

## NodeBB WebView App

Welcome to the documentation for the Android or actually NodeBB WebView App.

### Start:

- 1) Download and install Android Studio (JDK 7 or higher are required) , just like Android SDK Platform 22.
- 2) Open Android Studio and select to import an existing project. Now simply select the unpacked folder.
- 3) This is it. You have successfully opened the code in Android Studio and are now ready to edit the code.

### Code Structure:

Basically the code itself is very simple. You can find the main code in ***/app/src/main/java/com.webview.nodebb.***

Here a little explanation for all java files:

#### **- GTracker.java:**

Creates a Google Analytics instance for tracking.

#### **- MainActivity.java:**

Is what the end user sees. It calls all other java files and creates the actual application.

#### **- WebViewConfig.java:**

Almost self-explaining. Allows you to enable or disable specific elements, like Google Analytics or AdMob.

#### **- drawer/DrawerAdapter.java:**

Declares the functions of the drawer.

#### **- fragment/MainFragment.java:**

Actual WebView file.

- **fragment/TaskFragment.java:**

“Loads” site and handles callbacks.

- **utility/DownloadUtil.java:**

Download manager.

- **utility/MediaUtility.java:**

Gets file paths.

- **utility/NetworkInf.java:**

Checks the network type.

- **view/ViewState.java:**

Define view state.

Basically this is everything the code is built upon. As you can see it is basically extremely simple, but therefore also relatively easy to advance.

Let’s take a look at the XML files, at least the ones we should care about.

They can be found in ***app/src/main/res***.

For us only the folder “**values**” matters.

**admob.xml:**

Enter AdMob ad ID and test device ID, if wished.

**navigation.xml:**

Defines navigation elements. To add a new one simply create a new <item>.

Samples can be found in the file itself. To add icons simply right click on the res folder and select "Image Asset". Now you can import your pictures/icons. Be sure to select "Action Bar and Tab Icons".

**strings.xml:**

Defines app name and text to return on an error.

**themes.xml:**

Defines the different themes.

That's basically everything you need to know about the files.

### **F.A.Q.**

**Q.:** How do I change the package name?

**A.:** Right click on the java folder > Create Package. Now simply drag and drop the files to the new folder and also edit package name in the AndroidManifest.xml. Also clean/rebuild the project because of possible "R" errors.

**Q.:** How do I change the theme?

**A.:** Open the AndroidManifest.xml and change "android:theme".

**Q.:** How do I change the Launcher Icon?

**A.:** Same way as you added an app icon, however, this time make sure it is a "Launcher Icon".

**Q.:** How do I get an APK?

**A.:** First of all, you need to create a keystore by running:

```
keytool -genkey -v -keystore webviewapp.keystore -alias <YOURALIAS> -keyalg RSA -keysize 2048 -validity 36500
```

Then enter the alias and the password into **webviewapp.properties**.

Now run "**gradlew assemble**". The APK can be found at **app/build/outputs**.