



# Flutter로 Movie App 만들기

**19.12.19**



## 발표자 소개

이름 : 김정태

직업 : Android Framework 개발자  
(6년차)

# Overview

개발 기간 : Flutter 기본 학습 2 month + 개발 1.5 month = 3.5 month

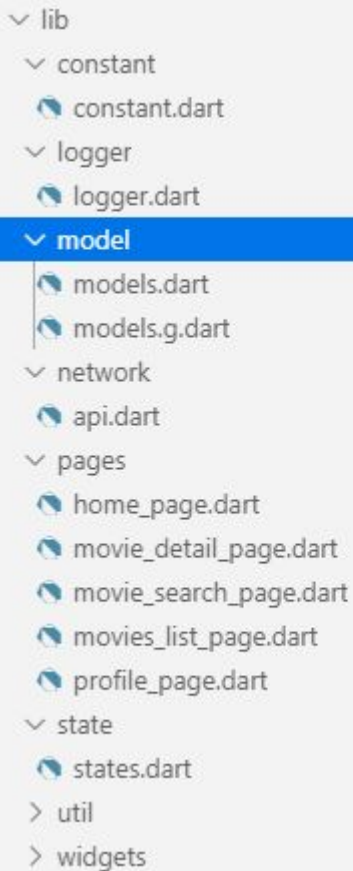
Library :

- Http
- Provider
- Expandable
- Logger
- Json\_annotation
- 

API : <https://developers.themoviedb.org/3>



# Package 구조



```

└─ lib
   └─ constant
      └─ constant.dart
   └─ logger
      └─ logger.dart
   └─ model
      ├── models.dart
      ├── models.g.dart
      └─ network
         └─ api.dart
   └─ pages
      ├── home_page.dart
      ├── movie_detail_page.dart
      ├── movie_search_page.dart
      ├── movies_list_page.dart
      └── profile_page.dart
   └─ state
      └─ states.dart
   > util
   > widgets

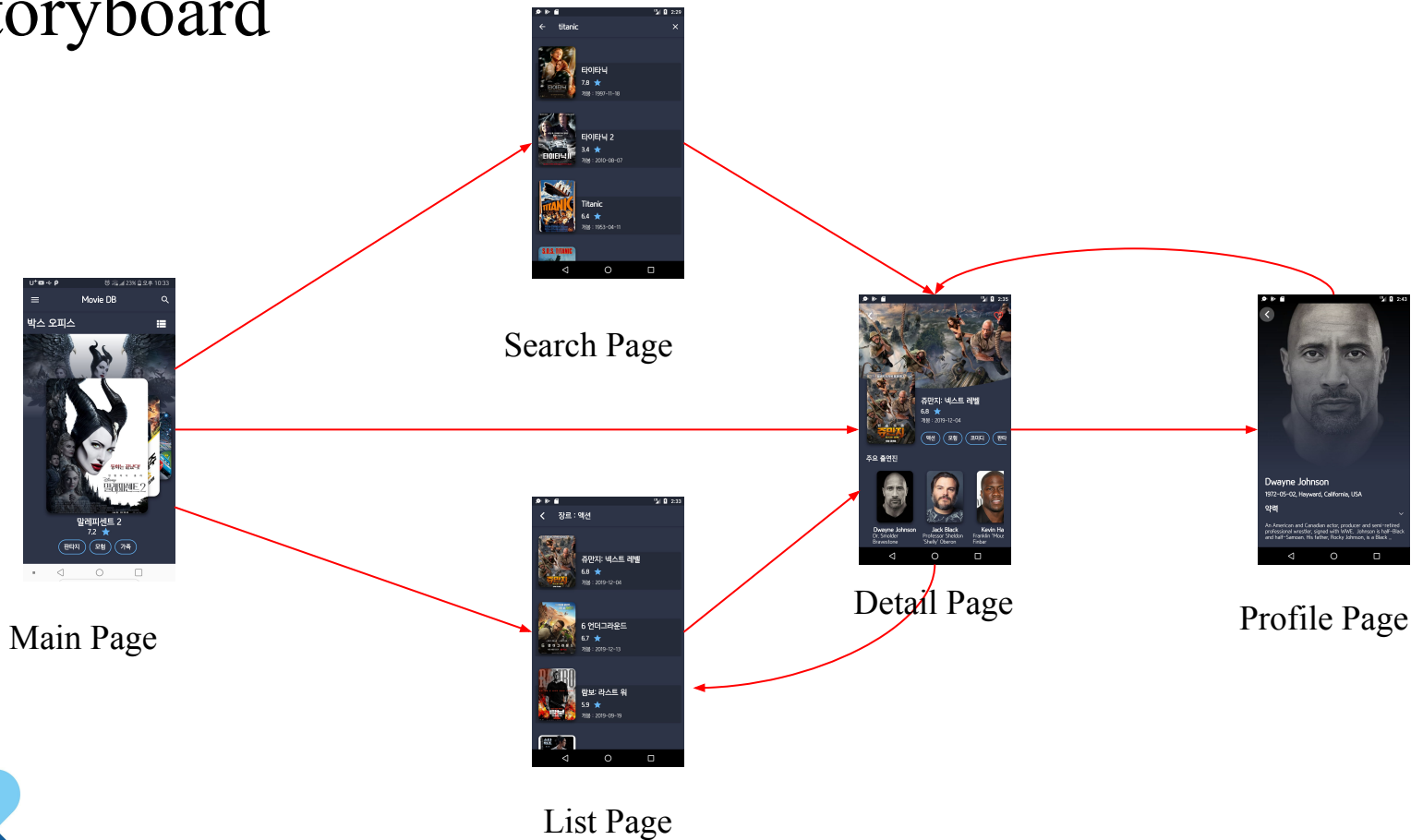
```

