**Android programmatically update application when a new version is available**

If you have apps that are hosted on your server and as time passes you make enhancements, fix bugs, etc. then comes the question how can I alert all my app users that a newer version is out there and make it real easy for them to download. You can email all your users with the link to the new version, that will be very cumbersome. Well that process needs to be built right into the app you deploy it for the first time.

|  |  |
| --- | --- |
| Here is an example how to do it, the basic idea is outlined below   1. Create a Web Service which your app can poll whenever the app launches or based on some time limit that can check if there is new version out there. 2. This Web service should return the lastest Version of the apk file that is hosted on the Server along with the URI of the application file that has the new version. 3. When your app gets the response from the Web Service, it will parse the JSON and check your app version to the lastest version that is available on the server. 4. If your app version is lower than the latest version it will prompt the user to start the download process. 5. The download of the new app is handled by the Download Manager. 6. The download manager will notify your app using Broadcast receiver when the download is complete. 7. Upon completion of the latest version of the application file the you can start the activity to install that file. 8. At this point user needs to say OKAY, lets do it. | [Android update application programmatically](http://1.bp.blogspot.com/-uJgMfSA3irc/UYsZpABgbpI/AAAAAAAADQk/3CThMpd0Fhg/s1600/Android+update+application+programmatically.png) |

**Java Servlet Web Service to check New Version - CheckAppVersion.java**

package com.as400samplecode;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.google.gson.JsonObject;

public class CheckAppVersion extends HttpServlet {

private static final long serialVersionUID = 1L;

public CheckAppVersion() {

super();

}

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

doPost(request,response);

}

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

PrintWriter out = response.getWriter();

response.setContentType("text/html");

//send a JSON response with the app Version and file URI

JsonObject myObj = new JsonObject();

myObj.addProperty("success", true);

myObj.addProperty("latestVersion", 2);

myObj.addProperty("appURI", "http://demo.mysamplecode.com/Servlets\_JSP/apps/MyAndroidApp.apk");

out.println(myObj.toString());

out.close();

}

}

**Application Manifest file - AndroidManifest.xml**

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

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

package="com.as400samplecode"

android:versionCode="1"

android:versionName="2.1.3" >

<uses-sdk

android:minSdkVersion="16"

android:targetSdkVersion="17" />

<application

android:allowBackup="true"

android:icon="@drawable/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/AppTheme" >

<activity

android:name="com.as400samplecode.MainActivity"

android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

**Application Layout file - activity\_main.xml**

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

<RelativeLayout 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" tools:context=".MainActivity"

android:padding="5dp">

<TextView android:id="@+id/versionName" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_alignParentLeft="true"

android:layout\_alignParentTop="true" android:layout\_marginTop="16dp"

android:text="What Version?" android:textAppearance="?android:attr/textAppearanceMedium"

android:textStyle="bold" />

<TextView android:id="@+id/textView1" android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:layout\_below="@+id/versionName"

android:layout\_marginTop="16dp" android:text="@string/copyright"

android:textAppearance="?android:attr/textAppearanceMedium" />

</RelativeLayout>

**IntentService for making Web Service requests - MyWebService.java**

package com.as400samplecode;

import java.io.ByteArrayOutputStream;

import java.io.IOException;

import org.apache.http.HttpResponse;

import org.apache.http.HttpStatus;

import org.apache.http.StatusLine;

import org.apache.http.client.ClientProtocolException;

import org.apache.http.client.HttpClient;

import org.apache.http.client.methods.HttpGet;

import org.apache.http.conn.params.ConnManagerParams;

import org.apache.http.impl.client.DefaultHttpClient;

import org.apache.http.params.HttpConnectionParams;

import org.apache.http.params.HttpParams;

import com.as400samplecode.MainActivity.MyWebReceiver;

import android.app.IntentService;

import android.content.Intent;

import android.util.Log;

public class MyWebService extends IntentService{

private static final String LOG\_TAG = "MyWebService";

public static final String REQUEST\_STRING = "myRequest";

public static final String RESPONSE\_STRING = "myResponse";

public static final String RESPONSE\_MESSAGE = "myResponseMessage";

private String URL = null;

private static final int REGISTRATION\_TIMEOUT = 3 \* 1000;

private static final int WAIT\_TIMEOUT = 30 \* 1000;

public MyWebService() {

super("MyWebService");

