카카오 로그인 API

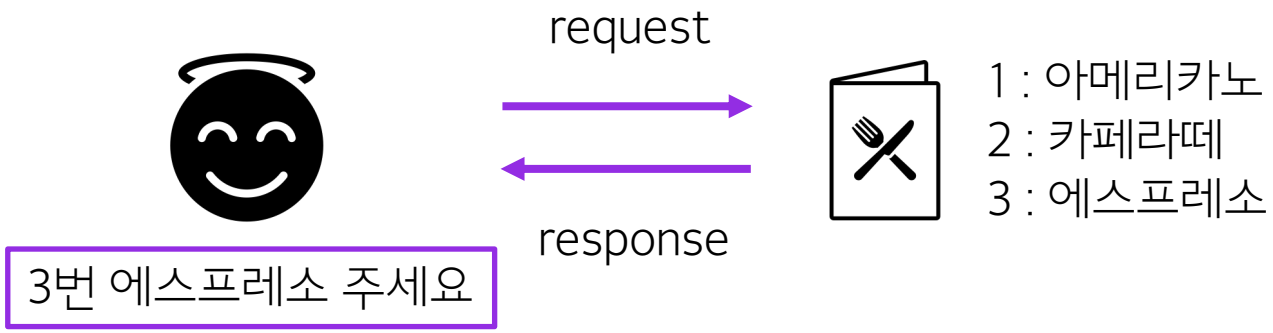
2019705O4 조담진

API

API(Application Programming Interface)

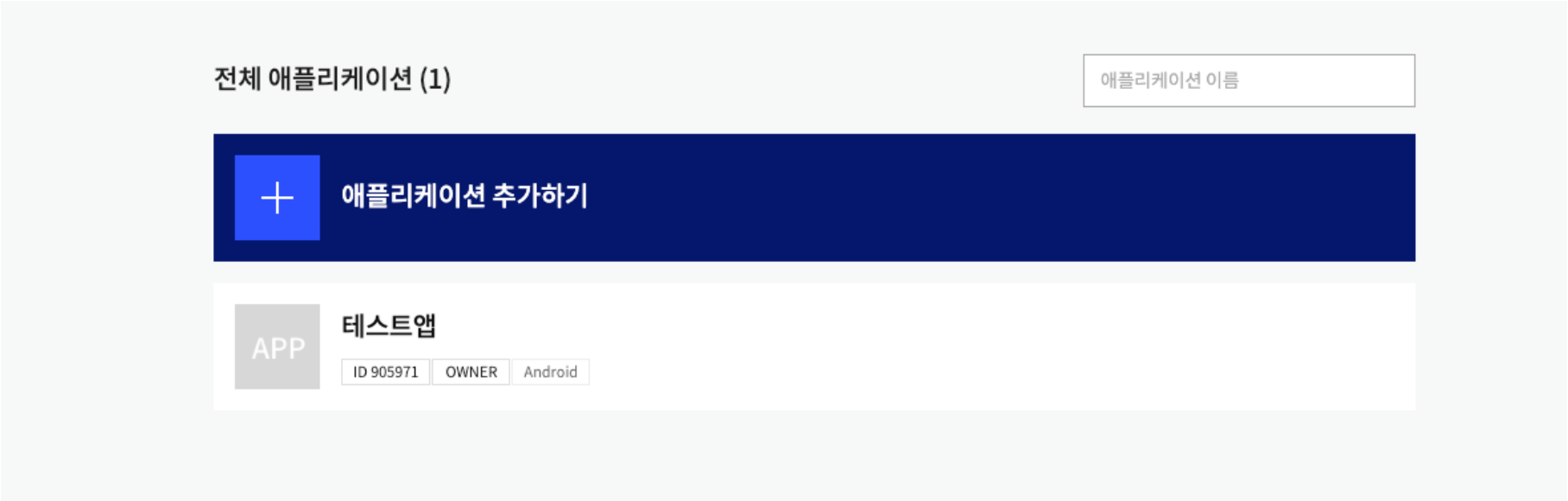
한 프로그램에서 다른 프로그램으로 데이터를 주고받기 위한 방법.

간단하게 말하면 메뉴판이라고 생각하면 된다.



