

Kakao Login 230118

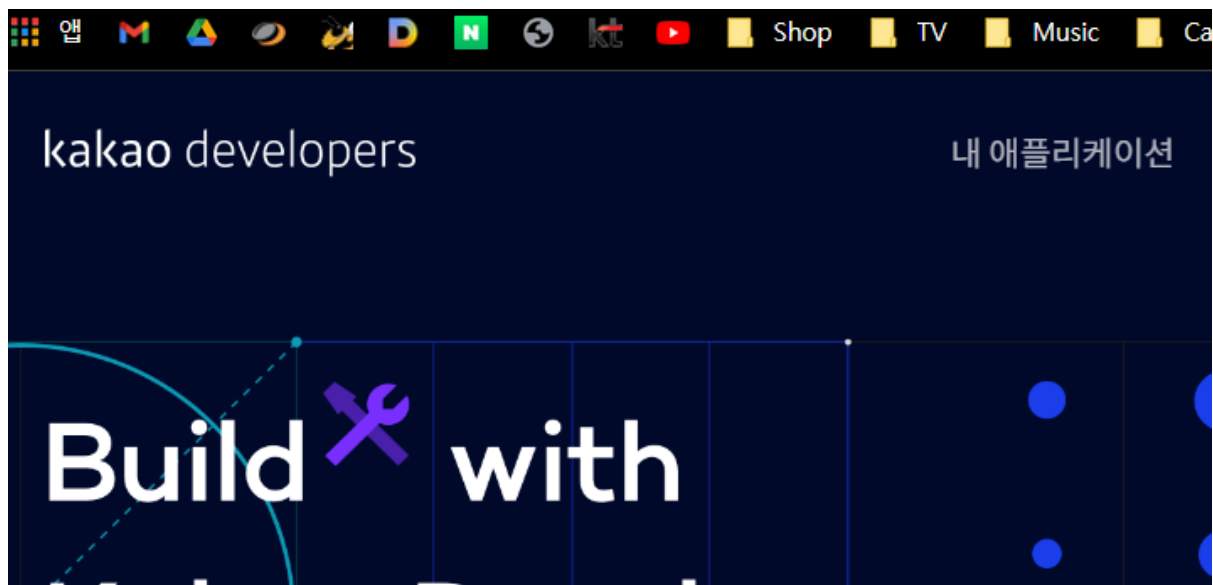
카카오 로그인

로그인 프로세스 전체를 카카오나 구글 등에 완전히 위임하게 되면 (자체 디비에 고객 정보가 전혀 없이) 마케팅 등이 어렵게 된다. 따라서 처음 사이트에 방문시 카카오 로그인을 했을때 그때 한번 회원 가입을 할 수 있게 하면 추후 로그인만 카카오를 통해서 허용하고 따라서 고객의 정보도 직접 보관할수 있어 다양한 프로모션 등의 고객 관리도 할수 있다.

1. 단지 고객 디비 관리 없이 카카오 로그인 만으로 회원인증
2. 로그인은 카카오로 인증을 할수 있지만 최초 접속시 회원 가입을 한번 진행하여 회원 관리/인증

카카오 로그인 → 해당 회원의 정보가 자체 디비에 존재하는지 확인 → 없으면 회원 가입 진행 → 있으면 로그인 성공(세션 만들기)

카카오 개발자 등록



내 애플리케이션으로 들어 가면 최초 개발자 등록(회원가입)을 진행하게 된다.

애플리케이션 추가하기

앱 아이콘

이미지
업로드

파일 선택

JPG, GIF, PNG

권장 사이즈 128px, 최대 250KB

앱 이름

loginexam

사업자명

Austin

- 입력된 정보는 사용자가 카카오 로그인할 때 표시됩니다.
- 정보가 정확하지 않은 경우 서비스 이용이 제한될 수 있습니다.

☒ [서비스 이용이 제한되는 카테고리, 금지된 내용, 금지된 행동](#) 관련 운영정책을 위반하지 않는 앱입니다.

취소

저장

JavaScript 키 1d944d4a9cb22483964d7e130a9b4c2a

플랫폼

설정된 플랫폼 정보가 없습니다. [플랫폼 설정하기](#)

Web 플랫폼 등록

사이트 도메인

JavaScript SDK, 카카오톡 공유, 카카오톡, 메시지 API 사용시 등록이 필요합니다.

여러개의 도메인은 줄바꿈으로 추가해주세요. 최대 10까지 등록 가능합니다. 추가 등록은 포럼으로 문의주세요.

