Flutter 주니어 질문 100제 part. 02

100 Flutter Junior Questions

조새벽

GDG Songdo orgernizer
Flutter Incheon orgernizer

https://www.linkedin.com/in/chochoq/

GDG 인천 송도 오픈카톡방

Flutter가 재미있는 이유

01

재미있는 UI 개발 경험

- 위젯 중심의 설계
- 다양한 앱 제작 가능 (Android, iOS, desk top app, web …)

02

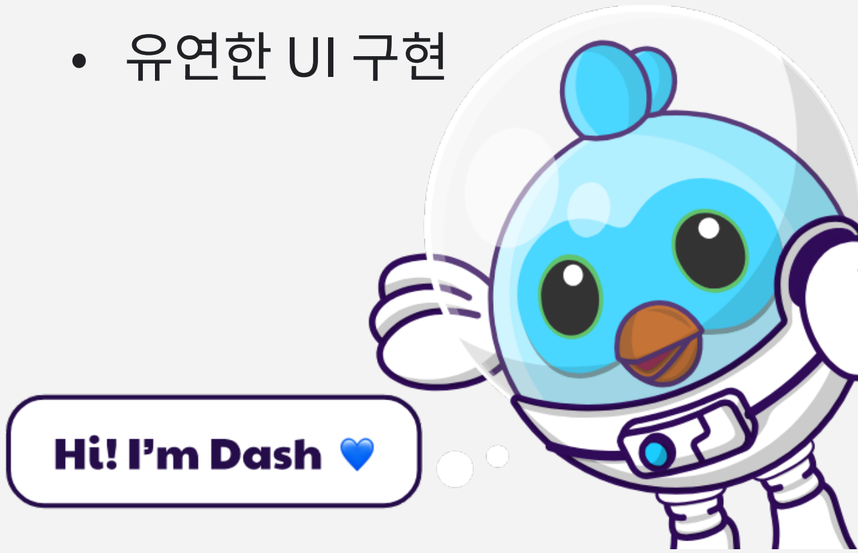
꾸준한 업데이트

- 다트 및 플러터의 발전
- 개발자와 함께 성장

03

커스터마이징의 자유도

- 다양한 위젯 제공
- 유연한 UI 구현



https://speakerdeck.com/chochoq/2023-dot-12-devfest-songdo-peulreoteo-junieo-jilmun-100seon?slide=2

질문 목차

1. Flutter에서 Dart 언어의 역할은 무엇인가요?
2. Flutter는 왜 Dart를 사용하는 건가요?
3. Dart랑 JS랑 비슷하다고 들었는데 왜 비슷하죠? 차이점이 뭐예요?
4. 플러터 프로젝트를 처음 만들었을 때 만들어지는 폴더 구조를 설명해 주세요!
5. pubspec.yaml 설명해 주세요!
6. 디버그라고 써 있는 빨간 배너는 뭐예요?
7. 프로필 모드는 뭐고 언제 사용하는 거예요?
8. 릴리즈 모드는 뭐고 언제 사용하는 거예요?
9. 플러터 앱 개발에서 핫 리로드랑 핫 리스타트가 뭐예요?

Google Developer Groups

https://speakerdeck.com/chochoq/2023-dot-12-devfest-songdo-peulreoteo-junieo-jilmun-100seon?slide=2

질 문 목 차

1. 플러터로 개발된 앱이 뭐가있어요? 소개해 주세요!
2. State가 뭐예요?
3. Key? Globalkey? 란 무엇인지? 그리고 어떻게 사용하는지
4. Flutter 를 하면서 제일 많이 만났던 게 null safety 에러였는데… 정확한 사용법이 궁금해요 ㅠㅠ
5. 플러터는 많은 라이브러리(패키지)와 함께 사용하는 것으로 유명한데요, 비슷한 라이브러리도 많은데 그 중에 나와 맞는 것을 잘 고르는 팁(?)

Q. 플러터로 개발된 앱이 뭐가있어요?
소개해 주세요!

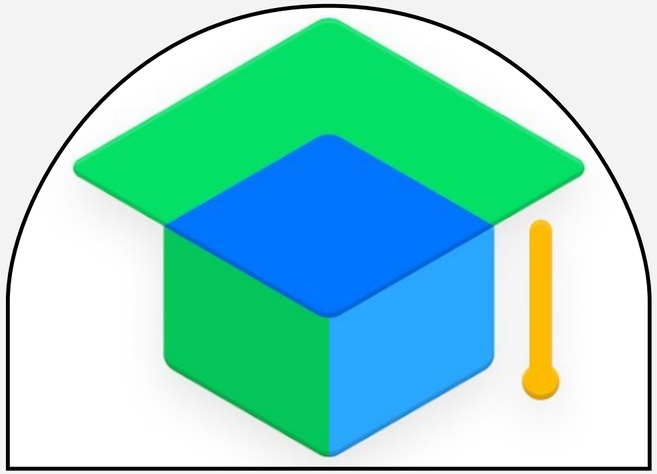
Flutter?

K DEVCON

Flutter?

Session No.	시작/종료시간	발표주제	발표자
Opening	12:30-12:40	키노트	다니엘 Flutter Incheon, @에어프레미아 랩스
Session 01	12:40-13:20	FFI로 연결하는 고(Go)와 플러터	리핵 Flutter/Dart GDE, Flutter/GDG Montreal, @Unity
	5분	쉬는시간	
Session 02	13:25-13:55	Web RTC	이정주 Flutter/GDG Incheon, @(주)퍼나니
	5분	쉬는시간	
Session 03	14:00-14:40	쇼어버드 작동 방식	Eric Seidel Founder of @Shorebird, ex-Co-Founder of Flutter Team
	5분	쉬는시간	
Session 04	14:45-15:25	풍성한 디자인 요청사항에 대응하기	박유진 ABC Studio @LINE
	10분	쉬는시간	
Session 05	15:35-16:05	플러터 렌더링 해부학	에이든 Flutter Seoul
Session 06	16:10-16:50	Flutter Web을 활용해 제품 개발 환경 개선하기	김종식 ABC Studio @LINE
	5분	쉬는시간	
Session 07	16:55-17:35	어느날 갑자기 앱이 터졌다	신원규 Flutter Seoul, @설로인
	10분	쉬는시간	
Session 08	17:45-18:25	Flutter Bloc 을 제품개발에 야무지게 적용하기	최정연 ABC Studio @LINE
	5분	쉬는시간	
Session 09	18:30-19:10	Empathetic Flutter	Craig Labenz & 이자영 Flutter Team @Google
	5분	쉬는시간	
Session 10	19:15-19:55	webOS 소개 및 Flutter 적용 계획	이동영 TV System App Team @LG

Flutter?

