

배달 앱(패캠마트)

2 Firebase 알아보기 Firebase Realtime Database

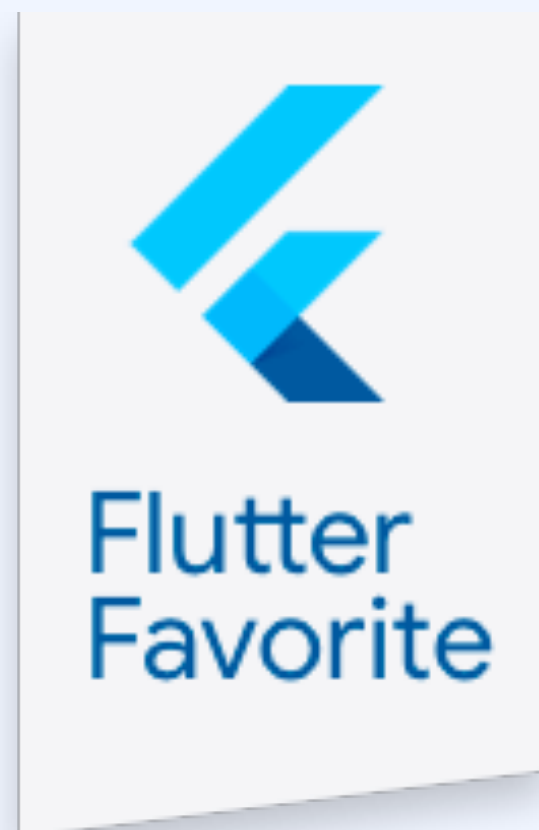
Firestore Realtime Database

- NoSQL
- 초기 Firestore Database 제품
- Realtime Database → Firestore
- JSON (JavaScript Object Notation) 형식의 데이터 저장 방식
- 실시간 데이터베이스
- 오프라인 지원

Firestore Realtime Database

설치 필요 패키지

1. https://pub.dev/packages/firebase_core
2. https://pub.dev/packages/firebase_database



firebase_database 10.2.3

Published 4 days ago • firebase.google.com Dart 3 compatible

SDK | FLUTTER | PLATFORM | ANDROID | IOS | MACOS | WEB

519

Readme | Changelog | Example | Installing | Versions | Scores

Flutter Database Plugin for Flutter

A Flutter plugin to use the [Firebase Database API](#).

To learn more about Firebase Database, please visit the [Firebase website](#)

pub v10.2.3

Getting Started

To get started with Firebase Database for Flutter, please [see the documentation](#).

Usage

To use this plugin, please visit the [Firebase Database Usage documentation](#)

Issues and feedback

Please file FlutterFire specific issues, bugs, or feature requests in our [issue tracker](#).

Plugin issues that are not specific to FlutterFire can be filed in the [Flutter issue tracker](#).

519 | 120 | 99%

LIKES | PUB POINTS | POPULARITY

Publisher

firebase.google.com

Metadata

Flutter plugin for Firebase Database, a cloud-hosted NoSQL database with realtime data syncing across Android and iOS clients, and offline access.

[Homepage](#)

[Repository \(GitHub\)](#)

[View/report issues](#)

[Contributing](#)

Documentation

[API reference](#)

License

BSD-3-Clause

Firestore Realtime Database

JSON 데이터 저장 형식

```
{
  "platform": {
    "android": {
      "name": 13,
      "sdk": { "version": 33 },
      "codeName" : "Tiramisu"
    },
    "android": {
      "name": 15,
      "sdk": { "version": 35 },
      "codeName" : "Vanilla Ice Cream"
    },
    "ios": { ... },
    "desktop": { ... }
  }
}
```

1. 데이터 중복을 최소화 하기
2. Depth를 최소화하기 == 평탄화 하기
3. 확장가능한 데이터 구조로 설계하기

Firestore Realtime Database

데이터 쓰기 - set()

주의: 데이터가 덮어쓰기 됩니다.

```
DatabaseReference ref = FirebaseDatabase.instance.ref("fast-campus/flutter");

await ref.set({
  "name": "Dreamwalker",
  "part": 5,
  "contents": {
    "firebase": "realtime-database"
  }
});
```

Firestore Database

데이터 쓰기 - push()

