MOBILE INTELLIGENCE 03

Android - Network

Minsoo Kim (Pukyong Nat'l Univ.)

Major In Industrial Data System & Engineering

부산대학교 PUSAN NATIONAL UNIVERSITY
Department of Industrial Engineering

국립부경대학교 PUKYONG NATIONAL UNIVERSITY
Department of Industrial and Data Engineering

Logistics of this Slide

- Network support in Android
  - Main classes in android.net package
  - Reading Network status: NetworkCallback class
  - Using additional Network: NetworkCallback, NetworkRequest class
  - Lab: Test Network Project
- Understanding Web App
  - Native App vs. Web App
  - Browsing Web Contents in Android
  - Adding WebView to Test Network Project
  - Using android.webkit.WebViewClient
  - Incorporating JavaScript
- XML Handling
  - Types of XML Parsers
  - DOM Parser Basics
  - XML DOM Tree
  - www.kma.go.kr: weather casting

Network support in Android

- Main classes in android.net package
  - Support many networks: 3G/4G/5G, Wi-Fi(P2P), Bluetooth, NFC, SIP, …
    - ConnectivityManager: monitor & notify changes on network's connectivity
      - {Application | Activity}.getSystemService(Context.CONNECTIVITY_SERVICE)
    - Manifest: android.permission.{INTERNET | ACCESS_NETWORK_STATE}
  - All Android apps have a system defined default network.
    - System responds to the app's networking request via the default network.
      - Network *myNet* = *cMgr*.getActiveNetwork(); ← can return null!
      - Usually prefer unlimited or fast networks rather than limited or slow networks.
    - Default network can be changed during the app lifecycle & notify for this.
      - *cMgr*.registerDefaultNetworkCallback(NetworkCallback {, Handler});
  - Network: one of the networks the device is currently connected to
    - Unable to use when disconnected, new network is used when reconnected.
      - Can communicate by obtaining Socket or Connection via SocketFactory or URL.
      - *myConn* = (HttpsUrlConnection) *myNet*.openConnection(new URL("*https://…*");
  - StrictMode: restrict tasks over Disk & Network in Main Thread
    - Introduced to avoid ANR exception due to the risk of blocking Main Thread
    - Can alleviate StrictMode by setting ThreadPolicy & VmPolicy, but it is not recommended. Use Thread, Handler or IntentService instead to relieve the burden of the Main Thread.

Network support in Android

- (Main classes in android.net package)
  - LinkProperties: provide Network's DNS server, IP address, Route info.
    - LinkProperties *linkP* = connMgr.getLinkProperties(*myNet*);
  - NetworkCapabilities: provide info on transports and their capability
    - Capability: Network's ability like MMS, NOT_METERED, INTERNET
      - NetworkCapabilities *netCap* = *cMgr*.getNetworkCapabilities(*myNet*);
      - Provide hasTransport(int *transportType*) and hasCapability(int *capability*)
        - » TRANSPORT_{CELLULAR | WIFI | BLUETOOTH | ETHERNET | VPN | USB … }
        - » NET_CAPABILITY_{MMS | WIFI_P2P | NOT_METERED | INTERNET | VALIDATED … }

```
private boolean isNetworkAvailable(Application app) {
  ConnectivityManager connMgr = (ConnectivityManager) app.getSystemService(Context.CONNECTIVITY_SERVICE);
  if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
    Network net = connMgr.getActiveNetwork();
    if (net == null) return false;
    NetworkCapabilities capa = connMgr.getNetworkCapabilities(net);
    return capa != null && (capa.hasTransport(NetworkCapabilities.TRANSPORT_WIFI) ||
      capa.hasTransport(NetworkCapabilities.TRANSPORT_CELLULAR) ||
      capa.hasTransport(NetworkCapabilities.TRANSPORT_ETHERNET) ||
      capa.hasTransport(NetworkCapabilities.TRANSPORT_BLUETOOTH));
  } else {  // NetworkInfo가 API level 29부터 deprecated되었음.
    NetworkInfo netInfo = connMgr.getActiveNetworkInfo();
    return netInfo != null && netInfo.isConnected();
  }
}
```

Network support in Android

- Reading Network status: NetworkCallback class
  - Get status event on default net by registering ConnectivityManager

```
connMgr.registerDefaultNetworkCallback(new ConnectivityManager.NetworkCallback() {
    @Override
    public void onAvailable(Network network) {
        Log.e(TAG, "The default network is now: " + network);
    }

    @Override
    public void onLost(Network network) {
        Log.e(TAG, "The Application lost default network. The last default network was " + network);
    }

    @Override
    public void onCapabilitiesChanged(Network network, NetworkCapabilities networkCapabilities) {
        Log.e(TAG, "The default network changed capabilities: " + networkCapabilities);
    }

    @Override
    public void onLinkPropertiesChanged(Network network, LinkProperties linkProperties) {
        Log.e(TAG, "The default network changed link properties: " + linkProperties);
    }
});
```

Create NetworkCallback and register it to ConnectivityManager to be notified of the state change on the default network.

Network support in Android

- Using additional Network: NetworkCallback, NetworkRequest class
  - Send NetworkRequest to ConnectivityManager for asking
    - registerNetworkCallback(NetworkRequest, NetworkCallback {, Handler})
      - To use a Handler different from the default thread, send it as an argument.
    - unregisterNetworkCallback(NetworkCallback) ← unregister
  - Frequently used NetworkCapabilities values
    - NET_CAPABILITY_MMS: network that can send MMS
    - TRANSPORT_WIFI_AWARE: for P2P Wi-Fi connection
    - NET_CAPABILITY_INTERNET, NET_CAPABILITY_VALIDATED: to send data to the server on the Internet
      - Check the existence of network with NetworkCallback doesn't mean that the app can send data → Some networks may not provide Internet connectivity!
      - After having proper NetworkCapabilities, check manifest permission
      - background networking needs CHANGE_NETWORK_STATE permission

```
NetworkRequest request = new NetworkRequest.Builder()
    .addCapability(NetworkCapabilities.NET_CAPABILITY_NOT_METERED)
    .addCapability(NET_CAPABILITY_INTERNET)
    .build();

connMgr.registerNetworkCallback(request, myNetworkCallback);
```

Ask a network with unlimited Internet connection

Network support in Android

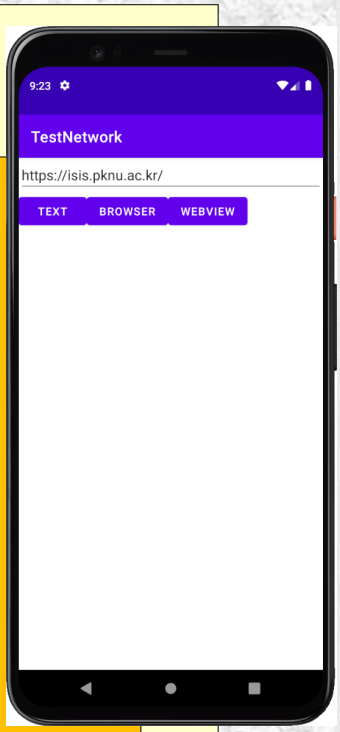
Lab: Test Network Project

- Edit activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/address"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="https://isis.pknu.ac.kr/" />

    ...

