Android Networking – Part 1

Minstrel Chiu

Outline

- Required Permissions
- HTTP Networking
- Threads & Parallelism
- Socket Networking

Required Permissions (1)

- Sample-Permission1 – Permission required demo

Required Permissions (2)

- android.permission.INTERNET (mandatory)
- android.permission.ACCESS_NETWORK_STATE (optional)

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

HTTP Networking (1)

- URL (Android Native)
- HttpURLConnection (Android Native)
- OkHttp (Square)
- Deprecated
  - ~~HttpClient, AndroidHttpClient~~ (Apache)
- No New Features
  - ~~Volley~~ (Google)

HTTP Networking (2)

- URL – RFC 2396 & RFC 2732
- http://www.example.com:1080/docs/resource1.html?s=test&k=123
  - Protocol – http
  - Host – www.example.com
  - Port – 1080
  - Path – /docs/resource1.html
  - File – /docs/resource1.html?s=test&k=123
  - Query – s=test&k=123
- Sample-URL-1 – URL parsing demo

HTTP Networking (3)

- HttpURLConnection GET
  1. Instantiate a object of URL class
  ```
  URL url = new URL("http://httpbin.org/stream/50");
  ```
  2. Call openConnection method
  ```
  HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
  ```
  3. Call connect method
  ```
  urlConnection.connect();
  ```
  4. Read website contents from InputStream
  ```
  InputStream is = urlConnection.getInputStream();
  ```
  5. Release the connection by disconnect
  ```
  urlConnection.disconnect();
  ```

HTTP Networking (4)

- HttpURLConnection Extras
  - Connection Timeout – setConnectTimeout
    - milliseconds
  - Read Timeout – setReadTimeout
    - Milliseconds
  - Do Output – setDoOutput, default is false
  - Request Method – setRequestMethod, default is GET
    - GET/POST/PUT/DELETE/HEAD/OPTIONS
  - Get Response Code – getResponseCode
    - 1xx/2xx/3xx/4xx/5xx

HTTP Networking (5)

- Sample-Http-1 – HttpURLConnection GET demo

HTTP Networking (6)

- HttpURLConnection POST
  - setDoOutput(true)
  - new Uri.Builder().appendQueryParameter(param1, value1).build().getEncodedQuery();
  - Write to OutputStream and then read from InputStream

HTTP Networking (7)

- Sample-Http-2 – HttpURLConnection POST demo

HTTP Networking (8)

- OkHttp GET – Much easier than HttpURLConnection
  1. Import latest OkHttp lib in *build.gragle*
     ```
     compile 'com.squareup.okhttp3:okhttp:3.7.0'
     ```
  2. Instantiate an OkHttpClient object
     ```
     OkHttpClient client = new OkHttpClient();
     ```
  3. Create a Request
     ```
     Request request = new Request.Builder().url("http://httpbin.org/stream/50").build();
     ```
  4. Execute a newCall and get Response
     ```
     Response response = client.newCall(request).execute();
     ```
     ```
     return response.body().string();
     ```

HTTP Networking (9)

- Sample-Http-3 – OkHttp GET demo

HTTP Networking (10)

- OkHttp POST– Much easier than HttpURLConnection
  - RequestBody body = new FormBody.Builder().add(param1, value1).build()
  - Request request = new Request.Builder().url(url).post(body).build()

HTTP Networking (11)

- Sample-Http-4 – OkHttp POST demo

Threads & Parallelism (1)

- Sample-Thread-1 – HttpURLConnection with StrictMode enabled demo
- StrictMode is a developer tools and usually been disabled in real products

Threads & Parallelism (2)

- NetworkOnMainThreadException
  - The exception that is thrown when an application attempts to perform a networking operation on its main thread.