A screenshot of a file explorer showing a package structure. The 'model' directory is highlighted in blue. The structure is as follows:

- lib
  - constant
    - constant.dart
  - logger
    - logger.dart
  - model
    - models.dart
    - models.g.dart
    - network
      - api.dart
  - pages
    - home\_page.dart
    - movie\_detail\_page.dart
    - movie\_search\_page.dart
    - movies\_list\_page.dart
    - profile\_page.dart
  - state
    - states.dart
  - > util
  - > widgets



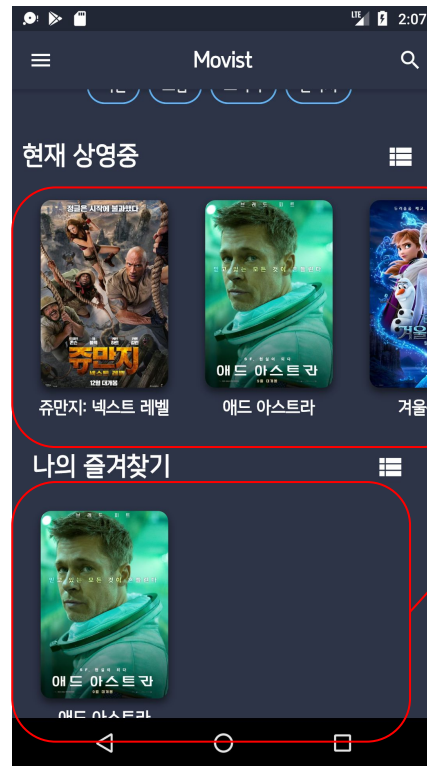
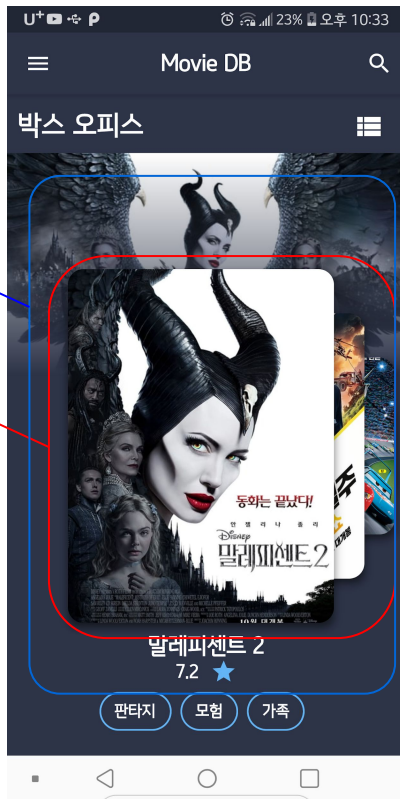
# Storyboard



# Main Page

PageView

Stack

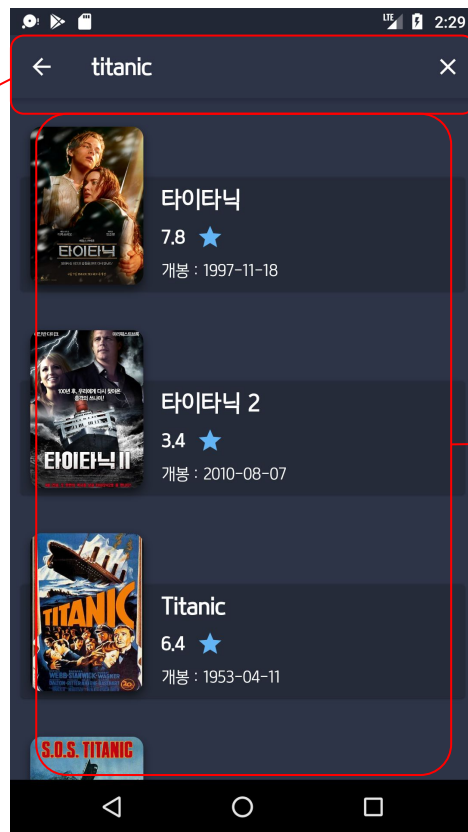


ListView  
(horizontal)



