**MIND STORM SOFTWARE PVT LTD**  
**Android Development Training**

**Hands-On Exercise – ex06 - Web View Activity.doc**

**Objective**This exercise demonstrates how to use the Android WebView to display HTML pages or custom HTML. This is useful to show HTML content like README or help, etc. It is also a powerful component and fully functional Javascript can be embedded inside it or even remotely loaded from a URL.

**Assumptions**

* Development Environment for Android (Java SDK, Eclipse, Android SDK) has been setup successfully.
* You are familiar with using Eclipse.
* Android SDK 4.x is available and Android Virtual Devices are already created.
* Start the Android Virtual Device to save time.
  + Click on **Window 🡪 Android SDK and AVD Manager**.
  + Select an **Android 4.x** compatible AVD and click on **Start**
  + Select **Scale display to real size** and provide a **Screen Size (in)** as **5 inches** or any other appropriate size for your development machine.

**Step by Step Instructions**

**Step 1 – Create the Android Project**

1. Create a new project. Click on **File🡪New 🡪 Android Application Project**
2. Enter **Project Name or Application Name** as **FileReadingExercise .**Click on **Next.**
3. Enter **Package Name** as **com.mindstorm.filereader .** Click on **Next**
4. Deselect (Uncheck) the **Create custom launcher icon.** Click on **Next.**
5. In **Create Activity**, go with the default options i.e. go with Create Activity and Blank Activity as selected. Click on **Next.**
6. On the **New Blank Activity,** let the Activity Name be **MainActivity** and change the Layout Name to **main**
7. Click on **Finish**
8. (Optional): Verify that the Project runs in your Emulator by **Right-click** the **Project** and **Run As 🡪 Android Application**

**Step 2 – Create the Layout**

Go to **/res/layout** folder and modify **activity\_main.xml** as shown below:

We have used the standard WebView view over here.

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<Spinner

android:id="@+id/spinnerSubjects"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:entries="@array/subjects" />

<Button

android:id="@+id/btnReadFile"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/readfile" >

</Button>

<WebView

android:id="@+id/fileContents"

android:layout\_width="match\_parent"

android:layout\_height="fill\_parent">

</WebView>

</LinearLayout>

**Step 3 – Create a sample HTML file**

In this example, we will ship a sample HTML files with the application itself. We can place these files in the /**assets** folder. Follow these steps:

1. Go to **/assets** folder.
2. **Right-click** and select **New 🡪 File.**
3. Enter the name as **english.html**
4. Modify the contents of the **english.html** as shown below:

<h2>English Notes</h2>

<ul>

<li>There are direct questions on synonyms, antonyms, idioms and phrases.</li>

<li>You can read the passage faster, if your vocabulary is good, else you’ll have to pause and think.</li>

<li>In comprehension passages, some questions are in the format of “what is the meaning of XYZ word in the sentence"</li>

</ul>

1. Go to **/assets** folder.
2. **Right-click** and select **New 🡪 File.**
3. Enter the name as **physics.html**
4. Modify the contents of the **physics.html** as shown below:

<h2>Physics Notes</h2>

<ul>

<li>The standard unit of mass is the kilogram.</li>

<li>The standard unit of time is the second.</li>

</ul>

**Step 4 – Code the Main Activity**

1. Modify the **MainActivity.java** code as shown below:

package com.mindstormsoftware.filereader;

import android.app.Activity;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.webkit.WebView;

import android.widget.Button;

import android.widget.Spinner;

import android.widget.Toast;

public class MainActivity extends Activity {

WebView fileContents;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

fileContents = (WebView)findViewById(R.id.fileContents);

//Read File

Button btnReadFile = (Button)findViewById(R.id.btnReadFile);

btnReadFile.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

try {

//Read the current selection for the subject

Spinner spinnerSubjects = (Spinner)findViewById(R.id.spinnerSubjects);

String strSelectedSubject = spinnerSubjects.getSelectedItem().toString();

//Update the Contents field in the UI

fileContents.loadUrl("file:///android\_asset/" + strSelectedSubject.toLowerCase() + ".html");

