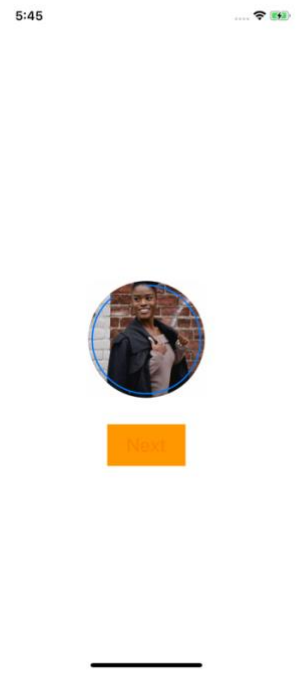
Image loading library

Project Description:

This is a xCode project then contains:

- Image loading library, can be found inside AvatarLib folder.
- The rest of the project is an example that uses this library as a one screen app with an Image view that make use of the library, and a next button that keep loading images from an example API (real backend).



To run the Project:

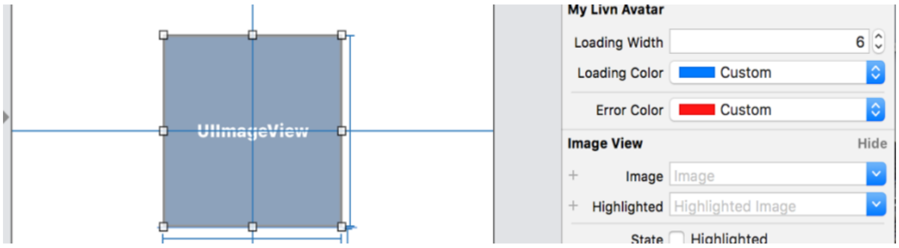
The project simply runs on xCode 10+, and can be deployed on any device/simulator running iOS 11+.

NOTE: No use of pods or any external libraries added.

Code Structure:

The library consists of the following components:

- MyLivnAvatar: The Main interface for the library, a class that extends apple UIImageView, and encapsulates all the required behaviors, this class have some properties that can be set from code or from Interface Builder, in order to manage how loading progress looks like.



My Livn Avatar

Loading Width 6

Loading Color 🟦 Custom

Error Color 🟥 Custom

Image View Hide

Image Image

Highlighted Highlighted Image

State □ Highlighted

- MyLivnImageLoader: the Class responsible for loading images, and can be separated from the library and used alone in any other project or with other classes, it have only one responsibility to download files and delegates the actions needed upon file download status to whoever using it.
- MyLivnCachManager: Caching manager also has one responsibility to handle file caching and can be separated too, and have two properties as required to manage maximum caching for the app:
    1. maximumNumberOfObjects
    2. maximumStorageSpace

Sample app API:

I used an external service that host images called : unsplash.com, it is a free API that can retrieve images, it needs authentication and this is added in the code, if for any reason this authentication doesn't work anymore please let me know and I will assist with this, the example API url will be found in a static variable called IMAGES_URL found in ViewController.m