

First, download & install the tools you need:

The very first thing you'll need is your copy of wp-android. If you don't have it, you can grab it here:

[http: link here](#), [blog post link?](#) [Direct download link?](#)

Extract the zip to a convenient location on your computer, and ensure that the content of your folder are like the image to the right.

Feel free to rename the containing folder to anything you like.

Time to setup your development environment:

First of all you'll need to download JDK 6. Please note that JDK 7 and JDK 8 will not work. It must be JDK 6.

Mac OS users: JDK 6 is installed as standard on all Mac's from Snow leopard and up. Reference: <https://developer.apple.com/library/mac/documentation/Java/Conceptual/Java14Development/02-JavaDevTools/JavaDevTools.html>

You'll also want to make sure you have Java 6 installed for running Eclipse, this can be downloaded here <http://support.apple.com/kb/DL1572>

Windows users: You can download JDK 6 here <http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html#jdk-6u45-oth-JPR>

Download and extract the zip files (windows) or disk image, .dmg (mac) and follow the on screen instructions.

All installed? Great! Let's move on then.

Next up your need to get Eclipse installed, this can be downloaded here <http://eclipse.org/downloads/>

Select your operating system from the drop down box on the right side, in this case it's Mac OS X. Directly under this you'll see "Eclipse IDE for Java Developers". Click the appropriate link on the right hand side (32 bit or 64 bit).

Now click the main download link "[United Kingdom] UK Mirror Service (http)" in my case, this starts off the download of a .tar.gz file, open this with Archive Utility.

Once unzipped you should see a folder called "Eclipse" double click that to open the folder and then you should see the contents like the screenshot to the right.

Double click "Eclipse" to launch it. On Mac, if you get a error message "Eclipse" can't be opened because it is from an unidentified developer. There are two ways round this.

In Finder, hold down ctrl and then click "Eclipse" (keep ctrl held down) and then click open.

Alternatively you can edit your security preferences, click the apple icon > system preferences > security and privacy, click the lock to make changes (you'll need to enter your administrator password and be logged on with a administrator account if you aren't already.

On the first tab "General" you'll see a option called "Allow apps downloaded from:".

“Mac App Store”

“Mac App Store and identified developers”

“Anywhere”

Pick “Anywhere” confirm and then click the lock to prevent further changes.

Now that’s sorted it’s time to open Eclipse.

Go to your eclipse folder and double click “Eclipse” to launch it.

You’ll be prompted to name your workspace, important: this is a location on your computer, so you should keep the full path for example /Users/Username/Documents/workspace, change “workspace” to what you want.

Once entered, click ok and Eclipse will continue launching. You’ll then see a welcome screen (see screenshot to the right).

Installing the ADT plugin.

Congrats! You have Eclipse installed and launched, now it’s time to install the ADT plugin (Android Development tools).

The Android developers site has a great guide for this here <http://developer.android.com/sdk/installing/installing-adt.html> but let’s run through it just to make sure.

Open up your Eclipse workspace and click “Help” then “Install new software” click that.

You should now have something that looks like the screenshot to the right.

Where you see “Work with: type or select a site”. Copy and paste “<https://dl-ssl.google.com/android/eclipse/>” and click “Add” It should now load the available packages. You’ll need all these and they’re checked by default.

So click “Next” here you’ll need to select the licensing agreement. Once accepted click “Finish”.

The ADT packages will now install, while the packages are installing you may get a message that reads “Warning: You are installing software that contains unsigned content. The authenticity or validity of this software cannot be established. Do you want to continue with the installation?” Just click “ok” to continue.

Once the ADT packages have been installed successfully, you’ll get a prompt to restart Eclipse (see screenshot to the right). You’ll need to restart Eclipse for the changes to take effect so click “Yes”.

Congrats you’ve successfully installed and configured Eclipse, now it’s time to get your App running!

Now you’ll need to get the correct Android SDK’s installed, there’s a fantastic guide on that here <http://developer.android.com/tools/help/sdk-manager.html>

You’ll want SDK’s 8-22, follow the above guide and install them, then restart Eclipse.

In Eclipse, click File > import > existing Android code into workspace. Click “Next”. Now click on browse and locate the WP for Android package you downloaded earlier, you then want to open that and import the WorkspaceEdublogs folder, so click that (don’t open it).