    <LinearLayout
        android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"/>

</LinearLayout>
```

add

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <Button android:id="@+id/textButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="doIt"
        android:text="text" />
    <Button android:id="@+id/browserButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="doIt"
        android:text="browser" />
    <Button android:id="@+id/webviewButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="doIt"
        android:text="WebView" />
</LinearLayout>
```



Network support in Android

(Lab: Test Network Project)

- Edit ActivityMain.java

```
public class MainActivity extends AppCompatActivity {
    LinearLayout container;
    EditText etUrl;
    TextView textView;
    ...
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etUrl = (EditText) findViewById(R.id.address);
        container = (LinearLayout) findViewById(R.id.container);
    }
    public void doIt(View button) {
        switch(button.getId()) {
            case R.id.textButton:
                Log.d("NETWORK", "get Text button clicked!");
                break;
            case R.id.browserButton:
                Log.d("NETWORK", "browser button clicked!");
                break;
            case R.id.webviewButton:
                Log.d("NETWORK", "webview button clicked!");
                break;
        }
    }
}
```

add

```
protected void onResume() {
    super.onResume();
    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    try {
        for (Network network : connMgr.getAllNetworks()) {
            NetworkCapabilities capa =
                connMgr.getNetworkCapabilities(network);
            Log.d("NETWORK", capa.toString());
        }
        Network activeNet = connMgr.getActiveNetwork();
        Log.d("NETWORK", "active(default) network: " + activeNet);
    } catch (Exception e) {
        Log.e("NETWORK", e.toString());
    }
    connMgr.registerDefaultNetworkCallback(myCallback);
}
protected void onStop() {
    super.onStop();
    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    connMgr.unregisterNetworkCallback(myCallback);
}
```

Network support in Android

- (Lab: Test Network Project)
  - (Edit ActivityMain.java)

```
public class MainActivity extends AppCompatActivity {
   LinearLayout container;
   EditText etUrl;
   TextView textView;
   ...
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentVi
      etUrl = (EditT
      container = (
   }
   public void dol
      switch(buttor
         case R.id.te
            Log.d("N
            break;
         case R.id.b
            Log.d("N
            break;
         case R.id.w
            Log.d("N
            break;
      }
   }
}
```

add

```
ConnectivityManager.NetworkCallback myCallback = new ConnectivityManager.NetworkCallback() {
   public void onAvailable(Network network) {
      Log.e("NETWORK", "The default network is now: " + network);
   }
   public void onLost(Network network) {
      Log.e("NETWORK", "The Application lost default network. The last default network was " + network);
   }
   public void onCapabilitiesChanged(Network network, NetworkCapabilities networkCapabilities) {
      Log.e("NETWORK", "The default network changed capabilities: " + networkCapabilities);
   }
   public void onLinkPropertiesChanged(Network network, LinkProperties linkProperties) {
      Log.e("NETWORK", "The default network changed link properties: " + linkProperties);
   }
};
```


Network support in Android

- (Lab: Test Network Project)
  - Add permission to AndroidManifest.xml and run!

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="kr.ac.sme.pknu.testnetwork">

   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
   <application
      android:allowBackup="true"
      android
      android
      android
      android
      android
      <activit
         <int
            <
            <
         </in
      </activi
   </applicat
</manifest>
```

add

Check for NetworkCallback is working

```
Run: app
W/knu.testnetwor: Accessing hidden method Landroid/view/View;->computeFitSystemWindows(Landroid/graphic
W/knu.testnetwor: Accessing hidden method Landroid/view/ViewGroup;->makeOptionalFitsSystemWindows()V (1
D/NETWORK: [ Transports: WIFI Capabilities: NOT_METERED&INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VPN&VALIDAT
    [ Transports: CELLULAR Capabilities: MMS&SUPL&DUN&FOTA&IMS&CBS&IA&INTERNET&NOT_RESTRICTED&TRUSTED&N
D/NETWORK: active(default) network: 101
E/NETWORK: The default network is now: 101
    The default network changed capabilities: [ Transports: WIFI Capabilities: NOT_METERED&INTERNET&NOT
E/NETWORK: The default network changed link properties: {InterfaceName: wlan0 LinkAddresses: [fe80::15:
D/OpenGLRenderer: Skia GL Pipeline
D/HostConnection: HostConnection::get() New Host Connection established 0xe5016fe0, tid 22704
D/HostConnection: HostComposition ext ANDROID_EMU_CHECKSUM_HELPER_v1 ANDROID_EMU_native_sync_v2 ANDROID
I/ConfigStore: android::hardware::configstore::V1_0::ISurfaceFlingerConfigs::hasWideColorDisplay retrie
    android::hardware::configstore::V1_0::ISurfaceFlingerConfigs::hasHDRDisplay retrieved: 0
I/OpenGLRenderer: Initialized EGL, version 1.4
D/OpenGLRenderer: Swap behavior 1
TODO  Terminal  Database Inspector  Profiler  4: Run  Build  6: Logcat
```

Network support in Android

- (Lab: Test Network Project)
  - Add processDownload() to MainActivity.java
    - Method to download a file and save it to internal storage.

```
private void processDownload(final TextView tv) {
   try {
      URL url = new URL(etUrl.getText().toString());
      HttpsURLConnection conn = (HttpsURLConnection) url.openConnection();
      conn.connect();

      InputStream is = new BufferedInputStream(url.openStream(), 1024);
      FileOutputStream fos = openFileOutput("download.dat", Context.MODE_PRIVATE);
      byte[] buffer = new byte[1024];
      int count = 0;
      while ((count = is.read(buffer)) != -1)
         fos.write(buffer, 0, count);
      fos.flush();
      fos.close();
      is.close();
   } catch(Exception e) {
      Log.d("THREAD", "Download Error: " + e.toString());
   }
}
```

Create a FileOutputStream for a file on Internal Storage

Network support in Android

- (Lab: Test Network Project)
  - Modify doIt() method in the MainActivity.java

```
public class MainActivity extends AppCompatActivity {
   LinearLayout container;
   EditText etUrl;
   TextView textView;
   ...
   public void doIt(View button) {
      switch(button.getId()) {
         case R.id.textButton:
            Log.d("NETWORK", "get Text button clicked!");
            ...
            break;
         case R.id.browserButton:
            Log.d("NETWORK", "browser button clicked!");
            break;
         case R.id.webviewButton:
            Log.d("NETWORK", "webview button clicked!");
            break;
      }
   }
}
```

add

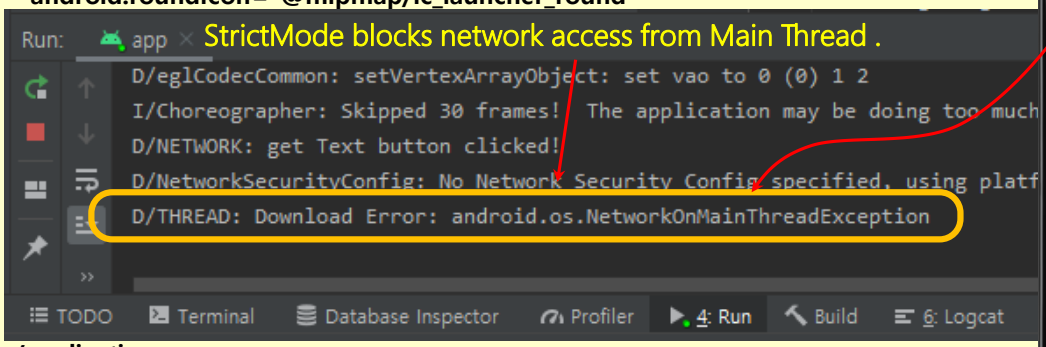
```
textView = new TextView(this);
textView.setText("This is test!");
container.removeAllViews();
container.addView(textView);
processDownload(textView);
```

Network support in Android

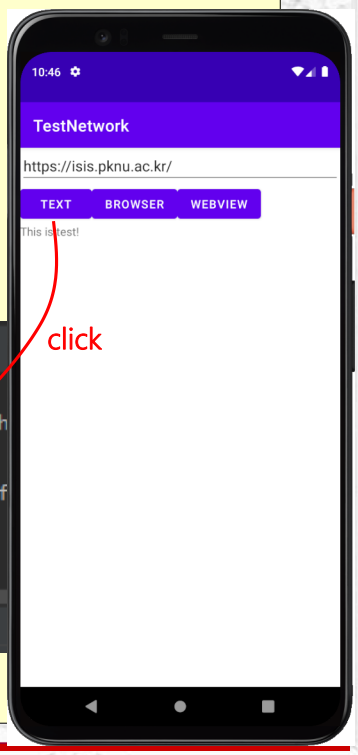
- (Lab: Test Network Project)
  - Add permission to AndroidManifest.xml and run → 'TEXT' button click!

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="kr.ac.sme.pknu.testnetwork">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
```

add




```
    </application>
</manifest>
```



Network support in Android

- (Lab: Test Network Project)
  - Alleviate StrictMode at the MainActivity.onCreate() and run!
    - Not a recommended approach!

```
public class MainActivity extends AppCompatActivity {
    ...
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etUrl = (EditText) findViewById(R.id.address);
        container = (LinearLayout) findViewById(R.id.container);

        StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder().permitNetwork().build());
    }
    ...
}
```

add

  - Android Studio: [View → Tool Windows → Device File Explorer]
    - data/data/*<package name>*/files/download.dat file!
      - download.dat right click → 'Save As...', copy file from emulator to outside.