# Search Page

SearchDelegate



ListView  
(Vertical)



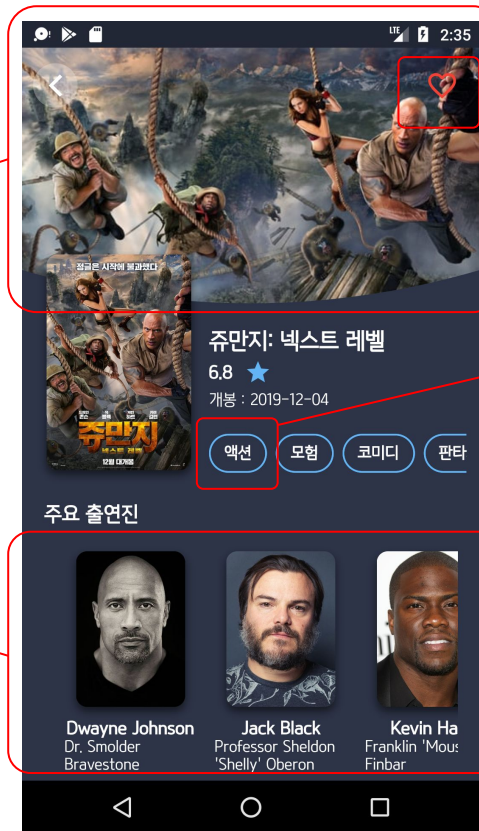
# Detail Page

ImageIcon

ClipRRect

Chip

ListView  
(Horizontal)





# Profile Page

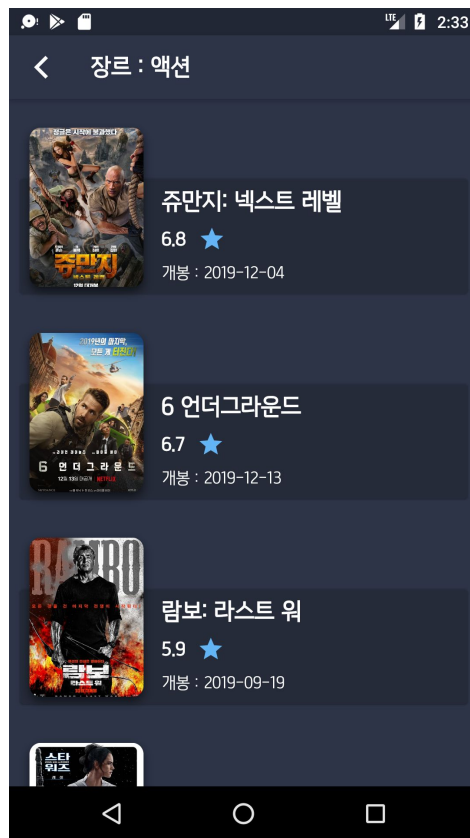
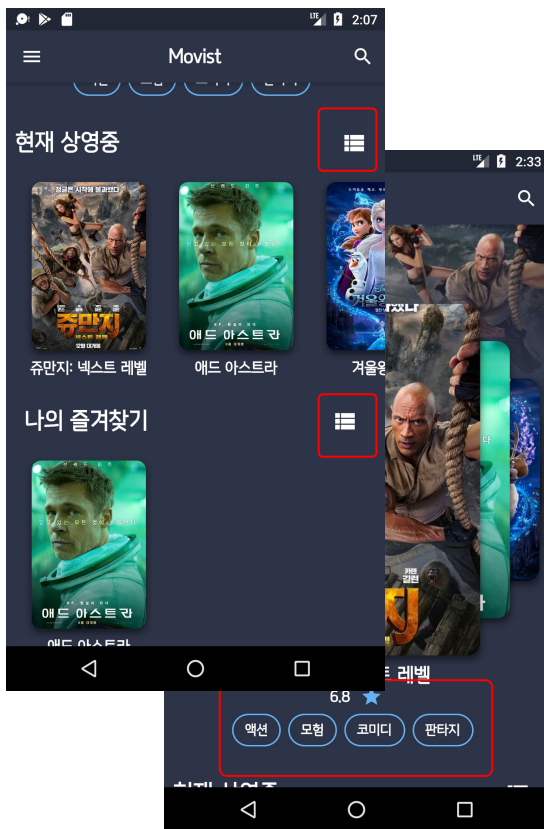