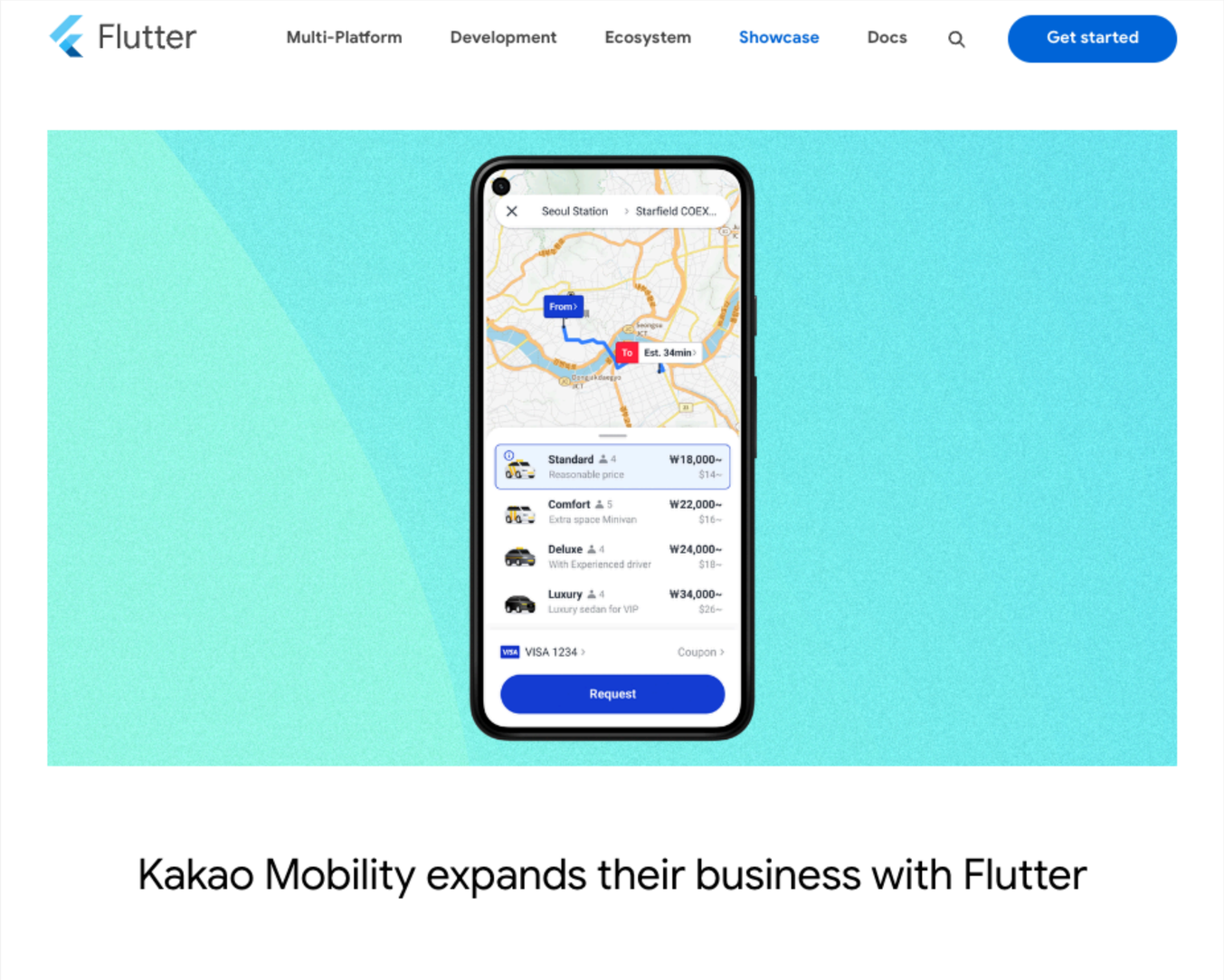


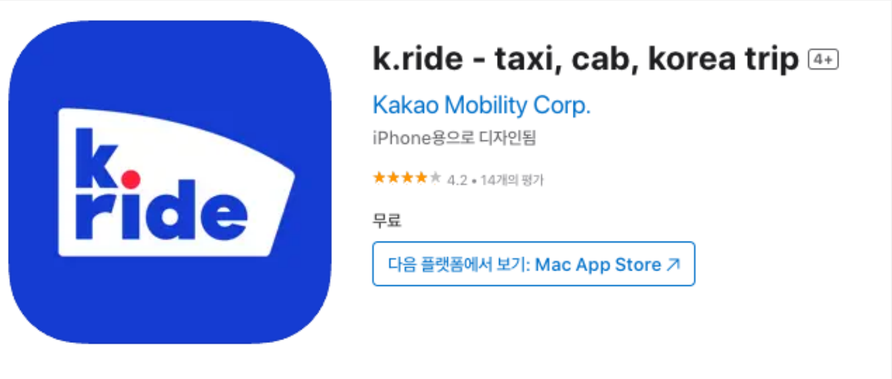
네이버 지식인 엑스퍼트

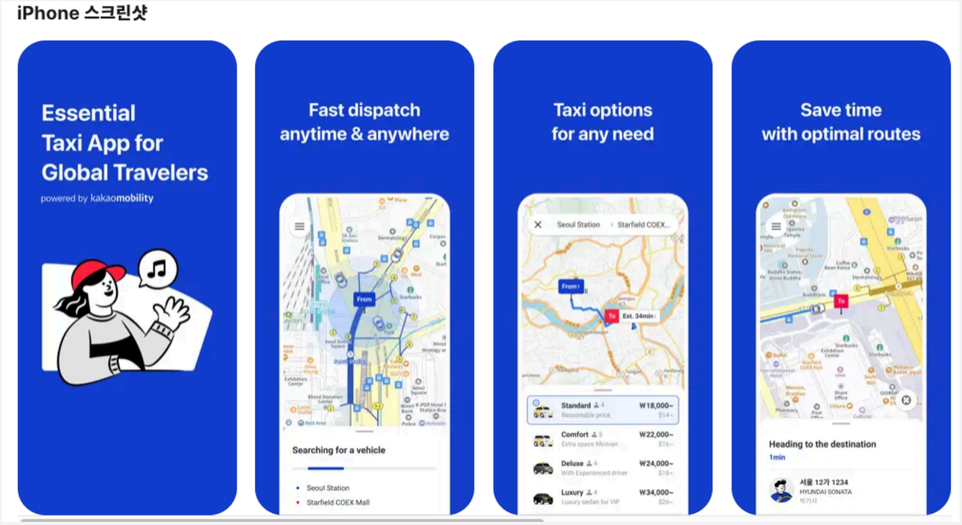
네이버 블로그

중고나라

Flutter?


