Classes

# Client

## Utility classes

Most of these classes are a collection of static functions used throughout that app.

* **BitmapUtils** – contains function(s) for dealing with bitmaps
* **GlideUtils** – contains function(s) for working with Glide (image processing library)
* **InputValidationUtils** – contains function(s) for validating user input using Regex patterns
* **ListUtils** – contains some custom utility functions and various high level list functions that aren’t available for the minimum SDK the app can be deployed on as such needed custom implementation.
* **UIUtils** - contains function(s) for dealing with UI components such as getting cleaned text from a text field and hiding elements.
* **RequestCodes** – this class contains codes for the activity result and permission result requests the app will make.
* // todo – move this to the ui package.

**AppScreen** – an enum containing the different screens that the app can show. Each of these corresponds to a specific fragment which will be hosted in the single Activity that the app contains. Each enum member contains an instance of its corresponding fragment. These fragment instances are used for navigation within the app.

## Local Data Stores

These classes contain functionality of saving and retrieving local data. Some of them deal with persisted data and some only save to RAM.

* **DataSets** – this class deals with data kept only in RAM. It includes a reference to the currently logged in user and a list of all categories products can be assigned to. Both these values are received via a login response (through 3 different endpoints). The references to these objects never change, only the internal members do.

This is used throughout the app to get user information and to display list of categories.

* **InitializedStore** – a superclass containing the infrastructure for singleton classes that need initialization before use.
* **TokenStore**, a subclass of InitializedStore – this class deals with persisting the jwt token received via server request to SharedPreferences.

This token is used in most server requests and received from most server responses (only request/responses where the token is unknown will leave it out such as registering)

As the token may refresh at any request (as a security measure), it is returned in server responses and persisted locally.

This token is used to identify the user for most server request and allows such things as automatic login for a user who has logged in in the past and hasn’t logged out.