexpected code " + response);
}

...
```

```
do … while (bytesRead != -1
            && bytesDownloaded < totalSize)
```

Learn more

- Connect to the Network Guide
- Managing Network Usage Guide
- HttpURLConnection reference
- ConnectivityManager reference
- InputStream reference

What's Next?

- Concept Chapter: 7.2 Internet connection
- Practical: 7.2 AsyncTask and AsyncTaskLoader

END