SingleChildScrollView

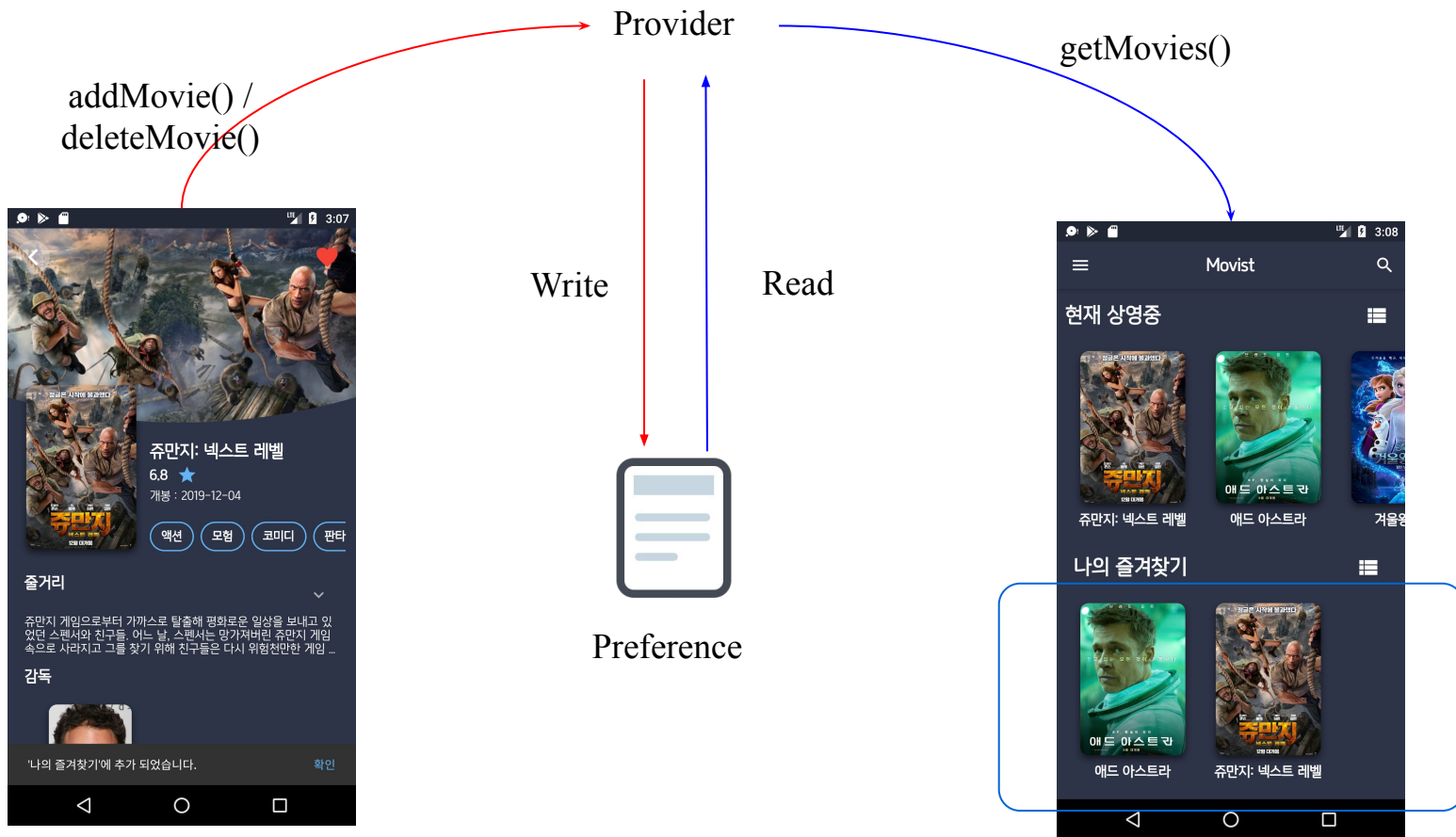
Container  
(No Transparent)



