

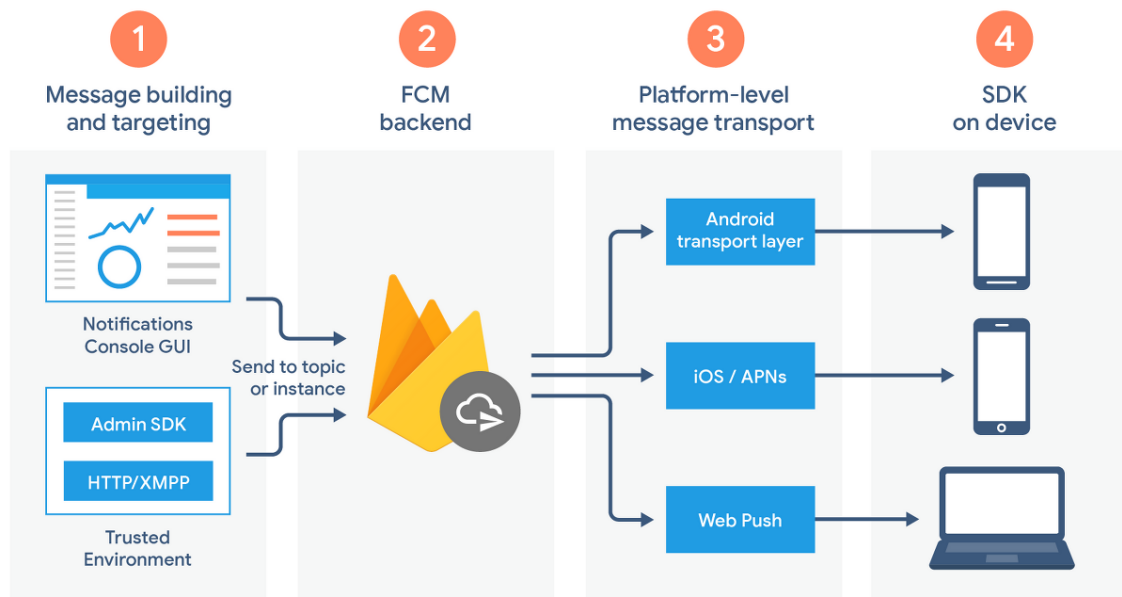
2. 외부 서비스 활용(Firebase)

이 프로젝트에서는 외부 서비스로 Firebase를 이용하여 알림기능을 구현했음

FCM(Firebase Cloud Messaging)이란?

- 메시지를 안정적으로 클라이언트 인스턴스에게 전송할 수 있는 교차 플랫폼 메시징 솔루션
- 무료로 제공되고 있기 때문에 부담없이 사용 가능

FCM 전송과정



- 애플리케이션 서버에서 전송된 메시지가 클라이언트에게 직접 전송되는 것이 아니라 중간에 FCM backend서버를 거쳐서 클라이언트에게 전송됩니다