Flutter
Multi-Platform
Development
Ecosystem
Showcase
Docs
Get started
Seoul Station
Starfield COEX...
From
To
Est. 34min
Standard
Reasonable price
₩18,000~
$14~
Comfort
Extra space Minivan
₩22,000~
$16~
Deluxe
With Experienced driver
₩24,000~
$18~
Luxury
Luxury sedan for VIP
₩34,000~
$26~
VISA 1234
Coupon
Request
Kakao Mobility expands their business with Flutter

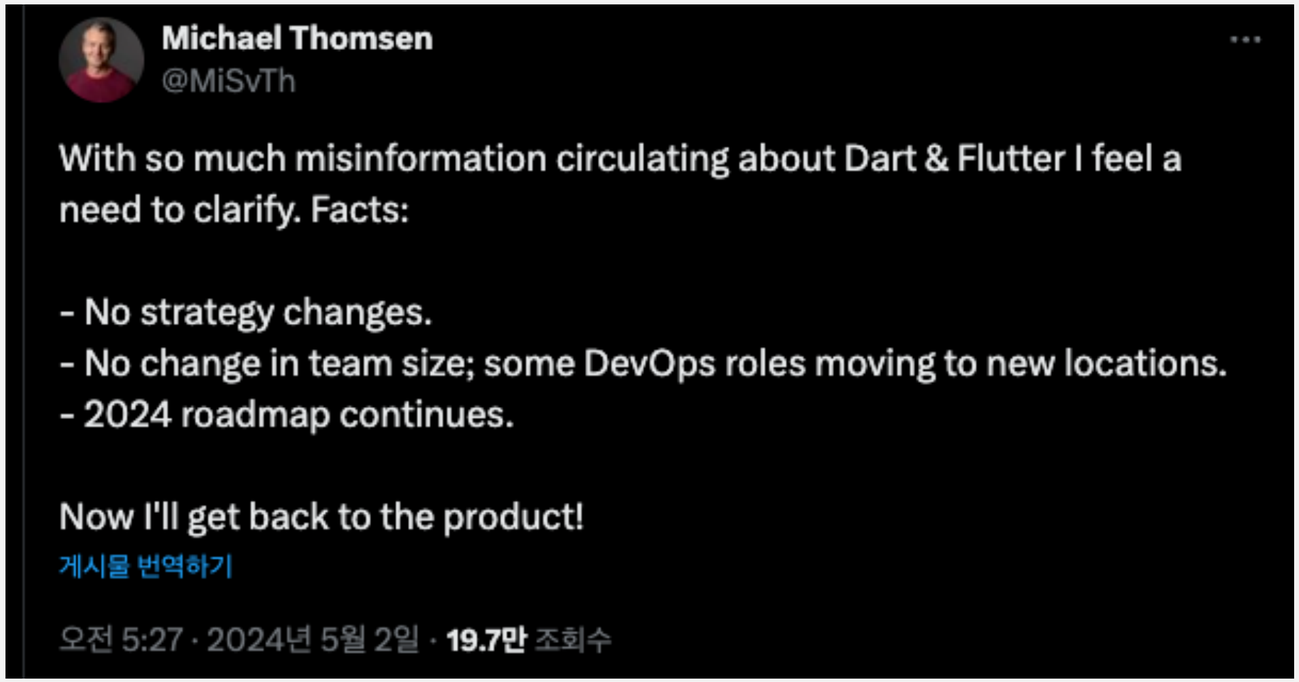

k.ride - taxi, cab, korea trip
4+
Kakao Mobility Corp.
iPhone용으로 디자인됨
4.2 • 14개의 평가
무료
다음 플랫폼에서 보기: Mac App Store


iPhone 스크린샷
Essential Taxi App for Global Travelers
powered by kakaomobility
Fast dispatch anytime & anywhere
Searching for a vehicle
Taxi options for any need
Save time with optimal routes
Heading to the destination

Flutter?

Flutter news
Coding Chef
구글이 플러터 개발자를
해고하고 플러터를 폐기한다고?
5:34
Google
구글 Flutter 직원들 해고
플루터 개발 그만둬야 하나?
14:37
Lay-Offs
Flutter 팀 해고
플러터의 미래는?
7:01

Flutter?



팩트:

- 전략 변경 없음.
- 팀 규모 변경 없음; 일부 DevOps 역할이 새로운 위치로 이동.
- 2024 로드맵 계속 됨.

https://x.com/MiSvTh/status/1785767966815985893

Flutter?

K DEVCON

Flutter Meetup Organizer Call

Thursday Oct 10, 2024 · 4pm – 5pm (Pacific Time - Los Angeles)

Hey party people 👋

This month's guest speaker is, Flutter Product Manager, Kevin Moore ✨

대충 플러터 네트워크 소식과
플러터 관계자(매니저, PM, GDE)와 질문타임을 가지거나
플러터의 향방에 대한 이야기를 합니다🔥

Q. State가 뭐예요?

Stateless / Stateful

위젯의 상태가 변경되지않고 정적이다

Stateless / Stateful

위젯의 상태가 변경되지않고 정적이다

Stateless

Stateless / Stateful

데이터가 실시간으로 변경되면 위젯이 UI를 동적으로 업데이트를 한다

Stateless / Stateful

데이터가 실시간으로 변경되면 위젯이 UI를 동적으로 업데이트를 한다

Stateful

Stateless / Stateful

Stateless widget안에 Stateful widget을 넣으면 Stateful widget안의 상태 값은 변경되지않는다.