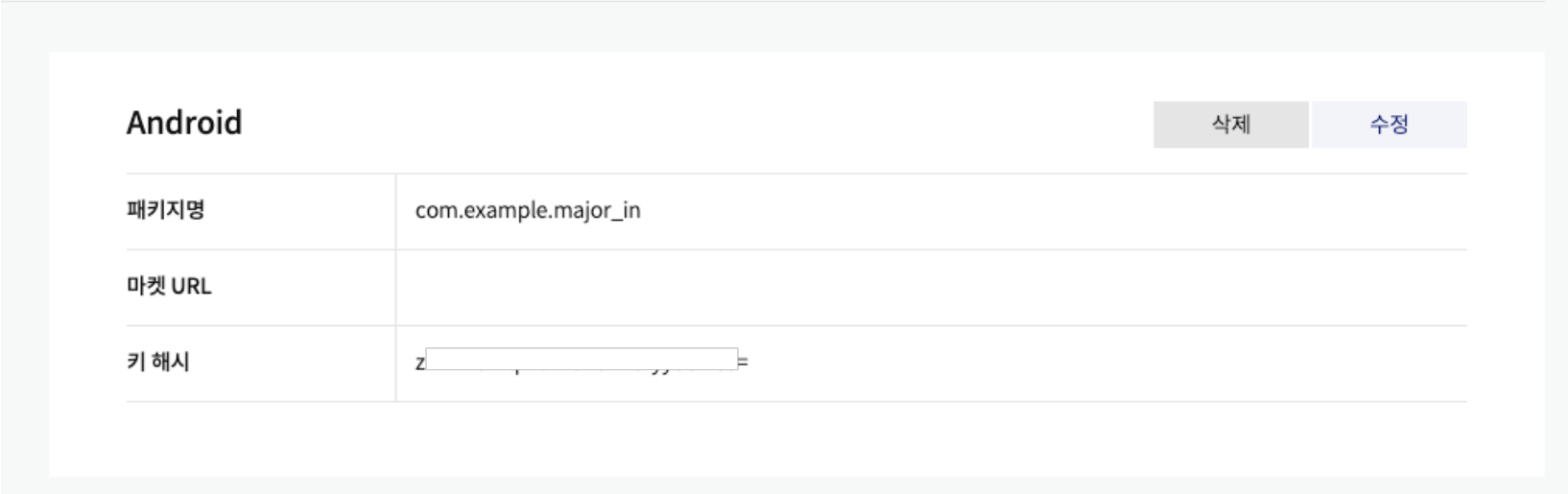
Kakao Developers 앱

- 애플리케이션 추가하기



Kakao Developers 앱

- Android 플랫폼 등록하기



- 패키지명은 사용할 프로젝트명으로 설정하기
- PC/노트북 각각 고유의 키 해시로 설정하기
- 키 해시 구하는 방법은 여러가지 있음

Kakao Developers 앱

- 로그인 활성화 및 동의항목(권한) 설정하기

Android SDK 설정

- Gradle 설정을 통해 카카오 API를 사용하기 위한 SDK Repository를 추가하기
- settings.gradle 파일에서 설정하기(android studio 최신 버전 기준)

```
dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
        maven { url 'https://devrepo.kakao.com/nexus/content/groups/public/' }
    }
}
```

모듈 설정

- build.gradle(Module:app)에서 필요한 라이브러리들 추가하기
- Android Studio 상단에 Sync Now 를 클릭해야 프로젝트에 포함됨

```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.5.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    //카카오 설정
//    implementation "com.kakao.sdk:v2-all:2.13.0" // 전체 모듈 설치, 2.11.0 버전부터 지원
    implementation "com.kakao.sdk:v2-user:2.13.0" // 카카오 로그인
//    implementation "com.kakao.sdk:v2-talk:2.13.0" // 친구, 메시지(카카오톡)
//    implementation "com.kakao.sdk:v2-story:2.13.0" // 카카오스토리
//    implementation "com.kakao.sdk:v2-share:2.13.0" // 메시지(카카오톡 공유)
//    implementation "com.kakao.sdk:v2-navi:2.13.0" // 카카오내비
//    implementation "com.kakao.sdk:v2-friend:2.13.0" // 카카오톡 소셜 피커, 리소스 번들 파일 포함
    //Glide 설정
    implementation "com.github.bumptech.glide:glide:4.11.0"
```

