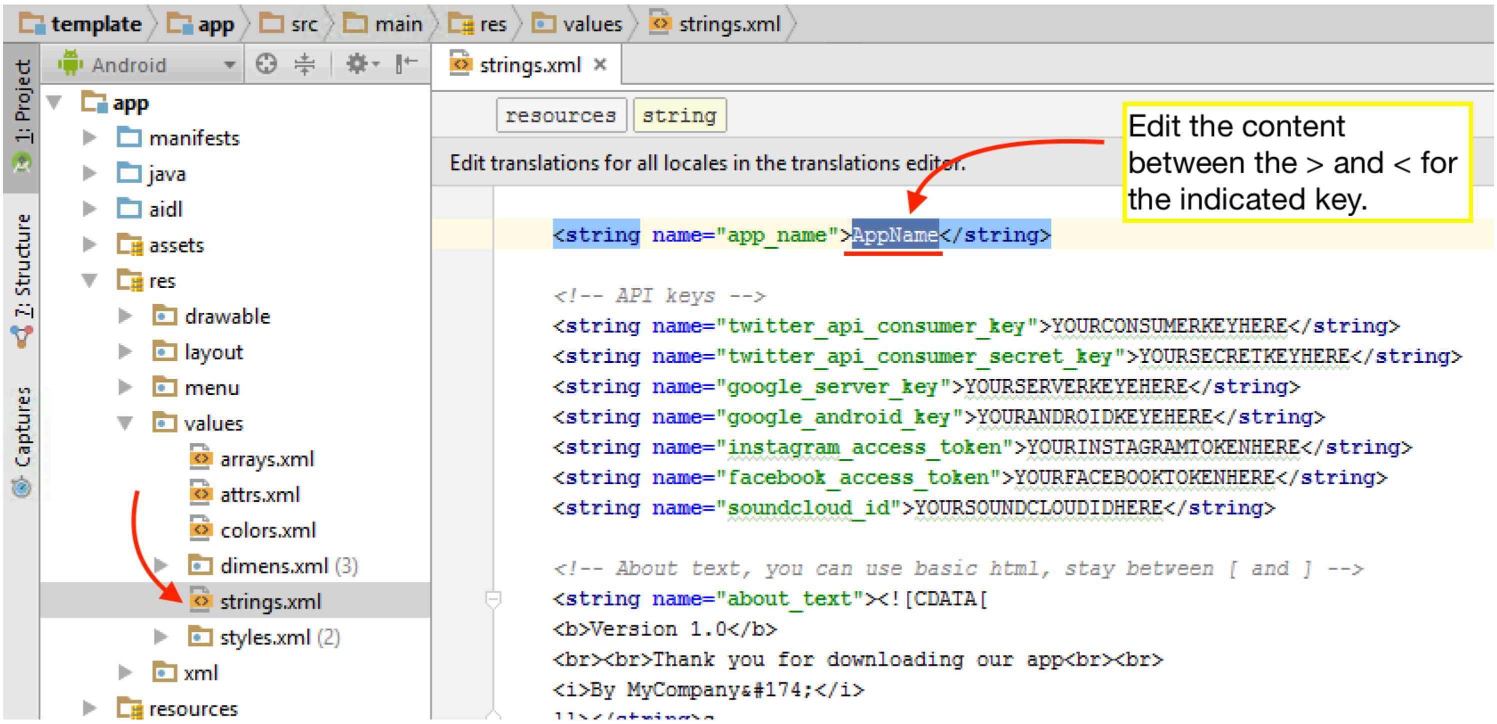
Configuring (/support/documentation/index.php/universal/4-configuring)

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

Below is illustrated how to change the value for the key "app_name". Simply enter your preferred value for this key between <string ..> and </string>.



Edit translations for all locales in the translations editor.

Edit the content between the > and < for the indicated key.

```
<string name="app_name">AppName</string>

<!-- API keys -->
<string name="twitter_api_consumer_key">YOURCONSUMERKEYHERE</string>
<string name="twitter_api_consumer_secret_key">YOURSECRETKEYHERE</string>
<string name="google_server_key">YOURSERVERKEYEHERE</string>
<string name="google_android_key">YOURANDROIDKEYEHERE</string>
<string name="instagram_access_token">YOURINSTAGRAMTOKENHERE</string>
<string name="facebook_access_token">YOURFACEBOOKTOKENHERE</string>
<string name="soundcloud_id">YOURSOUNDCLOUDIDHERE</string>

<!-- About text, you can use basic html, stay between [ and ] -->
<string name="about_text"><![CDATA[
<b>Version 1.0</b>
<br><br>Thank you for downloading our app<br><br>
<i>By MyCompany&#174;</i>
```

Change your apps name and about text

Open the file strings.xml in res --> values. Enter the preferred value in the tag for the key *app_name. E.g:*

```
<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.

Simply enter your RSS Feed url (if you want notifications) for the key *rss_push_url*

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.

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

If you do not want to use RSS based notifications in your app, simply leave the value for the key empty as shown above.

In-App purchases

Universal offers built-in capabilities to remove ads and unlock content after an in-app purchase. If you do not plan on using in-app purchases, you can remove the BILLING permission from Manifest.xml. If you do wish to use this functionality continue with the following steps:

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. Enter the license code between for the key *google_play_license*

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. Enter your product ID for the key *product_id*

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.

Setting up API keys

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 integratingTwitter 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.

```
<string name="twitter_api_consumer_key">YOURCONSUMERKEYHERE</string>
<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).
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:

```
<string name="google_server_key">YOURSERVERKEYEHERE</string>
<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 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 an 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:

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

Instagram Access Token

If you want to show Instagram posts in your app, using our integrated Instagram content provider, you will need an Instagram Access Token. *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. If you haven't already, create an Instagram Account.
2. Check this post on how to create an Access Token (/support/documentation/../community/index.php?action=artikel&cat=2&id=67&artlang=en).

Now that we have our Access Token, find the line below and replace the placeholder value:

```
<string name="instagram_access_token">YOURINSTAGRAMIDHERE</string>
```

SoundCloud Client ID

If you would like 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:

```
<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. Automatic detection frameworks of the Google Play store may reject your app if Ads are found in combination with Youtube. As a precaution you can set *ADMOB_YOUTUBE* in *Config.java* to *false*. You can always appeal to a Google rejection, since Ads are never shown next to a playing video.

1. Go to 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 for the key *ad_id*

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

To disable Admob simply do not entery any value for the key ad_id and leave it completely empty as shown above.

Preparing your Wordpress Blog for Universal

If you would like to show a Wordpress Blog in your app, you need to either install the JSON API plugin or use the Wordpress REST API.

Rest API

The Rest API is offered for Wordpress.com blogs by default, but it can also be used by self-hosted Wordpress blog that have the JetPack plugin (https://wordpress.org/plugins/jetpack/) installed. Make sure that you site is accessible over the REST API by visiting https://public-api.wordpress.com/rest/v1/sites/YOURBLOG

Where YOURBLOG can be your Blog's ID (e.g. 3584907) or you blogs domain without http (e.g. www.artifacting.com (http://www.artifacting.com) or en.blog.wordpress.com).

JSON API

If you have a self-hosted blog and you do not want to use Jetpack, you can use the JSON-API plugin as well.

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/".

Visual Appearance

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