Network support in Android

- (Lab: Test Network Project)
  - Complete processDownload() ← Read file from the Internal Storage.

```
private void processDownload(final TextView tv) {
  try {
    URL url = new URL(etUrl.getText().toString());
    HttpsURLConnection conn = (HttpsURLConnection) url.openConnection();
    conn.connect();

    InputStream is = new BufferedInputStream(url.openStream(), 1024);
    FileOutputStream fos = openFileOutput("download.dat", Context.MODE_PRIVATE);
    byte[] buffer = new byte[1024];
    int count = 0;
    while ((count = is.read(buffer)) != -1)
      fos.write(buffer, 0, count);
    fos.flush();
    fos.close();
    is.close();
    ...
  } catch(Exception e) {
    Log.d("THREAD", "Download Error: " +
  }
}
```

add

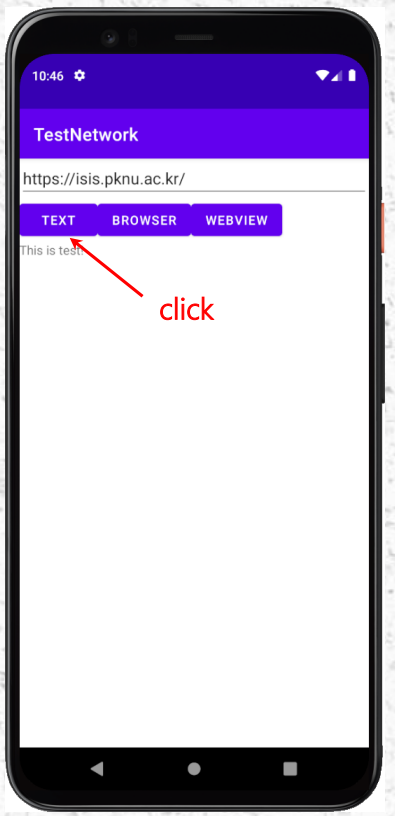
```
String line;
BufferedReader br = new BufferedReader(
    new InputStreamReader(
        openFileInput("download.dat"), StandardCharsets.UTF_8));
StringBuilder sb = new StringBuilder();
while ((line = br.readLine()) != null)
    sb.append(line).append("₩n");
br.close();

tv.setText(sb.toString());
deleteFile("download.dat");
```

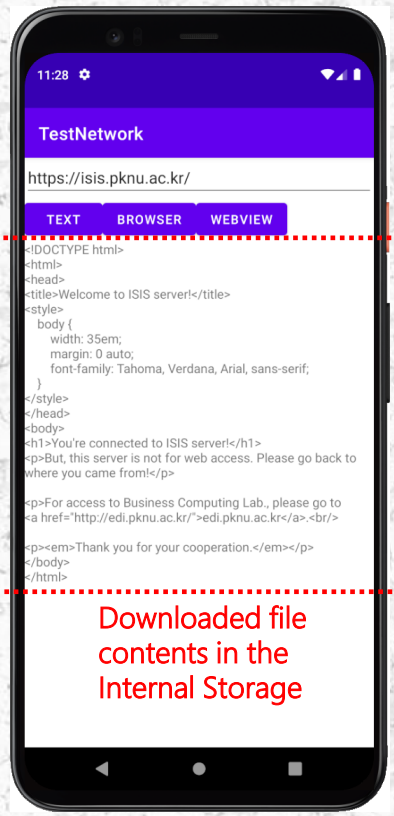
Get FileInputStream for Internal Storage file.

Delete a file from the Internal Storage

Network support in Android

- (Lab: Test Network Project)
  - Check how it works.





Network support in Android

- (Lab: Test Network Project)
  - Use separate Thread without alleviating StrictMode!
    - Comment out a StrictMode code and modify doIt() method

```
public class MainActivity extends AppCompatActivity {
    ...
    protected void onCreate(Bundle savedInstanceState) {
        ...
        // StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder().permitNetwork().build());
    }
    ...
    public void doIt(View button) {
        switch(button.getId()) {
            case R.id.textButton:
                Log.d("NETWORK", "get Text button clicked!");
                textView = new TextView(this);
                textView.setText("This is test!");
                container.removeAllViews();
                container.addView(textView);
                processDownload(textView);
                break;
            ...
        }
    }
    ...
}
```

Comment out

replace

```
new Thread() {
    public void run() {
        processDownload(textView);
    }
}.start();
```


Network support in Android

- (Lab: Test Network Project)
  - (Use separate Thread without alleviating StrictMode!)
    - Running will cause a CalledFromWrongThreadException
      - Because TextView.setText() is called from a thread that is not UI Thread!
    - Rerun after modifying processDownload() method

```
app
D/NETWORK: get Text button clicked!
D/NetworkSecurityConfig: No Network Security Config specified, using platform default
D/THREAD: Download Error: android.view.ViewRootImpl$CalledFromWrongThreadException: Only the original thread that created a view hierarchy can touch its views.
Terminal   Database Inspector   Profiler   4: Run   Build   6: Logcat
```

```
private void processDownload(final TextView tv) {
    try {
        ...
        while ((line = br.readLine()) != null)
            sb.append(line).append("₩n");
        br.close();

        tv.setText(sb.toString());
        deleteFile("download.dat");
    } catch(Exception e) {
        Log.d("THREAD", "Download Error: " + e.toString());
    }
}
```

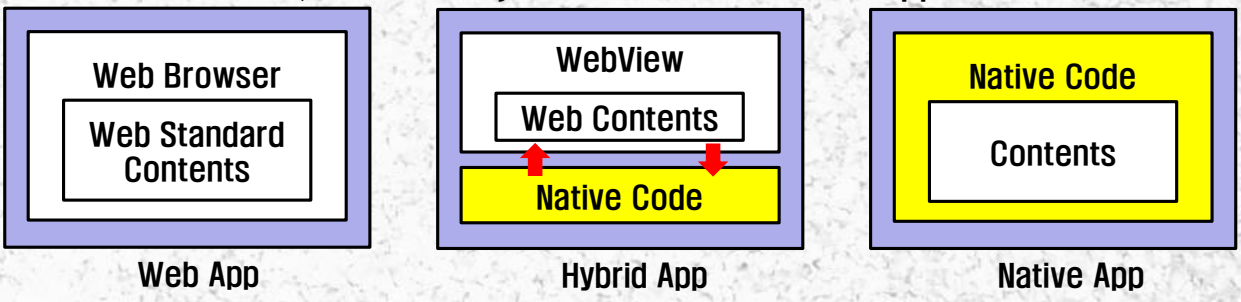
replace

```
runOnUiThread(new Runnable() {
    public void run() {
        tv.setText(sb.toString());
    }
});
```

Understanding Web App

Native App vs. Web App

- Native App: develop with Android SDK → installed on the device
  - Good performance, free to access whole Android functionalities
- Web App: develop with web standards → use Web Browser w/o install
  - Easy support for various display settings by using Browser
  - Fast development for surf-style apps → save development cost
    - Android provides WebView widget that is implemented with WebKit engine.
- Hybrid App: Embedding WebView inside of Native App
  - Distributable via App Market, combine the advantages of Native & Web App
  - Special design of enclosed web pages are needed. By defining an interface to Native App, Java Script can call Android's API
  - Have some advantages of Web App, but it can be slow due to the overhead of WebView, and memory restriction can also be applied.



