

KOSA Mini Proj1

Servlet과 JSP를 사용한
회원&게시글 관리 웹 어플리케이션

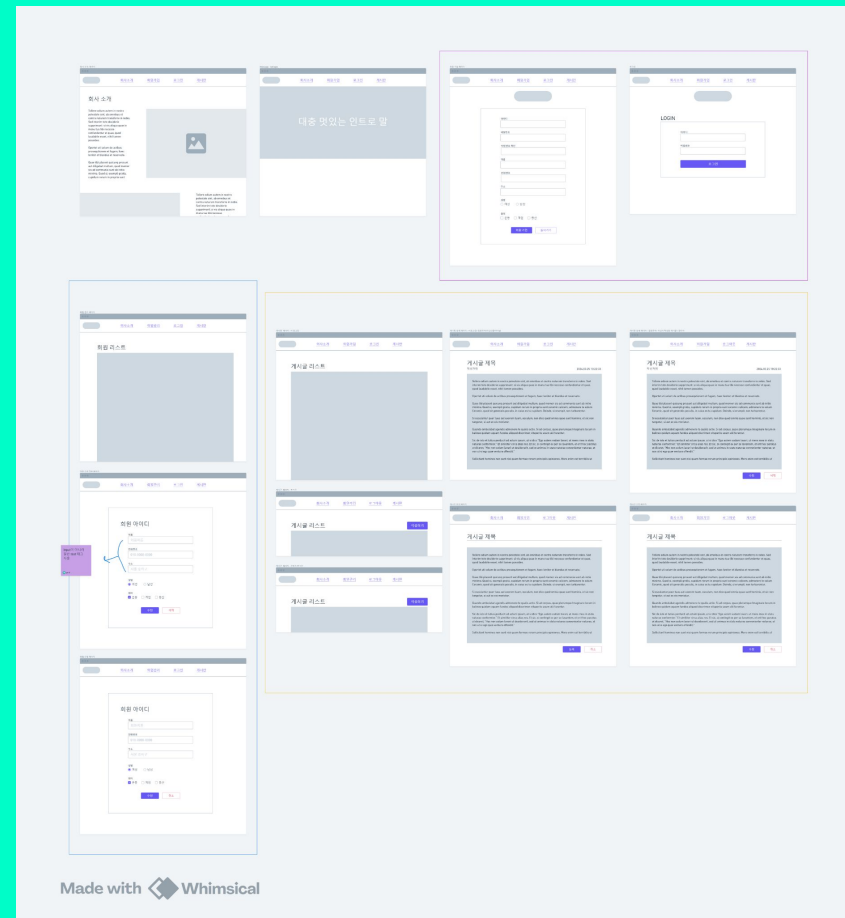
안아영

스토리 보드

스토리보드

Whimsical을 이용해 스토리보드를 작성

- 기능에 중심하기 위해서 빠르고 간단하게 만들 수 있는 툴
- 공유와 파일 출력이 편리



데이터 베이스

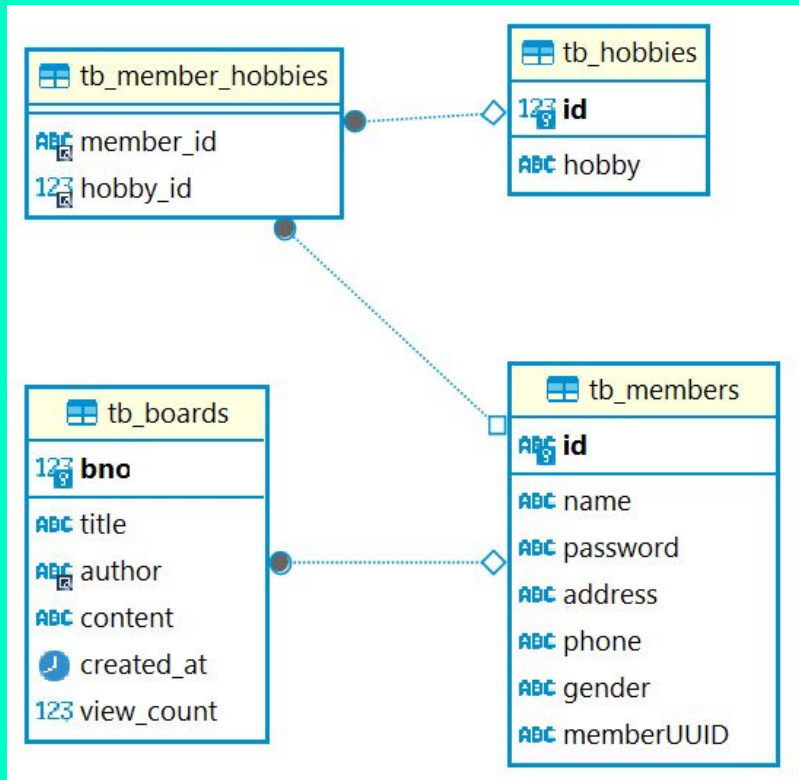
데이터 베이스

MariaDB 사용

한 회원이 여러 취미를 가질 수 있게 위해서,
유저와 취미의 연관 관계만 정의한
tb_member_hobbies 테이블을 생성

tb_members

- 자동 로그인 기능을 위한 memberUUID



데이터 베이스

게시글을 작성한 회원이 탈퇴한 경우,
게시글의 **author** 컬럼 데이터를 NULL로 처리

- ON DELETE SET NULL

회원이 가지고 있는 취미를 나타내는
tb_member_hobbies 테이블에서 회원과
취미 데이터가 삭제된다면 해당 테이블에서
그 데이터를 사용하는 데이터가 삭제되도록
처리

- ON DELETE CASCADE

```
CREATE TABLE tb_boards (  
  bno int auto_increment primary key,  
  title varchar(100) not null,  
  author varchar(100),  
  content text not null,  
  created_at timestamp default CURRENT_TIMESTAMP,  
  view_count int default 0,  
  FOREIGN KEY (author) REFERENCES tb_members(id) ON DELETE SET NULL  
);
```

```
CREATE TABLE tb_member_hobbies(  
  member_id varchar(100),  
  hobby_id int,  
  foreign key (member_id) references tb_members(id) on delete cascade,  
  foreign key (hobby_id) references tb_hobbies(id) on delete cascade  
);
```

기능 소개

기능 소개 - 회원

회원 가입

- 중복 아이디 체크

회원 정보 수정&삭제

- 로그인 사용자 or 관리자

회원 전체 리스트

- 관리자

자동 로그인

세션

게시판

게시물 리스트 & 검색

게시물 읽기

- viewCount 증가

게시물 생성

- 로그인한 사용자만

게시물 수정&삭제

- 자신의 게시물
- 관리자

코드

- Connection Pool
- 데이터 화면 출력
- 취미를 다루는 방법

Connection Pool

기존 코드

- 각각 DAO에 Connection이 따로 존재
- Connection을 끊고 다시 연결하는 행위는 리소스를 소모한다.
- 불필요한 연결에 의한 서버 자원 낭비

