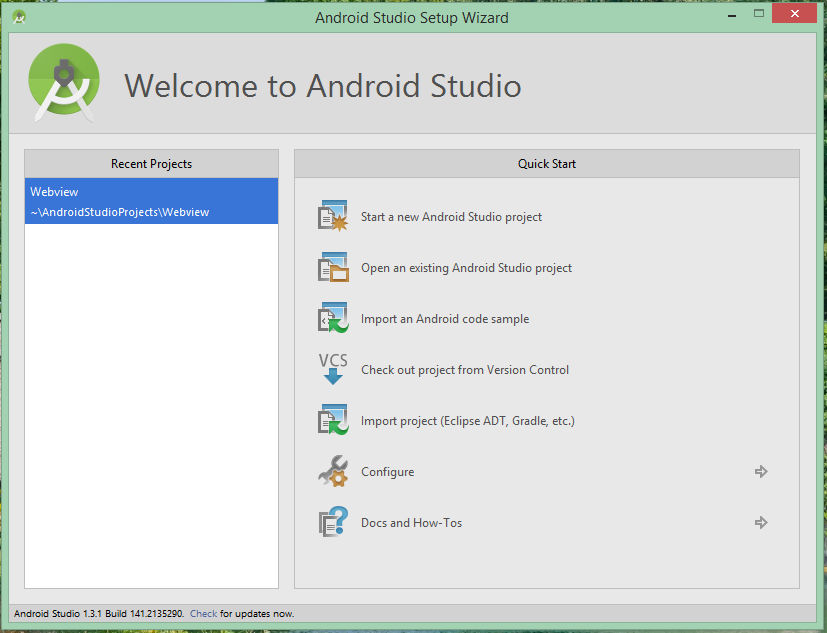
**Material Webview step by step instructions**

1. Download and install Android Studio <https://developer.android.com/sdk/index.html>
2. Install all required packages

* Click *Tools->Android->SDK Manager* in the menu
* Please enable the following options:
  + „Android 5.1 Lollipop“ in tab „SDK Platform“
  + „Android SDK Build Tools“,“Android SDK Platform Tools-22“,  
    “Android Support Repository, Rev xx“, „Google Repository, Rev xx“  
    „Google USB Driver, Rev xx“
* Click „Apply“ and follow the instructions