}

@Override

protected void onHandleIntent(Intent intent) {

String requestString = intent.getStringExtra(REQUEST\_STRING);

Log.v(LOG\_TAG, requestString);

String responseMessage = "";

try {

URL = requestString;

HttpClient httpclient = new DefaultHttpClient();

HttpParams params = httpclient.getParams();

HttpConnectionParams.setConnectionTimeout(params, REGISTRATION\_TIMEOUT);

HttpConnectionParams.setSoTimeout(params, WAIT\_TIMEOUT);

ConnManagerParams.setTimeout(params, WAIT\_TIMEOUT);

HttpGet httpGet = new HttpGet(URL);

HttpResponse response = httpclient.execute(httpGet);

StatusLine statusLine = response.getStatusLine();

if(statusLine.getStatusCode() == HttpStatus.SC\_OK){

ByteArrayOutputStream out = new ByteArrayOutputStream();

response.getEntity().writeTo(out);

out.close();

responseMessage = out.toString();

}

else{

Log.w("HTTP1:",statusLine.getReasonPhrase());

response.getEntity().getContent().close();

throw new IOException(statusLine.getReasonPhrase());

}

} catch (ClientProtocolException e) {

Log.w("HTTP2:",e );

responseMessage = e.getMessage();

} catch (IOException e) {

Log.w("HTTP3:",e );

responseMessage = e.getMessage();

}catch (Exception e) {

Log.w("HTTP4:",e );

responseMessage = e.getMessage();

}

Intent broadcastIntent = new Intent();

broadcastIntent.setAction(MyWebReceiver.PROCESS\_RESPONSE);

broadcastIntent.addCategory(Intent.CATEGORY\_DEFAULT);

broadcastIntent.putExtra(RESPONSE\_MESSAGE, responseMessage);

sendBroadcast(broadcastIntent);

}

}

**Application Activity - MainActivity.java**

package com.as400samplecode;

import org.json.JSONException;

import org.json.JSONObject;

import android.net.ConnectivityManager;

import android.net.NetworkInfo;

import android.net.Uri;

import android.os.Bundle;

import android.os.Environment;

import android.app.Activity;

import android.app.AlertDialog;

import android.app.DownloadManager;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.IntentFilter;

import android.content.pm.PackageInfo;

import android.content.pm.PackageManager.NameNotFoundException;

import android.util.Log;

import android.view.Menu;

import android.widget.TextView;

public class MainActivity extends Activity {

private static final String LOG\_TAG = "AppUpgrade";

private MyWebReceiver receiver;

private int versionCode = 0;

String appURI = "";

private DownloadManager downloadManager;

private long downloadReference;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//Overall information about the contents of a package

//This corresponds to all of the information collected from AndroidManifest.xml.

PackageInfo pInfo = null;

try {

pInfo = getPackageManager().getPackageInfo(getPackageName(), 0);

}

catch (NameNotFoundException e) {

e.printStackTrace();

}

//get the app version Name for display

String version = pInfo.versionName;

//get the app version Code for checking

versionCode = pInfo.versionCode;

//display the current version in a TextView

TextView versionText = (TextView) findViewById(R.id.versionName);

versionText.setText(version);

//Broadcast receiver for our Web Request

IntentFilter filter = new IntentFilter(MyWebReceiver.PROCESS\_RESPONSE);

filter.addCategory(Intent.CATEGORY\_DEFAULT);

receiver = new MyWebReceiver();

registerReceiver(receiver, filter);

//Broadcast receiver for the download manager

filter = new IntentFilter(DownloadManager.ACTION\_DOWNLOAD\_COMPLETE);

registerReceiver(downloadReceiver, filter);

