# FCM 푸시 알림 시스템 다이어그램

결제 완료 시 FCM 푸시 메시지를 발송하는 전체 프로세스를 PlantUML로 표현한 다이어그램입니다.

**작성일**: 2026-01-23
**작성자**: 김택권

---

## 📋 목차



---

## 1. 다이어그램 종류

### 1.1 시퀀스 다이어그램

**파일**: `FCM_결제완료_푸시_시퀀스_다이어그램.puml`

결제 완료부터 푸시 알림 수신까지의 **시간 순서대로** 표현한 다이어그램입니다.

**용도**:

- 전체 프로세스의 순서 이해
- 각 단계별 상호작용 확인
- 디버깅 시 흐름 추적

### 1.2 시스템 아키텍처 다이어그램

**파일**: `FCM_시스템_아키텍처_다이어그램.puml`

시스템의 **구조와 컴포넌트 관계**를 표현한 다이어그램입니다.

**용도**:

- 시스템 전체 구조 파악
- 컴포넌트 간 의존성 이해
- 아키텍처 문서화

---

## 2. 시퀀스 다이어그램

### 2.1 주요 단계

```
1. 결제 완료
   ↓
2. Flutter 앱 → PushNotificationService 호출
   ↓
3. HTTP POST → FastAPI 서버
   ↓
4. FCMService → DB에서 FCM 토큰 조회
   ↓
5. Firebase Admin SDK 초기화 확인
   ↓
6. FCM 메시지 생성 및 Google Cloud FCM으로 발송
   ↓
7. Google FCM → APNs/GCM으로 전달
   ↓
8. 사용자 기기에 푸시 알림 수신
```

## 2.2 주요 상호작용

**단계 1: 결제 완료**

- **위치**: `lib/view/payment/purchase/toss_result_page.dart`
- **동작**: 결제 성공 시 `PushNotificationService.sendToCustomer()` 호출

**단계 2: HTTP 요청**

- **엔드포인트**: `POST /api/customer/{customerSeq}/push`
- **요청 본문**:

```
{
  "title": "결제 완료",
  "body": "예약번호: {orderId}",
  "data": {
    "type": "payment_complete",
    "reserve_seq": "..."
  }
}
```

**단계 3: FCM 토큰 조회**

- **테이블**: `device_token`
- **쿼리**: `SELECT fcm_token FROM device_token WHERE customer_seq = ?`
- **결과**: 고객의 모든 기기 FCM 토큰 목록

**단계 4: Firebase Admin SDK**

- **인증 파일**: `serviceAccountKey.json`
- **위치**: `fastapi/serviceAccountKey.json`

- **역할**: Google Cloud FCM 인증

**단계 5: FCM 메시지 발송**

- **서비스**: `FCMService.send_notification_to_customer()`
- **메서드**: `firebase_admin.messaging.send()`
- **대상**: Google Cloud FCM 서버

