

Internet mit Android

Problemstellung

- Habe ich derzeit Internet?
- Wie kann man Daten aus dem Internet herunterladen/hochladen?
- Wie viel kostet es den Benutzer bei aktueller Verbindung?
 - WLAN / Mobiles Netz / Roaming

Ziel

- Habe ich derzeit Internet?
- Wie kann man Daten aus dem Internet herunterladen/hochladen?
- Wie viel kostet es den Benutzer bei aktueller Verbindung?
 - WLAN / Mobiles Netz / Roaming
- Einfaches Feststellen, ob Verbindung möglich ist bzw. aktuelles Netz
- Einfache Schnittstelle um Daten zu transferieren
- Auf Netzwechsel reagieren

ConnectivityManager

- Überwachen von Verbindungen
- Broadcast-Intents bei Änderungen
- Failover bei Verlust: WLAN => Mobiles Netz
- API: Netzwerkstatus auslesen & Netzwerk für Traffic auswählen

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

HTTP Clients

- HTTPS
- Streaming Up-/Downloads
- Timeouts
- IPv6
- Connection Pooling / Caching

HttpURLConnection

- Empfohlen von Android
- Leichtgewichtig
- Probleme bei Version < Gingerbread

Apache HttpClient

- Deprecated seit API-Level 22
- Große & flexible API
 - Problem: schwierig zu verbessern ohne Kompatibilität zu beeinträchtigen

Demo

Permissions

AndroidManifest.xml

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

Verbindung prüfen

```
private boolean isOnline() {  
    ConnectivityManager connMgr = (ConnectivityManager)  
        getSystemService(Context.CONNECTIVITY_SERVICE);  
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();  
    return (networkInfo != null && networkInfo.isConnected());  
}
```

- `getActiveNetworkInfo()`: Erstes Interface oder null
- Speziell für WLAN | mobiles Netz prüfen:
 - `connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI | TYPE_MOBILE)`
- Alle Informationen in `networkInfo`:
 - `getTypeName()`, `getSubtypeName()`
 - `getExtraInfo()`, `isRoaming()`
 - ...

Network Status

connected

Type:	mobile
Subtype:	UMTS
Extra:	epc.tmobile.com
Roaming:	false
Reason:	connected

BroadcastReceiver

```
private class NetworkReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        setCurrentStatus(); // update GUI and networkInfo
    }
}

onCreate()
...
IntentFilter filter = new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION);
receiver = new NetworkReceiver();
this.registerReceiver(receiver, filter);
...
```

- Empfängt Änderungen des Netzes stößt Update der GUI an
- Auch in Manifest registrierbar:
 - ... <action android:name="android.net.conn.CONNECTIVITY_CHANGE" /> ...
 - Aber: kann Anwendung aufwecken (evtl. gewollt)

GET Request mit HTTPClient

```
... try {  
    HttpParams httpParameters = new BasicHttpParams();  
    HttpConnectionParams.setConnectionTimeout(httpParameters, 10000); // Default == 0  
    HttpConnectionParams.setSoTimeout(httpParameters, 15000);  
    HttpClient httpClient = new DefaultHttpClient(httpParameters);  
    HttpGet httpGet = new HttpGet("http://www.mi.hs-rm.de/~pbart001/wsgi/highscore.wsgi/");  
    HttpResponse httpResponse = httpClient.execute(httpGet); // make request  
    int response = httpResponse.getStatusLine().getStatusCode();  
    if (response == HttpStatus.SC_OK) {  
        // receive as InputStream  
        is = httpResponse.getEntity().getContent();  
        // Convert InputStream into a string  
        String contentAsString = convertInputStreamToString(is);  
    }  
} catch (IOException e) { ... }
```

- Einfacher GET Request
- HttpGet / HttpPost / HttpPut / ...
- Content als String

GET Request mit HttpURLConnection

```
...
InputStream is = null; HttpURLConnection conn = null;
try {
    URL url = new URL("http://www.mi.hs-rm.de/~pbart001/wsgi/highscore.wsgi/get");
    conn = (HttpURLConnection) url.openConnection();
    conn.setReadTimeout(10000); // Default == 0
    conn.setConnectTimeout(15000);
    conn.setRequestMethod("GET");
    conn.connect(); // make request
    if (conn.getResponseCode() == HttpURLConnection.HTTP_OK) {
        is = conn.getInputStream();
        String contentAsString = convertInputStreamToString(is); // Convert into a String
    } catch (IOException e) {...} finally {... conn.disconnect() ... }
```

- Einfacher GET Request
- Content als String

GET Request Timeout

```
...
InputStream is = null; HttpURLConnection conn = null;
try {
    URL url = new URL("http://jonastheis.de/timeout.php");
    conn = (HttpURLConnection) url.openConnection();
    conn.setReadTimeout(10000); // Default == 0
    conn.setConnectTimeout(15000);
    conn.setRequestMethod("GET");
    ...
} catch (SocketTimeoutException e) {
    ... // Show nice error to user
} finally {... conn.disconnect() ... }
```

- Alles wie bisher: nur zusätzliche Exception fangen
- SocketTimeoutException

POST Request

```
... try { ... // as before
    conn.setRequestMethod("POST"); conn.setDoOutput(true); // very important
    String params="comment_text=" + URLEncoder.encode("Fancy Android!", "UTF-8");
    conn.setFixedLengthStreamingMode(params.getBytes().length);

    PrintWriter out = new PrintWriter(conn.getOutputStream());
    out.print(paramsString); out.close();
    conn.connect(); // make request
    if (conn.getResponseCode() == HttpURLConnection.HTTP_OK) {
        is = conn.getInputStream();
        String contentAsString = convertInputStreamToString(is); // Convert into a String
    } catch (IOException e) {...} finally {... conn.disconnect() ... }
```

- Einfacher POST Request
- `conn.setDoOutput(true)`: muss gesetzt werden, sonst kein `OutputStream`
- Content-Length setzen (nicht `conn.setRequestProperty("Content-Length", length)`):
 - `conn.setFixedLengthStreamingMode(params.getBytes().length)`
 - Größe nicht bekannt: `setChunkedStreamingMode(int)`

Best Practices für Webanfragen

- Requests niemals auf dem MainThread!
 - `NetworkOnMainThreadException`
- Immer sicherstellen, dass Verbindung vorhanden ist
- Service
- Viel Traffic: Einstellungen für Nutzer anbieten
 - z.B. Bilderupload nur im WLAN
- Aussagekräftige Fehlermeldungen für Nutzer bei z.B. Timeout
- Traffic minimieren

Tipps zur Android Debug Bridge

- WLAN / mobiles Netz in adb an-/ausschalten
- Auf dem Device: Entwickleroptionen – Root Zugriff für ADB erlauben
- adb shell: su
 - svc wifi (enable | disable)
 - svc data (enable | disable)

Fragen?

Quellen

- <http://developer.android.com/training/basics/network-ops/connecting.html#connection>
- <http://developer.android.com/training/basics/network-ops/managing.html>
- <http://courses.coreservlets.com/Course-Materials/pdf/android/Android-Networking-2.pdf>
- <http://www.vogella.com/tutorials/AndroidNetworking/article.html>
- <http://javatechig.com/android/android-networking-tutorial#4-application-manifest-permissions>
- <http://developer.android.com/reference/java/net/URLConnection.html>
- <http://candrews.integralblue.com/2011/09/best-way-to-use-httpclient-in-android/>
- <http://android-developers.blogspot.de/2011/09/androids-http-clients.html>