Understanding Web App

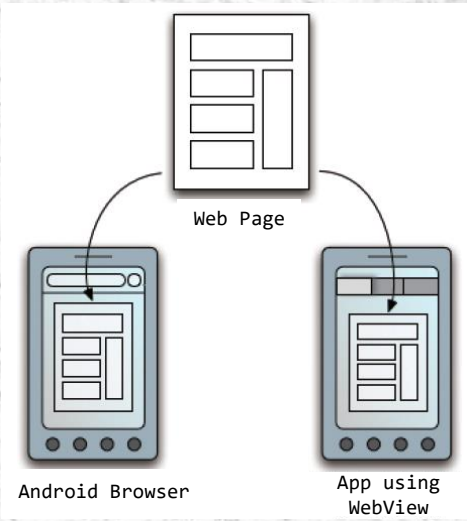
Browsing Web Contents in Android

1. Run a browser app using Intent
   - Uri *uri* = Uri.parse("https://some.domain.com/resource");
   - startActivity(new Intent(Intent.ACTION_VIEW, *uri*));
2. Use a WebView
   - After obtaining a WebSetting using getSettings(), detailed setting is possible.
   - loadUrl(String *urlString*): load contents using URL address string.
   - loadData(String *data*, String *mimeType*, String *encoding*): give data to load
   1. Use by defining in the Layout XML

```
<WebView
    android:id="@+id/webview"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
...
```

   2. Create WebView in the Java code

```
public void onCreate(Bundle saveInstanceState) {
    ...
    WebView webView = new WebView(this);
    setContentView(webView);
    ...
}
```



Understanding Web App

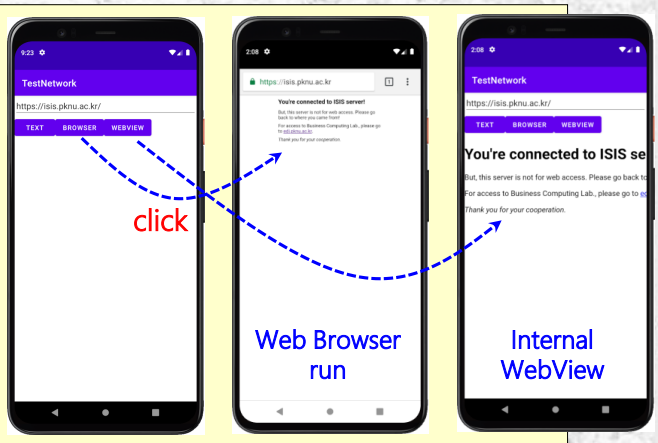
Adding WebView to Test Network Project

- Modify doIt() of MainActivity & add webView data member

```
public class MainActivity extends AppCompatActivity {
    LinearLayout container;
    EditText etUrl;
    TextView textView;
    WebView webView;          ← add
    ...
    public void doIt(View button) {
        switch(button.getId()) {
            case R.id.textButton:
                ...
            case R.id.browserButton:
                Log.d("NETWORK", "browser button clicked!");
                ...           ← add
                break;
            case R.id.webviewButton:
                Log.d("NETWORK", "webview button clicked!");
                ...           ← add
                break;
        }
    }
    ...
}
```

```
Uri uri = Uri.parse(etUrl.getText().toString());
startActivity(new Intent(Intent.ACTION_VIEW, uri));
```

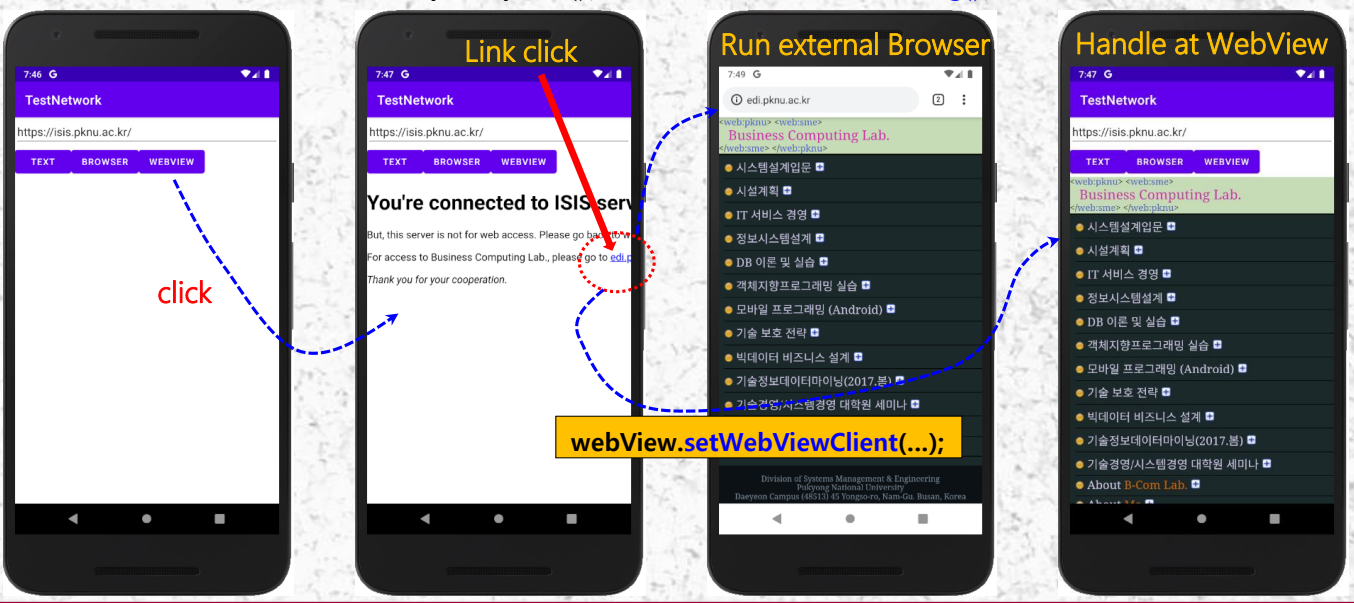
```
webView = new WebView(this);
container.removeAllViews();
container.addView(webView);
webView.getSettings().setLoadsImagesAutomatically(true);
webView.getSettings().setJavaScriptEnabled(true);
webView.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
webView.loadUrl(etUrl.getText().toString());
```



Understanding Web App

Using android.webkit.WebViewClient

- Click a link inside of WebView → run a Browser!
- WebViewClient: decide contents and further processing on WebView
  - Use WebView.setWebViewClient(…)
  - Provides callback for requests (e.g.: state change, error, …) in WebView
    - onPageStarted(), onPageFinished(), onLoadResource(), onReceivedError(), shouldInterceptRequest(), shouldOverrideUrlLoading(), …




```
webView.setWebViewClient(...);
```

Understanding Web App

- (Using android.webkit.WebViewClient)
  - Add TestBrowser class to MainActivity class

```
public class MainActivity extends AppCompatActivity {
    ...
    private class TestBrowser extends WebViewClient {
        @Override
        public boolean shouldOverrideUrlLoading(WebView view, WebResourceRequest request) {
            if(request.getUrl().getHost().contains("pknu.ac.kr")) { // This is my website, so do not override.
                Log.d("NETWORK", "shouldOverrideUrlLoading: return false;");
                return false; // Let my WebView load this page!
            }
            // Otherwise, the link is not for a page on my site, so launch another Activity that handles URLs.
            Log.d("NETWORK", "shouldOverrideUrlLoading: return true;");
            startActivity(new Intent(Intent.ACTION_VIEW, request.getUrl()));
            return true;
        }
        @Override
        public void onReceivedError(WebView view, WebResourceRequest request, WebResourceError error) {
            super.onReceivedError(view, request, error);
            Log.d("NETWORK", "Web Page Loading Error: " + error.getDescription());
            view.loadData("<html><body>Something wrong!</body></html>", "text/html", "utf-8");
        }
    }
    ...
}
```