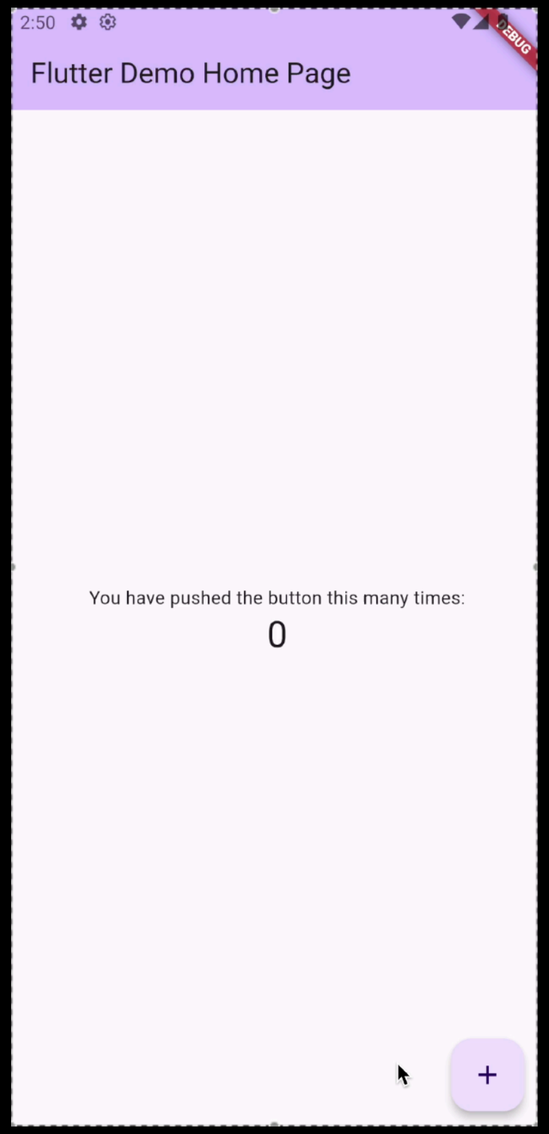
Stateless / Stateful

Stateless widget안에 Stateful widget을 넣으면 Stateful widget안의 상태 값은 변경되지않는다.

변경된다

State가 뭐예요?

```
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}
```



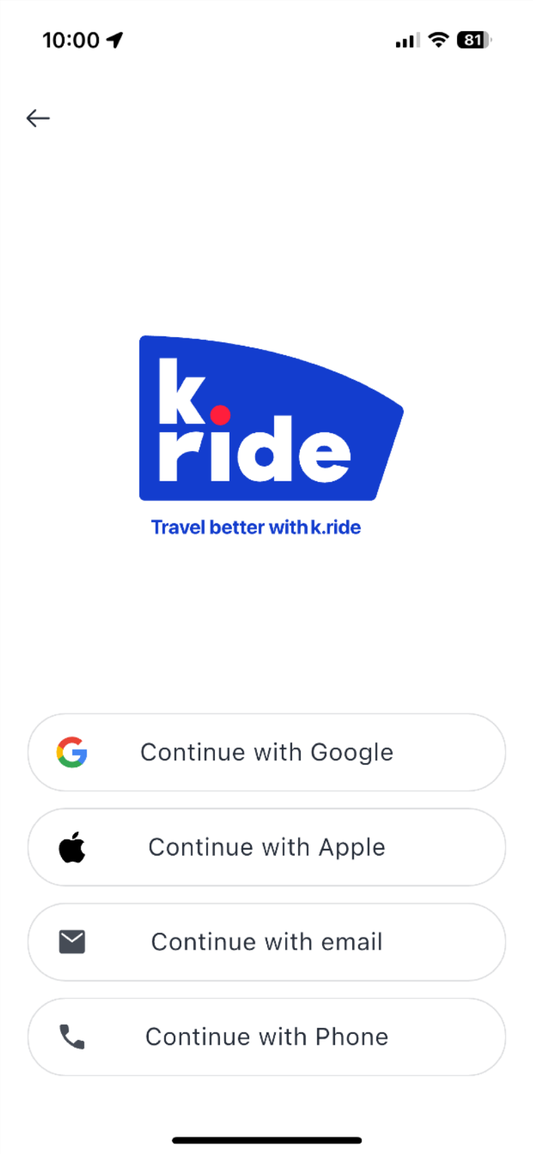
State

- Stateless widget
- Stateful widget
- 상태는 시간이 지남에 따라 변경되는 데이터
- UI의 모양에 영향을 줌
- 위젯 상태가 변경되면 Flutter는 새 상태를 반영하기 위해 위젯을 다시 그림

State가 뭐예요?

Stateless

- 상태(내용)가 변경되지 않음
- 이는 위젯의 수명 동안 변경되지 않는 정적 콘텐츠를 표시하는 데 주로 사용
- 내부 상태가 없으며 텍스트, 이미지, 아이콘 및 사용자 상호 작용에 변경되거나 응답할 필요가 없는 기타 UI 요소와 같은 정보를 표시하는 데 자주 사용
- 예 : 정적 요소인 Text, Icon, Image, 등에 사용



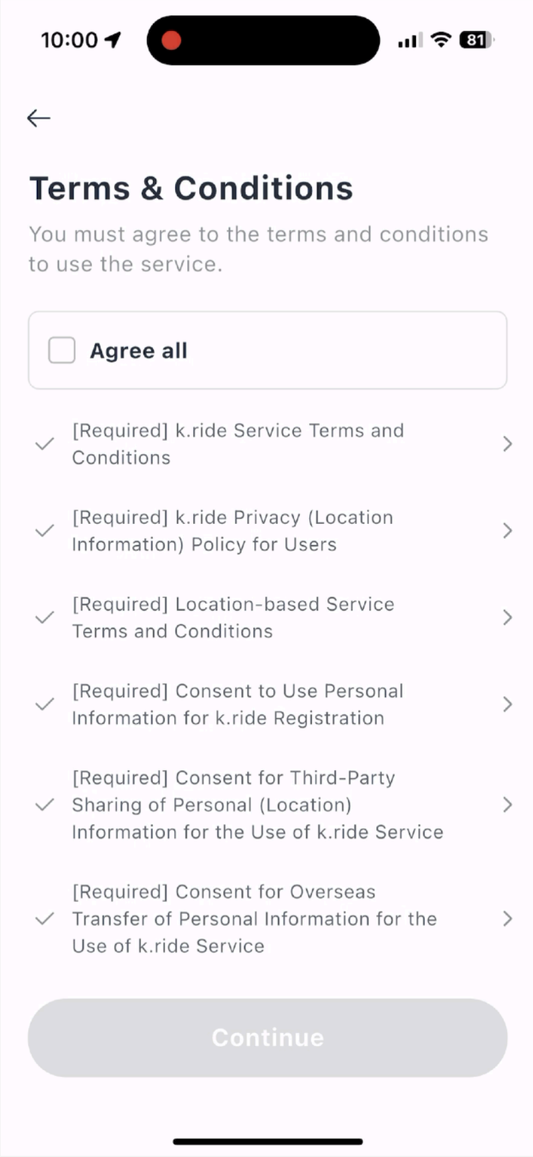
Stateless

```
class MyTextWidget extends StatelessWidget {
  final String text;

  MyTextWidget(this.text);

  @override
  Widget build(BuildContext context) {
    return Text(text);
  }
}
```

State가 뭐예요?