예시: (O) <https://example.com> (X) <https://www.example.com>

<https://localhost:8080>

기본 도메인

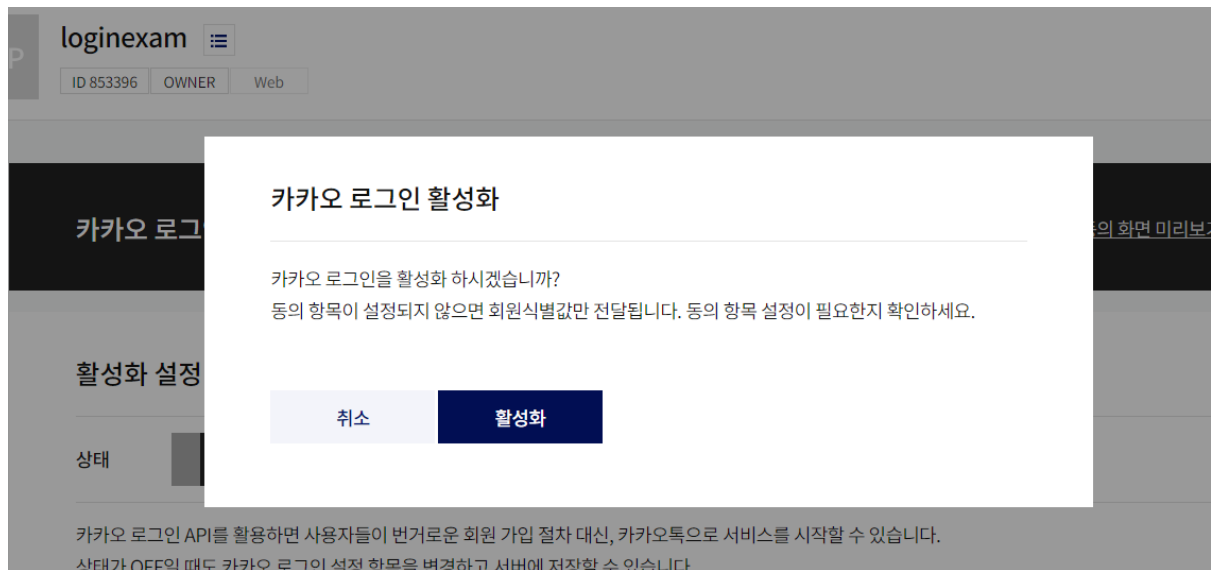
기본 도메인은 첫 번째 사이트 도메인으로, 카카오톡 공유와 카카오톡 메시지 API를 통해 발송되는 Web 링크 기본값으로 사용됩니다.

<https://localhost:8080>

취소

저장

활성화 클릭



OpenID Connect 활성화 설정

상태

ON

카카오 로그인의 확장 기능인 OpenID Connect를 활성화합니다.

이 설정을 활성화하면 카카오 로그인 시 사용자 인증 정보가 담긴 ID 토큰을 액세스.

Redirect URI

Redirect URI

카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다.

여러개의 URI를 줄바꿈으로 추가해주세요. (최대 10개)

REST API로 개발하는 경우 필수로 설정해야 합니다.

예시: (O) <https://example.com/oauth> (X) <https://www.example.com/oauth>

`http://localhost:8080`

취소

저장

카카오 로그인에 성공했을 때 넘겨 받을 데이터 설정

동의항목

카카오 로그인으로 서비스를 시작할 때 동의 받는 항목을 설정합니다. 미리 보기를 통해 사용자에게 보여질 화면을 확인할 수 있습니다. 사업자 정보를 등록하여 비즈 앱으로 전환하고 비즈니스 채널을 연결하면 권한이 없는 동의 항목에 대한 검수 신청을 할 수 있습니다.

[비즈니스 설정 바로가기](#)

개인정보

항목 이름	ID	상태	
닉네임	profile_nickname	● 사용 안함	설정
프로필 사진	profile_image	● 사용 안함	설정

로그인 성공시 닉네임, 이메일, 성별 데이터를 넘겨 받게 된다.

[비즈니스 설정 바로가기](#)

개인정보

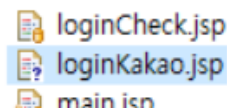
항목 이름	ID	상태	
닉네임	profile_nickname	● 필수 동의	설정
프로필 사진	profile_image	● 사용 안함	설정
카카오계정(이메일)	account_email	● 선택 동의	설정
이름	name	○ 권한 없음	
성별	gender	● 선택 동의 [수집]	설정
연령대	age_range	● 사용 안함	설정
생일	birthday	● 사용 안함	설정
출생 연도	birthyear	○ 권한 없음	

닉네임, 카카오회정(이메일), 성별 [설정]

카카오 사이트 설정 끝.