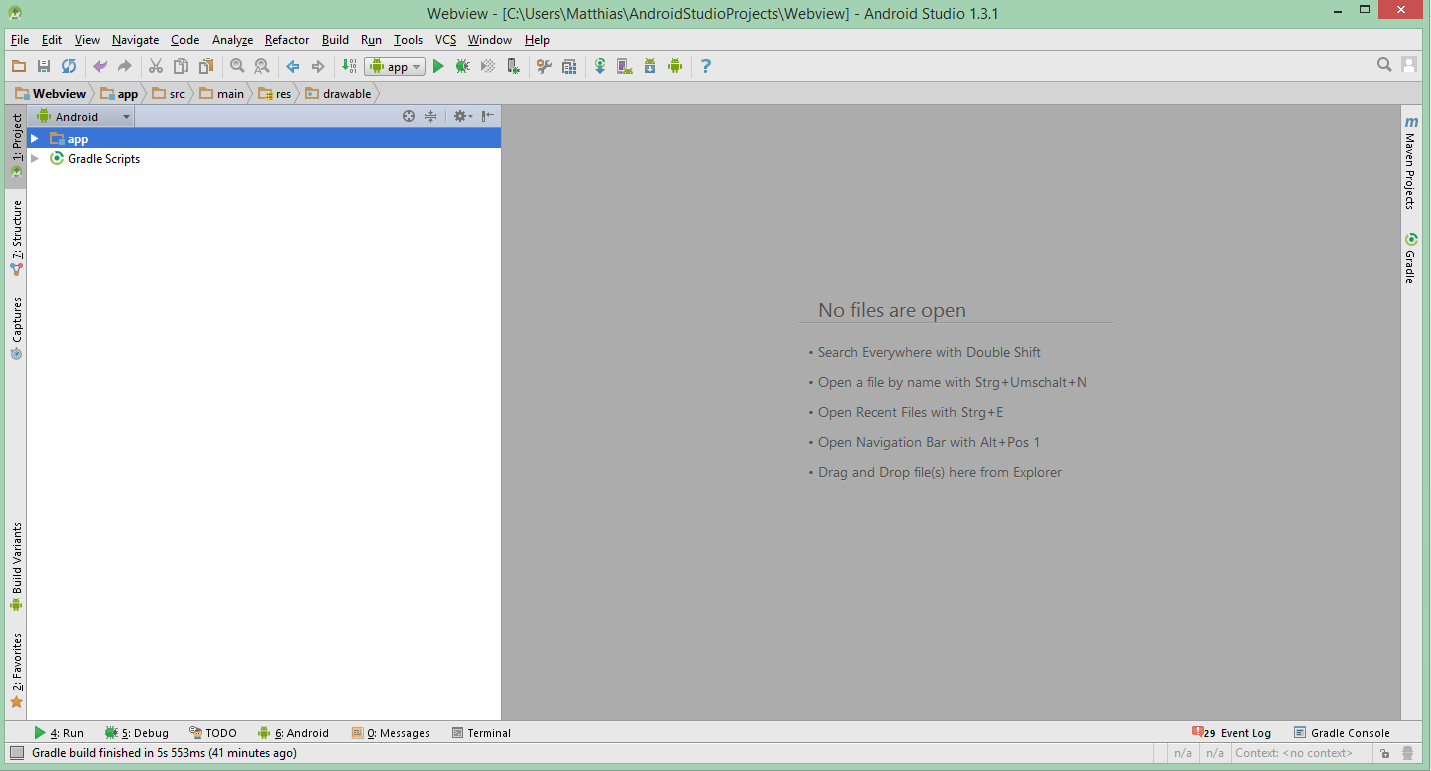
1. Import Project

* Open Android Studio
* You‘ll see this window



* Click *Open an existing Android Studio project*
* Follow the instructions (select <extracted zip>/Android Studio project/Project/Webview)

1. Customize your app



The Android Studio window should look like this. If the navigaton on the left is missing, press ALT-1 (Win) / CMD-1 (Mac). Make sure, the dropdown in the left upper corner shows „Android“.

Change app name, ad unitID and translations

Doubleclick on *app->res->values->strings.xml* in the navigator

The file should look like this:

<**resources**>  
  
  
 <**string name="app\_name"**>Webview</**string**>  
 <**string name="banner\_ad\_unit\_id"**>xxx</**string**>  
  
 <**string name="navigation\_drawer\_open"**>Open navigation drawer</**string**>  
 <**string name="navigation\_drawer\_close"**>Close navigation drawer</**string**>  
 <**string name="refresh"**>refresh</**string**>  
 <**string name="share\_using"**>Share using</**string**>  
 <**string name="share"**>Share</**string**>  
  
</**resources**>

**Change App name:**

Replace „Webview“ (value for „app\_name“) with the name you want.

**Change Ad unit ID**

Insert your AdMob unitID as value for „banner\_ad\_unit\_id“

**Change Translations**

You can change every value in the file as you want. (Do not change something else if you don’t know what you are doing!)

Change general options

Doubleclick on *app->java->net.matthi.webview->Pages*  in the navigator

Let’s have a look at this section:

***//White Design*public boolean useWhiteFont = false;  
  
*//URLs included in your app*public String homeDomains = "";  
  
*//Error message*public String error = "<center> The page couldn't be loaded </center>";  
  
*//Share*public boolean share = true;  
public boolean shareAsAction = false;  
public String shareText = "Look at this! http://www.example.org/";  
  
*//Misc*public String localEncoding = "UTF-8";  
  
*//Ads*public boolean useAds = false;  
*// Enter your unit id at /res/values/strings.xml -> banner\_ad\_unit\_***

Description

Replace **public boolean useWhiteFont** = **false**; with **public boolean useWhiteFont** = **true**; if you want white font and icons in the actionbar.

You can add more domains to your app. Links to them won’t be opened in a new window.

E.g. : **public** String **homeDomains** = **"www.example.com"**;

(The domain must be [www.domain.com](http://www.domain.com), don‘t use http:// here. Paths ([www.test.com/123](http://www.test.com/123)) are not supported)

The message **"<center> The page couldn't be loaded </center>"** will be displayed, if the page couldn’t be loaded. Feel free to change it as you like.

**public boolean share = true;  
public boolean shareAsAction = false;  
public String shareText = "Look at this!** [**http://www.example.org/**](http://www.example.org/)**";**

Here you can enable the share-button. (share = true/false). If you set **shareAsAction = true,** the share-button is visible as an icon in the actionbar.

Replace **Look at this!** [**http://www.example.org/**](http://www.example.org/)with the message you want to share.

***//Misc*public String localEncoding = "UTF-8";**

The encoding for offline-pages. UTF-8 should be good for the most ones.

***//Ads*public boolean useAds = false;  
*// Enter your unit id at /res/values/strings.xml -> banner\_ad\_unit\_id***

Change to **public boolean useAds = true;** if you want to use ads.

