20230118 JDBC & Kakao Login



카카오 로그인

로그인 프로세스 전체를 카카오, 구글 등에 완전히 위임하게 되면 (자체 DB에 고객 정보X) 마케팅이 어렵게 된다.

따라서 처음 방문시 카카오 로그인을 했을 때 회원 가입을 할 수 있게 하고 카카오를 통한 로그인을 허용하면 고객의 정보도 직접 보관할 수 있다. 이를 통해 다양한 프로모션 등 고객관리도 할 수 있다.

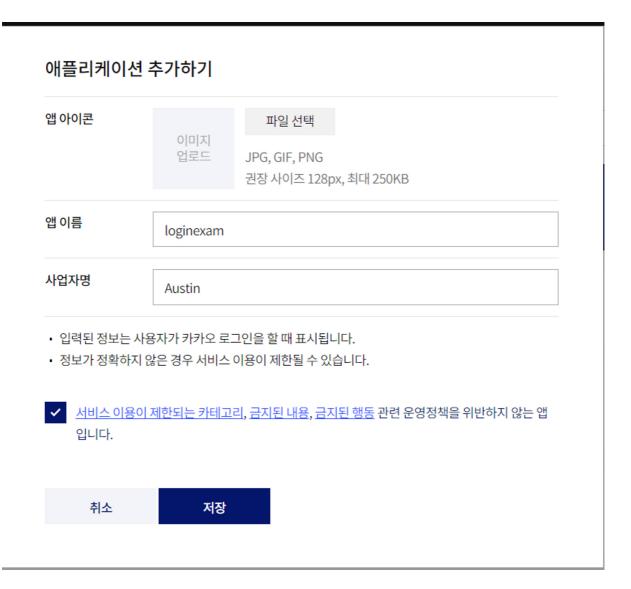
- 1. 고객 DB 관리 없이 카카오 로그인 만으로 회원 인증
- 2. 로그인은 카카오로 할 수 있지만 최초 적습시 회원가입을 진행해 회원 관리/인증
 카카오 로그인 → 해당 회원의 정보가 DB에 존재하는지 확인 → 없으면 회원가입 진행
 → 있으면 로그인 성공 (세션 만들기)

카카오 로그인 설정

카카오 개발자 등록

https://developers.kakao.com/

애플리케이션 추가



Web 플랫폼 등록

Web 플랫폼 등록

사이트 도메인

JavaScript SDK, 카카오톡 공유, 카카오맵, 메시지 API 사용시 등록이 필요합니다. 여러개의 도메인은 줄바꿈으로 추가해주세요. 최대 10까지 등록 가능합니다. 추가 등록은 포럼 으로 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 발송5지의 Web 링크 기본값으로 사용됩니다.

https://localhost:8080

취소

저장

<mark>카카오 로그인 & OpenID Connect 활성화</mark>

Redirect할 페이지 설정

Redirect URI

Redirect URI

카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다.

여러개의 URI를 줄바꿈으로 추가해주세요. (최대 10개)

REST API로 개발하는 경우 필수로 설정해야 합니다.

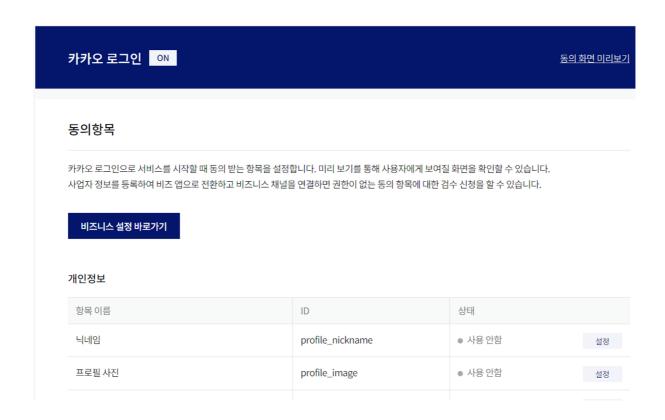
예시: (O) https://example.com/oauth (X) https://www.example.com/oauth

http://localhost:8080

취소

저장

카카오 로그인시 넘겨 받을 정보 설정



JavaScript ₱ e85f1f60d5ae784c7e84ade4beae9fd5

자체 로그인 페이지 \rightarrow loginCheck.jsp \rightarrow DAO \rightarrow DB table (로그인 성공/실패) 자체 로그인 페이지 (카카오 호출) \rightarrow 카카오 로그인 페이지 (아이디, 비번) \rightarrow 로그인 성공/실패

JavaScript를 통한 카카오 로그인 설정 - https://developers.kakao.com/docs/latest/ko/kakaologin/js

```
<%@ 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>
</al>
<a id="custom-login-btn" href="javascript:kakaoLogin()">
<img src="kakao.png">
</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
             + '&mgender=' + kakaoAccount.gender;}
     });
   });
 }
</script>
</body>
</html>
```

우리 DB에 해당 회원이 존재하는지 여부 확인 후 exist 메서드로 확인 후 존재하면 (0) 세션 처리 (로그인 상태) 존재하지 않으면 회원 가입 페이지로 이동 회원 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 sesstion.setAttribute("sid",memail);
} else {
  response.sendRedirect("signupForm.jsp?"); <- 카카오 데이터는 변경할수 없도록 설정
}
```

exist 메서드

```
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 카카오로그인으로 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);
}
```