

# 실시간 퀴즈앱

## 2 Firebase realtime database (실시간 데이터베이스) 설계

## Firestore Realtime Database 리뷰

- NoSQL
- JSON (JavaScript Object Notation) 형식의 데이터 저장 방식
- 실시간 데이터베이스
- 오프라인 지원

## Firestore Realtime Database 리뷰

### JSON 데이터 저장 형식

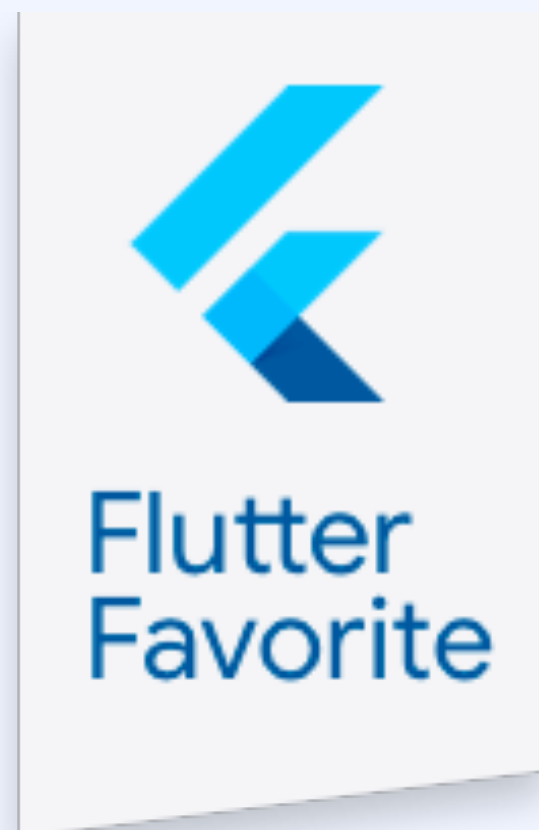
```
{
  "platform": {
    "android": {
      "name": 13,
      "sdk": { "version": 33 },
      "codeName" : "Tiramisu"
    },
    "android": {
      "name": 15,
      "sdk": { "version": 35 },
      "codeName" : "Vanilla Ice Cream"
    },
    "ios": { ... },
    "desktop": { ... }
  }
}
```

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

## Firestore Realtime Database 리뷰

### 설치 필요 패키지

1. [https://pub.dev/packages/firebase\\_core](https://pub.dev/packages/firebase_core)
2. [https://pub.dev/packages/firebase\\_database](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

Firestore Database Plugin for Flutter

A Flutter plugin to use the Firestore Database API.

To learn more about Firestore Database, please visit the Firestore website

pub v10.2.3

Getting Started

To get started with Firestore Database for Flutter, please see the documentation.

Usage

To use this plugin, please visit the Firestore 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 Firestore 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 리뷰

### 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 Realtime Database 리뷰

### 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 리뷰

### 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 리뷰

### 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 리뷰

### 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 설계

### 설계 조건

- 퀴즈 묶음에 문제내용을 저장해야함.
  - **1 quiz : N problem**
    - 1개의 문제에 N개의 문제 선택지와 정답을 저장해야함.
      - **1 problem : N option**
  - 퀴즈에 참가 가능한 **핀코드**를 생성하고 저장할 수 있어야함.
  - 퀴즈의 실시간 문제 **풀이 상태를 체크**해야함.
    - 퀴즈 시작과 종료 처리
  - 퀴즈에 **참가자 정보를 실시간으로 확인**해야함.
  - 퀴즈 **참가자의 문제 풀이 정보를 저장**할 수 있어야함.
    - 풀이 순서에 맞게 점수를 차등적으로 배분해야함.

## Firestore Realtime Database 설계

- 퀴즈
  - 경로: quiz
  - 키: 랜덤 생성
- 퀴즈 상세
  - 경로: quiz\_detail
  - 키: 랜덤 생성
- 퀴즈 상태
  - 경로: quiz\_state
  - 키: 퀴즈 상세의 키를 참조

## Firestore Realtime Database 설계

- 퀴즈
    - 경로: quiz
1. code: pin code = 참가자 입장 코드
  2. generateTime: 퀴즈 생성 시간
  3. quizDetailRef: 퀴즈 문제 묶음 참조 키
  4. Timestamp: 퀴즈 생성시간(Unix timestamp)
  5. uid : 퀴즈 문제 생성자

▼ -NZsSqXnYWywfp8r-q4m

code: "608989"

generateTime: "2023-07-09 11:24:57.047"

quizDetailRef: "-NZsSqWszZgV3fzD60qZ"

timestamp: 1688869497047

uid: "tgH0LjpaniDRqeYQhD1K7..."

## Firestore Realtime Database 설계

- 퀴즈 상세
  - 경로: quiz\_detail

1. code: pin code = 참가자 입장 코드

2. problems: 퀴즈 문제 목록

- answer: 정답명
- answerIndex: 정답인덱스
- options: 선택지 목록
- Title: 문제

```

    ▼ quiz_detail
      ▼ -NZsSqWszZgV3fzD60qZ
        code: "608989"
        ▼ problems
          ▼ 0
            answer: "2"
            answerIndex: 1
            ▼ options
              0: "1"
              1: "2"
              2: "3"
              3: "4"
            title: "플러터"
  
```

## Firestore Realtime Database 설계

- 퀴즈 상태
  - 경로: quiz\_state

1. Current: 현재 문제 풀이의 인덱스
2. Solve: 문제 별 풀이 결과 저장 **목록**
  1. Answer : 사용자가 선택한 정답
  2. Correct : 풀이 정답 여부
  3. Name: 닉네임
  4. Timestamp: 시간정보
  5. Uid: 사용자 uid
3. State: 퀴즈 시작 상태
4. Triggers : 문제별 시간 처리 **목록**
5. User: 참가자 정보 저장 **목록**

```

    quiz_state
      -NZsXkXFeay0MVC8DV24
        current: 1
        quizDetailRef: "-NZsXkXFeay0MVC8DV24"
        solve
          0
            -NZsY0L07-C9Yr7jqcFR
              answer: "2"
              correct: false
              name: "12345"
              timestamp: 1688870851993
              uid: "i5A0RkB2Bp9U1tgWBPS6U..."
              state: true
        triggers
          0
            end: 1688870856798
            start: 1688870851798
        user
          -NZsXvVEnpa8wZ1J-RUg
            name: "12345"
            uid: "i5A0RkB2Bp9U1tgWBPS6U..."
    
```