

Flutter

- <https://flutter.dev>
- <https://pub.dev>



Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase.

Flutter는 상태(**State**)에서 시작하여 상태(**State**)로 끝난다.

[Flutter](#), [Framework](#), [Mobile](#), [Hybrid](#)

필수 Library

Alternate reflection

- Build runner
 - https://pub.dev/packages/build_runner

Dart는 Reflection이 존재하지 않음. 이를 보완하기 위해 직접 대응하는 소스 코드를 생성한다. 이에 의존하는 라이브러리로는 [Json Serializable](#), [MobX](#) 등이 있다.

아쉽게도 Java의 Lombok 혹은 Kotlin에 "data" class에 해당하는 라이브러리는 완성도가 좋지 않다.

JSON

[Json Serializable](#)

아래 기술되어 있는 MobX, Retrofit은 해당 라이브러리를 지원한다.

State management

공식 홈페이지에서는 Provider 사용을 권장하지만 완벽히 Widget으로부터 분리된 상태 관리를 원한다면 외부 Library도 좋은 선택이라 생각한다.

- 참조 : <https://flutter.dev/docs/development/data-and-backend/state-mgmt/options>

아키텍처 패턴, Boilerplate등을 고려했을 때 MobX가 가장 최적이었다.

- MobX
 - <https://pub.dev/packages/mobx>
 - https://pub.dev/packages/flutter_mobx
 - https://pub.dev/packages/mobx_codegen

REST

- [Retrofit](#)
- [Chopper](#)

둘 다 Android의 Retrofit을 영향 받아 구현한 라이브러리인데, `BaseUrl`을 Annotation 기반으로 선언하는 편리함에 있어서 Retrofit이 훨씬 좋은 디자인으로 생각된다.

Tips

사용자 상호작용 없이 상태 변경

- <https://stackoverflow.com/questions/54846280/calling-setstate-during-build-without-user-interaction>

MobX, Redux 등을 이용할 때 FCM을 받거나 Timer를 작동시켜 Widget을 변경 하려는 경우 아래와 같은 오류 메시지를 접하게 된다.

```
setState() or markNeedsBuild() called during build.
```

이 경우 아래와 같이 Widget이 Rendering 된 후 상태를 변경시키면 된다.

```
WidgetsBinding.instance.addPostFrameCallback((_) => setState(() {}));
```

From:

<http://wiki.ledyx.xyz/> - **Ledyx WIKI**

Permanent link:

<http://wiki.ledyx.xyz/doku.php?id=mobile:flutter>

Last update: **2021/02/07 05:52**