Handle inside of WebView

Delegate handling to other component

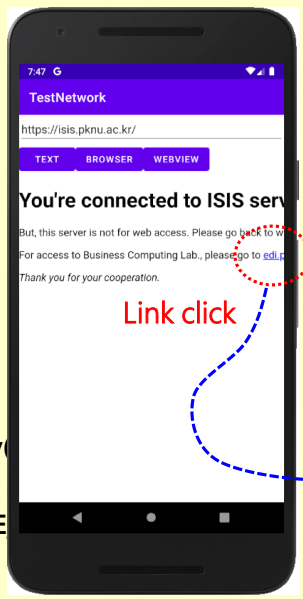
Compose contents using data

Understanding Web App

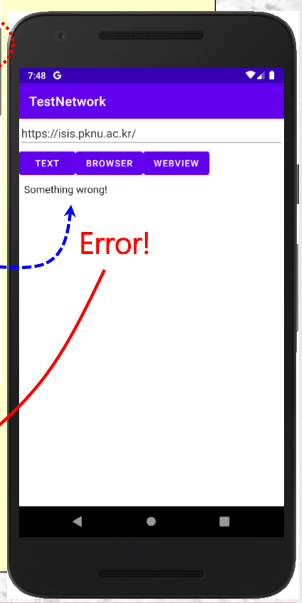
- (Using android.webkit.WebViewClient)
  - Modify doIt() method in the MainActivity and run

```
public class MainActivity extends AppCompatActivity {
    ...
    public void doIt(View button) {
        switch(button.getId()) {
            ...
            case R.id.webviewButton:
                Log.d("NETWORK", "webview button clicked!");
                webView = new WebView(this);
                container.removeAllViews();
                container.addView(webView);
                webView.setWebViewClient(new TestBrowser());
                webView.getSettings().setLoadsImagesAutomatically(
                webView.getSettings().setJavaScriptEnabled(true);
                webView.setScrollBarStyle(View.SCROLLBARS_INSIDE
                webView.loadUrl(etUrl.getText().toString());
                break;
        }
}
```

add



Link click



Error!

Since Android v9, WebView restricts http:// access!

```
/kr.ac.sme.pknu.testnetwork D/eglCodecCommon: setVertexArrayObject: set vao to 1 (1) 0 0
/kr.ac.sme.pknu.testnetwork D/NETWORK: Web Page Loading Error: net::ERR_CLEARTEXT_NOT_PERMITTED
/kr.ac.sme.pknu.testnetwork D/eglCodecCommon: setVertexArrayObject: set vao to 1 (1) 4 10
/kr.ac.sme.pknu.testnetwork D/eglCodecCommon: setVertexArrayObject: set vao to 0 (0) 4 6
```

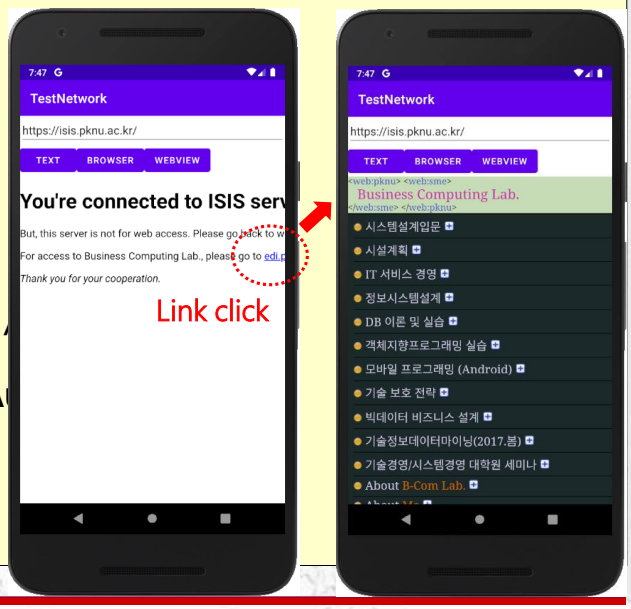
Profiler | 4: Run | Build | 6: Logcat

Understanding Web App

- (Using android.webkit.WebViewClient)
  - Modify Manifest file and test

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"package="kr.ac.sme.pknu.testnetwork">
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:usesCleartextTraffic="true"
        android:theme="@style/Theme.TestNetwork">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAU
            </intent-filter>
        </activity>
    </application>
</manifest>
```

add



Understanding Web App

- (Using android.webkit.WebViewClient)
  - Add onKeyDown() method to MainActivity
    - Combine WebView's goBack() for BACK Button

```
public class MainActivity extends AppCompatActivity {
    ...
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        // Check if the key event was the Back button and if there's history to the WebView.
        if ((keyCode == KeyEvent.KEYCODE_BACK) && webView != null && webView.canGoBack()) {
            webView.goBack();
            return true;
        }
        // If not, hubble up to the default system behavior. (probably exit the activity)
        return super.onKeyDown(keyCode, event);
    }
    ...
}
```

  - Configuration change restarts Activity(onDestroy → onCreate) and clears old settings → Set minor config for Activity to handle directly
    - <activity android:configChanges="orientation|screenSize|keyboardHidden" ...>
    - Override Activity.onConfigurationChanged(), if not, corresponding config change is discarded and Activity can avoid restarting!

Understanding Web App

Incorporating JavaScript

1. JavaScript code → Android code: register object to handle invocation
   - Use WebView.addJavaScriptInterface(Object *object*, String *name*)
   - To enable JS code call, add @JavascriptInterface to method!

```
public class WebAccess {
    Context mContext;

    WebAccess(Context c) {
        mContext = c;
    }

    // Show a toast from the web page.
    @JavascriptInterface
    public void showToast(String msg) {
        Toast.makeText(mContext, msg, Toast.LENGTH_SHORT).show();
    }
}
```

```
WebView webView = (WebView) findViewById(R.id.webview);
webView.addJavascriptInterface(new WebAccess(this), "Android");
```

register

```
<input type="button" value="Show Toast"
    onClick="webToast('Hello Android!')" />
<script type="text/javascript">
    function webToast(message) {
        Android.showToast(message);
    }
</script>
```

test_webapp.html

call

- To use HTML, JS, CSS file by saving into app: use assets folder
  - Android Studio: [File → New → Folder → Assets Folder] menu
    - assets folder is created besides the res folder, not under the res folder!
  - WebView.loadUrl("file:///android_asset/test_webapp.html");

Understanding Web App

(Incorporating JavaScript)

- Modify MainActivity