Connection Pool을 활용하면,

- Connection을 WAS에 미리 생성
- WAS가 Connection을 관리
- 필요할 때, 준비된 Connection을 사용하기만 하면 됨

```
// 기존방식 코드 예제
public class BoardDAO {
    private static Connection conn = null;
    // ..PreparedStatement들..
    static {
        try {
            // 1. 클래스를 로드
            Class.forName("org.mariadb.jdbc.Driver");

            // 2. 데이터베이스와 연결
            conn = DriverManager.getConnection( //
                "jdbc:mariadb://localhost:3306/miniproj_db", //
                "bituser", "1004");

            System.out.println("MariaDB 연결 성공");
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Connection Pool

WAS에서 관리하기 때문에 WAS의 설정을
변경해줘야한다.

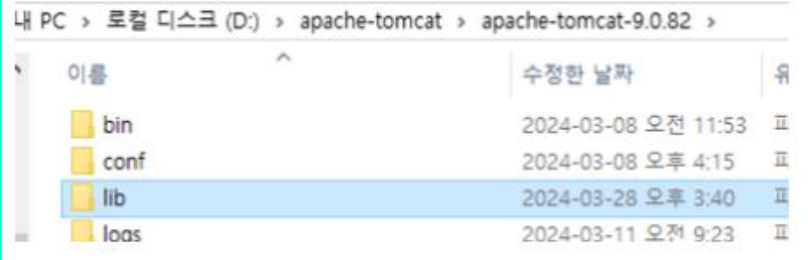
- server.xml
- context.xml



서버에서 JDBC 드라이버를 로드하기 때문에
서버의 라이브러리 폴더에 mariaDB
드라이버 jar 파일을 넣어줘야한다.

```
<GlobalNamingResources>
  <!-- Editable user database that can also be used by
        UserDatabaseRealm to authenticate users
    -->
  <!-- 여기에 서버 미리에 설정할 리소스를 설정해준다. -->
  <Resource driverClassName="org.mariadb.jdbc.Driver" 로드할 드라이버
        url="jdbc:mariadb://localhost:3306/miniproj_db" 디비URL
        username="bituser" 디비 접속 정보
        password="1004" 디비 비밀번호
        name="jdbc/mini_proj1_db" 해당 자원의 이름
        type="javax.sql.DataSource" 해당 자원의 타입, 커넥션풀을 위해서 DataSource를 씬
        maxActive="500" 최대 500개의 커넥션을 가진다는 의미
        maxIdle="100" 최소 100개가 돌아가게 보장하겠다.
        maxWait="5000" /> 5초 기다리겠다.
  </GlobalNamingResources>
```

```
<!-- 서버/context.xml-->
<ResourceLink
  global="jdbc/mini_proj1_db"
  name="jdbc/mini_proj1_db"
  type="javax.sql.DataSource" />
```



Connection Pool

프로젝트의 `web.xml`에 서버의 리소스를 사용하기 위한 정보를 설정해야한다.

- `<res-ref-name>` : 사용하고자 하는 리소스
- `<res-type>` : 사용하고자 하는 리소스의 타입
- `<res-auth>` : 리소스에 대한 권한

XML ▾

```
<!--web.xml-->
<resource-ref>
  <description>DataSource example</description>
  <res-ref-name>jdbc/mini_proj1_db</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

Connection Pool

DB와 Connection하는 코드를
BaseDAO라는 부모 클래스에 작성

BoardDAO, MemberDAO 등의 DAO는
BaseDAO를 상속

DataSource를 사용해 커넥션 풀 관리

- Connection Pool을 관리하는 객체
- DataSource 객체를 통해서 필요한 Connection을 획득, 반납 등의 작업

```
/**
 * 모든 DAO가 공유해서 사용할 Connection
 */
public class BaseDAO {
    public static Connection conn = null;
    static {
        try {
            // Connection Pool을 사용한다.
            InitialContext ic= new InitialContext();
            //2. lookup()
            DataSource ds = (DataSource) ic.lookup("java:comp/env/jdbc/mini_proj1_db");
            //3. getConnection()
            conn = ds.getConnection();
            conn.setAutoCommit(false); // auto Commit false
            System.out.println("MariaDB 연결 성공");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

화면에 출력하는 방법

회원의 필드 중 Gender를 Gender라는 Enum 타입을 사용

- DB 테이블에서 Gender를 저장할 때, **varchar(1)**, 'F' or 'M' 제약조건
- 하지만 화면에는 female과 male로 표시하고 싶음
- 요청을 받을 때도, F와 M이 아니라 female과 male로 받고 싶음

```
private Gender gender;  
public static enum Gender {  
    F,  
    M  
}
```