//check of internet is available before making a web service request

if(isNetworkAvailable(this)){

Intent msgIntent = new Intent(this, MyWebService.class);

msgIntent.putExtra(MyWebService.REQUEST\_STRING, "http://demo.mysamplecode.com/Servlets\_JSP/CheckAppVersion");

startService(msgIntent);

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.activity\_main, menu);

return true;

}

@Override

public void onDestroy() {

//unregister your receivers

this.unregisterReceiver(receiver);

this.unregisterReceiver(downloadReceiver);

super.onDestroy();

}

//check for internet connection

private boolean isNetworkAvailable(Context context) {

ConnectivityManager connectivity = (ConnectivityManager) context.getSystemService(Context.CONNECTIVITY\_SERVICE);

if (connectivity != null) {

NetworkInfo[] info = connectivity.getAllNetworkInfo();

if (info != null) {

for (int i = 0; i < info.length; i++) {

Log.v(LOG\_TAG,String.valueOf(i));

if (info[i].getState() == NetworkInfo.State.CONNECTED) {

Log.v(LOG\_TAG, "connected!");

return true;

}

}

}

}

return false;

}

//broadcast receiver to get notification when the web request finishes

public class MyWebReceiver extends BroadcastReceiver{

public static final String PROCESS\_RESPONSE = "com.as400samplecode.intent.action.PROCESS\_RESPONSE";

@Override

public void onReceive(Context context, Intent intent) {

String reponseMessage = intent.getStringExtra(MyWebService.RESPONSE\_MESSAGE);

Log.v(LOG\_TAG, reponseMessage);

//parse the JSON response

JSONObject responseObj;

try {

responseObj = new JSONObject(reponseMessage);

boolean success = responseObj.getBoolean("success");

//if the reponse was successful check further

if(success){

//get the latest version from the JSON string

int latestVersion = responseObj.getInt("latestVersion");

//get the lastest application URI from the JSON string

appURI = responseObj.getString("appURI");

//check if we need to upgrade?

if(latestVersion > versionCode){

//oh yeah we do need an upgrade, let the user know send an alert message

AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);

builder.setMessage("There is newer version of this application available, click OK to upgrade now?")

.setPositiveButton("OK", new DialogInterface.OnClickListener() {

//if the user agrees to upgrade

public void onClick(DialogInterface dialog, int id) {

//start downloading the file using the download manager

downloadManager = (DownloadManager)getSystemService(DOWNLOAD\_SERVICE);

Uri Download\_Uri = Uri.parse(appURI);

DownloadManager.Request request = new DownloadManager.Request(Download\_Uri);

request.setAllowedNetworkTypes(DownloadManager.Request.NETWORK\_WIFI);

request.setAllowedOverRoaming(false);

request.setTitle("My Andorid App Download");

request.setDestinationInExternalFilesDir(MainActivity.this,Environment.DIRECTORY\_DOWNLOADS,"MyAndroidApp.apk");

downloadReference = downloadManager.enqueue(request);

}

})

.setNegativeButton("Remind Later", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

// User cancelled the dialog

}

});

//show the alert message

builder.create().show();

}

}

} catch (JSONException e) {

e.printStackTrace();

}

}

}

//broadcast receiver to get notification about ongoing downloads

private BroadcastReceiver downloadReceiver = new BroadcastReceiver() {

@Override

public void onReceive(Context context, Intent intent) {

//check if the broadcast message is for our Enqueued download

long referenceId = intent.getLongExtra(DownloadManager.EXTRA\_DOWNLOAD\_ID, -1);

if(downloadReference == referenceId){

Log.v(LOG\_TAG, "Downloading of the new app version complete");

//start the installation of the latest version

Intent installIntent = new Intent(Intent.ACTION\_VIEW);

installIntent.setDataAndType(downloadManager.getUriForDownloadedFile(downloadReference),

"application/vnd.android.package-archive");

installIntent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(installIntent);

}

}

};

}