```
public class MainActivity extends AppCompatActivity {
    ...
    @JavascriptInterface
    public void showToast(String message) {
        Toast.makeText(this, message, Toast.LENGTH_LONG).show();
    }
    @JavascriptInterface
    public void logCat(String message) {
        Log.d("NETWORK", message);
    }
    public void doIt(View button) {
        switch(button.getId()) {
            ...
            case R.id.webviewButton:
                Log.d("NETWORK", "webview button clicked!");
                webView = new WebView(this);
                webView.addJavascriptInterface(this, "WebAPI");
                container.removeAllViews();
                container.addView(webView);
            ...
    }
    ...
}
```

add

add

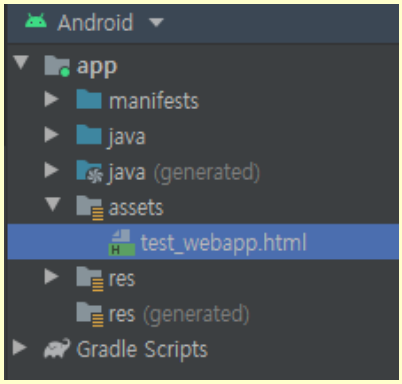
add

Understanding Web App

- (Incorporating JavaScript)
  - Create assets folder, add test_webapp.html

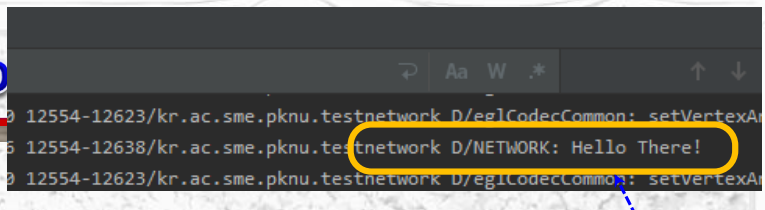
test_webapp.html

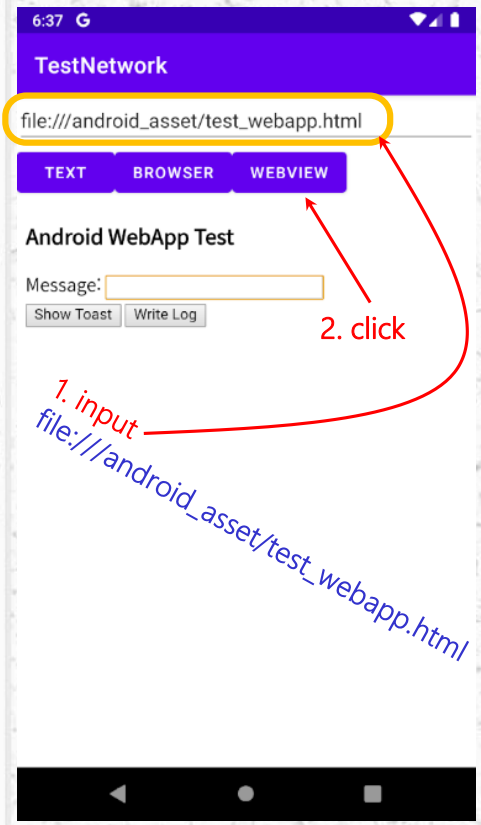
```
<!doctype html>
<html lang="en">
  <head><meta charset="utf-8"><title>Android WebApp Test using WebView</title></head>
  <body><H3>Android WebApp Test</H3>Message: <input type="text" id="message" size="20" /><br>
    <input type="button" onClick="webToast()" value="Show Toast" />
    <input type="button" onClick="webLog()" value="Write Log" />
    <script type="text/javascript">
      function webToast() {
        var message = document.getElementById('message').value;
        WebAPI.showToast(message);
      }
      function webLog() {
        var message = document.getElementById('message').value;
        WebAPI.logCat(message);
      }
      function showAlert(message) {
        alert(message);
      }
    </script>
  </body>
</html>
```

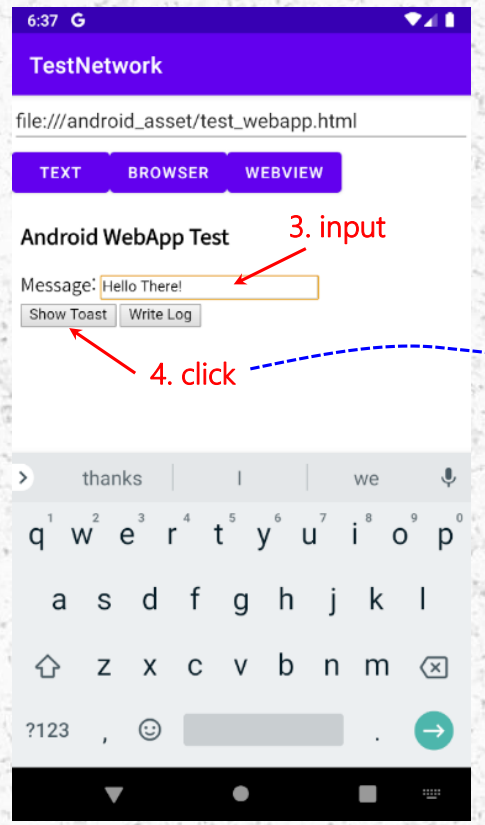


Understanding Web App

- (Incorporating JavaScript)
  - Run & Test







Understanding Web App

- (Incorporating JavaScript)
  2. Android code → JavaScript code: first of all, enable JavaScript
     - WebView.evaluateJavascript(String *script*, ValueCallback *callback*)
       - In older version: used WebView.loadUrl("javascript:alert('hello')") style
       - For alert() to work, WebChromeClient needs to be set!

```
WebView webView = new WebView(this);
webView.getSettings().setJavaScriptEnabled(true);
webView.setWebChromeClient(new WebChromeClient() {
    @Override
    public boolean onJsAlert(WebView view, String url, String message, JsResult result) {
        return super.onJsAlert(view, url, message, result);
    }
});
webView.evaluateJavascript("alert('it works!')", null);
...
```

Override JavaScript's alert() task

Call JavaScript's alert() function

- Using android.webkit.WebChromeClient
  - Full screen support, WebView's window opening & closing, showing JavaScript dialog to user, Host App's UI change related task
  - WebView.setWebChromeClient(...)
    - onProgressChanged(), onReceivedIcon(), onReceivedTitle(),onJsBeforeUnload(), onJsConfirm(), onJsAlert(), onJsPrompt(), onJsTimeout(), openFileChooser(), getDefaultVideoPoster(), onRequestFocus(), onReachedMaxAppCacheSize(), ...

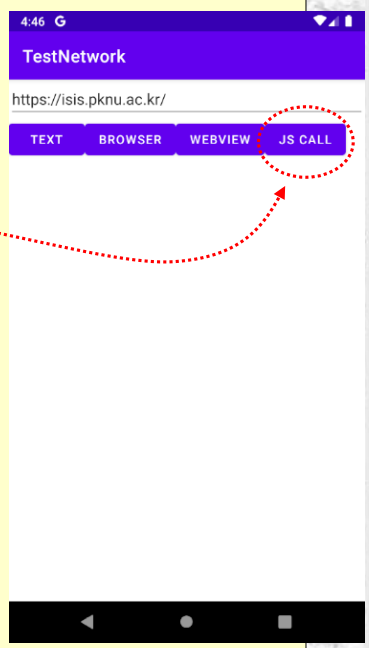
Understanding Web App

- (Incorporating JavaScript)
  - Add a Button to activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    ...
    <LinearLayout
        ...
        <Button
            android:id="@+id/webviewButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="doIt"
            android:text="WebView" />
        ...
    </LinearLayout>

    <LinearLayout
        android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"/>
</LinearLayout>
```

add

```
<Button
    android:id="@+id/jsCallButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="doIt"
    android:text="JS call" />
```



Understanding Web App

- (Incorporating JavaScript)
  - Modify doIt() code in MainActivity

```
public class MainActivity extends AppCompatActivity {
    ...
    public void doIt(View button) {
        switch(button.getId()) {
            ...
            case R.id.webviewButton:
                Log.d("NETWORK", "webview button clicked!");
                webView = new WebView(this);
                webView.addJavascriptInterface(this, "WebAPI");
                container.removeAllViews();
                container.addView(webView);
                webView.setWebViewClient(new TestBrowser());
                webView.getSettings().setLoadsImagesAutomatically(true);
                webView.getSettings().setJavaScriptEnabled(true);
                webView.setWebChromeClient(new WebChromeClient());
                webView.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
                webView.loadUrl(etUrl.getText().toString());
                break;
            ...
        }
    }
    ...
}
```