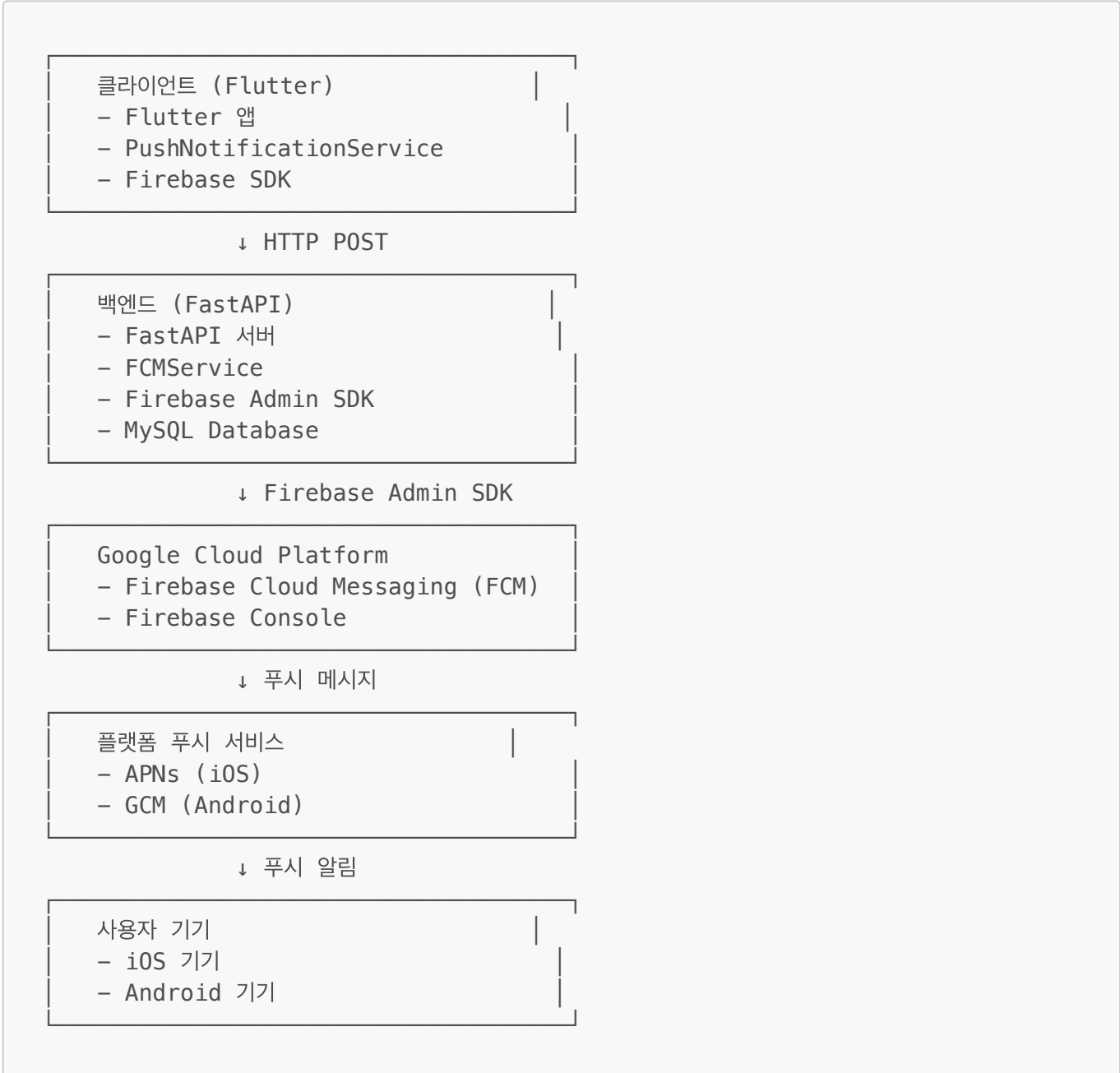
**단계 6: 플랫폼 푸시 서비스**

- **iOS**: Apple Push Notification Service (APNs)
- **Android**: Google Cloud Messaging (GCM)

---

## 3. 시스템 아키텍처 다이어그램

### 3.1 주요 레이어



### 3.2 컴포넌트 설명

**클라이언트 레이어**

- **Flutter 앱**: 사용자 인터페이스 및 비즈니스 로직
- **PushNotificationService**: 백엔드 API 호출 유틸리티
- **Firebase SDK**: FCM 토큰 관리 및 메시지 수신

**백엔드 레이어**

- **FastAPI 서버**: RESTful API 엔드포인트 제공
- **FCMService**: FCM 메시지 생성 및 발송 로직
- **Firebase Admin SDK**: 서버 측 Firebase 인증 및 메시지 발송
- **MySQL Database**: FCM 토큰 및 고객 정보 저장

**Google Cloud 레이어**

- **Firebase Cloud Messaging**: 메시지 라우팅 및 전달
- **Firebase Console**: 프로젝트 설정 및 인증 키 관리

**플랫폼 레이어**

- **APNs**: iOS 기기로 푸시 알림 전달
- **GCM**: Android 기기로 푸시 알림 전달

---

## 4. 주요 컴포넌트 설명

### 4.1 PushNotificationService

**위치**: `lib/utils/push_notification_service.dart`

**주요 메서드**:

- `sendToCustomer()`: 고객의 모든 기기에 푸시 발송
- `sendToToken()`: 특정 FCM 토큰에 푸시 발송

**역할**:

- HTTP 요청 생성 및 전송
- 응답 처리 및 에러 핸들링

### 4.2 FCMService

**위치**: `fastapi/app/utils/fcm_service.py`

**주요 메서드**:

- `send_notification_to_customer()`: 고객의 모든 기기에 푸시 발송
- `send_notification()`: 단일 기기에 푸시 발송

**역할**:

- DB에서 FCM 토큰 조회
- Firebase Admin SDK 초기화
- FCM 메시지 생성 및 발송

### 4.3 Firebase Admin SDK

**인증 파일**: `fastapi/serviceAccountKey.json`

**역할**:

- Google Cloud FCM 인증
- 서버 측 메시지 발송 권한 획득

**주의사항**:

- Git에 커밋하지 않음 (`.gitignore`에 추가)
- 보안에 주의 (절대 공개 저장소에 업로드 금지)

