Using Internet Resources

Connecting to Internet resources

- First thing first
  - Before you can access Internet resources, you need to add an INTERNET uses-permission node to your application manifest, as shown in the following XML snippet:

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

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

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

Opening an Internet Data Stream

- Whatever you do such as downloading an image/audio/video or text file or loading a web page, you need to open an internet data stream

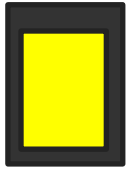
```
String myFeed = getString(R.string.my_feed);
try {
  URL url = new URL(myFeed);

  // Create a new HTTP URL connection
  URLConnection connection = url.openConnection();
  HttpURLConnection httpConnection = (HttpURLConnection)connection;

  int responseCode = httpConnection.getResponseCode();
  if (responseCode == HttpURLConnection.HTTP_OK) {
    InputStream in = httpConnection.getInputStream();
    processStream(in);
  }
}
catch (MalformedURLException e) {
  Log.d(TAG, "Malformed URL Exception.");
}
catch (IOException e) {
  Log.d(TAG, "IO Exception.");
}
```

Internet Resources

- Android offers several ways to leverage Internet resources
  - Use of a WebView
    - Includes a WebKit-based browser within an Activity
    - <WebView h= w= src=‘’/>



  - Use client-side APIs
    - Google Firebase APIs: Interact directly with database server processes
    - Google Firebase Cloud API

Implement Own Processing Technique

- Implement Own Processing Technique
  - You can process remote XML feeds to extract and process data using a Java-based XML parser, such as SAX or the XML Pull Parser
  - You can process remote text/json formatted data to extract and process data using a Java-based Json parser

Why do we need Own Processing Technique?

- With Internet connectivity and a WebKit browser,
  - Is there any reason to create native Internet-based applications when you could make a web-based version instead.

Native and Internet-based Application

- There are a number of benefits to creating thick- and thin-client native applications rather than relying on entirely web-based solutions:

- **Bandwidth** — Static resources such as images, layouts, and sounds can be expensive on devices with bandwidth restraints
  - By creating a native application, you can limit the bandwidth requirements to changed data only

- **Caching** — With a browser-based solution, a patchy Internet connection can result in intermittent application availability
  - A native application can cache data and user actions to provide as much functionality as possible without a live connection and synchronize with the cloud when a connection is reestablished

Native and Internet-based Application

- **Reducing battery drain**—Each time your application opens a connection to a server, the wireless radio will be turned on (or kept on)
  - A native application can bundle its connections, minimizing the number of connections initiated
  - The longer the period between network requests, the longer the wireless radio can be left off

- **Native features**—Android devices are more than simple platforms for running a browser
  - They include location-based services, Notifications, widgets, camera hardware, background Services, and hardware sensors
  - By creating a native application, you can combine the data available online with the hardware features available on the device to provide a richer user experience

Native and Internet-based Application

- Modern mobile devices offer a number of alternatives for accessing the Internet
  - **Mobile Internet**—GPRS, EDGE, 3G, 4G, and LTE Internet access is available through carriers that offer mobile data.
  - **Wi-Fi**—Wi-Fi receivers and mobile hotspots are becoming increasingly common.

Native and Internet-based Application

- If you use Internet resources in your application, remember that your users' data connections are dependent on the communications technology available to them
  - EDGE and GSM connections are notoriously low-bandwidth, whereas a Wi-Fi connection may be unreliable in a mobile setting

- Optimize the user experience by limiting the quantity of data transmitted and ensure that your application is robust enough to handle network outages and bandwidth limitations.

Problem Discussion

- **Two devices**
  - Both are connected to the same WiFi router
  - Router is not connected to the internet (cable is unplugged)
  - Both devices are not capable to access internet via network data

- Can these two devices communicate with each other?

END