프로젝트에 FCM을 적용한 부분

- 안내데스크가 상담 요청을 하면 해당 안내데스크와 같은 지역에 속하는 상담사들에게 실시간으로 알림을 보내기 위해서 사용

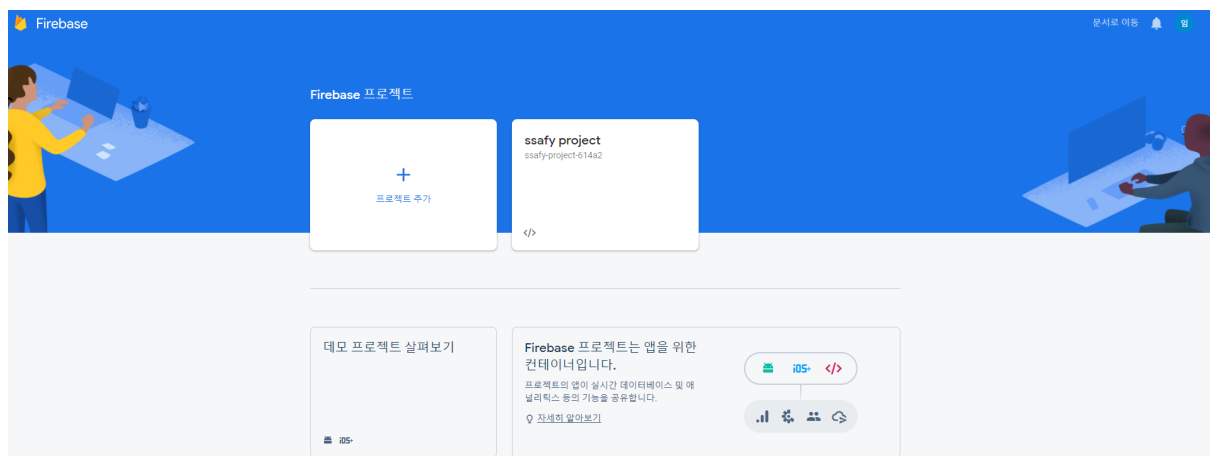
적용함으로써 얻은 이점

- 폴링(polling)방식과 같이 주기적으로 상담사(클라이언트)가 서버에 요청을 통해 새로운 알림이 들어왔는지 확인하지 않고 FCM이 클라이언트에게 전송될때만 클라이언트가 서버에 요청을 하여 기존 방식보다 서버의 부담이 감소되는 이점이 생김