인터넷 사용 권한 설정

- API를 통해 카카오 서버와 통신하기 위해 설정해야 함
- AndroidManifest.xml 에서 설정하기

```
<!-- 인터넷 사용 권한 설정-->
<uses-permission android:name="android.permission.INTERNET" />
```

Java 8 설정

- Java 8 사용을 위한 설정을 해줘야 함(대부분 설정되어 있어서 확인만 해주기)

```
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}
```

SDK 초기화

- Android SDK를 사용하기 위해서는 가장 먼저 네이티브 앱 키로 초기화를 해야 함
- Application 을 상속한 클래스 만들기
- KaKao Developers 사이트에서 만든 앱의 네이티브 앱 키를 사용함

```
public class KakaoApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();
        KakaoSdk.init( context: this,  appKey: "                네이티브 앱 키                );
    }
}
```

SDK 초기화

- AndroidManifest.xml에서 애플리케이션을 등록해준다

```
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="Major_in"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.Major_in"
    android:name=".KakaoApplication"
    tools:targetApi="31">
```

Redirect URI 설정

- 카카오 로그인 기능을 구현하기 위함
- AndroidManifest.xml에 액티비티 추가하기

```
<activity android:name="com.kakao.sdk.auth.AuthCodeHandlerActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data android:host="oauth"
            android:scheme="                                    " />
    </intent-filter>
</activity>
```

Kakao + 네이티브 앱 키

UI 구성

- 1. 로그인이 되어있을 경우

  - 카카오 계정의 프로필사진과 닉네임을 보여준다.

  - 로그아웃 버튼을 Visible한다.

- 2. 로그인이 되어있지 않을 경우

  - 로그인 버튼을 Visible한다.

기능 구현

- 로그인 여부에 따른 화면

```
private void updateKakaoLoginUi() {
    //로그인이 되어있는지 확인하고 invoke 메소드를 콜백함
    UserApiClient.getInstance().me(new Function2<User, Throwable, Unit>() {
        @Override
        public Unit invoke(User user, Throwable throwable) {
            //로그인 상태일 경우
            if (user != null) {
                //얻을 수 있는 계정의 다양한 정보들
                Log.i(TAG, msg: "invoke: id=" + user.getId()); //계정 아이디
                Log.i(TAG, msg: "invoke: id=" + user.getKakaoAccount().getEmail()); //계정 이메일
                Log.i(TAG, msg: "invoke: id=" + user.getKakaoAccount().getProfile().
                        getThumbnailImageUrl()); //계정 프로필 이미지
                Log.i(TAG, msg: "invoke: id=" + user.getKakaoAccount().getProfile().
                        getNickname()); //계정 닉네임

                nickName.setText(user.getKakaoAccount().getProfile().getNickname());
                //글라이드 오픈소스 이미지뷰 사용하기
                Glide.with(profileImage).load(user.getKakaoAccount().getProfile().
                        getThumbnailImageUrl()).circleCrop().into(profileImage);

                loginButton.setVisibility(View.GONE);
                logoutButton.setVisibility(View.VISIBLE);
            //로그인 상태가 아닐 경우
            } else {
                nickName.setText(null);
                profileImage.setImageBitmap(null);

                loginButton.setVisibility(View.VISIBLE);
                logoutButton.setVisibility(View.GONE);
            }
            return null;
        }
    });
}
```

