Ruben Mimoun

Moveo Test

Navigation :

The app navigation is done by using a router, instantiating view controllers from different storyboards(Auth, Notes, Error), according to the protocol they are linked to.

Each ***instantiate*** method infer the T type of the desired view controller. The Router ***push*** method checks first if the instantiated view controller type exists in the the stack before adding it otherwise it pops it from the stack.

Design Patterns :

The app uses **MVP** and **MVVM** design patterns. The *LoginViewController*, and *RegistrationViewController*, are not using any kind of data, but are processing inputs to log in or to register. The business logic of the inputs has been pull out to their presenter.

NoteListViewController, *NotesMapViewController*, *NotesEditViewController* are getting data delivered through **MVVM** coupled with FirebaseFirestore. Each type of view models contains an *Observable<T?>* object. Once the object's value is defined, we trigger the listening by inserting a closure as a parameter to the ***bind*** method it holds. This method triggers the closure, updates the value with on ***didSet*** triggers closure. This way, when the data value of the Observable is updated, the closure containing the UI update is triggered as well.

The list view controllers, is given a view model getting all the notes together. When data is received it creates and array of notes view model containing data for each cell, used to reload the tableview and render the number of the cell and from which we pull the associated note view model to insert in each cell.

The process is very similar to the map screen data displaying.

The map screen and the list screen are using the same view model object kind, and the annotations placed of the map, the cells in the table and the edit screen are using the same view model object kind.

Data Handling :

As previously said data is mainly handled with FirebaseFirestore, although image uploading is managed with FirebaseStorage.

The project doesn't contains, any pods nor *file.xcworkspace*. I have implemented a ***Swift Package*** wrapping Firebase data methods I have been working on and you can find here :

https://github.com/boulomiel/FirebaseWrapperSPM.git

All the data handling methods and contained into the *FirebaseDataManager* singleton. And can be applied to any *<T: FirebaseCodable>* object. *FirebaseCodable* being a type from the SPM.

Each user is given a collection of his own from FirebaseFirestore, he cannot reach other users collections.

Images are stored in Storage according to the currentUserEmail and the timestamp and it's being uploaded.

Authentication handling :

Authentication methods are contained into the *FirebaseAuthManager*. Each methods returns their success or failure through a completion handler. The *SplashScreenViewController*, checks if the user is logged in, using an ***authStateListener*** before routing him the next screens.

General UI Informations :

As said earlier, we use different Storyboards.

Auth : contains a *UIPageViewController*, in which the login and and register screens are inserted.

Notes : contains a *UITabBarController*, to which are linked the list screen and the map screen. For a matter of scalability if the list screen or the map screen are to have “brothers” they are both wrapped into a *UINavigationController*. The *NotesEditViewController* is pushed when : a cell is tapped in the screen, an annotation is tapped in the map screen or when we tap the right navigation item in the list screen.

If no data has been retrieved, both the map screen and the list screen are overlaid with a *NoNotesView*. It’s an animated view containing a figure image pointing to the right navigation item.

Etiam Sit Amet Est

Donec Quis Nunc

Lorem ipsum dolor sit amet, ligula suspendisse nulla pretium, rhoncus tempor fermentum, enim integer ad vestibulum volutpat. Nisl rhoncus turpis est, vel elit, congue wisi enim nunc ultricies sit, magna tincidunt. Maecenas aliquam maecenas ligula nostra, accumsan taciti. Sociis mauris in integer, a dolor netus non dui aliquet, sagittis felis sodales, dolor sociis mauris, vel eu libero cras.

Arcu habitasse elementum est, ipsum purus pede porttitor class, ut adipiscing, aliquet sed auctor, imperdiet arcu per diam dapibus libero duis. Enim eros in vel, volutpat nec pellentesque leo, temporibus scelerisque nec. Ac dolor ac adipiscing amet bibendum nullam, lacus molestie ut libero nec, diam et, pharetra sodales, feugiat ullamcorper id tempor id vitae. Mauris pretium aliquet, lectus tincidunt. Porttitor mollis imperdiet libero senectus pulvinar. Etiam molestie mauris ligula laoreet, vehicula eleifend. Repellat orci erat et, sem cum, ultricies sollicitudin amet eleifend dolor nullam erat. Leo vitae diam est luctus, ornare mauris urna, vitae sodales et ut facilisis dignissim, imperdiet in diam, quisque.

Leo vitae diam est luctus, ornare mauris urna, vitae sodales et ut facilisis dignissim, imperdiet in diam, quisque adipiscing nec posuere feugiat ante velit. Vivamus leo quisque. Neque mi vitae, nulla cras diam lacus, nibh pellentesque libero. Dolor at venenatis in, ac in quam purus diam mauris, dolor leo vehicula at commodo. Turpis condimentum varius aliquet accumsan, sit nullam in turpis augue, vel tristique, metus id consequat orci penatibus. Ipsum vehicula euismod aliquet, pharetra sed nulla hamenaeos pede ut ipsum, facilisis dui pede nibh et arcu, imperdiet condimentum bibendum maecenas sollicitudin, dolor pharetra rutrum ultrices nunc. Lobortis cras, dui morbi nulla habitant suscipit, mauris pede ante at integer pretium, amet potenti necessitatibus sapien natoque dolor, tincidunt et hendrerit ligula. Non convallis molestie sodales in, augue integer nulla, condimentum egestas mauris est justo ornare, elit sed ornare, vel nec. Nec nam tortor et donec, augue et euismod non risus sem. Montes et metus adipiscing consectetuer nunc. Non libero nam dolor. Nascetur quis ut, tristique libero sit, ac ut in et felis convallis. Pellentesque dignissim amet commodo, nec turpis dignissim torquent, laoreet orci unde aptent tenetur, dolor sit. Sed sed mauris duis. Quis enim ut, dolor id arcu explicabo ligula, quisque. Arcu habitasse elementum est, ipsum purus pede porttitor class, ut adipiscing, aliquet sed auctor, imperdiet arcu per diam dapibus libero duis. Enim eros in vel, volutpat nec pellentesque leo, temporibus scelerisque nec. Ac dolor ac adipiscing amet bibendum nullam.