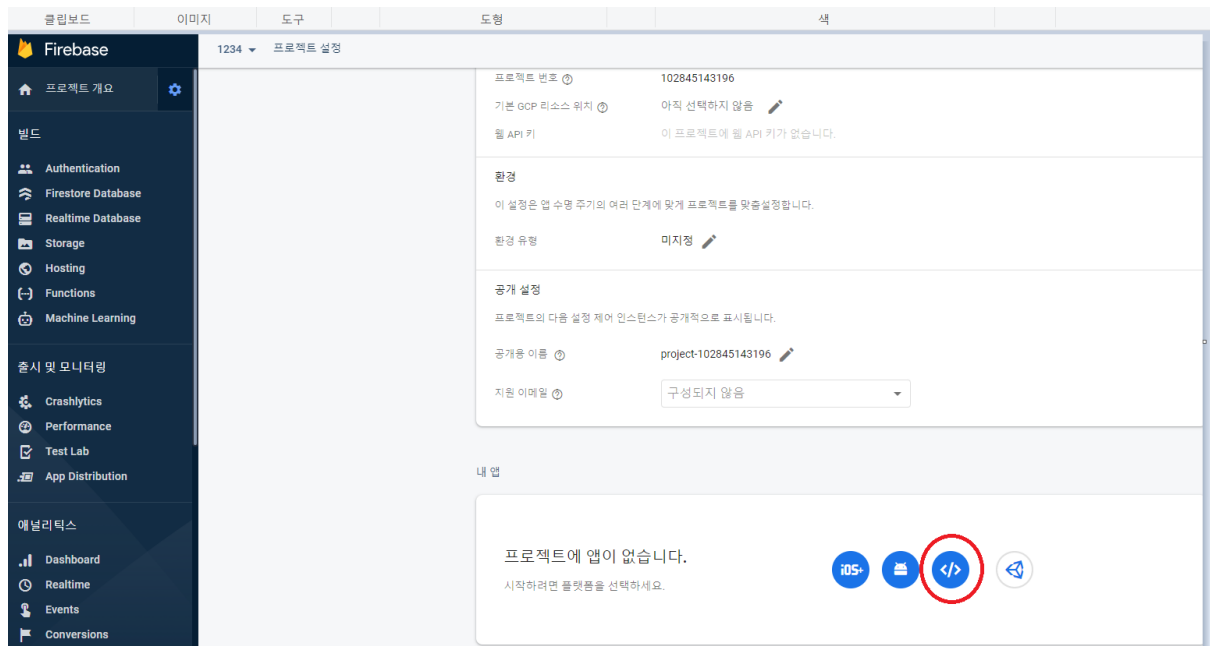
프로젝트에 사용을 위한 세팅

1. 프로젝트 추가

- <https://console.firebase.google.com/?hl=ko> 사이트에 들어가서 로그인 후 **프로젝트 추가**

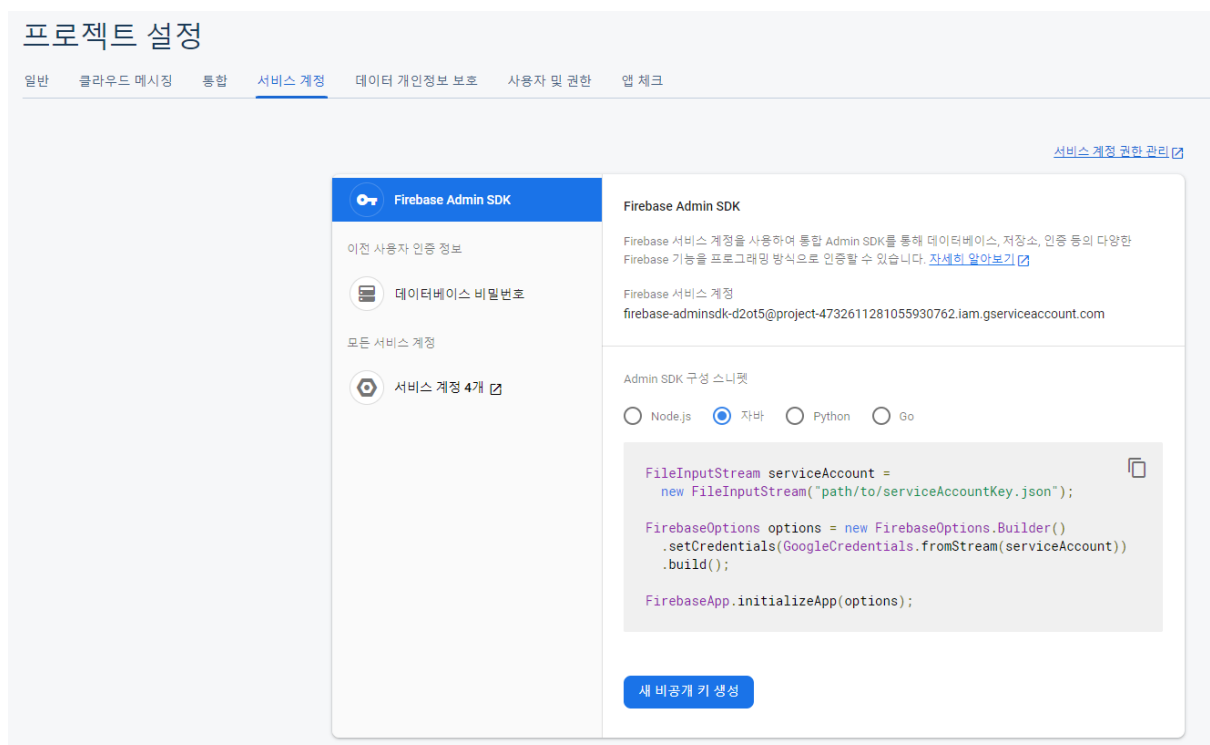


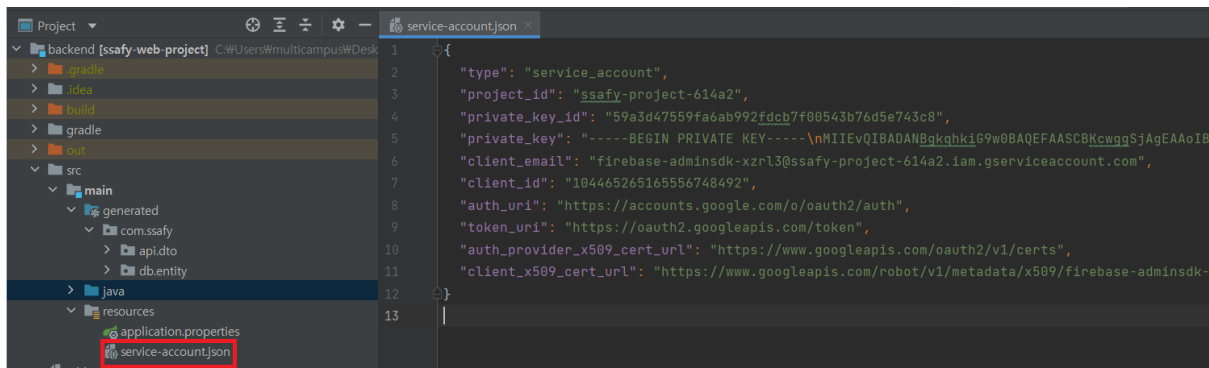
2. 생성된 프로젝트 설정에 들어가서 **웹앱** 추가



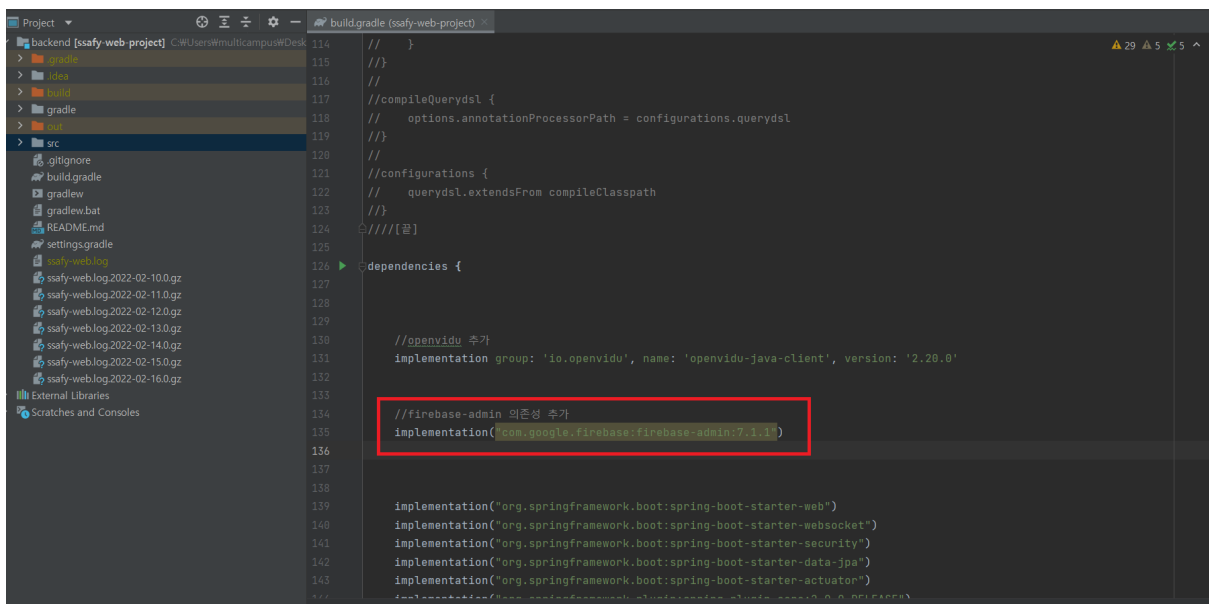
백엔드 세팅

1. 프로젝트 설정 → 서비스 계정으로 가서 **새 비공개 키 생성** 후 백엔드 프로젝트의 resources 폴더에 저장





2. 의존성 추가

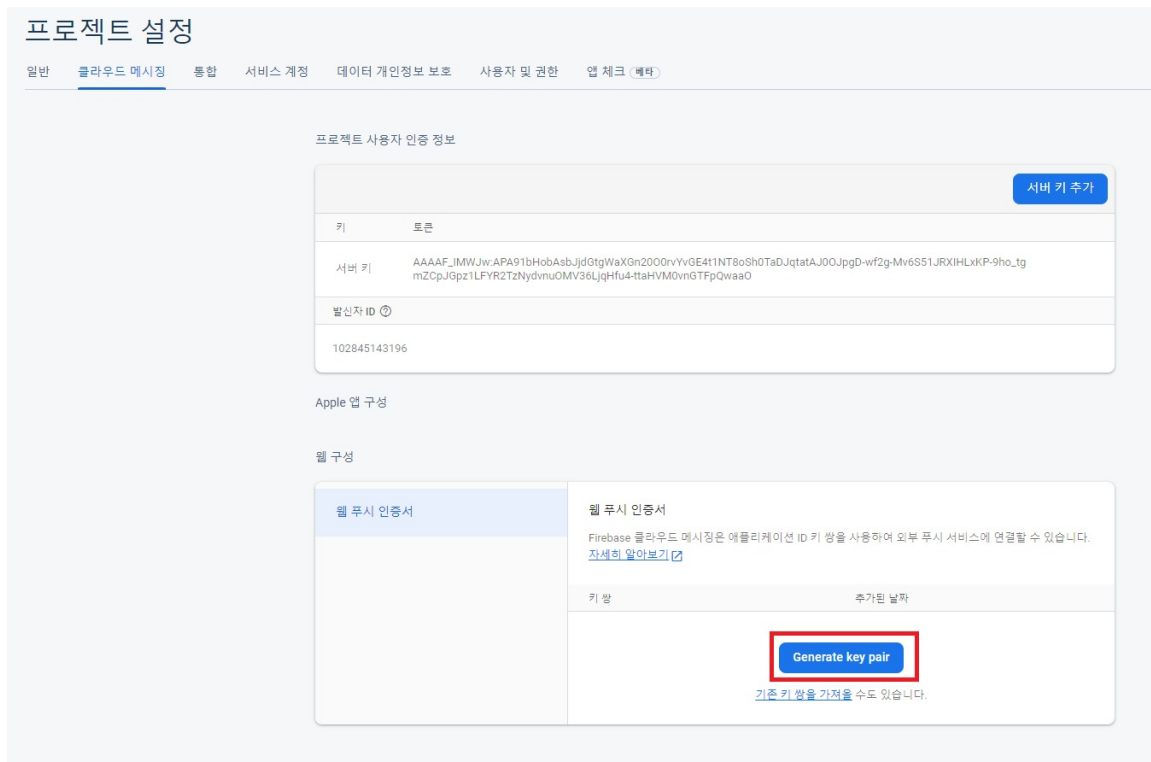


3. 어플리케이션이 시작될 때 비공개 키 파일(service-account.json)의 인증정보를 이용해서 Firebase를 초기화하도록 코드 설정 후 FCM 사용

