

# Gymbros: Firebase 백엔드 개발 가이드

firebase 사용할 것임

## 2. 백엔드에서 해야 할 작업 정리함

아래 단계대로 Firebase 프로젝트 세팅하고 핵심 기능 구현하면 됨

### 1단계: Firebase 프로젝트 생성 및 앱 연동

1. [Firebase 콘솔](#) 접속해서 새 프로젝트 생성함
2. 필수 서비스 활성화함
  - Authentication: 왼쪽 메뉴에서 Authentication 이동 → 시작하기 누름 → 이메일/비밀번호 로그인 방식 활성화함
  - Firestore Database: Firestore Database 이동 → 데이터베이스 만들기 클릭함. 처음엔 테스트 모드로 시작하면 보안 규칙 신경 안 써도 됨(나중에 프로덕션 모드로 바꿔야 함)
3. Flutter 앱에 Firebase 연동 필요함. 프로젝트 설정에서 Android, iOS 앱 추가해야 하고, 생성된 설정 파일은 프론트엔드에게 전달해야 함(자세한 건 3번 항목 참고)

### 2단계: Firestore 데이터 구조 설계 (가장 중요함)

Firestore는 데이터를 컬렉션/문서 단위로 저장하는 NoSQL DB임. 아래 구조 참고해서 설계하면 됨

- **users** 컬렉션: 사용자 정보 저장
  - 각 문서는 한 명의 사용자, 문서 ID는 Firebase Authentication에서 제공하는 UID 사용
  - 경로: `users/{user_uid}`
  - 데이터 예시:

```
{
  "email": "user@example.com",
  "username": "Gymbros",
  "goals": ["Muscle Gain", "Weight Loss"],
  "gender": "Male",
  "age": 28,
  "height": 175.5,
  "weight": 70.0,
  "experienceLevel": "Intermediate"
}
```

- **workouts** 컬렉션: 운동 종류 및 정보 저장
  - 경로: `workouts/{workout_id}`
  - 데이터 예시:

```
{
  "name": "Bench Press",
  "muscleGroup": "Chest",
  "difficulty": "Intermediate",
  "description": "Lying on a bench, lower a barbell to the chest, then press it back up.",
  "videoUrl": "https://youtube.com/..."
}
```

- **logs** 서브컬렉션: 특정 사용자의 운동 기록 저장
  - users 문서 하위 컬렉션으로 생성하면 관리 편함
  - 경로: `users/{user_uid}/logs/{log_id}`
  - 데이터 예시:

```
{
  "workoutId": "workout_id_for_bench_press",
  "workoutName": "Bench Press",
  "date": "2025-07-17T10:00:00Z",
  "sets": 5,
  "reps": 10,
  "weight": 60
}
```

### 3단계: Firestore 보안 규칙 설정 (데이터 보호)

이 단계가 앱 보안에 가장 중요함. 보안 규칙은 인증 안 된 사용자가 데이터 읽거나 수정 못 하게 막아줌

- Firestore Database > 규칙 탭에서 설정함
- 핵심 규칙:
  1. 사용자는 자신의 user 정보만 읽고 쓸 수 있음
  2. 모든 사용자는 workouts 컬렉션의 운동 정보만 읽을 수 있음(쓰기 불가, 관리자만 가능)
  3. 사용자는 자신의 logs만 생성/읽기/수정/삭제 가능
- 규칙 예시:

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /users/{userId} {
      allow read, write: if request.auth.uid == userId;

      match /logs/{logId} {
        allow read, write, create, delete: if request.auth.uid == userId;
      }
    }
  }
}
```

```
match /workouts/{workoutId} {  
  allow read: if request.auth != null;  
  allow write: if false;  
}  
}  
}
```

## 4단계: 운동 추천 로직 개발 (Cloud Functions)

운동 추천 로직은 Firebase Cloud Functions로 구현함

- HTTP 요청으로 호출되는 함수 하나 만들
- 함수 호출 시 해당 사용자의 프로필(goals, experienceLevel 등) Firestore에서 읽어옴. workouts 컬렉션의 운동들과 비교해서 적합한 운동 목록 조합해서 반환함
- 추천 기준은 논의해서 결정하면 됨

---

## 3. 프론트엔드에게 전달해야 할 정보

아래 정보 전달해야 Flutter 앱 개발 원활하게 진행 가능함

### 1. Firebase 연동 설정 파일

- Android: `google-services.json`
- iOS: `GoogleService-Info.plist`
- (Firebase 프로젝트에서 앱 추가하면 다운로드 가능)

### 2. 확정된 Firestore 데이터 구조

- 데이터 구조 변경 시, 최종 컬렉션/필드(데이터) 이름 전달 필요. 앱에서 동일한 이름으로 데이터 요청해야 함

### 3. 운동 추천 Cloud Function의 URL

- 운동 추천 함수 개발/배포 후, 호출 가능한 고유 URL 주소 전달 필요

가이드 참고해서 개발 진행하면 됨.