```
action: "insert"  
address: "서울시 동대문구 새동"  
gender: "female"  
hobbies: {1: "요리", 2: "낚시", 3: "게임"}  
id: "new1"  
name: "새사람"  
password: "1234"  
phone: "010-1234-1234"
```

```
CREATE TABLE tb_members (  
    id varchar(100) primary key,  
    name varchar(20) not null,  
    password varchar(100) not null,  
    address varchar(255),  
    phone varchar(100),  
    gender varchar(1) not null,  
    constraint ck_gender check (gender in ('M','F'))  
);
```

화면에 출력하는 방법

데이터를 바인딩하기 위해서 사용하는

Jackson은 **getter**와 **Setter**를 사용해 매핑함

ex) private String name;

- getName()
- setName(String name)

따라서 내가 원하는 값으로 **get**하고 **set**하기
위해 **getter**와 **setter**를 정의

DB에서는 'F'와 'M'으로만 저장하기 때문에
DB에 insert를 할 때 사용할 **getDBGender()**

```
public String getGender() {  
    if(this.gender.equals(Gender.F)) return "female";  
    return "male";  
}  
  
public void setGender(String gender) {  
    if(gender.equals("female")) this.gender = Gender.F;  
    else this.gender = Gender.M;  
}  
  
public String getDBGender() {  
    if(this.gender.equals(Gender.F)) {  
        return "F";  
    }  
    return "M";  
}
```

화면에 출력하는 방법

화면에 출력되는 모습

회원 리스트

아이디	이름	성별
ADMIN	관리자	male
female	여자1	female
female2	여자1	female
female3	여자1	female
female4	여자	female
male2	남자	male
male4	남자	male
male5	남자	male
male6	남자	male
TEST2	박덕구	male

데이터 베이스

id	gender
ADMIN	M
female	F
female2	F
female3	F
female4	F

프론트에 보내주는 데이터 형식

```
<label>성별:</label>
<input type="radio" id="female" name="gender" value="female" checked>
<label for="female">여성</label>
<input type="radio" id="male" name="gender" value="male">
<label for="male">남성</label>
```


취미를 요청하는 방법

유저가 가진 취미를 의미하는 필드

- `Map<Integer, String> hobbies;`

Map 형식이기 때문에 view에서 hobbies에 대한 요청을 보낼 때, 형식도 Map 구조를 가지고 있어야함

그렇지않다면 우측처럼 ObjectMapper에서 예외가 발생된다.

```
public class MemberVO {  
    private String id;  
    private String name;  
    private String password;  
    private String address;  
    private String phone;  
    private Gender gender;  
    private Map<Integer, String> hobbies; // 모든 취미
```

```
SEVERE: 경로가 []인 컨텍스트의 서블릿  
[com.miniproj1.members.MemberServlet]을(를) 위한 Servlet.service() 호  
출이 예외를 발생시켰습니다.  
com.fasterxml.jackson.databind.exc.MismatchedInputException: Cannot  
deserialize value of type  
java.util.LinkedHashMap<java.lang.Integer,java.lang.String> from  
Array value (token JsonToken.START_ARRAY)  
at [Source: REDACTED  
(StreamReadFeature.INCLUDE_SOURCE_IN_LOCATION disabled); line: 7,  
column: 16] (through reference chain:  
com.miniproj1.members.MemberVO["hobbies"])
```

취미를 요청하는 방법

common.js의 formToSerialize 메서드

- 기존의 코드는 배열로 반환
- input의 type이 checkbox인 경우 내가 필요한 코드는 키-값 구조

기존 메서드에서 checkbox 타입인 경우를 if문으로 별도 처리

```
{
  ... "id": "new1",
  ... "name": "새로운사람",
  ... "phone": "010-2222-2222",
  ... "password": "1111",
  ... "address": "서울시 동대문구",
  ... "hobbies": {
    ... "1": "요리",
    ... "2": "낚시"
  },
  ... "action": "insert"
}
```

