Network Operations


Repo: https://github.com/karlmorris/AndroidNetwork
Overview

- Network operations in Android
- Patterns and examples
- Requirements and considerations
Networks in Android

• Mobile devices can belong/connect to many networks
  – e.g. Mobile, WiFi

• ConnectivityManager provides access to information and services regarding networks
  – Responds to queries, auto switches

• You can build applications with granular control over network usage
Ensuring Network is Connected

• Clean and proper

```java
ConnectivityManager connMgr = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
if (networkInfo != null && networkInfo.isConnected()) {
    // fetch data
} else {
    // display error
}
```
Ensuring Network is Connected

• Quick and dirty

    try {
        // fetch data
    } catch (Exception e) {
        // display error
    }
Network Operations and Threads

- Threads allow asynchronous operations
- Useful for offloading slow or heavy tasks
- Mandatory for network operations
Basic Example

Thread t = new Thread() {
    public void run() {
        try{  
            URL url = new URL(http://www.example.com);
            BufferedReader reader = new BufferedReader(
                new InputStreamReader(
                    url.openStream()));
            String response = reader.readLine();
            Message msg = Message.obtain();
            msg.obj = response;
            responseHandler.sendMessage(msg);  
        } catch (Exception e) {}  
    }
};
t.start();
Binary Example

...

Bitmap bitmap = BitmapFactory.decodeStream(is);
msg.obj = bitmap;
responseHandler.sendMessage(msg);
...

Handler:
...

ImageView imageView = (ImageView) findViewById(R.id.image_view);
imageView.setImageBitmap((Bitmap) msg.obj);
...

Network Permissions

android.permission INTERNET
- Allows applications to open network sockets

android.permission.ACCESS_NETWORK_STATE
- Allows applications to access information about networks
Considerations

• Implement a Preference Activity and respond to preference changes
  - Use an intent filter to specify the manager activity:
    ACTION_MANAGE_NETWORK_USAGE
• Detect network changes and change behavior accordingly
Other considerations

- Change update frequency based on battery life
- API 23 introduced Doze and App Standby
  - Network operations are suspended when device is unplugged and unused for a long period
  - Wake locks are ignored