### 4.4 device_token 테이블

**스키마**:

```
CREATE TABLE device_token (
  device_token_seq INT PRIMARY KEY AUTO_INCREMENT,
  customer_seq INT NOT NULL,
  fcm_token VARCHAR(255) NOT NULL,
  device_type ENUM('ios', 'android') NOT NULL,
  device_id VARCHAR(255),
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);
```

**역할**:

- 고객의 FCM 토큰 저장
- 여러 기기 지원 (같은 고객의 여러 기기)
- 기기별 토큰 관리

---

## 5. 데이터 흐름

### 5.1 결제 완료 → 푸시 발송

```
1. 사용자 결제 완료
   ↓
2. toss_result_page.dart
   ↓
```

```
3. PushNotificationService.sendToCustomer()
   ↓
4. HTTP POST /api/customer/{customerSeq}/push
   ↓
5. FastAPI 엔드포인트 수신
   ↓
6. FCMService.send_notification_to_customer()
   ↓
7. DB에서 FCM 토큰 조회
   ↓
8. Firebase Admin SDK로 메시지 발송
   ↓
9. Google Cloud FCM 수신
   ↓
10. APNs/GCM으로 전달
   ↓
11. 사용자 기기 수신
```

## 5.2 메시지 구조

**HTTP 요청 (Flutter → FastAPI)**

```
{
  "title": "결제 완료",
  "body": "예약번호: 123",
  "data": {
    "type": "payment_complete",
    "reserve_seq": "123",
    "screen": "payment_detail"
  }
}
```

**FCM 메시지 (FastAPI → Google FCM)**

```
messaging.Message(
  token="fcm_token_here",
  notification=messaging.Notification(
    title="결제 완료",
    body="예약번호: 123"
  ),
  data={
    "type": "payment_complete",
    "reserve_seq": "123",
    "screen": "payment_detail"
  }
)
```

# 6. 다이어그램 사용 방법

## 6.1 PlantUML 뷰어 설치

### VS Code 확장 프로그램

1. VS Code 열기
2. 확장 프로그램 검색: "PlantUML"
3. "PlantUML" 확장 프로그램 설치
4. `.puml` 파일 열기
5. `Alt + D` 또는 우클릭 → "Preview PlantUML Diagram"

### 온라인 뷰어

- PlantUML Online Server
- `.puml` 파일 내용을 복사하여 붙여넣기

### 로컬 설치

```
# Java 설치 필요
brew install plantuml

# 다이어그램 생성
plantuml FCM_결제완료_푸시_시퀀스_다이어그램.puml
```

## 6.2 다이어그램 수정

### 시퀀스 다이어그램 수정

1. `FCM_결제완료_푸시_시퀀스_다이어그램.puml` 파일 열기
2. PlantUML 문법으로 수정
3. 미리보기로 확인

### 아키텍처 다이어그램 수정

1. `FCM_시스템_아키텍처_다이어그램.puml` 파일 열기
2. 컴포넌트 및 관계 수정
3. 미리보기로 확인

## 6.3 이미지로 내보내기

### VS Code에서

1. 다이어그램 미리보기 열기
2. 우클릭 → "Export Diagram"
3. PNG, SVG, PDF 등 선택

**명령줄에서**

```
plantuml -tpng FCM_결제완료_푸시_시퀀스_다이어그램.puml
plantuml -tsvg FCM_시스템_아키텍처_다이어그램.puml
```

---

## 7. 참고 문서

- FastAPI FCM 단발 푸시 테스트 가이드
- FCM 메시지 데이터 활용 가이드
- FCM 백엔드 푸시 알림 발송 가이드
- FCM 로컬 저장소 및 서버 연동 가이드

---

## 8. 변경 이력

- **2026-01-23**: 초기 문서 작성
  - 시퀀스 다이어그램 생성
  - 시스템 아키텍처 다이어그램 생성
  - 문서 작성

---

## 요약

이 문서는 결제 완료 시 FCM 푸시 알림을 발송하는 전체 프로세스를 PlantUML 다이어그램으로 표현합니다.

**주요 다이어그램**:

1. **시퀀스 다이어그램**: 시간 순서대로 프로세스 표현
2. **시스템 아키텍처 다이어그램**: 컴포넌트 구조 및 관계 표현

**주요 흐름**:

1. 결제 완료 → Flutter 앱
2. HTTP 요청 → FastAPI 서버
3. FCM 토큰 조회 → MySQL DB
4. 메시지 발송 → Google Cloud FCM
5. 푸시 전달 → APNs/GCM
6. 알림 수신 → 사용자 기기