



Flutter folder structure 참고 repo

<https://github.com/zubairehman/flutter-boilerplate-project>

2000+ star를 받은 것으로 보아 많은 사람들의 공감을 얻었다고 판단이 됩니다.
(일단 제가 본 flutter template들 중 가장 interaction이 많습니다)

Flutter를 처음 잡으면서 폴더 배치도 난항을 겪었던 요소 중 하나였기 때문에,
reference를 두고 고치는 방향을 택하는 것이 맞다고 판단했습니다.

- 해당 repo에서 라이브러리 사용, 로직 등 다른 요소도 참고할 것이 있을 수도 있지만, 일단은 folder structure에 관한 고찰만 기록합니다.
- 230618 현재 Readme와 소스 코드의 sync가 맞지 않는 것으로 추정됩니다.

해당 repo에서 folder structure에 대한 설명

1. Constants

앱에서 사용하는 모든 상수를 모아 두는 폴더.
분류에 따라 다른 파일을 생성하여 보관한다.

2. Data

앱에서 사용하는 모든 비즈니스 로직이 들어간다.
여기서는 `local`, `network`, `sharedpref` 로 나누어 디렉토리를 생성하였고
root에 `repository.dart` 파일을 생성하여 각 unit들을 합쳐서 사용할 수 있게 했다.

3. Stores

앱의 전역 상태에 관련된 항목이 들어간다.

예시로 `login` 디렉토리가 있는데 실제 소스에서는 `user` 로 바뀐듯?
기능에 따라 디렉토리를 따로 분류해서 배치를 해 둔 것 같다.

4. UI

실제 화면을 구성하는 소스가 들어간다.
스크린에 따라 화면을 구분해 두었으며,
해당 스크린에서만 쓰는 widget은 따로 폴더를 파서 배치했다.

5. Utils

공통으로 사용하는 로직이나 유틸 등이 들어간다.

6. Widgets

여러 화면에 걸쳐서 사용되는 widget들이 들어간다.

7. `routes.dart`

8. `main.dart`

사건 (안 읽어도 됨)

- Utils, Widgets 폴더에 들어가는 기준은 독립성으로 판단하면 될 것 같다.

앱의 다른 코드에 의존하지 않고,
과장 좀 보태서 해당 파일을 그대로 다른 앱으로 복붙해서 사용할 수 있으면 OK

해당 앱에서 1번만 사용했더라도, 다른 앱에서 그대로 재사용이 된다면
그건 Widgets 폴더에 있는 게 맞다고 생각...

dependency에도 최대한 영향을 받지 않는 것이 좋음.

`shared_preferences` 와 같이 안 쓸 수가 없는 라이브러리나,
모두가 합의해서 해당 그룹에서 모든 앱에 사용하기로 한 라이브러리 등은 예외

- Stores 폴더와 Data 폴더의 공존 여부는 고민을 해 보아야 할 듯
- 폴더 이름만 보고는 직관적으로 분류가 와닿지 않아,
약간의 네이밍이나 구조 수정이 필요할지도 모르겠다.

ex) Data → Logic 으로 이름 변경

ex) Utils와 Widgets 폴더를 Common 폴더 아래에 배치

그 외...

<https://github.com/SimpleBoilerplates/Flutter>

`riverpod` 와 `go_router` 등을 사용했지만 폴더 구조가 난해함

몇 개 더 찾아보기는 했는데 지금까지는 소개한 구조가 제일 깔끔한듯...
더 좋은 ref가 있을지는 모르겠습니다.