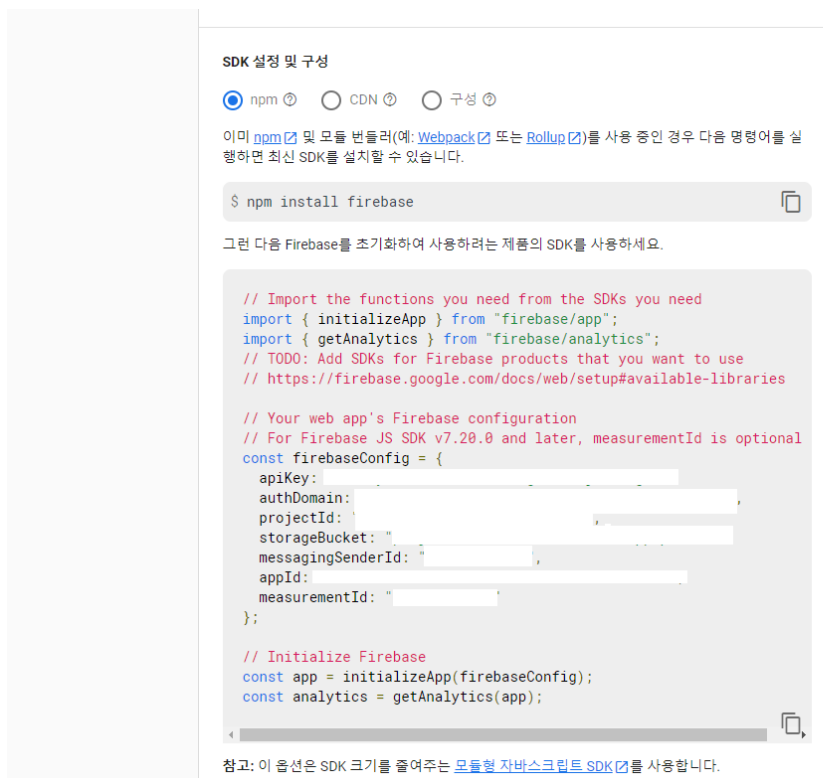
```
1 package com.ssafy.common.firebase;
2
3 import ...
4
5 @Slf4j
6 @Component
7 public class FCMInitializer {
8
9     private String googleApplicationCredentials;
10
11     @PostConstruct
12     public void initialize() throws IOException {
13         ClassPathResource resource = new ClassPathResource("service-account.json");
14
15         try (InputStream is = resource.getInputStream()) {
16             FirebaseOptions options = FirebaseOptions.builder()
17                 .setCredentials(GoogleCredentials.fromStream(is))
18                 .build();
19
20             if (FirebaseApp.getApps().isEmpty()) {
21                 FirebaseApp.initializeApp(options);
22                 log.info("FirebaseApp initialization complete");
23             }
24         }
25     }
26 }
27 }
```

프론트 세팅

1. 웹 푸시 인증서 키 생성



2. 프로젝트 설정 → 일반 → 내앱의 SDK 설정 정보 사용



3. firebase 버전 8.10.0 사용

- 8.10.0은 공식문서에 나와 있고, 5개월전까지 업데이트 되었기에 선택