자체 로그인 페이지(아이디, 비번) → loginCheck.jsp → DAO → DB Table (로그인 성공/실패)

자체 로그인 페이지(카카오 호출) → 카카오 로그인 페이지(아이디, 비번) → 로그인 성공/실패



```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<a id="custom-login-btn" href="javascript:kakaoLogin()">

</a>

<script src="https://developers.kakao.com/sdk/js/kakao.js"></script>

<script>
//발급 받은 키
Kakao.init("1d944d4a9cb22483964d7e130a9b4c2a");

function kakaoLogin() {
    window.Kakao.Auth.login({
        // 개발자 사이트에서 체크한 항목들과 반드시 일치해야 한다.
        scope:'profile_nickname, account_email, gender',
        success: function (authObj) {window.Kakao.API.request({url:'/v2/user/me',
            success:res => {const kakaoAccount = res.kakao_account;
                location.href='loginKakaoCheck.jsp?mname=' + kakaoAccount.profile.nickname
                    + '&memail=' + kakaoAccount.email
                    + '&gender=' + kakaoAccount.gender;}}
        });
    }
});
}

</script>

</body>
</html>
```



성공 !!! [austiny@naver.com] 님 로그인 성공하였습니다.

```

<!--
<!-- 성공 !!!
<!--
<%
String mname = request.getParameter("mname");
String memail = request.getParameter("memail");
String mgender = request.getParameter("mgender");

session.setAttribute("sid", memail);

String sid = (String)session.getAttribute("sid");

%>

[ <%=sid%> ] 님 로그인 성공하였습니다.

</body>
</html>

```

자체 로그인 페이지(카카오 호출) → 카카오 로그인 페이지(아이디, 비밀번호) → 로그인 성공/실패

// 우리 디비에 해당 회원이 존재하는지 여부 확인 exist 메서드로 확인 후
// 존재하면 (0) 세션 처리 (로그인 상태)
// 디비에 존재하지 않으면 회원 가입 페이지로 이동 가입 유도

loginCheckKakao.jsp


```

String mname = request.getParameter("mname");
String memail = request.getParameter("memail");
String mgender = request.getParameter("mgender");

// 우리 디비에 해당 회원이 존재하는지 여부 확인 exist 매서드로 확인 후
// 존재하면 (0) 세션 처리 (로그인 상태)
// 디비에 존재하지 않으면 회원 가입 페이지로 이동 가입 유도

int result = memberDAO.exist(memail);

if (result == 0) {
    out.print("로그인 성공"); <- 디비에 회원 정보 가 존재 하는 경우 단순히 로그인 처리
    session.setAttribute("sid", memail);
} else {
    response.sendRedirect("signupForm.jsp?"); <- 카카오 데이터는 변경할수 없도록 설정
}

```

exist 매서드

```

public static int exist(String memail) throws NamingException, SQLException {

    //C R U D
    String sql = "SELECT memail FROM member WHERE memail=?";

    //커넥션 풀 사용
    Connection conn = ConnectionPool.get();

    PreparedStatement pstmt = conn.prepareStatement(sql);

    pstmt.setString(1, memail);

    ResultSet rs = pstmt.executeQuery();

    if (!rs.next()) return 1; //디비에 회원 정보가 전혀 없는 경우
    return 0; // 디비에 존재하는 경우

}

```

로그인에는 카카오톡으로 성공하였으나 우리 디비에는 존재하지 않는 경우 회원 가입 폼으로 이동하면서 카카오 에서 받은 회원 데이터를 함께 넘긴다.

```

int result = memberDAO.exist(memail);

if (result == 0) {
    out.print("로그인 성공");
} else {
    response.sendRedirect("signupForm.jsp?mname=" + mname + "&memail=" + memail + "&mgender=" + mgender);
}

```

```
<select class="form-select" name="mgender" aria-label="Default select example">
<%
if (mgender.equals("male")) {
%>
<option value="male">남성</option>
<%}else{%>
<option value="female">여성</option>
<%} %>
</select>
```

한글 깨질 경우

```
response.sendRedirect("signupForm.jsp?mname=" + java.net.URLEncoder.encode(mname) +
"&memail="+memail+"&mgender="+mgender);
```