

# Configuring (/support/documentation/index.php/universal/4-configuring)

Before we can change the actual content of the application. We got to set some basic information, like API keys and your apps name.

## Change your apps name and about text

Open the file strings.xml in res --> values. Enter the required value in the tag, so for example, your apps name would be:

```
1 | <string name="app_name">The Awesome Band News</string>
```

You can find the *about\_text* string below the API key strings. Change your apps about text, by using html code between the `<![CDATA[` and `]]>` tags.

## Setting up notifications

You can enter one RSS Feed (this may also be based on a youtube channel or twitter timeline) that users will be notified about when it updates, with a notification.

Find the line below, and type your RSS Feed url (if you want notifications) between `<string name="rss_push_url">` and `</string>`

```
1 | <string name="rss_push_url"></string>
```

You can also change the frequency the RSS feed is checked for updates by editing the frequency in the line below. As you can see, the default value is 5.

## In-App purchases

Universal offers built-in capabilities to remove ads and unlock content after an in-app purchase. You need to perform a few steps to enable in app purchases:

1. Create a new app on Google Play (or open an existing app you would like to update).
2. Go to 'Services and API's' and copy your app's license code (RSA-code).
3. Find the following line:

```
1 | <string name="google_play_license"></string>
```

4. Enter the license code between `<string name="google_play_license">` and `</string>`.

Now you need to finish the documentation and compile your apk, and continue with the following steps after you've uploaded your APK to Google play.

1. Go to 'In-App products'
2. Add a new product (managed/standalone product) and come up with a unique product ID (usually something like com.yourcompany.yourapp.product)
3. Follow the steps to configure the product
4. Find the following line:

```
1 | <string name="product_id"></string>
```

5. Enter your product ID between `<string name="product_id">` and `</string>`

After submitting your the app with the updated product id to Google Play, you should be able to make in-app purchases within a few hours.

**Important** In-App purchases require an Extended CodeCanyon license.

## Entering your Twitter api keys

If you want to show Tweets inside your app, using our integrated Twitter content provider, you will need Twitter API keys. *If you have no plans of integrating Twitter or if you are only using embedded tweets (e.g. in your WordPress posts or WebView), you can continue to the next heading.*

1. Go to <https://dev.twitter.com/apps> (<https://dev.twitter.com/apps>) and login
2. Click 'Create a new app' (<https://apps.twitter.com/app/new>)
3. Fill the form to create application
4. After completion, you should see the application settings screen. If this is not the case, manually browse to this screen.
5. Go to the 'Keys and tokens' tab
6. Locate the Consumer Key and Secret key
7. Find the lines below in Strings.xml, and replace the placeholder values with the retrieved keys.

```
1 <string name="twitter_api_consumer_key">YOURCONSUMERKEYHERE</string>
2 <string name="twitter_api_consumer_secret_key">YOURSECRETKEYHERE</string>
```

## Entering your Server and Android api keys.

In order to use Youtube and/or Maps in your application, you need to configure your Server and Android API keys in the Google Developer console.

1. For instructions on creating your Server and Android API keys click here ([/support/documentation/raw/google\\_api.html](https://support.google.com/android/answer/9000000)).
2. Using the information above, you should be able to have retrieved an Android Key (based on your keystore) and a Server Key.
3. Find the line below, and enter the the api key like shown below:

```
1 <string name="google_server_key">YOURSERVERKEYEHERE</string>
2 <string name="google_android_key">YOURANDROIDKEYEHERE</string>
```

## Facebook Access Token

If you want to show Facebook posts in your app, using our integrated Facebook content provider, you will need a Facebook access token. *If you have no plans of integrating Facebook or if you are only using embedded posts (e.g. in your WordPress posts or WebView), you can continue to the next heading.*

1. Go to [developers.facebook.com](https://developers.facebook.com) and click on Log In in the top right. Log in using your personal Facebook credentials.
2. If this is your first time signing in to the Facebook Developer portal then click on Register Now. Registering is a quick and easy process which will take less than a couple of minutes. If you're already registered then you can skip ahead to step 9.
3. Accept the Facebook terms and click Continue.
4. Enter your phone number to confirm your account.
5. Facebook will send you an automated text message containing a confirmation code. Enter it in the box and click Confirm.
6. Choose to share your phone number with Only Me (unless you wish to share it with publicly or with friends).
7. If there is a step asking you to provide some information about yourself. You can skip this next step by clicking Skip.
8. Click Done.