You’ll import screen should now look like this one to the right, you should see:

actionbarsherlock
menudrawer
wp-android
wp-android/tests

Uncheck wp-android/tests (we don't need this folder).

And now click "Finish".

The project has now been imported successfully and is ready to use, note that you'll see multiple warnings these can be safely ignored, they won't stop the app from running.

Modifying your app:

Now it's time to modify your app.

Click the arrow next to wp-android on the left, then click androidmanifest.xml.

You'll notice that along the tabs at the bottom there's Manifest, Application, Permissions, Instrumentation, Androidmanifest.xml, we actually want the xml code, so again click the AndroidManifest.xml tab.

You should see this code at the very top, lines 1-6.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="org.edublogs.android"
    android:installLocation="auto"
    android:versionCode="1"
    android:versionName="1.0" >
```

Where you see "org.edublogs.android" "Version code" and "Version name" change these as needed. Important your package name should end .android, you normally use your domain backwards.

So if your domain is example.com your package would be com.example.android

As this is your first release of the app, version code should be 1 and version name should be 1.00

Now lines 8-10, you should see this code.

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="17" />
```

Change that so the minSdkVerion equals 14, rather than 8.

Scroll down a little further in the file, specifically to lines 54-60. You'll see this code.

```
<application
    android:name="WordPress"
    android:allowBackup="false"
```

```
android:hardwareAccelerated="true"
android:icon="@drawable/app_icon"
android:label="edublogs"
android:theme="@style/WordPress" >
```

Change edublogs to your app name, for example your business name.

Once you've made your changes, in Eclipse go to File > Save as and confirm the overwriting of the existing file.

You'll notice now there's over 2,000 errors! Don't panic, this is expected and perfectly normal, it's because by renaming your app, the sub packages and references are still pointing to their old destination so we need to update those to get this up and running. Follow the instructions and you'll be set. But first let's finish modifying our app.

Open up res >> values >> strings.xml. Scroll right down to the end, the important code you want to change is line numbers 455-463 which is

```
<!-- WORDPRESS CONFIGURATION -->
<string name="readerURL">https://en.edublogs.org/reader/mobile/v2</string>
<string name="readerLoginURL">http://edublogs.org/wp-login.php</string>
<string name="helpURL">http://help.edublogs.org/android/</string>
<string name="wpcomXMLRPCURL">http://edublogs.org/xmlrpc-
JepexZak0e.php</string>
<string name="wpcomLoginURL">http://edublogs.org/wp-login.php</string>
<string name="signupURL">https://edublogs.org/signup/?ref=wp-android</string>

</resources>
```

Change those URL's to your own site's domain. Then file >> save as.

Now let's get our App images modified.

Before we can actually see our customized app in a virtual device, we need to actually customize some stuff. So let's do the stuff that we will see on the screen of the virtual device once we launch it: images.

Minimize the **Eclipse** program window to your system tray for now. We'll get back to it in a bit.

Browse to your **workspace** folder and open it.
Open the **wp-android** folder, then the **res** folder inside

that.

Inside **res** are a number of **drawable** folders. Those folders contain all the app images. You'll want to replace several with your own custom stuff so your app looks the way you want it to.

If you take a look through some of those folders, you'll see several images of varying sizes with the Edublogs logo or icon.

Those are the images you want to replace. You gotta check them carefully too, because some are actually transparent :)

While there are a couple of dozen images that need to be swapped out, you don't need to make them all individually yourself.

There's a really cool timesaver available online to automatically create all the standard images you need for your Android app: ***Makeappicon.com***

<http://makeappicon.com/>

As an added bonus, the online tool also creates all variations needed for iOS and iOS7 devices!

Simply upload a jpg, png or psd of the image you want to use (best size is 1024x1024px) and let the magic happen!

Once you've created your icon, you'll receive a zipped folder by email with all your icons pre-formatted and properly named for a new Android app.

Extract the contents of the zip and, inside, you'll see 3 folders.

Open the **android** folder and you'll find several folders with names just like the ones in your app that need editing. Inside each folder is a version of your icon of the exact size you need for the corresponding folder in the app.

Each icon you replace with your own must have the exact **same name as the original** in the app, otherwise the app won't find them.

You will need to create a few extra icons for those that *Makeappicon* can't know about (this is a custom app after all). Simply make copies of any one of them and resize to fit where needed.

Now in Eclipse in the Package Explorer (left hand side). click the arrow next to src, highlight each individual file one by one and rename.

