

Firestore와 Flutter 통합 개발

Firestore 기반 Flutter 애플리케이션 개발

학습목표

- Firebase 플랫폼의 기본 개념과 Flutter 통합 방법 이해
- Firebase 프로젝트 설정 및 개발 환경 구성 방법 습득
- Firestore 데이터베이스와 인증 기능 구현 능력 향상

Firestore 플랫폼 소개

- Firestore는 iOS, Android, 웹, Unity, Flutter를 네이티브로 지원하는 백엔드 플랫폼
- 데이터베이스, 인증, 스토리지, 테스트 등 다양한 제품군 제공
- Google I/O 2022 이후 Flutter를 1급 지원 플랫폼으로 채택하여 강화된 통합 지원
- Spark(무료)와 Blaze(유료) 두 가지 요금제 운영으로 유연한 사용 가능

Firestore 프로젝트 설정

```
# Firebase CLI 설치
npm install --location=global firebase firebase-tools

# Firebase 로그인
firebase login

# Firebase 프로젝트 초기화
firebase init
```

Firestore CLI 설치 설명

Firestore CLI는 Firestore 서비스를 로컬 환경에서 관리하기 위한 도구입니다.

주요 기능:

- Firestore 프로젝트 생성 및 관리
- Firestore 에뮬레이터 실행
- Firestore 호스팅 배포
- Firestore Functions 관리

Firestore 에뮬레이터 설정

```
# 에뮬레이터 시작
firebase emulators:start

# 특정 서비스만 시작
firebase emulators:start --only hosting
```

에뮬레이터 설정 설명

Firebase 에뮬레이터는 로컬 개발 환경에서 Firebase 서비스를 테스트할 수 있게 해주는 도구입니다.

필수 요구사항:

- Node.js 설치
- Java 설치
- Firebase CLI 설치
- Firebase 프로젝트 초기화 완료

Flutter Firebase 설정

```
# FlutterFire CLI 설치
dart pub global activate flutterfire_cli

# Firebase 구성
flutterfire configure --project=[PROJECT_ID]
```


Flutter Firebase 통합 설명

FlutterFire CLI는 Flutter 프로젝트와 Firebase를 연동하기 위한 도구입니다.

주요 기능:

- Firebase 설정 파일 자동 생성
- 플랫폼별 설정 자동화
- Firebase 서비스 초기화 코드 생성

데이터베이스 통합

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_core/firebase_core.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}
```

데이터베이스 통합 설명

Firestore는 Firebase의 NoSQL 데이터베이스입니다.

초기화 과정:

1. Firebase Core 초기화
2. Firestore 인스턴스 생성
3. 데이터베이스 연결 설정

요약

- Firebase는 Flutter 앱 개발을 위한 강력한 백엔드 플랫폼을 제공
- Firebase CLI와 FlutterFire CLI를 통해 간편한 프로젝트 설정 가능
- 에뮬레이터를 통한 로컬 테스트와 실제 서비스 배포 지원

END