고유한 키를 생성해서 하위에 데이터를 둘때 사용

```
DatabaseReference parentRef = FirebaseDatabase.getInstance().ref("temp");  
DatabaseReference newChildRef = parentRef.push();  
newChildRef.set({"test": 10});
```

Firestore Realtime Database

데이터 업데이트 - update()

/// 방법 1

```
DatabaseReference ref = FirebaseDatabase.instance.ref("fast-campus/flutter");  
await ref.update({  
  "part": 8,  
});
```

/// 방법 2

```
DatabaseReference ref2 = FirebaseDatabase.instance.ref("users");  
await ref2.update({  
  "fast-campus/flutter/part": 19,  
  "fast-campus/flutter/contents/firebase": "host",  
});
```

Firestore Realtime Database

데이터 읽기 - get() (Future)

```
final ref = FirebaseDatabase.instance.ref();  
final snapshot = await ref.child('패스트캠퍼스/$flutterId').get();  
if (snapshot.exists) {  
    print(snapshot.value);  
} else {  
    print('No data available.');
```


Firestore Realtime Database

데이터 읽기 - onValue() (Stream)

```
DatabaseReference ref = FirebaseDatabase.instance.ref('flutterId');
ref.onValue.listen((DatabaseEvent event) {
    final data = event.snapshot.value;
    updateStarCount(data);
});
```

Firestore Realtime Database

데이터 읽기 - 이벤트 처리

이벤트	일반적인 용도
onChildAdded	항목 목록을 검색하거나 항목 목록에 대한 추가를 수신 대기합니다. 이 이벤트는 기존 하위 요소마다 한 번씩 트리거된 후 지정된 경로에 하위 요소가 새로 추가될 때마다 다시 트리거됩니다. 리스너에는 새 하위 요소의 데이터를 포함하는 스냅샷이 전달됩니다.
onChildChanged	목록의 항목에 대한 변경사항을 수신 대기합니다. 이 이벤트는 하위 노드가 수정될 때마다 트리거됩니다. 여기에는 하위 노드의 하위 요소에 대한 수정이 포함됩니다. 이벤트 리스너에 전달되는 스냅샷에는 하위 요소의 업데이트된 데이터가 포함됩니다.
onChildRemoved	목록의 항목 삭제를 수신 대기합니다. 이 이벤트는 바로 아래 하위 요소가 삭제될 때 트리거됩니다. 콜백 블록에 전달되는 스냅샷에는 삭제된 하위 요소의 데이터가 포함됩니다.
onChildMoved	순서가 지정된 목록의 항목 순서 변경사항을 수신 대기합니다. 현재의 정렬 메서드에 따라 항목 순서 변경 원인이 된 onChildMoved 이벤트가 항상 onChildChanged 이벤트를 뒤따릅니다.

출처: <https://firebase.google.com/docs/database/flutter/lists-of-data?hl=ko>

Firestore Realtime Database

데이터 읽기 - 데이터 정렬

방법	용도
orderByChild()	지정된 하위 키 또는 중첩된 하위 경로의 값에 따라 결과를 정렬합니다.
orderByKey()	하위 키에 따라 결과를 정렬합니다.
orderByValue()	하위 값에 따라 결과를 정렬합니다.

출처: <https://firebase.google.com/docs/database/flutter/lists-of-data?hl=ko>

Firestore Realtime Database

데이터 읽기 - 데이터 필터링

방법	용도
limitToFirst()	정렬된 결과 목록의 시작 부분부터 반환할 최대 항목 수를 설정합니다.
limitToLast()	정렬된 결과 목록의 맨 끝 부분부터 반환할 최대 항목 수를 설정합니다.
startAt()	선택한 정렬 기준 메서드에 따라 지정된 키 또는 값보다 크거나 같은 항목을 반환합니다.
startAfter()	선택한 정렬 기준 메서드에 따라 지정된 키 또는 값보다 큰 항목을 반환합니다.
endAt()	선택한 정렬 기준 메서드에 따라 지정된 키 또는 값보다 작거나 같은 항목을 반환합니다.
endBefore()	선택한 정렬 기준 메서드에 따라 지정된 키 또는 값보다 작은 항목을 반환합니다.
equalTo()	선택한 정렬 기준 메서드에 따라 지정된 키 또는 값과 동일한 항목을 반환합니다.

출처: <https://firebase.google.com/docs/database/flutter/lists-of-data?hl=ko>