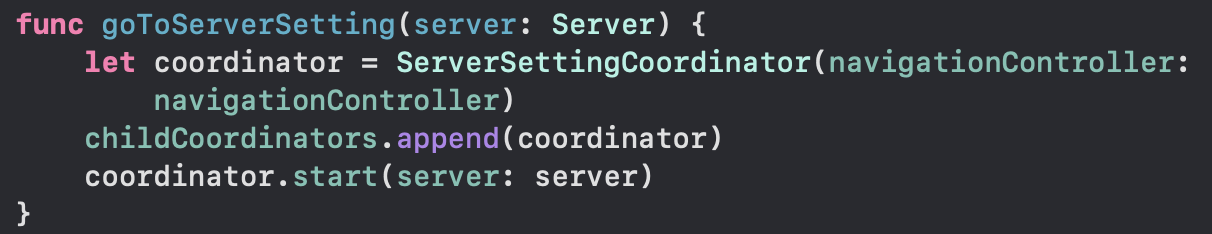
프로젝트를 진행하면서 고민해왔던 것들에 대해서 더 좋은 방법이 있는지 궁금해서 질문 드립니다.

#### 1. viewController의 책임과 관련해서

프로젝트 설계를 MVVM으로 설계 했고, ViewController(이하 vc)와 ViewModel(이하 vm)의 책임에 대해서 많은 고민이 되어 질문 드립니다.

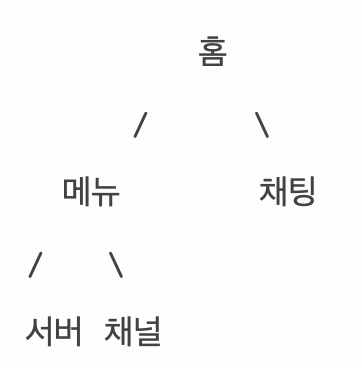
저의 경우, coordinator로 화면 전환 시 vc를 통해 값 데이터를 아래와 같이 전달 받고 있습니다.



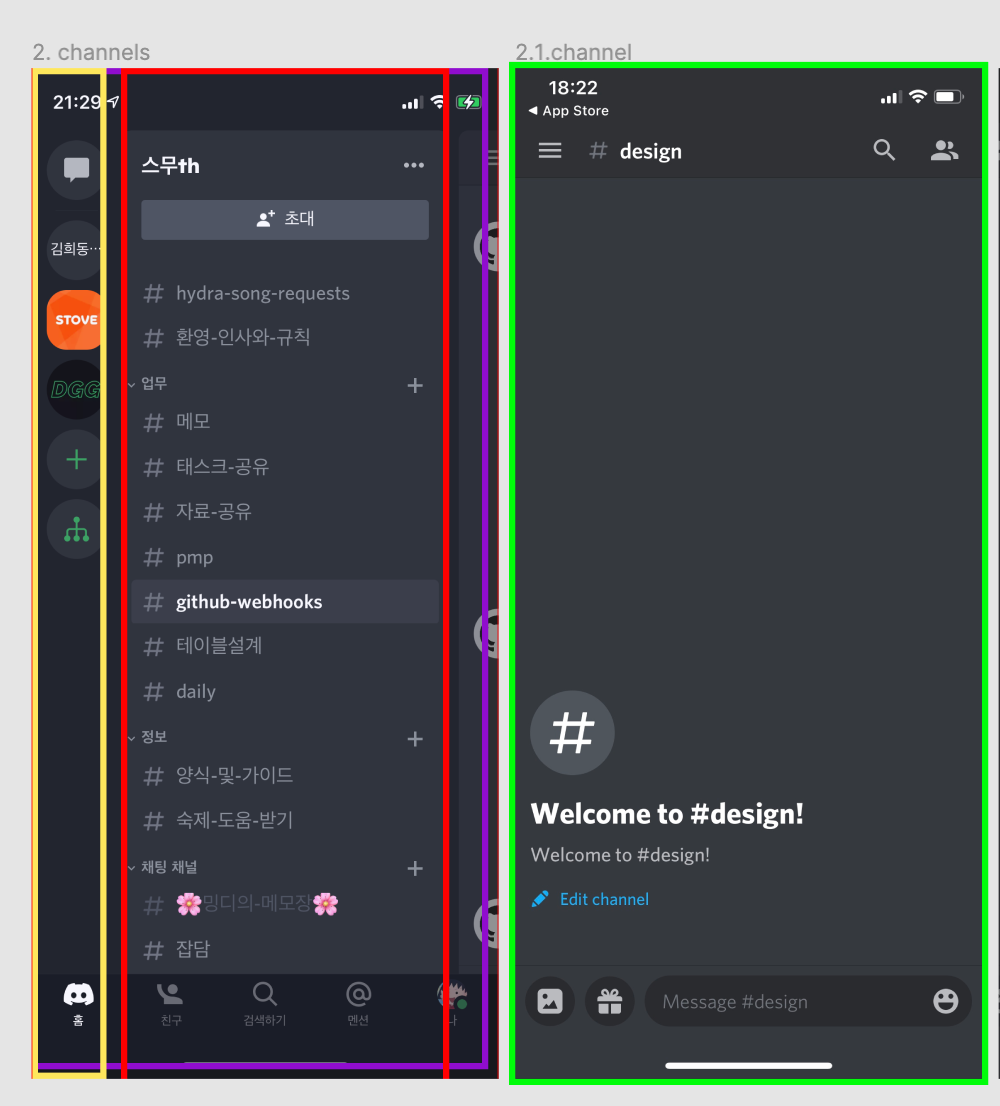
이렇게 전달 하면서, vc가 데이터를 들고 각각 view와 vm에 바인딩을 하는데, 그렇게 되면 vc가 데이터(모델)을 가지고 있는 격이라, MVVM의 Model을 vc가 들고 있는 경우라고 생각이 되어집니다. 보통 현업에서 vc와 vm의 책임을 어느 정도 선에서 타협하고, 어떻게 설계하는지 궁금합니다.