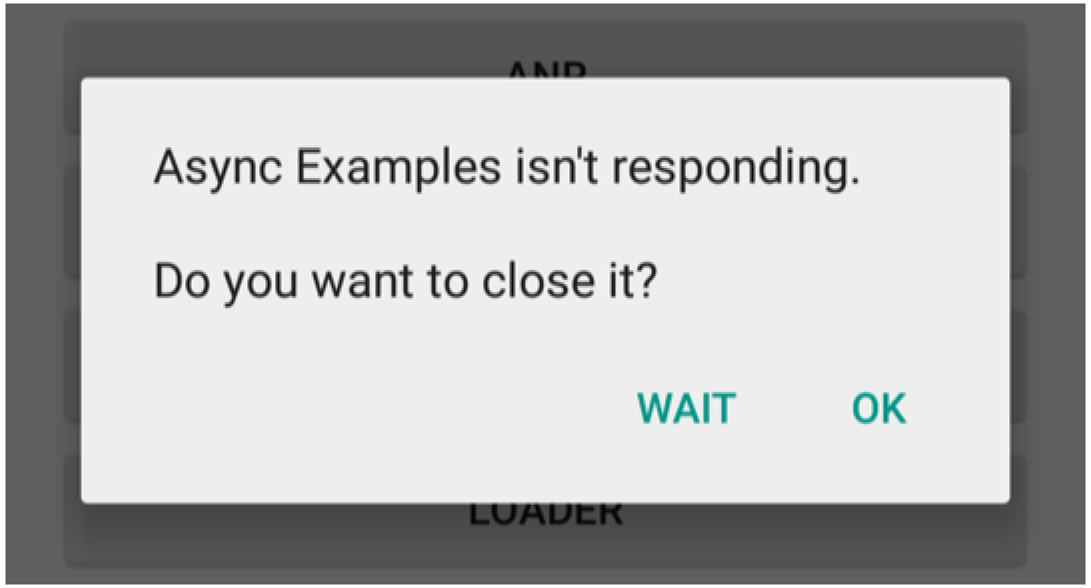
    This is only thrown for applications targeting the Honeycomb SDK or higher. Applications targeting earlier SDK versions are allowed to do networking on their main event loop threads, but it's heavily discouraged.

Threads & Parallelism (3)

- Main Thread
  - Runs application code from the queue one by one
  - a.k.a. UI Thread
- Background Thread
  - Application can have many background threads
  - For operations to perform that are not instantaneous
  - a.k.a. Worker Thread
- Rules
  - Do not block the UI thread
  - Do not access the Android UI toolkit from outside the UI thread

Threads & Parallelism (4)

- ANR (Application Not Responding)
  - 5 second input event timeout



Async Examples isn't responding.

Do you want to close it?

Threads & Parallelism (5)

- Thread
- Thread + Handler
- AsyncTask

Threads & Parallelism (6)

- Sample-Thread-2 – HttpURLConnection on background thread demo

Threads & Parallelism (7)