기능 구현

- 로그인 버튼 클릭 이벤트 처리

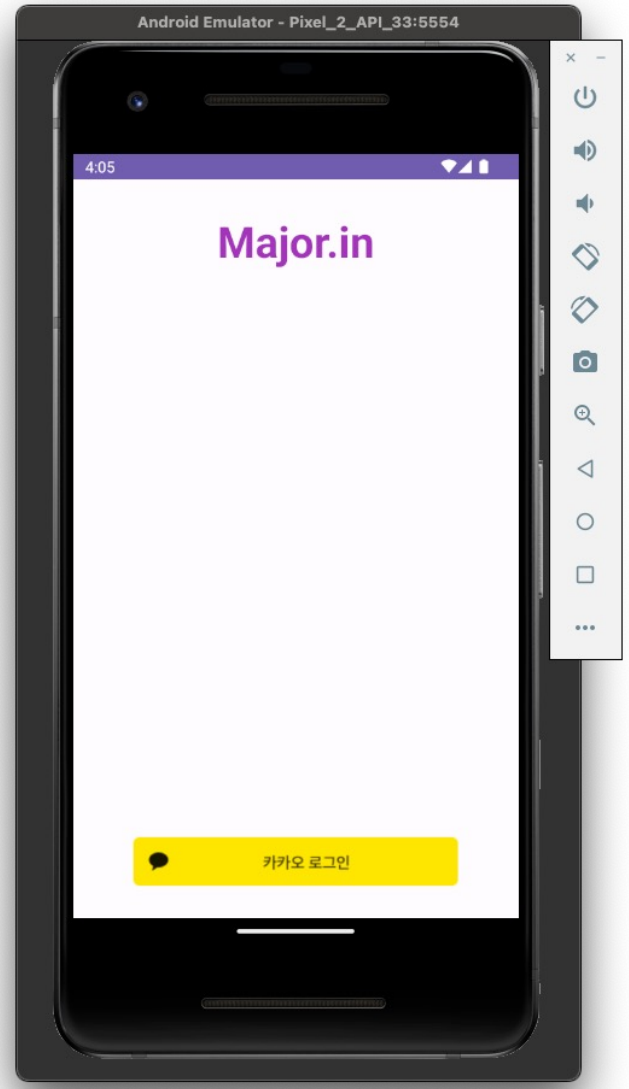
```
//콜백 객체, 로그인 결과에 대한 처리
Function2<OAuthToken, Throwable, Unit> callback = new Function2<OAuthToken, Throwable, Unit>() {
    @Override
    public Unit invoke(OAuthToken oAuthToken, Throwable throwable) {
        //토큰이 전달되면 로그인 성공, null이면 실패 (디테일한 처리인 듯)
        if (oAuthToken != null) {
            // TBD
        }
        //결과가 오류가 있다면 처리 (이것도)
        if (throwable != null) {
            // TBD
        }
        updateKakaoLoginUi();
        return null;
    }
};

loginButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //설치 여부 확인 (ture : 카카오톡 설치된 경우. 카카오톡을 띄워서 로그인, false : 아닌 경우. 카카오톡 홈페이지를 통해
        if (UserApiClient.getInstance().isKakaoTalkLoginAvailable( context: MainActivity.this)) {
            //파라미터로는 context, login 결과를 처리하기 위한 invoke 콜백(function2)
            UserApiClient.getInstance().loginWithKakaoTalk( context: MainActivity.this, callback); //
        } else {
            UserApiClient.getInstance().loginWithKakaoAccount( context: MainActivity.this, callback);
        }
    }
});
```