#### 2. 하나의 화면에 여러 데이터를 전달해야하는 상황 관련

디스코드의 경우 하나의 화면에서 여러 메인 컴포넌트를 기반으로 데이터를 관리하고 있습니다.

그래서, 화면을 기준으로 다음과 같이 나눠서 설계 하였습니다. (아래 그림 참고)

이렇게 설계 했을 때 home vc가 중간에 하위 vc(채팅, 메뉴)들을 매니징 하는 역할로 존재하게 됩니다.



노랑 - 서버(커뮤니티) vc

빨강 - 채널 vc

– 노랑+빨강 - 메뉴 vc

초록 - 채팅 vc

보라 - 홈 vc

Q1. 보통 화면 간 subView들이 계층적으로 이뤄지는 경우, home vc와 같이 중간 매니징 하는 역할의 vc가 존재하는게 일반적일까요?

Q2. 그래서 화면 간 전환은 home vc가, 그리고 값 데이터 전달은 menu vc를 기준으로 해서 전달하고 있는데, 메뉴 vc 내에서는 Rx로, 그리고 같은 계층 (메뉴 <-> 채팅) 간은 delegate를 사용해서 전달하고 있습니다. 모두 Rx로 통일해서 하고 싶은데, Binder를 사용 해서 주입이 가능한 것을 확인하였지만 Binder는 view 로직 관련해서 값을 주입할 때 주로 사용한다고 합니다. 이럴 경우 어떻게 해서 값을 전달 해야 하나요?

#### 3. table/collection ViewController 관련

table/collection view controller의 경우 datasource를 기반으로 해서 그려주게 됩니다. view와 관련된 것들(layout, cell appearence 등)과 데이터 부분을 명확히 분리하고 싶은데, viewController에서 관리하는 경우가 많더라고요. 이런 경우, viewcontroller가 비대해져서 나중에 코드 관리하기가 힘들어질 수 있을 것 같다는 생각이 들었습니다.

만약 table/colllection view는 파일을 따로 둬서 책임을 명확하게 두고 진행한다고 하였을 때 datasource는 viewcontroller에 두고 사용하나요? 아니면 view 파일에 같이 선언하고 사용하나요?