

Configuring

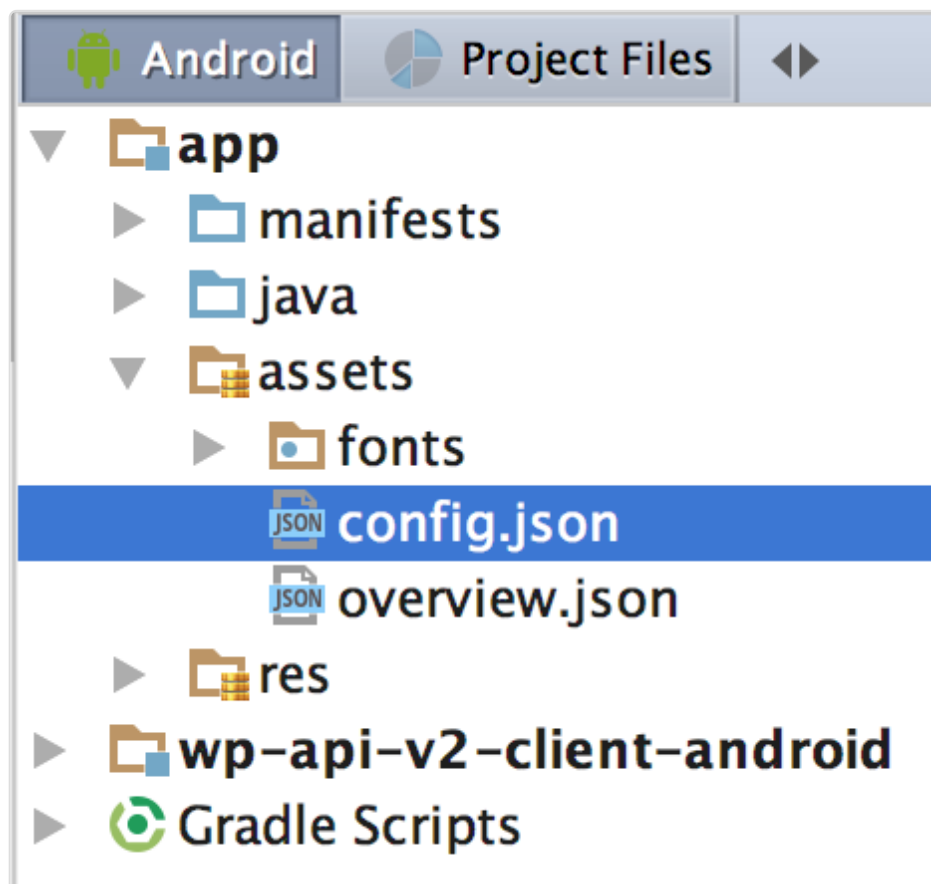
We can now configure our app to use the configuration we created in the previous step. In addition, we will set some basic information, like API keys and your apps name.

Set to use your Configuration

You can put the configuration file locally in your app (recommended), or host it on your own server online.

Local Configuration (recommended)

If you'd like to use a local configuration file (or would like to add local overview configurations) you should add the Configuration JSON files to your app's assets. First, find the placeholder config.json file in your template assets, as shown below, and remove it. Now add your own config.json file (and possibly your overview json files) to the assets folder of your project, the result should again look as shown below:



Online Configuration file

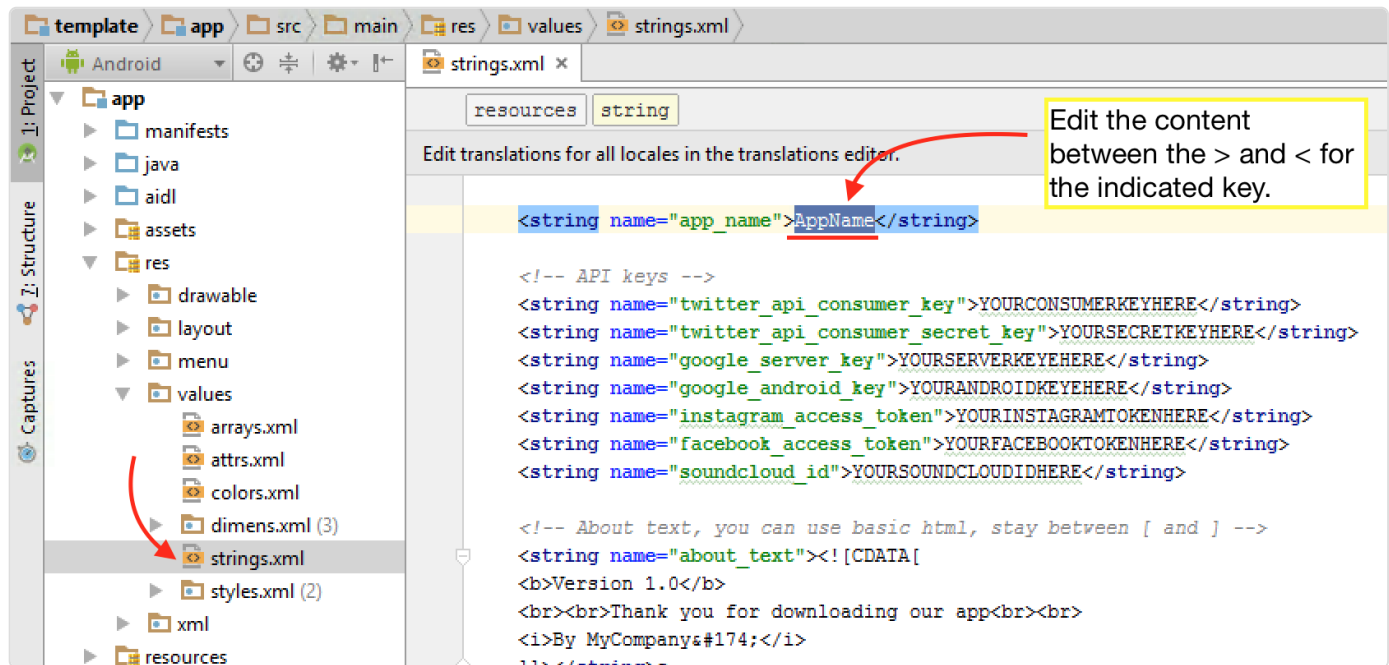
You can configure Universal to load a configuration file from the web, which has the advantage that you can update your app's content dynamically, without users having to update their app. Upload your JSON file to your web server if you haven't already, and make sure that it is reachable by url. Next, open Config.java in `java/com.sherdle.universal/` and set `CONFIG_URL` to point to your main configuration.

For example, if your main configuration JSON can be found at <http://blog.com/configuration.json>:

```
public static String CONFIG_URL = "http://blog.com/configuration.json";
```

Configuring values in the Strings file

In the next steps will set some predefined values in the resources file (Strings.xml found in res/values/). 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>.



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 editing the html code between the <![CDATA[and]]> tags.

Setting up notifications

If you'd like to send notifications to your users, you can enable OneSignal support by following the steps below. If you do not want OneSignal support, you can simply skip these steps.

1. Go to OneSignal.com (<http://onesignal.com>) and sign in (or sign up if you do not have a account). And go to my apps (<https://onesignal.com/apps>).
2. Click Add a new app and follow the onscreen instructions.
3. In the step Configure Platform you will locate a Firebase Sender ID using these instructions (<https://documentation.onesignal.com/docs/generate-a-google-server-api-key>). Note this **Sender ID**, as we will be needing this later.
4. In the step Install SDK you will be provided with your **App ID**.
5. You can skip all the steps that require adding libraries or editing code. As the library has already been integrated into Universal.
6. Finally, enter your OneSignal App ID and Sender ID in strings.xml as shown below:

```
<string name="onesignal_app_id">YOURAPPIDHERE</string>
<string name="onesignal_google_project_number">YOURSENDERIDHERE</string>
```

Tip If you'd like to automate notifications (i.e. when your RSS feed updates) you can use a service like Zapier (<https://zapier.com/zapbook/onesignal/>) to do this for you. For WordPress, you can use the

WordPress OneSignal plugin, and enable the 'Send notifications additionally to iOS & Android platforms' option in the plugin settings.

If you do not want to use notifications in your app, simply leave the value empty in the Strings file as per default.

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 a Privacy Policy

If you provide a privacy policy url, the template will require users to agree to your policy before they can use your app. Please provide the url to your privacy policy, for the tag for the key *privacy_policy_url*, for instance:

```
<string name="privacy_policy_url">https://yoursite.com/privacy</string>
```

Your privacy policy should describe how you use cookies, and which 3rd party services your app uses. If you do not have a privacy policy, you can use a service like this (<https://app-privacy-policy-generator.firebaseapp.com/>) to create one.

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 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.

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

Google API keys.