Stateful

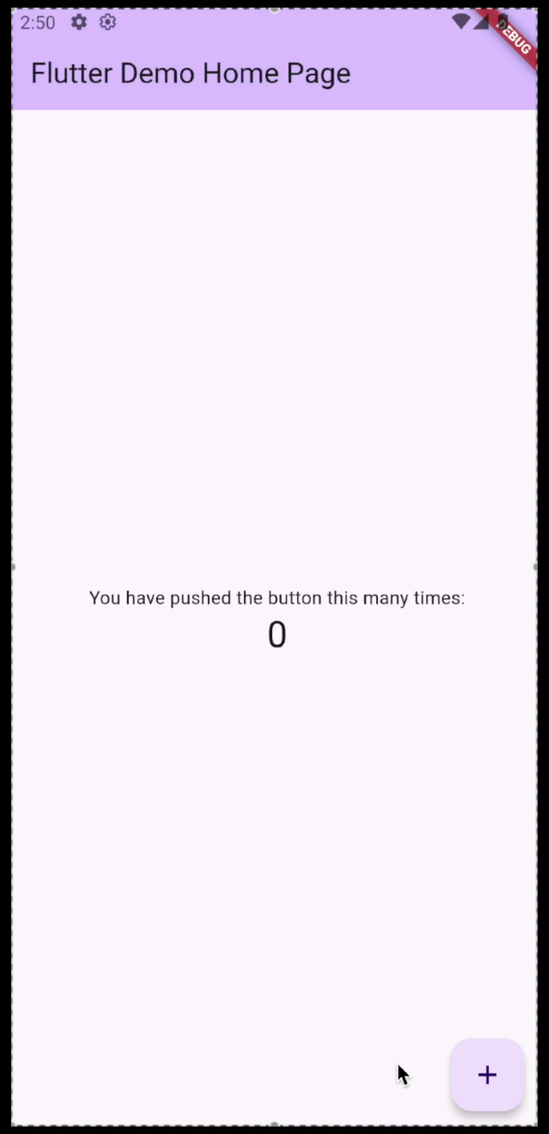
- 변경 가능하며 수명 동안 내부 상태를 변경가능
- 사용자 상호작용이나 기타 이벤트에 대응하여 위젯의 상태를 관리하고 업데이트해야 할 때 사용
- 일반적으로 동적 및 대화형 UI 구성 요소를 구축하는 데 사용
- 수정이 가능하고 위젯을 다시 빌드하는 작업을 트리거하는 해당 변경 가능 상태 객체(일반적으로 를 확장)를 가짐
- 예 : TextField, Checkbox, ListView, 등.



State가 뭐예요?

Stateful

```
class CounterWidget extends StatefulWidget {
  @override
  _CounterWidgetState createState() => _CounterWidgetState();
}

class _CounterWidgetState extends State<CounterWidget> {
  int count = 0;

  void incrementCounter() {
    setState(() {
      count++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
        Text('Count: $count'),
        ElevatedButton(
          onPressed: incrementCounter,
          child: Text('Increment'),
        ),
      ],
```



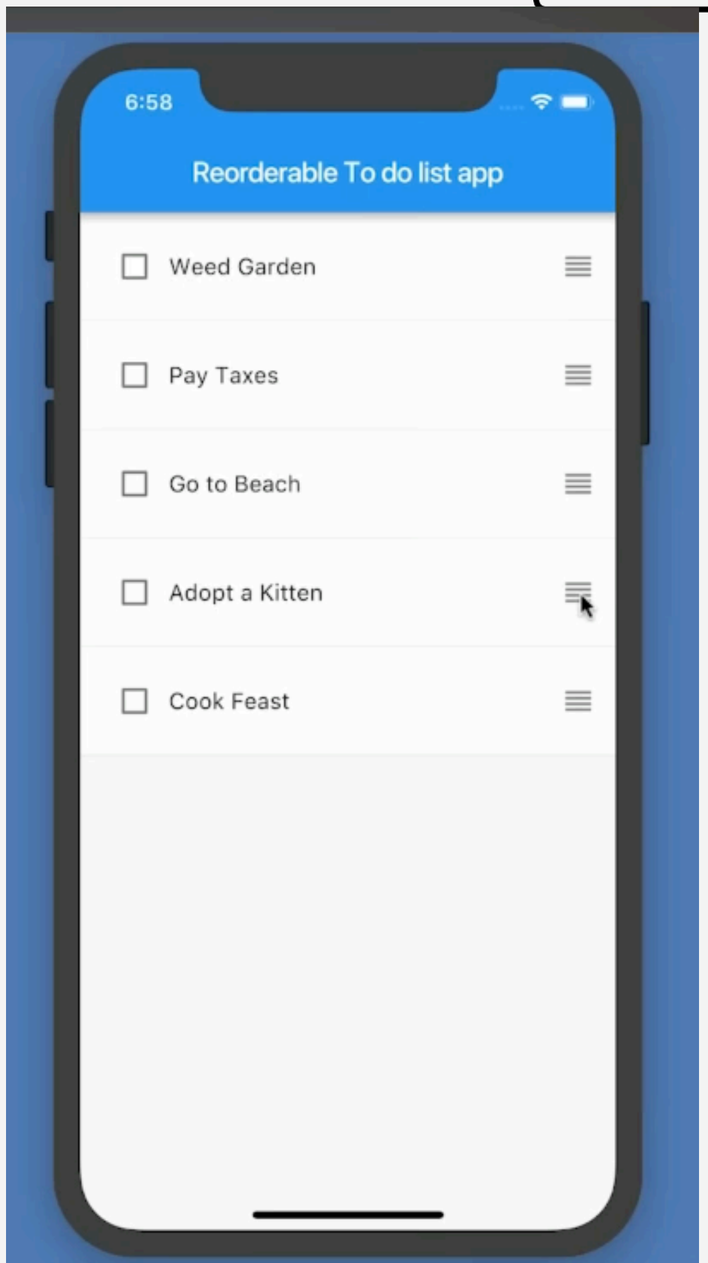
Stateless / Stateful

속성	Stateless 위젯	Stateful 위젯
가변 상태	값이 변하지 않음	값이 변할 수 있음
동적 동작	외부 이벤트에 반응하지 않는 정적 동작	사용자 상호작용이나 데이터 변경에 따라 동작 변화 가능
재빌드 트리거	상태 변경 시 재빌드를 트리거하지 않음	상태를 반영하기 위해 재빌드를 트리거함
사용 사례	텍스트, 아이콘, 이미지 등 정적 콘텐츠 표시	동적 UI 요소 처리, 인터랙티브 컴포넌트
성능	매우 효율적이고 가벼움	상태 관리로 인해 약간의 오버헤드가 있을 수 있음
코드 복잡도	간단하고 최소한의 보일러플레이트	상태를 관리하고 상태 변경 로직 구현 필요
사용 유연성	자주 업데이트가 필요 없는 컴포넌트에 적합	자주 변하는 컴포넌트에 필수적
예시 위젯	텍스트, 아이콘, 이미지	체크박스, 텍스트필드, 슬라이더