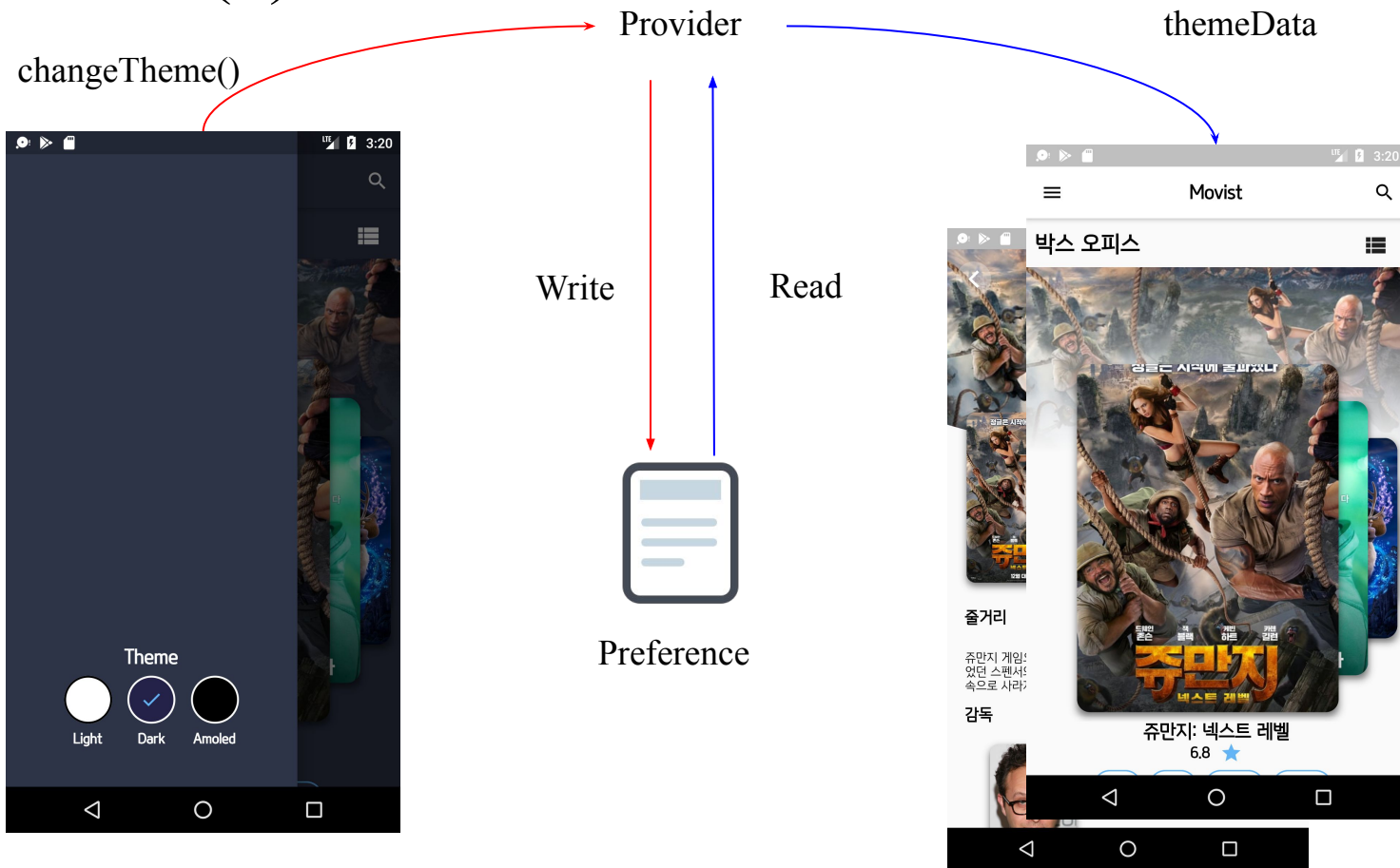
# List Page



# Favorite



# Theme 변경 (1)



# Theme 변경 (2)

```
void main() {

  runApp(
    MultiProvider(
      providers: [
        ChangeNotifierProvider(builder: (_) => FavoriteState()),
        ChangeNotifierProvider(builder: (_) => ThemeState()),
      ],
      child: MyApp()),
  );
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    ThemeData themeData = Provider.of<ThemeState>(context).themeData;

    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: themeData,
      home: HomePage(),
    );
  }
}
```



# Conclusion

## Pros

- UI 개발이 쉽다.
- 비 동기 연산에 대해서 처리가 매우 간단하다.
- Hot reload

## Cons

- Declarative programming이 생소한 경우 이해하기가 어렵다.
- Dart라는 언어를 새로 배워야 한다.
- 조금이라도 복잡한 UI를 그리게 되면 들여쓰기 Hell이 펼쳐진다.
- Bug나 막히는 부분이 있을 때 참고 할만한 자료가 적다.