```
//radio와 checkbox인 경우는 반드시 선택된 것만 대상으로함
if (element.type == 'radio' || element.type == 'checkbox') {
  if (element.checked) {
    if (typeof data[element.name] == 'undefined') {
      //문자열 1건 추가
      data[element.name] = element.value;
    } else if (typeof data[element.name] == 'string') {
      //문자열 값을 배열로 변경
      data[element.name] = [data[element.name], element.value];
    } else if (typeof data[element.name] == 'object') {
      //배열에 문자열 값을 추가
      data[element.name].push(element.value);
    }
  }
}
```

```
// checkbox는 map구조로 받고 싶음
} else if (element.type === 'checkbox') {
  if (element.checked) { // 체크된 요소만 전송
    if (typeof data[element.name] === 'undefined') {
      data[element.name] = {};
      data[element.name][element.id] = element.value;
    } else {
      data[element.name][element.id] = element.value;
    }
  }
}
```

취미를 요청하는 방법

다 괜찮은 줄 알았지만...

checkbox가 하나인 경우에서 문제가 발생

- ex. 자동 로그인 체크

수정한 formToSerialize 를 사용하면

autoLogin: {autoLogin:"Y"} 라는 객체로
만들어져서 요청을 보내게 됨

그러나 MemberVO에 있는 자동 로그인 체크
여부에 대한 필드의 타입은 String

=> ObjectMapper에서 다시 예외 발생

LOGIN

아이디:

비밀번호:

로그인

자동로그인체크 ☐

```
public class MemberVO {  
    private String id;  
    private String name;  
    private String password;  
    private String address;  
    private String phone;  
    private Gender gender;  
    private Map<Integer, String> hobbies; // 모든 취미  
  
    // 자동 로그인을 위한 UUID  
    private String memberUUID;  
    // 자동로그인 체크 여부  
    private String autoLogin;  
}
```

취미를 요청하는 방법

해결 방법

1. formToSerialize를 쓰지 말고 하나하나 input의 요소를 가져와 직접 값을 JSON.stringify 해주기
2. formToSerialize를 수정하기
3. MemberVO를 변경하기

html의 dataset을 활용

- 체크박스가 하나인 경우 dataset을 활용해 input에 속성으로 data-single 값을 설정
- 만약 dataset에 single이 있는 경우에 다르게 처리

느낀점 : 그냥 배열로 할 걸

```
<div>
  <label for="autoLogin">자동로그인체크</label>
  <!-- 체크박스가 하나인 경우 -->
  <input type="checkbox" data-single="true" name="autoLogin" value="Y">
</div>
```

```
// checkbox는 map구조로 받고 싶음
} else if (element.type === 'checkbox') {
  if (element.checked) { // 체크된 요소만 전송
    if (typeof data[element.name] === 'undefined') {
      data[element.name] = {};
      // id가 없는 경우를
      if(element.dataset.single){
        data[element.name] = element.value;
        return data;
      }
      data[element.name][element.id] = element.value;
    } else {
      data[element.name][element.id] = element.value;
    }
  }
}
```

후기

후기

- 금방 끝날 줄 알았지만, 생각해야할 것들이 많았던 프로젝트
- JSP를 사용하는 게 많이 까다로움
- 다음 프로젝트는 예외 처리에 더 신경쓰고 싶음
- CSS 부분을 거의 하지않음 😊
 - 그럼에도 정말 바빴음!

 RATTY

[회사소개](#) [로그아웃](#) [마이페이지](#) [회원관리](#) [게시판](#)

회원 리스트

[새 글 작성하기](#)

[검색](#)

no	제목	작성자	작성일	조회수
3	아영 최고!	박덕구	2024-03-31 17:17:55	0
4	아영 최고!	박덕구	2024-03-31 17:19:08	0
5	아영 최고!	박덕구	2024-03-31 17:19:08	0
6	아영 최고!	박덕구	2024-03-31 17:19:08	0
7	아영 최고!	박덕구	2024-03-31 17:19:09	0

감사합니다.