Content (/support/documentation/index.php/universal/5-content)

We can now continue to change to content of our application. In this step, we determine what 'items' are in our app (things users can select in the navigation drawer).

Open the file Config.java in java/com.sherdle.universal/

You can change what is in the menu by adding Sections and Items. The order that you add sections and items to your list, is the same order as it will appear in your app.

Some placeholder content is given as an example, you might want to remove this (between '*//DONT MODIFY ABOVE THIS LINE*" and "*//IT's Suggested to not ....*").

Adding a Section

```
i.add(new NavItem("Section", NavItem.SECTION));
```

You can use this code to add a section. Where 'Section' is the sections name.

Adding Items

You can add unlimited items to your app. A line to add an item to your configuration file is formatted like this:

```
i.add(new NavItem(NAME, DRAWABLE, NavItem.ITEM, PROVIDER, new String[]{PARAMS}));
```

Where:

> *NAME:* The name of your item as it appears in the menu. You can type this name directly in the configuration between quotes or reference to a String using R.string.stringname.
>
> *DRAWABLE:* This is the icon of your item. You can put a 0 if you do not want to use an icon, or put reference to a drawable using R.drawable.drawablename. It's very important that you create an appropriately sized drawable for all screen densities (compound drawable) if you are planning to use icons. You can look at the dimensions we used with ic_details.png
>
> *PROVIDER:* This is the type of item you would like to load, details on what to enter here will be provided below for each type of item.
>
> *PARAMS:* This can be various parameters to help determine the provider what content to load. Parameters are always Strings (between quotes) and separated by comma's for example: *"param 1","param 2","param 3", "etc".* Details on which parameters to provide are given below for each type of item.

Wordpress item

If you would like to show the latest Wordpress posts or a category in your app, you can add a Wordpress item to the menu. You can use the following values for your configuration:

> *PROVIDER:* WordpressFragment.class
>
> *PARAMS:* As first parameter enter the url to your Wordpress blog (starting with http:// not ending with a slash) to use the JSON API (https://wordpress.org/plugins/json-rest-api/) or enter the Blog ID or name of your blog (as established in the previous step of the documentation) to use the Jetpack (https://jetpack.com/support/json-api/)/Wordpress REST API (https://developer.wordpress.com/docs/api/); The second parameter is a category slug (which you can leave empty for all posts combined). The last and third parameter is optional, and this is a Disqus formatted string.

An example for the Wordpress.com Blog (or Jetpack blog) *https://en.blog.wordpress.com*, to show posts from the category 'events':

```
i.add(new NavItem("Wordpress Category", 0, NavItem.ITEM, WordpressFragment.class, new
String[]{"en.blog.wordpress.com", "events"}));
```

If we would like to show all posts from the category 'food' from *https://someblog.com* that has the JSON API plugin installed to the Wordpress installation at main domain:

```
i.add(new NavItem("Wordpress Category", 0, NavItem.ITEM, WordpressFragment.class, new
String[]{"http://someblog.com", "food"}));
```

If we would like to show all recent posts from *https://myblog.com* that has the JSON API plugin installed to the Wordpress installation at the subdirectory blog, and we would like to use Disqus for comments:

```
i.add(new NavItem("Wordpress Category", 0, NavItem.ITEM, WordpressFragment.class, new
String[]{"http://myblog.com/blog", "", "http://myblog.disqus.com/;myblog;%d
http://www.myblog.com/?p=%d"}));
```

Notice how the disqus formatted string is formatted like: *disqus url;shortname;identifier pattern*. Where the post ID in the identifier pattern is replaced with *%d*. This information can usually be found in the source code of a webpage that embeds disqus comments.

Adding a Youtube item

If you would like to show a Youtube playlist in your app, you can add a Youtube item the the menu. You can use the following values for your configuration:

> *PROVIDER:* YoutubeFragment.class
>
> *PARAMS:* Respectively: the playlist ID of the Youtube Playlist you would like to display; and optionally a Youtube Channel ID if you would like to enable search on your channel.