9. Now click on Create New App.
10. Click on advanced setup.
11. Enter your App Name. This can be anything you like. Click Continue.
12. If prompted with a security check, Fill in the Security Check and click Continue.
13. Your App should now be set up. Copy your App ID and App Secret. Please note that you need to click 'Show' next to the App Secret before copying.

We can use this ID en Secret, to generate an Access token. If you wish you can use this PHP script (<http://stackoverflow.com/a/20514503/1683141>) or this Python script (<http://blog.lwolf.org/blog/2014/06/16/obtaining-never-expiring-access-token-to-post-on-facebook-page/>) to obtain a valid access token over the Facebook API. You can also use the tool available at this website (<https://smashballoon.com/custom-facebook-feed/access-token/>)(at the bottom) or follow the tutorial here (<https://www.rocketmarketinginc.com/blog/get-never-expiring-facebook-page-access-token/>).

Now that we have our access token, find the line below and enter the access token by replacing the placeholder value:

```
1 | <string name="facebook_access_token">YOURFACEBOOKTOKENHERE</string>
```

?

## Instagram Client ID

If you want to show Instagram posts in your app, using our integrated Instagram content provider, you will need an Instagram client ID. *If you have no plans of integrating Instagram or if you are only using embedded posts (e.g. in your WordPress posts or WebView), you can continue to the next heading.*

1. Navigate to the Instagram developer page (/support/documentation/If you want to show Facebook posts in your app, using our integrated Facebook content provider, you will need Twitter API keys and access tokens. If you have no plans of integrating Facebook or if you are only using embedded posts (e.g. in your WordPress posts or WebView), you can continue to the next heading.) and login if you are not logged in already.
2. Click Register your Application button
3. If this is your first application, register as developer by filling the information asked, accept the API terms of use and brand guidelines and click Sign up buttons. If you are not seeing this page, you can skip this step.
4. Click Register your Application button again to proceed.
5. Click Register a New Client button
6. In the Basic section, fill the fields as described below:
  - Application Name:* Name of your application. You can enter whatever name you like, but it's recommended to enter your website or company name in this field.
  - Description:* Description of your website/company you entered in Application Name field.
  - Website URL:* Open your website homepage in browser, click the address bar so that website url gets highlighted, copy this url and paste it in this option. Please make sure that this url includes http:// or https:// and there is no trailing slash (for example: <http://mywebsite.com> or <http://www.mywebsite.com>) (<http://www.mywebsite.com>)).
  - Redirect URI(s):* Enter exactly the same website url as in Website URL option above
  - Contact email:* Specify your email
7. Fill Captcha (Do not click "Register" button)
8. Navigate to Security section and configure the options as described below
  - Disable Implicit OAuth:* Uncheck this option
  - Enforce signed requests:* Keep this option unchecked
9. Click Register button
10. Copy the Client ID of the application you just created

Now that we have our Client ID, find the line below and replace the placeholder value:

```
1 | <string name="instagram_client_id">YOURINSTAGRAMIDHERE</string>
```

?

