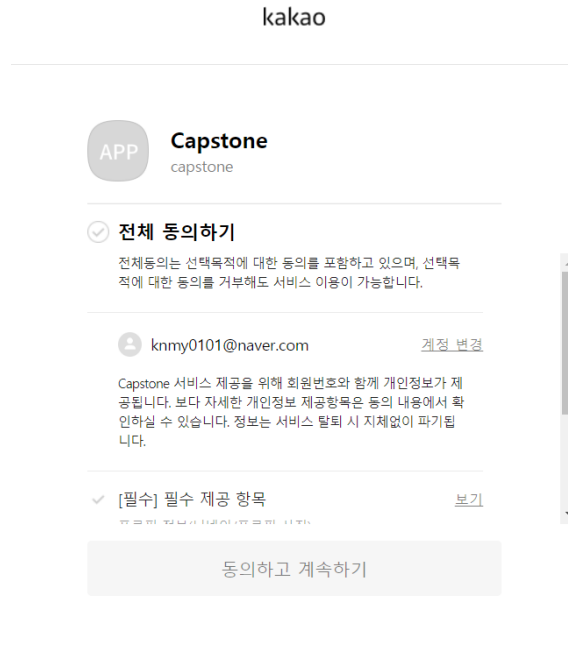


카카오 로그인 API

1. Node.js 서버에 구현하여 앱에서 서버로 카카오 API 요청

처음에 이 방법으로 진행했는데 문제점이 꽤 많네요. 우선 시도는 해봐서 결과만 보여드리면 아래처럼 브라우저에서는 잘 됩니다.



하지만 앱 내에서는 카카오 API 요청 시 위 처럼 로그인 페이지의 html 파일이 넘어오는데 이걸 웹뷰로 처리하는데에도 문제가 많았고, 실제 카카오 로그인을 수행하면 기본 웹 브라우저가 하나 더 켜져서 문제입니다. (앱에서 카카오 로그인 수행하면 크롬이 켜지면서 로그인 되었다고 나옴.)

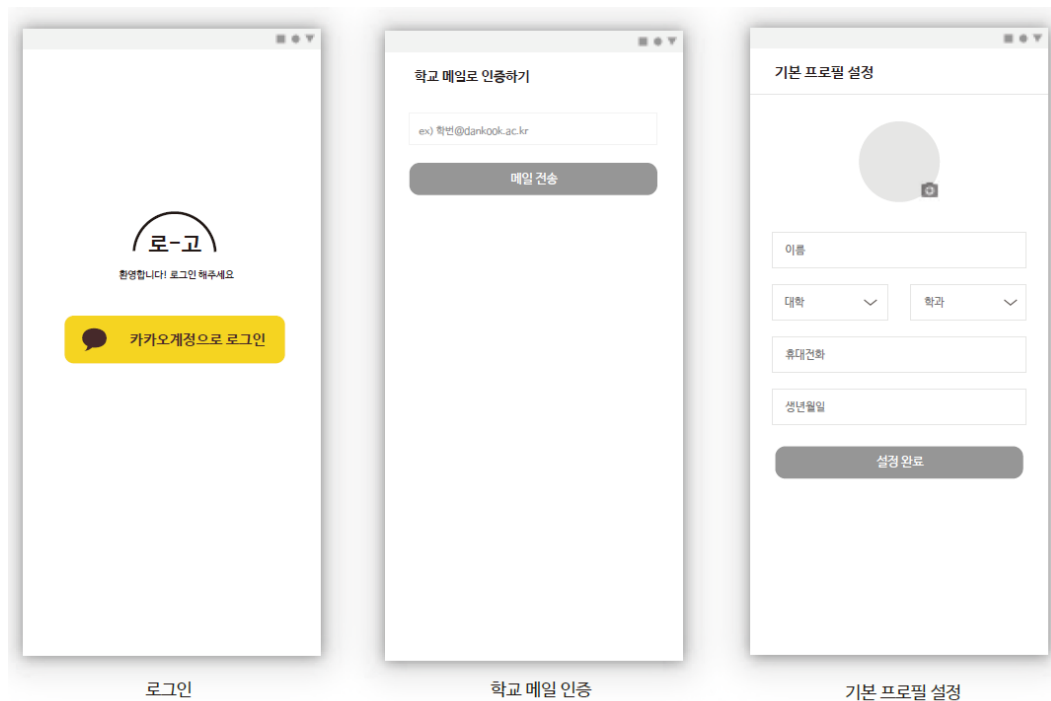
그래서 구글링 좀 해보니 일반적인 방법이 아니라 합니다. 보통 카카오 API는 앱에서 바로 요청한다고 하네요

2. React Native 카카오 로그인 라이브러리 이용(<https://www.npmjs.com/package/@react-native-seoul/kakao-login>)

그래서 카카오 로그인 라이브러리 이용해보니까 이 방법이 훨씬 나을 것 같네요. 우선 현재 KAKAO DEVELOPER 사이트에서 키해시, 패키지명 세팅도 끝났고, 앱 repo 클론해서 구현은 다 해놓긴 했습니다. 해나님이 보기에 문제 없으면 커밋하고 PR 날릴게요. 그리고 카카오회계 이메일 알려주시면 KAKAO DEVELOPER 초대 보낼게요. 아래에 간단한 구현과정도 설명해두었습니다.

