Combine#

- **Apple introduce combine in WWDC 2019**
- **The minimum supporting version of iOS is iOS 13**

Why Combine?

It removes the dependencies of the external libraries like RxSwift and RxCocoa & it providing the flexibility of reactive programming.

What is Combine?

Combine is a framework similar to the Rx which allows you to write functional Reactive code.

It provides you declarative APIs.

In layman's terms : **Functional reactive programming** is something that allows you to process values over the time to process something that will be happening at some point in time some asynchronous stuff so you can relate it with network calls or you can relate it with the notifications. Because we don't know that at what time we will be receiving the the notifications but we don't know that what time that particular event will be taking place. So anything where we want to process the values over time that can fall under the category of function reactive programming, and how does it works, it works on the principle of **pub-sub** model i.e. **Publisher-Subscriber model**. So we have **Publisher** which produces the value and then we have **Subscriber** one or more which consumes that value which produced by the **Publisher** and **Combine** has introduced third part into this which we call as **Operators**. So operators are nothing but the different kind of **Publishers** which take the inputs which receives the value from some different Publisher and they produce some different value by performing some operations on it.

Publishers

Publishers is a protocol and it declares that any entity that confirms to this protocol it can transmit a sequence of values over time now this protocol has two associatedTypes that is one for output and the other one for failure. so the output will be the type of value it can produce and the failure will be the type of error which can be encountered.

Subscriber#

It is also the protocol and it declares that any entity that confirms to this particular protocol that is Subscriber. It can receive input from Publisher now similar to the Publisher it also has two associatedTypes to make the Publisher and Subscriber work in he sync. The input type and the failure type of Subscriber must match the output type and the failure type of its corresponding Publisher.

Operators#

Operators are the methods that are called on Publishers and their return Publishers so they are used for manipulating the values by changing them/adding them/removing them and they can be chained together so you will see this in action when we will be making the web service calls, once we receive the response from the server then we perform some operations on that particular response. We either map it or we parse it and all these things will be done by operators.

What is @Published ?

@Published is a **property-wrapper** which wraps our properties that can be a password or anything else. it drives that properties and it automatically write **willSet** block, the same willSet block that we write in our computed properties so that willSet block get automatically written when we use the thread publish property wrapper with our properties and then whenever the value of this particular property will change the observers will automatically notified about that new value and is available if you want to do some operation if you want reload or anything else you can do.

Zip & CombineLatest#

Zip is a publisher which takes other publishers as the parameters or as the arguments and the returns the tuple Just like CombineLatest.

The difference between Zip & CombineLatest is that Zip will return only when all of the Publishers will complete their execution just like Dispatch Group works. I am relating this with dispatch group so that it is become easier to understand but it is not exactly like that. The internal working of Zip is different than what is from the Dispatch Group, so the difference is that Zip will pass on the values to the downstream only when all of the Publishers have completed their execution while in case of the CombineLatest it will pass the values of any of the publisher changes