Q. Key? Globalkey? 란 무엇인지? 그리고 어떻게 사용하는지

Key

- UI가 변경될 때 Flutter가 위젯을 구별하기 위해 사용하는 식별자
- 동일한 유형의 위젯들을 추가, 제거, 또는 재정렬할 때 사용
- 예를 들어, 할 일 목록(To-do list) 앱에서 항목의 우선순위를 바꾸거나 완료된 항목을 제거할 때 키를 사용



Key

- ValueKey: 간단하고 고유한 데이터(예: 텍스트)가 있을 때 사용.
- ObjectKey: 복잡한 데이터 조합이 있을 때 사용.
- UniqueKey: 각 위젯이 서로 고유해야 할 때 사용. 하지만 랜덤 숫자를 키로 사용하는 것은 바람직하지 않음.
- PageStorageKey: 사용자의 스크롤 위치를 저장하기 위한 특수한 키.
- GlobalKey: 상태를 잃지 않고 부모 위젯이 바뀌는 경우나, 다른 부분의 위젯 상태에 접근해야 할 때 사용. 앱 전체에서 위젯 상태에 액세스 가능

ValueKey 예

```
class ToDoList extends StatefulWidget {
  @override
  _ToDoListState createState() => _ToDoListState();
}

class _ToDoListState extends State<ToDoList> {
  List<String> tasks = ['Buy groceries', 'Walk the dog', 'Do laundry'];

  @override
  Widget build(BuildContext context) {
    return ReorderableListView(
      children: tasks
          .map((task) => ListTile(
                key: ValueKey(task),
                title: Text(task),
              ))
          .toList(),
      onReorder: (oldIndex, newIndex) {
        setState(() {
          if (newIndex > oldIndex) newIndex -= 1;
          final task = tasks.removeAt(oldIndex);
          tasks.insert(newIndex, task);
        });
      },
```

Key

ObjectKey 예

```
class Contact {
  final String id;
  final String name;
  final String phone;

  Contact(this.id, this.name, this.phone);
}

class ContactList extends StatelessWidget {
  final List<Contact> contacts = [
    Contact('1', 'Alice', '123-456-7890'),
    Contact('2', 'Bob', '987-654-3210')
  ];

  @override
  Widget build(BuildContext context) {
    return ListView(
      children: contacts
          .map((contact) => ListTile(
                key: ObjectKey(contact),
                title: Text(contact.name),
                subtitle: Text(contact.phone),
              ))
          .toList(),
```

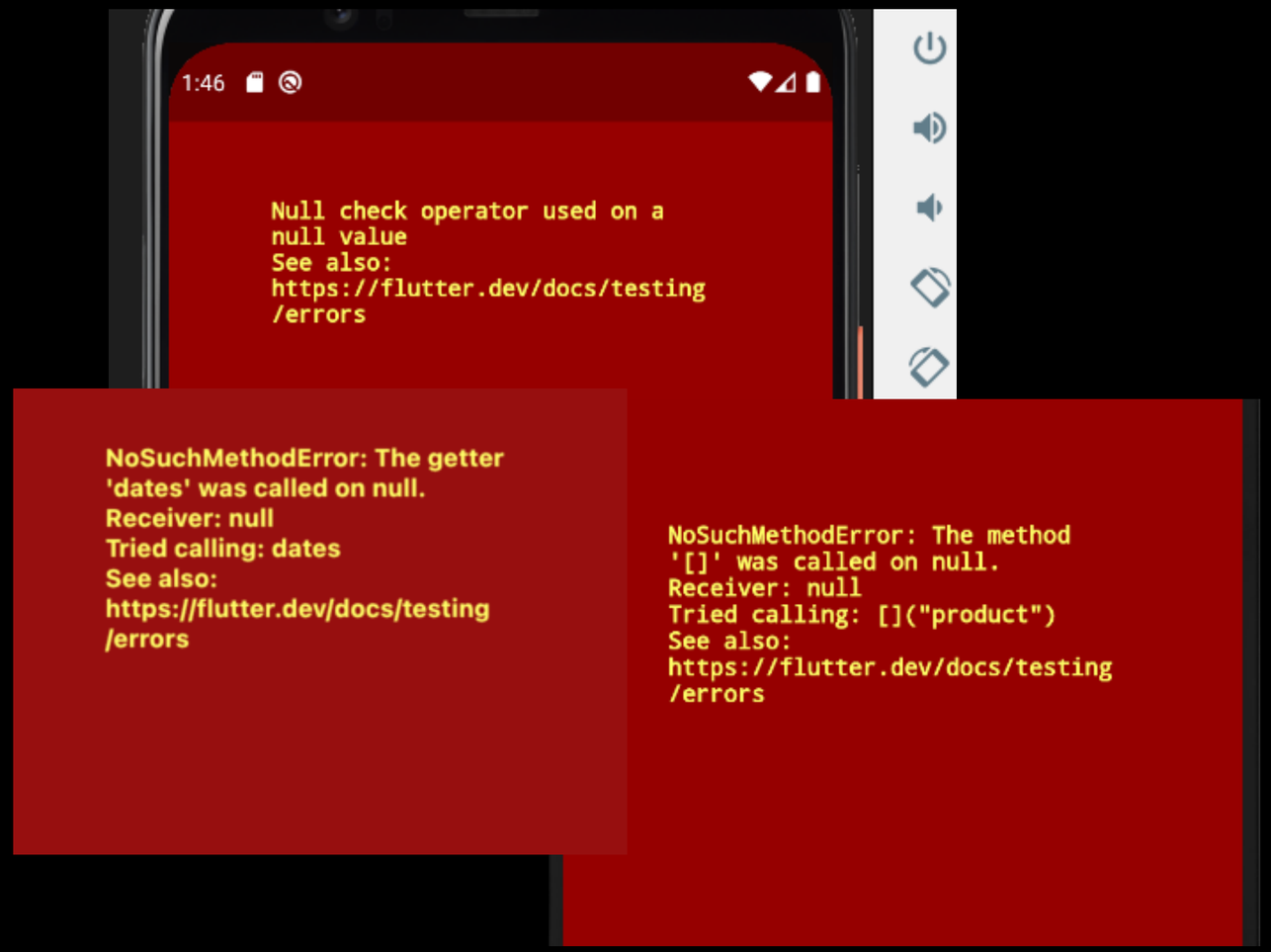
Q. Flutter 를 하면서 제일 많이 만났던 게 null safety 에러였는데... 정확한 사용법이 궁금해요 ㅠㅠ