이렇게 진행되면 해나님이 하실 과정은 유저가 아래 와이어프레임에서 '카카오 계정으로 로그인'

버튼 클릭하면 4페이지에 존재하는 kakaoHelper.js 모듈의 login 메서드 호출하고 얻어온 프로필 정보(result 변수)에서 id필드 가져오고 기본 프로필 설정 프레임에서 입력받은 값이랑 같이 POST로 서버에 요청하면 될 것 같아요.(학교 메일 인증은 아직 어떻게 될지 모르니 생략)



그리고 kakaoHelper.js는 어디에 둘지 몰라서 EventPage 폴더 안에다 만들어 뒀는데 이거는 아무데나 적당히 옮기시면 될 거 같아요. 타입스크립트는 잘 몰라서 그냥 js로 만들었습니다.

구현 과정

1~5 과정은 따로 안보셔도 될거 같고 6번 RN 소스코드만 보시면 될 듯 합니다.

1. 라이브러리 설치

```
npm install @react-native-seoul/kakao-login
```

2. manifest 파일 설정

application 태그 안에 아래처럼 allowBackup을 true로 바꿔주세요.

```
android:allowBackup="true"
```

3. SDK 설치

맨아래 maven{url 'kakao.com~~어쩌구 저쩌구'} 추가하고 싱크 맞추기

```
allprojects {  
    repositories {  
        mavenLocal()  
        maven {  
            // All of React Native (JS, Obj-C sources, Android binaries) is installed from npm  
            url("$rootDir/../node_modules/react-native/android")  
        }  
        maven {  
            // Android JSC is installed from npm  
            url("$rootDir/../node_modules/jsc-android/dist")  
        }  
  
        google()  
        jcenter()  
        maven { url 'https://www.jitpack.io' }  
        maven { url 'http://devrepo.kakao.com:8088/nexus/content/groups/public/' }  
    }  
}
```

아래 내용 복사 가능합니다.

```
maven { url 'http://devrepo.kakao.com:8088/nexus/content/groups/public/' }
```

4. android/app/src/res/values의 strings.xml에 kakao_app_key 정의

```
<resources>  
    <string name="app_name">swcproject</string>  
    <string name="kakao_app_key">6f50d43c8959abe7aefa4610126888a4</string>  
</resources>
```

아래 내용 복사 가능합니다.

```
<string name="kakao_app_key">6f50d43c8959abe7aefa4610126888a4</string>
```

5. manifest 수정

```
<application  
    android:name=".MainApplication"  
    android:label="swcproject"  
    android:icon="@mipmap/ic_launcher"  
    android:roundIcon="@mipmap/ic_launcher_round"  
    android:allowBackup="true"  
    android:theme="@style/AppTheme">  
    <meta-data  
        android:name="com.kakao.sdk.AppKey"  
        android:value="@string/kakao_app_key" />  
</application>
```

복사용

```
<meta-data  
    android:name="com.kakao.sdk.AppKey"  
    android:value="@string/kakao_app_key" />
```

위까지 하면 세팅은 다 끝났습니다.

6. RN 소스코드

KakaoHelper.js

```
import KakaoLogins from '@react-native-seoul/kakao-login';

if(!KakaoLogins){
  console.error('module is not linked');
}

var kakaoHelper = {
  login: loginFunction,
  logout: logoutFunction,
};

function loginFunction(){
  KakaoLogins.login().then(result=>{
    console.log(`###Login Result:${JSON.stringify(result)}`);
    KakaoLogins.getProfile().then(result=>{
      console.log(`Profile:${JSON.stringify(result)}`);
      return result;
    }).catch(err=>{
      console.log(`profile err:${JSON.stringify(err)}`);
    });
  }).catch(err=>{
    console.log(`Login err:${JSON.stringify(err)}`);
  });
};

function logoutFunction(){
  KakaoLogins.logout().then(result=>{
    console.log(`###Logout Result:${JSON.stringify(result)}`);
  }).catch(err=>{
    console.log(`Login err:${JSON.stringify(err)}`);
  });
};

module.exports=kakaoHelper;
```

위 모듈은 카카오 로그인, 로그아웃을 제공하는데 로그인 함수 내부를 보면 KakaoLogins.login()으로 로그인 시도하고 성공하면 결과 출력후 다시 KakaoLogins.getProfile()로 유저 정보를 얻어옵니다.

위 모듈은 로그인하고 싶은 곳가서 모듈 가져오고 모듈.login() 수행하면 됩니다. 저는 그냥 테스트용으로 EventPage 폴더 안에 index.tsx에서 테스트 해보고 주석처리 해놨습니당.

```
// stack navi를 위한 prop
import {StackNavigationProp} from '@react-navigation/stack';
type NavigationProp = StackNavigationProp<MainPageParamList, 'MainPage'>;
interface Props {
  navigation: NavigationProp;
}

//카카오 로그인
const kakaoLogin=require('./KakaoHelper');
kakaoLogin.login();
```

아래는 결과화면

