

Kakao Login

✓ LINUX

✓ 카카오 개발자

✓ 카카오 로그인

LINUX

UBUNTU 22.10

Download Ubuntu Desktop | Download | Ubuntu

The open-source desktop operating system that powers millions of PCs and laptops around the world. Find out more about Ubuntu's features and how we support developers and organisations below.

 <https://ubuntu.com/download/desktop>



UTM 4.0.9

<https://github.com/utmapp/UTM/releases>

카카오 개발자

Kakao Developers

Start with kakao platform +service api products Start with kakao platform + service api products 인간은 바로 사람과 사람이 서로 기대어 구성됩니다. 상생은 상대의 부정이 아닌 긍정을 통해 이룰 수 있는 세계입니다.

 <https://developers.kakao.com/>

kakao developers

전체 애플리케이션 (1)

애플리케이션 이름



애플리케이션 추가하기

APP

loginexam

ID 853391

OWNER

Web

카카오 로그인

ON

[동의 화면 미리보기](#)

활성화 설정

상태

ON

카카오 로그인 API를 활용하면 사용자들이 번거로운 회원 가입 절차 대신, 카카오톡으로 서비스를 시작할 수 있습니다.

상태가 OFF일 때도 카카오 로그인 설정 항목을 변경하고 서버에 저장할 수 있습니다.

상태가 ON일 때만 실제 서비스에서 카카오 로그인 화면이 연결됩니다.

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 토큰과 함께 발급받을 수 있습니다.

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	설정

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 할수 있게 하면 추후 로그인만 카카이를 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 카카으로 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)

loginKakao.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>loginKakao.jsp</title>
</head>
<body>
<a href="javascript:kakaoLogin()" id="custom-login-btn">
    
</a>
<script src="http://developers.kakao.com/sdk/js/kakao.js"></script>
<script>
    //발급받은 키
    Kakao.init("fc7abe9b9479ffa297474094a4964dcf");

    function kakaoLogin(){
```

```

window.Kakao.Auth.login({
  //체크한 항목과 반드시 일치 시키기
  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
        + '&mgender='+kakaoAccount.gender;}}
    });
  }
});
}
</script>
</body>
</html>

```

loginCheckKakao.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="jdbc.*"%>
<%
    request.setCharacterEncoding("utf-8");

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

    int result = memberDAO.exist(memail);

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

    session.setAttribute("sid", memail);
%>

```

exist method

```

public static int exist(String memail) throws NamingException, SQLException {
    String sql = "SELECT memail FROM member WHERE memail=?";

    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  
}
```