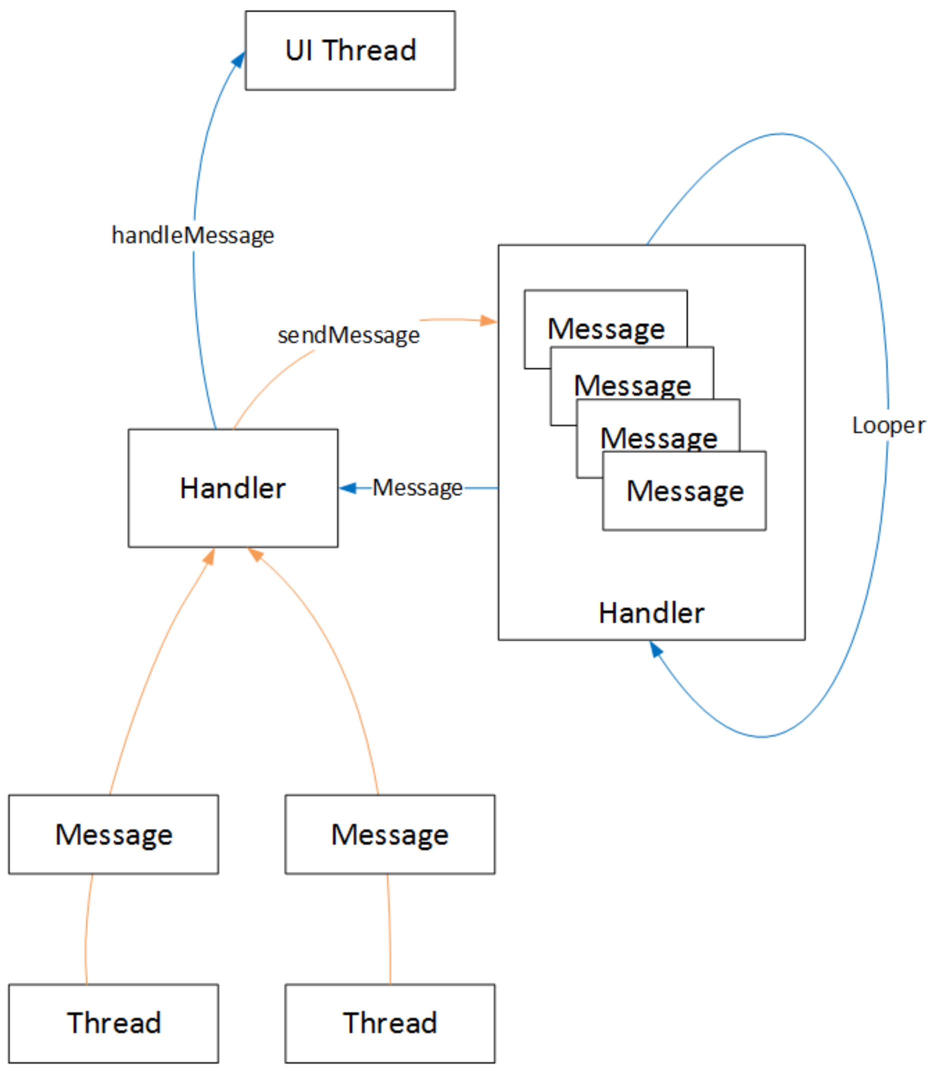
- CalledFromWrongThreadException
  - Only the original thread that created a view hierarchy can touch its views
- The tasks that you run on a thread from a thread pool aren't running on your UI thread, they don't have access to UI objects.

Threads & Parallelism (8)

- Sample-Thread-3 – Communicating with UI thread demo

Threads & Parallelism (9)

- Thread + Handler



Threads & Parallelism (10)

- Thread + Handler
  - Send Message
    - Handler. sendEmptyMessage(int what)
    - Handler. sendMessage(Message msg)
    - Handler. obtainMessage(int what, Object obj)
  - Handle Message
    - Handler.handleMessage(Message msg)

Threads & Parallelism (11)

- Sample-Thread-4 – HttpURLConnection with background thread and handler demo

Threads & Parallelism (12)