Null

Dart에서 null은 변수가 선언되었지만, 값이 할당되지 않은 경우
값/데이터가 없음을 의미

null safety 예

```
// null 될 수 없는 변수
var i = 42;
String name = getFileName();
final b = Foo();
```



null safety

nullable 변수 처리 방법

```
// 변수의 값이 null일 수 있음을 나타내려면 형식 선언에 ?를 추가
int? aNullableInt = null;

Color? favoriteColor;

void setBackground(Color? color){
    //…
}

List<Color?> favorites;
```

null safety

nullable 변수 처리 방법

```
// 변수의 값이 null일 수 있음을 나타내려면 형식 선언에 ?를 추가
int? aNullableInt = null;
Color? favoriteColor;
void setBackground(Color? color){
    //…
}
List<Color?> favorites;
```

```
// ??를 사용해 null일 때 대체 값을 제공하는 데 사용
String color = favoriteColor ?? 'Blue';

int? age;
int actualAge = age ?? 18;

List<Color> safeColors = favorites ?? [];

Map<String, int>? scores;

Map<String, int> safeScores = scores ?? {'Alice': 10, 'Bob': 20};
```

null safety

nullable 변수 처리 방법

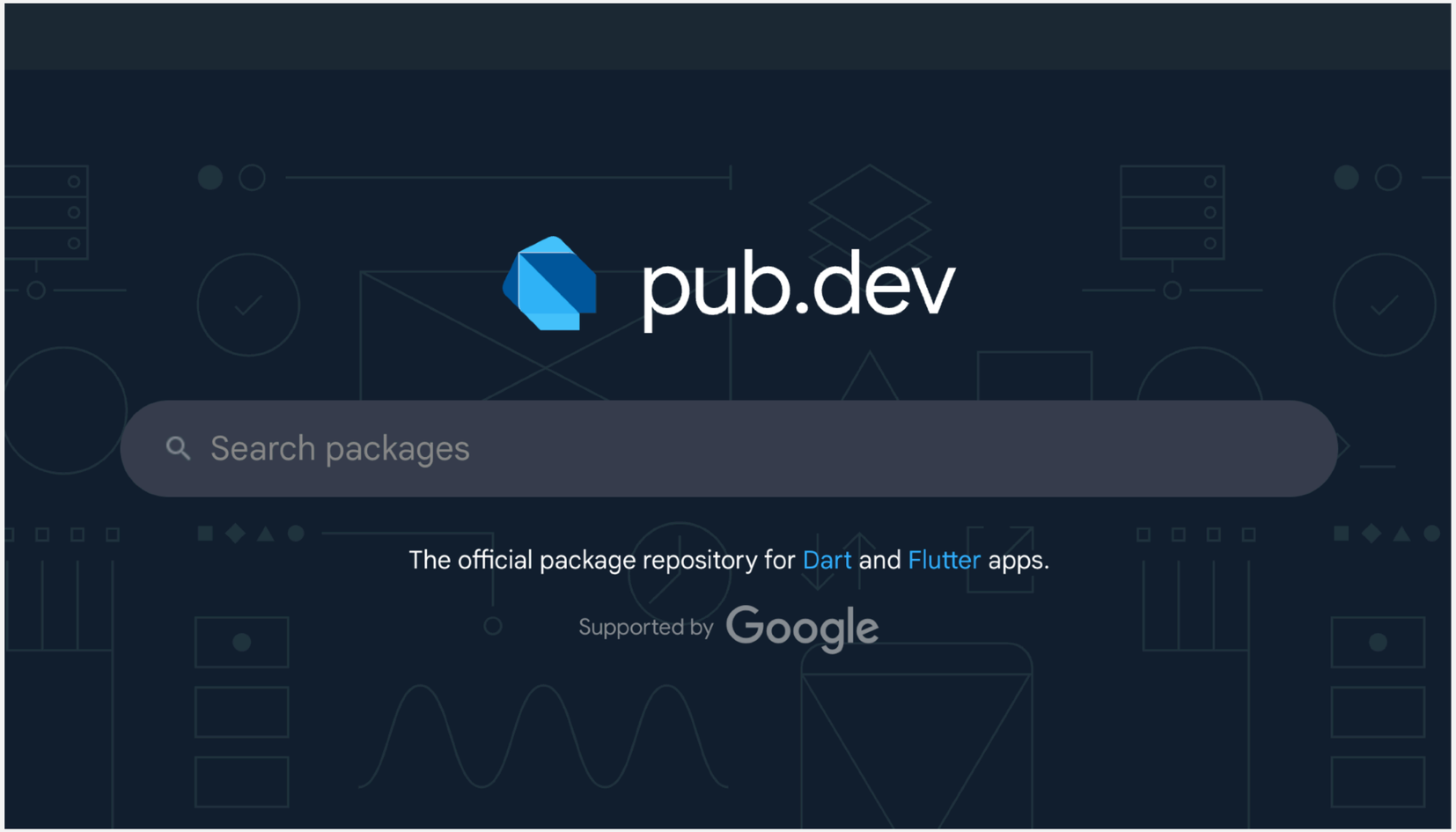
```
// 변수의 값이 null일 수 있음을 나타내려면 형식 선언에 ?를 추가
int? aNullableInt = null;
Color? favoriteColor;
void setBackground(Color? color){
    //…
}
List<Color?> favorites;

// ??를 사용해 null일 때 대체 값을 제공하는 데 사용
String color = favoriteColor ?? 'Blue';
int? age;
int actualAge = age ?? 18;
List<Color> safeColors = favorites ?? [];
Map<String, int>? scores;
Map<String, int> safeScores = scores ?? {'Alice': 10, 'Bob': 20};
```

```
// !을 사용해 null이 아님을 명시
String? name;
name = "John";
String nonNullableName = name!;

String? getCity() {
  return "New York";
}
String city = getCity()!;

List<int>? values = [1, 2, 3];
List<int> nonNullableValues = values!;
```