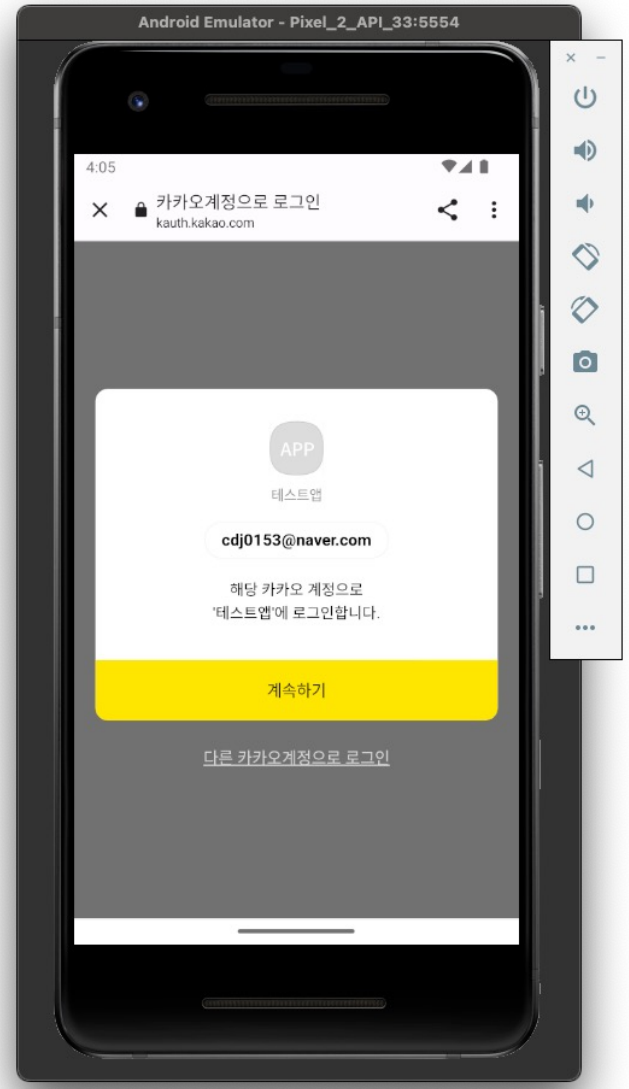
기능 구현

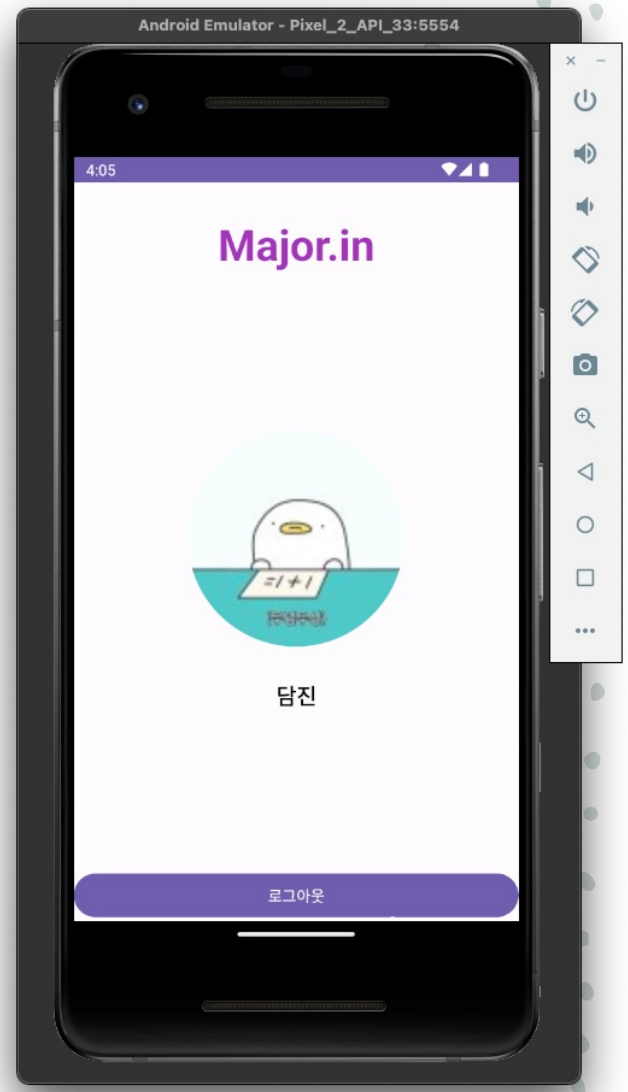
- 로그아웃 버튼

```
logoutButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        UserApiClient.getInstance().logout(new Function1<Throwable, Unit>() {
            @Override
            public Unit invoke(Throwable throwable) {
                updateKakaoLoginUi();
                return null;
            }
        });
    }
});
```

실행 화면


Android Emulator - Pixel_2_API_33:5554
4:05
Major.in
카카오 로그인


Android Emulator - Pixel_2_API_33:5554
4:05
카카오계정으로 로그인
kauth.kakao.com
APP
테스트앱
cdj0153@naver.com
해당 카카오 계정으로
'테스트앱'에 로그인합니다.
계속하기
다른 카카오계정으로 로그인


Android Emulator - Pixel_2_API_33:5554
4:05
Major.in
담진
로그아웃