Add your webpages to the navigation

Now we work on this section:



It’s constructed as following:

ID (menu,“Title“,“tabs“,“URL“,“OFFLINE\_URL“,Color,Icon,RefreshButton),

**ID**: This name is only used internally, you can call it as you like.

**menu**: true, if the page should be linked in the 3-dot menu, false, if you want it in the main navigation.

**Title**: The title of the page in quotation marks.

**Tabs:** null, if you don’t want tabs on this side. If you want them, have a look at „tabs“ at the end of this chapter.

**URL**: The url of the page in quotation marks.  
The URL must contain http://, „www“ and the complete path.  
E.g. „http://www.example.org/yourPage.html“

**OFFLINE\_URL**: The local file, that is displayed, when the page couldnt be loaded in quotation marks.  
if you don’t need it, write null.

|  |  |
| --- | --- |
| URL = <http://www.example.org>  OFFLINE\_URL = null | Content loads from the internet,  error-message if not available |
| URL = <http://www.example.org>  OFFLINE\_URL = example.html | Content loads from the internet,  uses local site if not available |
| URL = null  OFFLINE\_URL = example.html | Uses local site only |
| URL = null  OFFLINE\_URL = null | This isn’t possible – your app will probably crash |

**Copy the offline html-files to the app/assets directory !**

**Color:** The color of the Actionbar (and statusbar on kitkat/lollipop).  
You have the selection between 10 colors:

***Color.RED***   
***Color.GREEN***   
***Color.BLUE***   
***Color.YELLOW***   
***Color.CYAN\_DARK***   
***Color.CYAN\_LIGHT***   
***Color.GRAY\_LIGHT***   
***Color.GRAY\_DARK***  
***Color.PURPLE\_DARK***  
***Color.PURPLE\_LIGHT***

**Icon:** The icon, displayed in the navigation. Write null, if „menu“ is true.

You have the selection between 17 icons:

***Icon.HOME***   
***Icon.DOWNLOAD***   
***Icon.STAR***   
***Icon.QUESTION***   
***Icon.INTERNET***   
***Icon.LIST***   
***Icon.PERSON***   
***Icon.MAP***   
***Icon.SETTINGS***   
***Icon.SHOP***   
***Icon.TEXT***   
***Icon.MOVIE***

***Icon.CALENDAR***

***Icon.GALLERY***   
***Icon.EYE***   
***Icon.WORK***   
***Icon.PEOPLE***

**RefreshButton:** Write true, if you want a refresh-button (useful for e.g. newsfeeds).Otherwise write false.

**Tabs:**

If you want tabs on a page, write all tab-titles, seperated by „||“ in „tabs“.

E.g.:

“ Tab1||Tab2||Tab3||Tab3“

You also need the same count of URLs (and offline\_urls, if you use them on this page)

E.g.:

“http://www.example.org/site1|| http://www.example.org/site2|| http://www.example.org/site3“

**The last row must be closed with a semicolon !**

A complete configuration could look as the following:

HOME (false,“Home“,null,“http://www.example.org/home.html“,null,Color.RED,Icon.HOME,false),  
NEWS (false,“News“,“TechNews||Politics“,“http://www.technews.com||http://www.news.com“,null,Color.CYAN\_DARK,Icon.LIST,true),  
OFFLINE (false,“Offline Page“,null,“offline.html“,Color.CYAN\_DARK,Icon.DOWNLOAD,false),  
IMPRINT (true,“Imprint“,null,“http://www.example.org/imprint.html“,“imprint.html“,Color.GRAY\_LIGHT,Icon.TEXT,false);

**The existing configuration is meant as an example and should be overwritten!**

1. Change app-icon and navigation header image (or feel free to use mine)

* Rightclick to app (in the navigation)->new->Image Asset
* Resource name must be „ic\_launcher“
* The dialog should be self-explaining
* To change the header image oft he app-navigation open the „res“ folder in the navigation
* Rightclick to „drawable“ -> Show in Explorer (Or Finder)
* Open the „drawable“ directory
* Replace the „material.png“ file with your own file with the same name

1. Change the package-id

* Open Gradle Scripts/build.gradle (the 2nd one !) in the navigator
* Change the „applicationId“ (e.g. org.example.ourapp)
* Click „Sync Now“ at the top oft he Editor (this could take some time)

1. Test your app

* Click *Run->Run ‘app‘* in the menu
* Select a connected device or emulator
* Click OK
* The app should start on the selected device/emulator

Setup an Android emulator:

<http://developer.android.com/tools/devices/managing-avds.html>

Install driver for testing on a device

<http://developer.android.com/tools/extras/oem-usb.html#InstallingDriver>

1. Build your app

* Click *Build->Generate Signed APK* in the menu
* Create a new keystore or use an existing

<https://developer.android.com/tools/publishing/app-signing.html>

* Click Next
* Select the destination for your APK
* Click Finish

**Congratulations! Your app is ready for distribution.**

**Do you have any problems ?**

Feel free to write me at [**matthias.rupp@mnet-mail.de**](mailto:matthias.rupp@mnet-mail.de)

**Did you publish your app successfully?**

I would be happy, if you let me know ☺