## SoundCloud Client ID

If you would like to play SoundCloud tracks in your app, using our integrated SoundCloud player, you will need an SoundCloud Client ID. *If you have no plans of integrating SoundCloud you can continue to the next heading.*

1. Navigate to the SoundCloud developers page (<https://developers.soundcloud.com>) and login if you are not logged in already.
2. Select *Your Apps* and click on *Register new App*
3. Enter the required details and continue
4. You should now see the SoundCloud Client ID

Now that we have our Client ID, find the line below and replace the placeholder value:

```
1 | <string name="soundcloud_id">YOURSOUNDCLOUDIDHERE</string>
```

?

## Entering your Admob details / turning on ads

We have build-in support for Admob banners on all screens and devices. In order to activate Admob ads inside your app, you need to have an Ad UNIT ID. *If you don't want ads inside your application, you can continue to the next heading.*

### Warning for Youtube users

Google's policy on using Admob & Youtube together is very strict. If you do not own the shown videos, make sure to ask permission to the respective owners before showing their videos in your app (combined with Admob).

1. Go to [admob.com](https://admob.com) (<https://admob.com>)
2. Login and create a new android app (if asked, choose for "banner").
3. Find and write down your Ad Unit ID (<https://support.google.com/admob/answer/3016009?hl=en>).
4. Open the file strings.xml in res --> values
5. Enter the value for ad\_id in the line below between `<string name="ad_id">` and `</string>`.

```
1 | <string name="ad_id"></string>
```

?

## Installing the JSON Api plugin

If you where planning on using a Wordpress Blog in your app, you need to install the JSON API plugin.

1. First you will need to download the plugin, preferably the tested version, which is 1.1.1 at <http://wordpress.org/plugins/json-api/developers/> (<http://wordpress.org/plugins/json-api/developers/>).
2. Upload the `json-api` folder to the `/wp-content/plugins/` directory or install directly through the plugin installer.
3. Activate the plugin through the 'Plugins' menu in WordPress or by using the link provided by the plugin installer.
4. Enable user-friendly URLs from Settings > Permalinks, use the "Day and name" format.
5. Determine your API url and use it for the Wordpress items, it will most likely be something like: "yourblogurl.com/api/".

## Changing the colors

You can change your apps colors to create a custom look. The primary color is the color of the Toolbar, some buttons and headers. The primary dark color is used for e.g. the status bar.

1. Open res/colors.xml
2. Change the myColorPrimary color code to a color code of your choice (toolbar).

3. Change the myColorPrimaryDark color code to a color code of your choice (status bar).
4. Change the myAccentColor color code to a color code of your choice (e.g. spinners and buttons).

## App icon

Now we need to put a png file with your icon named 'ic\_launcher.png' in the following folders:

- res/drawable-mdpi - with a size of **48px x 48px**
- res/drawable-hdpi - with a size of **72px x 72px**
- res/drawable-xhdpi - with a size of **96px x 96px**
- res/drawable-xxhdpi - with a size of **144px x 144px**

You can use a tool, like this site: <http://romannurik.github.io/AndroidAssetStudio/icons-launcher.html> (<http://romannurik.github.io/AndroidAssetStudio/icons-launcher.html>) to help you resize your images or even create an icon.

## Drawer Header

Now we need to put a png file that will server as your drawer header image, named 'drawer\_header.png' in the following folders:

- res/drawable-mdpi - with a width of **200px**
- res/drawable-hdpi - with a width of **300px**
- res/drawable-xhdpi - with a width of **400px**
- res/drawable-xxhdpi - with a width of **600px**
- Put the original unscaled image in res/drawable as fallback

You can use a tool, like this site: <http://romannurik.github.io/AndroidAssetStudio/icons-generic.html> (<http://romannurik.github.io/AndroidAssetStudio/icons-generic.html>) to help you resize your images. If you do not want to use a drawer header image, and use the classic (below the ActionBar) drawer, change 'true' to 'false' in USE\_NEW\_DRAWER in Config.java

You can now continue. We'll configure the content (config.java) in the next step.

---

*Video for Eclipse: <https://www.youtube.com/watch?v=bv0CmaR2DK8>*