For example, if we would like to show the Youtube Playlist UU7V6hW6xqPAiUfataAZZtWA which is available on the channel UC7V6hW6xqPAiUfataAZZtWA:

```
i.add(new NavItem("Youtube Channel", 0, NavItem.ITEM, VideosFragment.class, new String[]
{"UU7V6hW6xqPAiUfataAZZtWA","UC7V6hW6xqPAiUfataAZZtWA"}));
```

Do you want to show your Channel, and not just a playlist? We made a small guide (/support/documentation/raw/youtube_setup.html) on how to find your ChannelId and the Playlist ID of your channel.

Adding a RSS item

If you would like to display an RSS feed in your app, you can use the following values:

> *PROVIDER:* RssFragment.class
>
> *PARAMS:* Your RSS url

For example, if we would like to show the following RSS feed; https://googleblog.blogspot.nl/atom.xml (https://googleblog.blogspot.nl/atom.xml)we can use:

```
i.add(new NavItem("RSS", 0, NavItem.ITEM, RssFragment.class, new String[]
{"https://googleblog.blogspot.nl/atom.xml"}));
```

Adding a Facebook item

If you would like to show a list of Facebook posts of a Facebook page you can use the following values:

> *PROVIDER:* FacebookFragment.class
>
> *PARAMS:* The Facebook ID of the page you would like to display. You can find your Facebook ID using a site like this (http://findmyfacebookid.com/), this (http://findfacebookid.com/) or this (https://lookup-id.com/).

For example, if we would like to show the posts of ID 104958162837:

```
i.add(new NavItem("Facebook", 0, NavItem.ITEM, FacebookFragment.class, new String[]
{"104958162837"}));
```

Adding a Instagram item

If you would like to show a list of your Instagram post, you can use the following values:

> *PROVIDER:* InstagramFragment.class
>
> *PARAMS:* The Instagram user ID of the user you would like to display. A way to find the user ID is described in this post (/support/documentation/../community/index.php?action=artikel&cat=2&id=67&artlang=en).

For example, if we had the user ID 1067259270:

```
i.add(new NavItem("Instagram", 0, NavItem.ITEM, InstagramFragment.class, new String[]
{"1067259270"}));
```

Adding a Twitter item

If you would like to show a list of tweets in your app, you can use the following values:

> *PROVIDER:* TweetsFragment.class
>
> *PARAMS:* You can use the username of a user you would like to display tweets from or display the results of a search query or hastag by putting *?* in front of your query. For example, using *android* as parameter will result in the tweets that the account with username *android* has posted, but using *?android* as parameter will result in all search results for *android*. Similarly you can use *?#android* to display all the results for *#android*.

For example, if we would like to show the tweets of user Android:

```
i.add(new NavItem("Twitter", 0, NavItem.ITEM, TweetsFragment.class, new String[]
{"Android"}));
```

Adding a WebView item

If you would like to show a web page inside your app, you can use the following values:

> *PROVIDER:* WebviewFragment.class
>
> *PARAMS:* The url of the webpage/site you would like to display.

For example, to show google.com in our app:

```
i.add(new NavItem("Webview", 0, NavItem.ITEM, WebviewFragment.class, new String[]
{"http://www.google.com"}));
```

Adding a Tumblr item

If you would like to show the images of a Tumblr user in your app, you can use the following values:

> *PROVIDER:* TumblrFragment.class
>
> *PARAMS:* The username of the user you would like to display.