To rename highlight your chosen file by clicking on it's name once, then click "Refactor" >> "Rename". When you click "Rename", a new pop up window will show, enter in your new name for that package. Uncheck "update references" and click ok.

Note: Don't rename org.xmlrpc.android this should stay the same, only rename the .org part if your package starts with another extension, .com for example.

Important: you should only change the .org and the .edublogs parts, the rest should rename the same.

Now you've updated all your package names, go click file >> Project >> Clean. Once it's completed cleaning the project, you'll still see errors, this is because the important references are still pointing to the old package names.

You need to change the imports in each one of those files, don't worry there aren't 1000+ lines to change, fixing one file, will fix hundreds of the errors.

For example let's take the code from this path src/org.edublogs.android/WordPress.java

Lines 1-30 should be

```

package org.edublogs.android;

import java.io.UnsupportedEncodingException;
import java.net.URLDecoder;
import java.util.HashMap;
import java.util.Map;
import java.util.UUID;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.support.v4.app.NotificationCompat;
import android.support.v4.content.IntentCompat;
import android.util.Log;
import android.graphics.BitmapFactory;

import com.google.android.gcm.GCMBaseIntentService;

import org.edublogs.android.ui.notifications.NotificationsActivity;
import org.edublogs.android.ui.posts.PostsActivity;
import org.edublogs.android.util.ImageHelper;
import org.edublogs.android.util.StringUtils;
import org.edublogs.android.WPComXMLRPCApi;

```

Notice how those last four imports all reference the old edublog's package name? Change those to your name, for example.

```
import org.edublogs.android.ui.notifications.NotificationsActivity;
```

Becomes

```
import com.example.android.ui.notifications.NotificationsActivity;
```

Once you've successfully modified your code, make sure to go to file >> save as, before moving onto the next one. Do this for every file with a red exclamation mark within the src folder.

When these changes are complete, go to Project >> Clean.

Your app should now be error free! You'll see 170 warnings, but don't worry you're safe to ignore these, they won't impact your ability to publish the app to the Google Play store.

Congrats! You now have a modified Android application for your WordPress site.

Exporting your app:

You now need to export your app as a Android Application (.apk).

In Eclipse, click File >> Export >> Android >> Export Android Application, then click “Next >”.

It'll now ask you to select your project, you should see wp-android as that's the main folder, click next.

Next we need to create a keystore, click “create a new keystore” Then click browse.

This should bring up a new window to enter your keystore name, your keystore name must end in .key

For example myapp.key.

Then enter a password and then repeat the password in the confirm field and click next.

Next you need to fill in the required information to generate your key. The required fields are Alias, password (the one you just created), confirm (your password repeated), validity in years.

And then either First and Last Name or Organisational unit. Once all this has been filled in click “Next >”.

You'll now need to enter a destination for your .apk file, make sure this is somewhere you can get to easily, as you'll need the .apk to upload to the Google Play Store.

Click “browse” enter a name, yourapp.apk and save.

Congrats your App is now complete and ready to upload to the Google Play Store!

Getting your app into the Google Play Store:

Go to Google Play Developers at <https://play.google.com/apps/publish/signup/> If you don't already have an account at Google Play,

you'll need to create one before continuing.

Click Add New Application, choose the Language and enter the Title of your app.

Click Upload new APK to Production.

Click Store listing in the left menu, then enter the Title and Description of your app that will be seen in the App store, and by other users in searches, etc.

Enter some Promo text if desired. This isn't a required field, but can be filled with any promotional text you'd like.

Recent changes is where you would enter any recent changes to the app: updated or new features etc. As this is the 1st version, you wouldn't enter anything here.

Add your Graphic Assets for each type (Phone, 7 inch tablet, 10 inch tablet), these are meant to be screenshots of your app in action :)

Then add your high res icon, featured graphic and promo graphic if required (only the high res icon is required here).

You can even add a promo video if you'd like.

Under Categorization, select an Application type. Then pick a Category, and select the Content Rating.

In the Contact details section, you can enter a Website or Email address, most people would enter a website here rather than email, though you could of course enter both.

If you have a Privacy policy, link to it, or check the box "Not submitting a privacy policy URL at this time"

Click Save :)

Now click Pricing & Distribution on the left menu. Select the Countries you want your app to be distributed to.

Finally, you must agree to the terms and conditions.

Once all this is done, click on Publish my App and you're done :)