Q. 플러터는 많은 라이브러리(패키지)와 함께 사용하는 것으로 유명한데요, 비슷한 라이브러리도 많은데 그 중에 나와 맞는 것을 잘 고르는 팁(?)

library

Pub.dev - flutter 공식 라이브러리 저장소



library

Pub.dev - flutter 공식 라이브러리 저장소

Flutter
공식 커뮤니티가
선정하고,
좋아하는
라이브러리 소개

Flutter Favorites

Some of the packages that demonstrate the highest levels of quality, selected by the Flutter Ecosystem Committee

font_awesome_flutter
The Font Awesome Icon pack available as Flutter Icons. Provides 2000 additional icons
fluttercommunity.dev

redux
Redux is a predictable state container for Dart and Flutter apps
fluttercommunity.dev

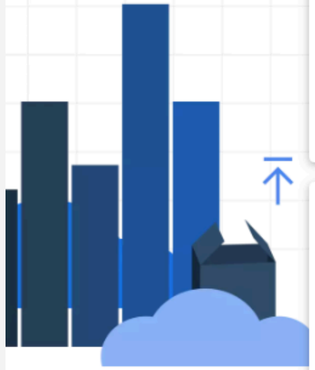
flutter_svg
An SVG rendering and widget library for Flutter, which allows painting and displaying
dnfield.dev

macos_ui
Flutter widgets and themes implementing the current macOS design language.
macosui.dev

VIEW ALL

Most popular packages

Some of the most downloaded packages over the past 60 days



share_plus
Flutter plugin for sharing content via the platform share UI, using the ACTION_SEND
fluttercommunity.dev

geolocator
Geolocation plugin for Flutter. This plugin provides a cross-platform (iOS, Android) API for
baseflow.com

sse
Provides client and server functionality for setting up bi-directional communication
tools.dart.dev

audioplayers
A Flutter plugin to play multiple audio files simultaneously
blue-fire.xyz

photo_view
Photo View provides a gesture sensitive zoomable widget. Photo View is largely used to
blue-fire.xyz

yaml_edit
A library for YAML manipulation with comment and whitespace preservation.
tools.dart.dev

60일동안
가장 많이 다운로드 된
라이브러리

library

Pub.dev - flutter 공식 라이브러리 저장소

flutter_naver_map 1.3.0+1

Published 16 days ago • note11.dev Dart 3 compatible

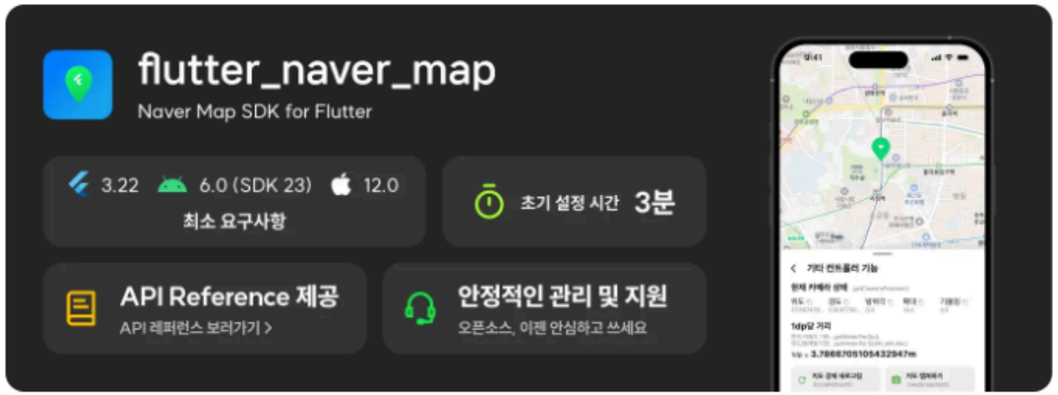
SDK FLUTTER PLATFORM ANDROID IOS

101

Readme Changelog Example Installing Versions Scores

flutter_naver_map

pub v1.3.0+1 pub v1.3.0+1 Stars 139



가이드 문서 바로가기 > Github Issue 바로가기 > 이슈 트래커 바로가기 >

101 LIKES | 160 PUB POINTS | 93% POPULARITY

Publisher

note11.dev

Metadata

Naver Map plugin for Flutter, which provides map service of Korea.

Repository (GitHub)

View/report issues

Documentation

Documentation

API reference

License

BSD-3-Clause (license)

101 LIKES | 160 PUB POINTS | 93% POPULARITY

Publisher

note11.dev

Metadata

Naver Map plugin for Flutter, which provides map service of Korea.

Repository (GitHub)

View/report issues

Documentation

Documentation

API reference

License

BSD-3-Clause (license)

library

Pub.dev - flutter 공식 라이브러리 저장소

Readme Changelog Example Installing Versions Scores

101 LIKES

160 / 160 PUB POINTS

93% POPULARITY

We analyzed this package 4 days ago, and awarded it 160 pub points (of a possible 160):

✓	Follow Dart file conventions	30/30
✓	Provide documentation	20/20
×	Platform support	20/20
✓	Pass static analysis	50/50
✓	Support up-to-date dependencies	40/40

→ 다트 파일의 컨벤션을 따르세요

→ 예시 문서와 주석을 제공하세요

→ 다양한 플랫폼을 지원

→ 코드에 오류, 경고 등의 문제가 없을 때

→ 최신 Dart, Flutter SDK를 지원

library

Open /// 오픈소스와 함께
Source /// 가치를 만드는 사람들의
Conference /// 이야기

24.11.16(Sat) 13 ~ 18시, 인천스타트업파크 6층 커넥트홀

*세션 시간은 추후 변동될 수 있습니다

13:00

Keynote: 제목은 키노트로 하겠습니다 근데 이제 오픈소스를 곁들인

김인제 – LINE+ 오픈챗 서버개발자
오픈소스 멘토링 멘토

13:10

토스의 오픈소스 이야기: 이 중 하나는 터지겠지

박서진 – 토스 Head of Frontend

14:00

식탁보 프로젝트의 어제, 오늘, 그리고 내일

남정현 – 식탁보 프로젝트
메가존 클라우드 소프트웨어 엔지니어
마이크로소프트 MVP

14:50

오픈소스 웹브라우저 개발자는 이렇게 일한다

임의상 – Chromium Committer

15:40

오픈소스 메커니즘: 오픈소스 생태계 전반에 대해

엄재웅(skydoves)
– GetStream Lead Android developer
Android & Kotlin GDE

16:30

오픈소스와 함께 공부하기

김소연 – flutter_naver_map 메인테이너

17:20

주니어도 한다! 업무에선 쓰지 못하는 오픈소스 기여하며 성장하기

안상혁 – LINE+ 소프트웨어 엔지니어
카프카, 스프링 카프카, 아르메리아 컨트리뷰터

감사합니다.

조새벽

SaeByeok Cho

GDG Songdo orgernizer
Flutter Incheon orgernizer