For example, if we would like to show the images of the user androidbackground:

```
i.add(new NavItem("Tumblr", 0, NavItem.ITEM, TumblrFragment.class, new String[]
{"androidbackgrounds"}));
```

Adding a Radio item

If you would like to let your users play a Radio station from within your app, you can use the following values:

> *PROVIDER:* MediaFragment.class
>
> *PARAMS:* The url to the stream

For example, if we have the ShoutCast stream http://yp.shoutcast.com/sbin/tunein-station.m3u?id=709809:

```
i.add(new NavItem("Radio Stream", 0, NavItem.ITEM, MediaFragment.class, new String[]{"http://yp.shoutcast.com/sbin/tunein-station.m3u?id=709809"}));
```

Adding a TV/Video Stream item

If you would like to let your users watch a live video stream or video from within your app, you can use the following values:

> *PROVIDER:* TvFragment.class
>
> *PARAMS:* The url to the stream

For example, if we have the Video stream http://149.255.152.110/discovery/index.m3u8:

```
i.add(new NavItem("Video Stream", 0, NavItem.ITEM, TvFragment.class, new String[]{"http://149.255.152.110/discovery/index.m3u8"}));
```

Adding a SoundCloud item

If you would like to play tracks of a SoundCloud user in your app, you can use the following values:

> *PROVIDER:* SoundCloudFragment.class
>
> *PARAMS:* Sound Cloud user ID. You can use a tool like this (http://timpietrusky.github.io/what-is-my-soundcloud-user-id/) or this (https://helgesverre.com/soundcloud/) to find the ID.

For example, if we have the user ID 13568105:

```
i.add(new NavItem("SoundCloud", 0, NavItem.ITEM, SoundCloudFragment.class, new String[]{"13568105"}));
```

Adding a Maps item

If you would like to display a map in your app, either of a single predefined location or of nearby search results, you can use the following values:

> *PROVIDER:* MapsFragment.class
>
> *PARAMS:* If you would like to display nearby results for place types (https://developers.google.com/places/supported_types) you can simply enter the place type as single parameter. If you would like to display a single pin on the map you can use the following parameters: HTML description of the location, title of the pin, body of the pin, latitude, longitude, zoom level.

For example, if we would like to display the location of our company named 'Company' with the description 'This is where our office is' at lat -33.864 and lon 151.206 with zoom level 13 and a nice HTML description:

```
i.add(new NavItem("Maps Query", 0, NavItem.ITEM, MapsFragment.class,
        new String[]{"<b>Adress:</b><br>SomeStreet 5<br>Sydney, Australia",
            "Company",
            "This is where our office is.",
            "-33.864",
            "151.206",
            "13"}));
```

Adding a Custom Intent item (Advanced)

If you would like to perform a custom action from the menu, you can add a Custom Intent to your menu to launch an app or url:

> *PROVIDER:* CustomIntent.class

PARAMS: The first parameter is the package name of the app to launch, the URL to open (this can be any any URL suitable for ACTION_VIEW and it will automatically be opened in the appropriate app) or the class of the Activity to open; The second parameter is the type of intent this is, you can either use CustomIntent.OPEN_URL, CustomIntent.OPEN_APP, CustomIntent.OPEN_ACTIVITY (no quotes)

For example, if we would like to open the Dialer with phone number 123456:

```
i.add(new NavItem("SoundCloud", 0, NavItem.ITEM, CustomIntent.class, new String[]{"tel:123456", CustomIntent.OPEN_URL}));
```

Items that require an in-app purchase

You can add items to your app that are exclusively available to user that have made an in-app purchase. This in-app purchase will not only remove ads, but can also unlock extra content.

To have an item require an in-app purchase to be made, in order for it to be used, you can add a boolean parameter with value 'true' to your configuration line. Considering the default configuration line, an item requiring an in app purchase would be formatted like this:

```
i.add(new NavItem("NAME", DRAWABLE, NavItem.ITEM, PROVIDER, new String[]{PARAMS}, true));
```

If the user has not made an in-app purchase, he can not access the content and he will be asked to make an in-app purchase to unlock & access the content. *Keep in mind that using in-app purchases in your app requires the Extended License from CodeCanyon.*