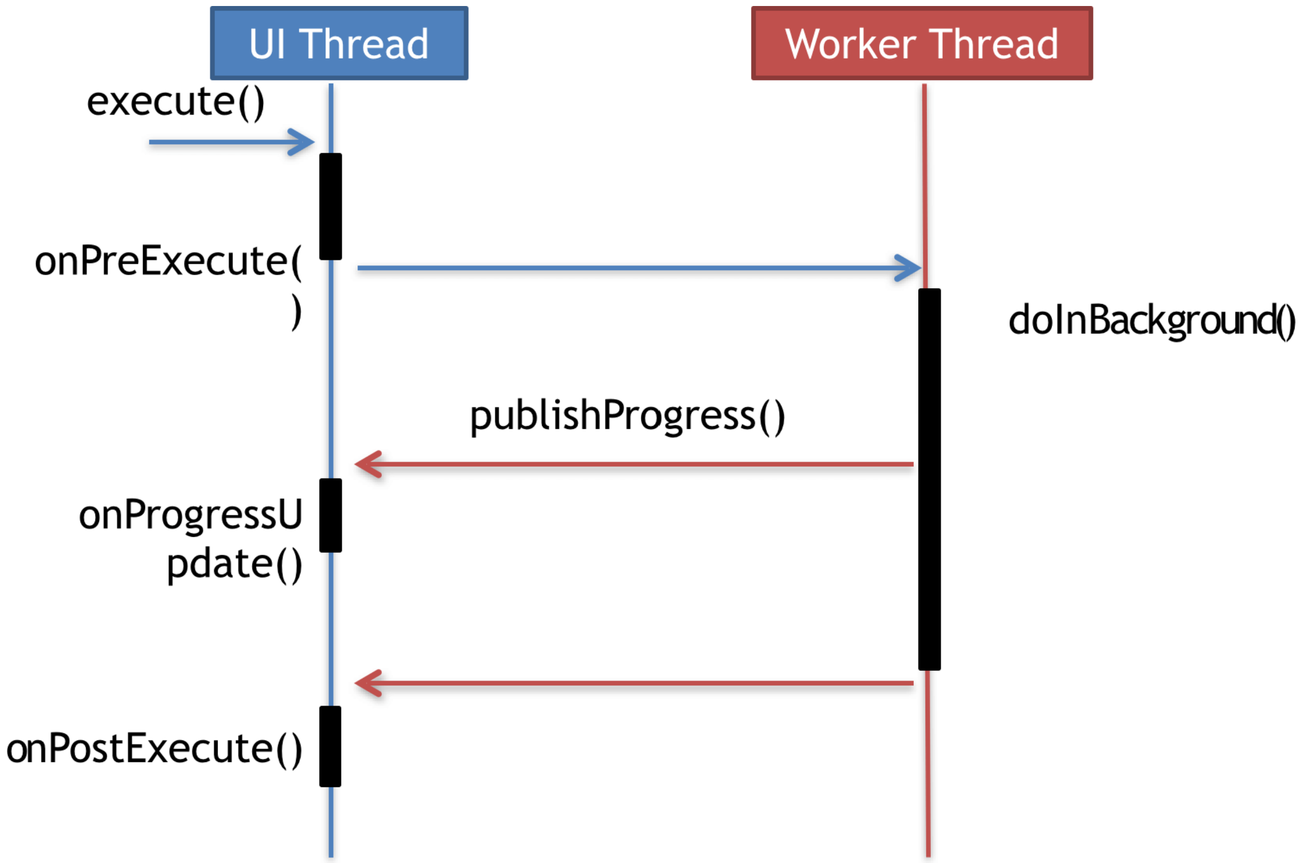
- AsyncTask
  - AsyncTask enables proper and easy use of the UI thread. This class allows you to perform background operations and publish results on the UI thread without having to manipulate threads and/or handlers.
- android.os.AsyncTask<Params, Progress, Result>

Threads & Parallelism (13)

- AsyncTask
  - onPreExecute – Runs on the UI thread before doInBackground
  - doInBackground – Runs on a background thread, and call publishProgress to publish updates on the UI thread
  - onProgressUpdate – Runs on the UI thread to get progress update from publishProgress
  - onPostExecute – Runs on the UI thread after doInBackground

Threads & Parallelism (14)

- AsyncTask



References: How To Simplify Networking In Android: Introducing The Volley HTTP Library

Threads & Parallelism (15)

- Sample-AsyncTask-1 – HttpURLConnection with AsyncTask demo
- Sample-AsyncTask-2 – OkHttp with AsyncTask demo

Threads & Parallelism (16)

- OkHttp + Background Thread
  1. AsyncTask
  2. OkHttp Callback
     - void onFailure(Call call, IOException e)
     - void onResponse(Call call, Response response)

Threads & Parallelism (17)

- Sample-OkHttp-Async-1 – OkHttp with AsyncTask demo
- Sample-OkHttp-Async-2 – OkHttp with Callback demo

Socket Networking (1)

- TCP Socket
  - FTP – port 20, 21
  - Telnet – port 23
  - DNS – port 53
  - HTTP – port 80
  - POP3 – port 110
- UDP Socket
  - DNS – port 53

Socket Networking (2)

- Socket
  1. Instantiate a object of Socket class with address and port
     ```
     Socket socket = new Socket("ptt.cc", 23);
     ```
  2. Read contents from InputStream
     ```
     InputStream is = socket.getInputStream();
     ```
  3. Release the connection by close
     ```
     socket.close();
     ```

Socket Networking (3)

- Sample-Socket-1 – Socket demo

Questions?

References

- Connecting to the Network
- Processes and Threads
- Communicating with the UI Thread
- Keeping Your App Responsive
- Udacity - Android Basics: Networking
- Google Samples - android NetworkConnect