In order to use \ 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 (<https://sherdle.com/help/google-api-keys/>).
2. Using the information above, you should be able to have retrieved an Android Key (based on your keystore) and a Server (General) 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>
```

Vimeo Access Token

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

1. First, you need an Access Token, generate one as is described here. (<http://hardcode.party/how-to/get-vimeo-access-token>)
2. Enter your Access token in Strings.xml as shown below.

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

```
<string name="vimeo_access_token">YOURVIMEOTOKENHERE</string>
```

Pinterest Access Token

If you want to show Pinterest posts in your app, using our integrated Pinterest content provider, you will need an Pinterest Access Token. *If you have no plans of integrating Pinterest you can continue to the next heading.*

1. If you haven't already, create a Pinterest Account.
2. Visit this page to learn how to create an Access Token (<https://developers.pinterest.com/docs/api/overview/>) (at least 'Read Public' is required).

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

```
<string name="pinterest_access_token">YOURPINTERESTTOKEN</string>
```

Flickr API Key

If you want to show Flickr images in your app, using our integrated Flickr content provider, you will need an Flickr API key. *If you have no plans of integrating Flickr you can continue to the next heading.*

1. If you haven't already, create a Flickr Account.
2. Sign into your Flickr account at flickr.com.
3. Visit the API Page (https://www.flickr.com/services/api/misc.api_keys.html) and click the 'apply for your key now online' link.
4. Select 'Apply for a non-commercial key'.
5. Enter some information about your apps name and why you need Flickr in your app. Agree with the conditions and click the submit button.
6. You should now see your Flickr Key on the screen. Copy this key.

Now that we have our Flickr API Key, find the line below and replace the placeholder value:

```
<string name="flickr_key">YOURFLICKRKEY</string>
```

Tumblr API Key

If you want to show Tumblr images in your app, using our integrated Tumblr content provider, you will need an Tumblr API key. *If you have no plans of integrating Flickr you can continue to the next heading.*

1. If you haven't already, create a Tumblr
2. Sign into your Tumblr account at tumblr.com.
3. Visit the API Page (<https://www.tumblr.com/oauth/apps>) and click the Register App button.
4. Follow the onscreen instructions to create the app and finish the registration.
5. For your newly created app, you should now see your OAuth Consumer Key on the screen. Copy this key.

Now that we have our Tumblr API Key, find the line below and replace the placeholder value:

```
<string name="tumblr_key">YOURTUMBLRKEY</string>
```

SoundCloud Client ID

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

SoundCloud currently has closed its API registration page. For now, please search the web for any publicly posted SoundCloud API keys or reach out to us for further assistance.

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>
```

WooCommerce Host and API

If you would like to integrate WooCommerce in your app, using our WooCommerce content provider, you will need to enter your WooCommerce host and API keys. *If you have no plans of integrating WooCommerce you can continue to the next heading.*

1. Ensure that you are using WooCommerce 3.5 or later, which supports the WooCommerce V3 API.
2. Follow the WooCommerce documentation (<https://docs.woocommerce.com/document/woocommerce-rest-api/>) on enabling your API and generating keys. You only need Read access, we recommend to use an Admin user.
3. In Strings.xml enter your Consumer Key for *woocommerce_consumer_key* and your Consumer Secret Key for *woocommerce_consumer_secret*.
4. Next, enter the url to your WordPress Installation with WooCommerce for *woocommerce_url* (without ending slash).

```
<string name="woocommerce_consumer_key">ck_XXXXXXX</string>
<string name="woocommerce_consumer_secret">cs_XXXXXXX</string>
<string name="woocommerce_url">http://mystore.com</string>
```

Please consult this article (<https://sherdle.com/help/woocommerce-doesnt-work/>) if you are having issue configuring WooCommerce, or if you want to learn how to configure additional details for your WooCommerce store, such as your currency.

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 advertisement (if asked, choose for "banner" or "interstitial" depending on what you'd like to use).
3. Find and write down your Ad Unit ID (<https://support.google.com/admob/answer/3016009?hl=en>) (not your App ID) of the advertisement.
4. Open the file `strings.xml` in `res -> values`
5. Enter the the ad unit ID of your banner advertisement for the key `ad_banner_id` or the ad unit ID of your interstitial advertisement for `ad_interstitial_id`
6. Repeat the steps above for banner or interstitial advertisements (depending on what you just configured) if you'd like.

```
<string name="admob_banner_id"></string>
<string name="admob_interstitial_id"></string>
```

To disable Admob banner ads and/or interstitial ads simply do not enter any value for the respective keys and leave it completely empty as shown above.

Firebase

In case you would like to use Firebase (only crash reporting, and basic analytics). Please create a new project in the Firebase Console (<https://console.firebase.google.com/u/0/>) and follow the on-screen instructions to add the `google-services.json` file to your project (No other steps or code changes as instructed are required).

Next, in your `Config.java` file, ensure that `FIREBASE_ANALYTICS` is set to 'true'.

Visual Appearance

Changing the colors

You can change your apps colors to create a custom look. If you are using the light theme (option `LIGHT_TOOLBAR_THEME` in `Config.java`), these colors will be used to color some UI elements. Otherwise, the primary color is also the color of the Toolbar and 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 serve 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 a 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.

Further appearance options

In `Config.java` you will find more options to determine how your app looks. For example, you can hide the sliding menu drawer or hide the top toolbar. Every option in this file is commented to help you understand the impact of the option.