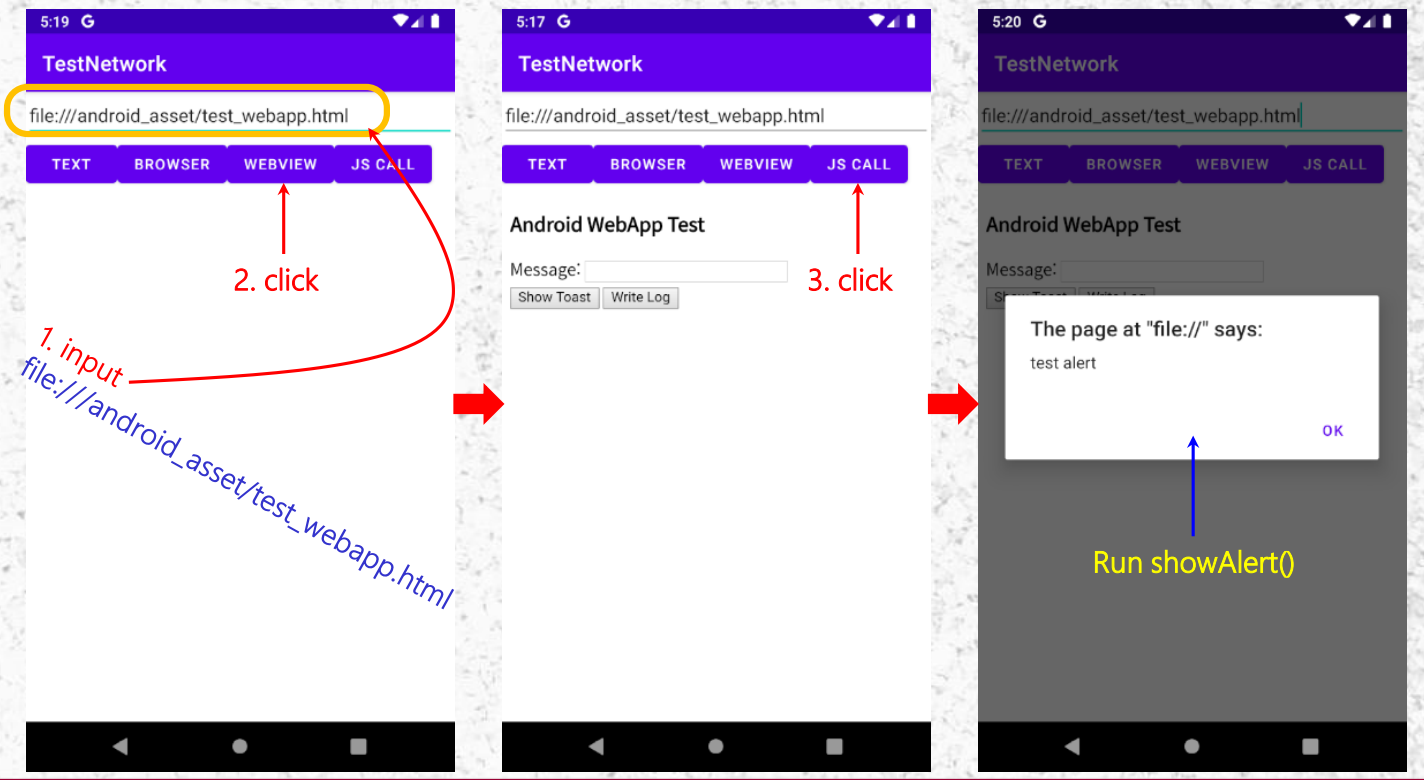
add

add

```
case R.id.jsCallButton:
    Log.d("NETWORK", "JS Call clicked!");
    if (webView != null)
        webView.evaluateJavascript("showAlert('test alert')", null);
    break;
```


Understanding Web App

- (Incorporating JavaScript)
  - Run & Test
    - First of all, load html file into WebView having JavaScript code to execute.



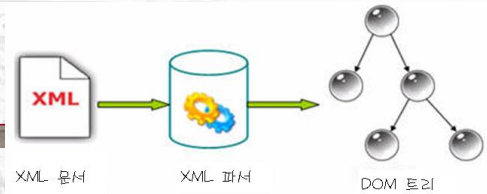
XML Handling

Types of XML Parsers

- Provide interface for application to read & handle XML document.
  - Java provides various parsers to handle XML document

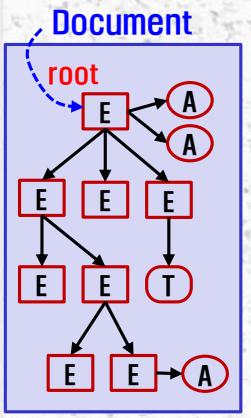
1. DOM Parser: construct whole tree of document in memory
   - Return a tree having a Document as the root, handle Node in various ways.
   - Can cause memory constraints because it keeps whole document in memory.
2. SAX Parser: process the document in an event-based triggering method
   - Call a callback for a token found in the document
   - Low memory profile, but it is difficult to reprocess past elements.
3. JDOM Parser: similar to DOM Parser, but more easier to handle.
   - Combine advantages of DOM and SAX API, low memory profile & fast speed
4. StAX Parser: similar to SAX Parser, but more efficient
   - StAX is a PULL API that application asks information to the parser, while SAX is a PUSH API that parser notifies information to the program.
5. XPath Parser: extract specific elements by specifying access path.
   - XPath: W3C recommended scheme to find data from the XML document.
6. DOM4J Parser: Java framework library for XML, XPath & XSLT parsing, it provides DOM, SAX and JAXP, etc…

XML Handling



DOM Parser Basics

- Document Object Model: W3C recommended model as a programming interface to access and modify XML style, structure and contents.
- Each node in DOM Tree inherits org.w3c.dom.Node interface.
  - Node type(Node.getNodeType()): Document, Element, Attribute, Text, …

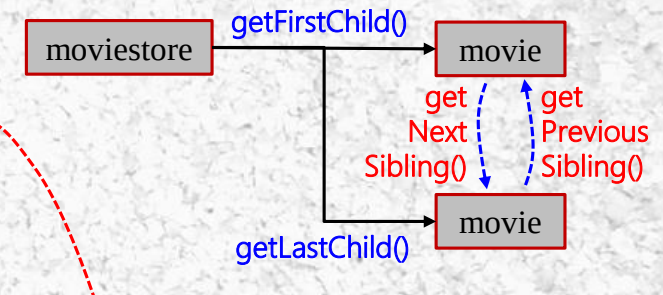


    - org.w3c.dom.Document: Whole tree obtained by parsing the document.
      - » Document.getDocumentElement() provides root element of the document.
      - » NodeList Document.getElementsByTagName(String)
    - Element: information construct defined as a tag in XML, <tag>…</tag> or <tag/>
      - » NodeList Element.getElementsByTagName(String)
      - » String getAttribute(String name): get value of name attribute
      - » Attr getAttributeNode(String name): get Attr Node object with name
    - NamedNodeMap Node.getAttributes(): return null if not an element node.
    - org.w3c.dom.Attr: Node object representing attribute of an element
      - » String getName(), Element getOwnerElement(), String getValue()
    - org.w3c.dom.Text: Node object representing text value of the element.
  - Obtaining child's Node list(org.w3c.dom.NodeList): Node.getChildNodes()
    - NodeList's method: Node item(int index), int getLength()
  - Useful methods in Node object
    - getFirstChild(), getLastChild(), getNextSibling(), getNodeName(), getNodeValue(), getParentNode(), getPreviousSibling(), hasAttributes(), hasChildNodes(), …