}

catch (Exception ex) {

showToast(ex.getMessage());

}

}

});

}

private void showToast(String msg) {

Toast.makeText(this, msg, Toast.LENGTH\_SHORT).show();

}

}

**Step 5 – Modify the AndroidManifest.xml**

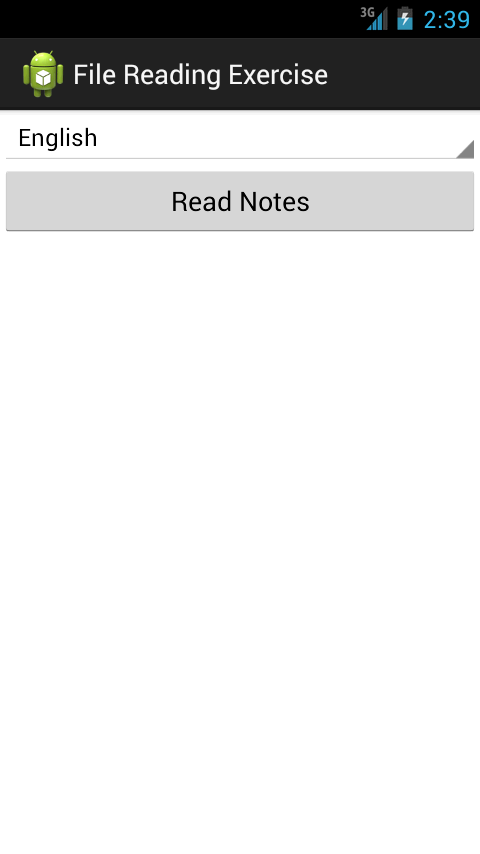
If you are going to load the HTML content from the web, it is important to add the required **permissions** in the **AndroidManifest.xml** file. Go to **AndroidManifest.xml** file and add the following permission outside of the <application> tag.

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

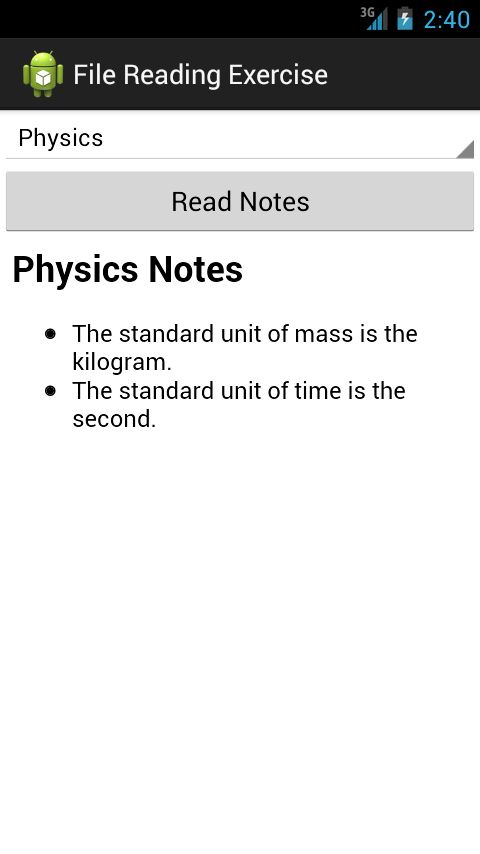
**Step 6 – Run the Example**

1. **Right Click** the **Project** in Eclipse.
2. Select **Run As 🡪 Android** Application
3. (Optional): If you have multiple compatible AVDs running, select the correct AVD. In our case it is the 4.x AVD.

You should see the MainActivity screen come up.



Select any of the two subjects : English or Physics from the list and it will read the contents of the file from the assets folder and show it as shown below:



**Experiment with the following:**

1. Uncomment some of the other ways to load HTML content as given in the code. And see it work.

**Summary**

This hands-on exercise demonstrated how to use the WebView view to load HTML content.

For more information: <http://developer.android.com/reference/android/webkit/WebView.html> and <http://developer.android.com/guide/webapps/webview.html>