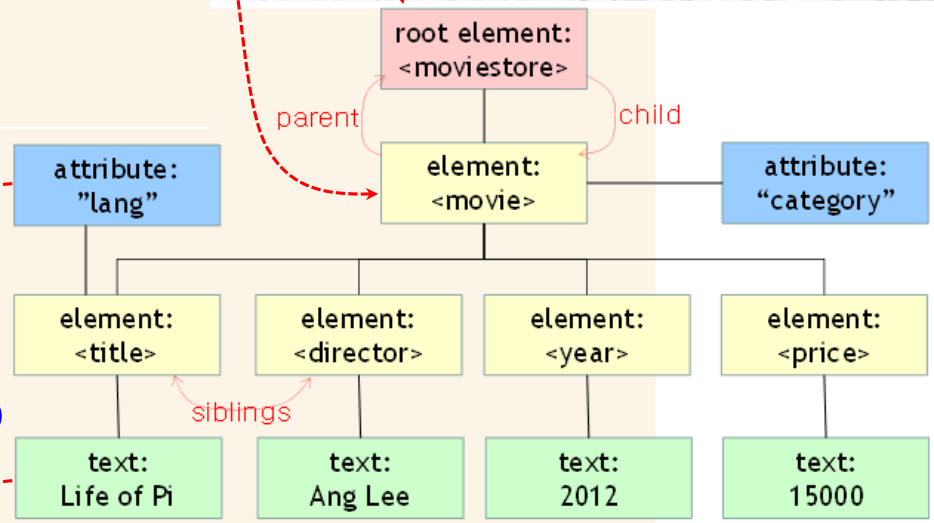
XML Handling

XML DOM Tree

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<moviestore>
  <movie category="drama">
    <title lang="en">Life of Pi</title>
    <director>Ang Lee</director>
    <year>2012</year>
    <price>15000</price>
  </movie>
  <movie category="fantasy">
    <title lang="en">Hobbit</title>
    <director>Peter Jackson</director>
    <year>2012</year>
    <price>20000</price>
  </movie>
</moviestore>
```



Document.getDocumentElement()



Node.getNodeValue()
Attr.getValue()

"en"

"Life of Pi"

Node.getNodeValue()
Text.getWholeText()

XML Handling

www.kma.go.kr: weather casting

https://www.kma.go.kr/wid/queryDFS.jsp?gridx=61&gridy=125

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<wid>
  ▶<header>
    ...
  </header>
  ▼<body>
    ▶<data seq="0">
      ...
    </data>
    ▶<data seq="1">
      ...
    </data>
    ▶<data seq="2">
      ...
    </data>
    ▶<data seq="3">
      ...
    </data>
    ▶<data seq="4">
      ...
    </data>
    ▶<data seq="5">
      ...
    </data>
    ▶<data seq="6">
      ...
    </data>
    ▶<data seq="7">
      ...
    </data>
    ▶<data seq="8">
      ...
    </data>
    ▶<data seq="9">
      ...
    </data>
    ▶<data seq="10">
      ...
    </data>
    ▶<data seq="11">
      ...
    </data>
    ▶<data seq="12">
      ...
    </data>
    ▶<data seq="13">
      ...
    </data>
    ▶<data seq="14">
      ...
    </data>
  </body>
</wid>
```

```
▼<header>
    <tm>202109200200</tm>
    <ts>0</ts>
    <x>61</x>
    <y>125</y>
  </header>
```

```
▼<data seq="0">
    <hour>6</hour>
    <day>0</day>
    <temp>20.0</temp>
    <tmx>28.0</tmx>
    <tmn>19.0</tmn>
    <sky>1</sky>
    <pty>0</pty>
    <wfKor>맑음</wfKor>
    <wfEn>Clear</wfEn>
    <pop>0</pop>
    <r12>0.0</r12>
    <s12>0.0</s12>
    <ws>1.7000000000000002</ws>
    <wd>3</wd>
    <wdKor>남동</wdKor>
    <wdEn>SE</wdEn>
    <reh>85</reh>
    <r06>0.0</r06>
    <s06>0.0</s06>
  </data>
```

```
▼<data seq="1">
    <hour>9</hour>
    <day>0</day>
    <temp>22.0</temp>
    <tmx>28.0</tmx>
    <tmn>19.0</tmn>
    <sky>1</sky>
    <pty>0</pty>
    <wfKor>맑음</wfKor>
    <wfEn>Clear</wfEn>
    <pop>0</pop>
    <r12>0.0</r12>
    <s12>0.0</s12>
    <ws>3.3000000000000003</ws>
    <wd>3</wd>
    <wdKor>남동</wdKor>
    <wdEn>SE</wdEn>
    <reh>70</reh>
    <r06>0.0</r06>
    <s06>0.0</s06>
  </data>
```

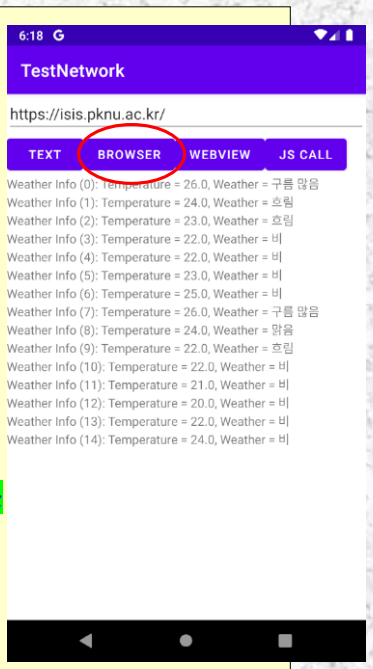
XML Handling

- (www.kma.go.kr: weather casting)
  - Use 'Test Network Project': *R.id.browserButton*
    - Modify doIt() of MainActivity class

```
public class MainActivity extends AppCompatActivity {
    ...
    public void doIt(View button) {
        switch(button.getId()) {
            ...
            case R.id.browserButton:
                Log.d("NETWORK", "browser button clicked!");
                Uri uri = Uri.parse(etUrl.getText().toString());
                startActivity(new Intent(Intent.ACTION_VIEW, uri));
                textView = new TextView(this);
                container.removeAllViews();
                container.addView(textView);
                GetXMLTask t = new GetXMLTask();
                t.execute("https://www.kma.go.kr/wid/queryDFS.jsp?gridx=61&gridy=125");
                break;
            case R.id.webviewButton:
            ...
        }
    }
    ...
}
```

delete

add



XML Handling

- (www.kma.go.kr: weather casting)
  - (Use 'Test Network Project': *R.id.browserButton*)
    - Add GetXMLTask class within MainActivity

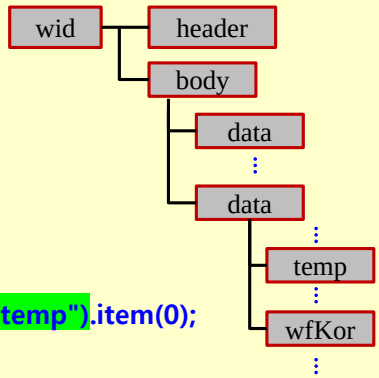
```
public class MainActivity extends AppCompatActivity {
    ...
    private class GetXMLTask extends AsyncTask<String, Void, Document> {
        protected Document doInBackground(String... urls) {
            URL url;
            Document doc = null;
            try {
                url = new URL(urls[0]);
                DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
                DocumentBuilder db = dbf.newDocumentBuilder();
                doc = db.parse(new InputSource(url.openStream()));
                doc.getDocumentElement().normalize();
            } catch (Exception e) {
                Log.e("NETWORK", e.getMessage());
            }
            return doc;
        }
        ...
    }
}
```

add

XML Handling

- (www.kma.go.kr: weather casting)
  - (Use 'Test Network Project': *R.id.browserButton*)
    - (Add GetXMLTask class within MainActivity)

```
private class GetXMLTask extends AsyncTask<String, Void, Document> {
    ...
    protected void onPostExecute(Document doc) {
        StringBuilder sb = new StringBuilder();
        NodeList nodeList = doc.getElementsByTagName("data");
        for (int i = 0; i < nodeList.getLength(); i++) {
            sb.append("Weather Info (").append(i).append("): ");
            Element dataElem = (Element) nodeList.item(i);
            Element tempElem = (Element) dataElem.getElementsByTagName("temp").item(0);
            Text textNode = (Text) tempElem.getChildNodes().item(0);
            sb.append("Temperature = ").append(textNode.getNodeValue()).append(", ");

            Element wfKorElem = (Element) dataElem.getElementsByTagName("wfKor").item(0);
            textNode = (Text) wfKorElem.getChildNodes().item(0);
            sb.append("Weather = ").append(textNode.getNodeValue()).append("₩n");
        }
        if (textView != null)
            textView.setText(sb.toString());
    }
}
```



XML Handling

- ((www.kma.go.kr: weather casting)
  - (Use 'Test Network Project': *R.id.browserButton*)