

Rest API 제작하기

프로젝트명 : B13RestAPI

준비사항

- 의존설정 : Spring Web, Lombok, JDBC API, Oracle Driver, Mybatis Framework
- JSP 사용을 위한 설정을 한다.
- Refresh Gradle Project 를 눌러 적용한다.

REST API란..??

- REST(REpresentational State Transfer)
 - 자원을 이름으로 구분하여 해당 자원의 상태를 주고받는 모든 것
- API(Application Programming Interface)
 - 애플리케이션 프로그래밍 인터페이스
- 즉, REST API는 컴퓨터와 컴퓨터, 서버와 클라이언트 등 다양한 애플리케이션 연결 구조에서 프로그래밍 인터페이스 규격에 맞춰 자원의 이름으로 구분하여 자원의 상태를 주고받는 행위를 말한다.

JSON의 4가지 형식 이해하기

뷰 : webapp/WEB-INF/views/home.jsp

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <%@ taglib prefix="c" uri="jakarta.tags.core" %>
4 <!DOCTYPE html>
5 <html>
6 <head>
7 <meta charset="UTF-8">
8 <title>Insert title here</title>
9 </head>
10 <body>
11     <h2>JSON의 4가지 형식 이해하기</h2>
12     <ul>
13         <li><a href="/">루트</a></li>
14         <li><a href="/restApi01.do">순수 객체*</a></li>
15         <li><a href="/restApi02.do">순수 배열*</a></li>
16         <li><a href="/restApi03.do">배열을 포함한 객체***</a></li>
17         <li><a href="/restApi04.do">객체를 포함한 배열**</a></li>
18     </ul>
```

DTO : com.edu.springboot.jsonype4.PersonVO.java

```
1 package com.edu.springboot.jsonype4;
2
3+ import lombok.AllArgsConstructor;
4
5
6 @Data
7 @AllArgsConstructor
8 public class PersonVO {
9     private String name;
10    private int age;
11    private String job;
12 }
```

컨트롤러 : com.edu.springboot.jsonype4.MyRestController.java

```
1 package com.edu.springboot.jsonype4;
2
3+ import java.util.ArrayList;
4
11
12 @RestController
13 public class MyRestController {
14
15     //JSON객체
16+ @GetMapping("/restApi01.do")
17     public Map<String, Object> restApi01(){
18         Map<String, Object> maps = new HashMap<>();
19
20         maps.put("key01", "홍길동");
21         maps.put("key02", "유비");
22         maps.put("key03", "손오공");
23         maps.put("key04", "강백호");
24         maps.put("key05", "둘리");
25
26         return maps;
27     }
```

```

29 //JSON배열
30 @GetMapping("/restApi02.do")
31 public List<String> restApi02(){
32     List<String> lists = new ArrayList<>();
33
34     lists.add("홍길동");
35     lists.add("유비");
36     lists.add("손오공");
37     lists.add("강백호");
38     lists.add("둘리");
39
40     return lists;
41 }
42

```

```

43 //JSON배열을 포함한 객체
44 @GetMapping("/restApi03.do")
45 public Map<String, Object> restApi03(){
46     Map<String, Object> maps = new HashMap<>();
47
48     List<String> lists1 =
49         new ArrayList<>(Arrays.asList("이순신", "세종대왕", "신사임당"));
50     maps.put("한국의위인", lists1);
51
52     List<String> lists2 =
53         new ArrayList<>(Arrays.asList("유비", "관우", "장비"));
54     maps.put("삼국지", lists2);
55
56     List<String> lists3 =
57         new ArrayList<>(Arrays.asList("손오공", "저팔계", "사오정"));
58     maps.put("서유기", lists3);
59
60     return maps;
61 }

```

```

63 //JSON객체를 포함한 배열
64 @GetMapping("/restApi04.do")
65 public List<PersonVO> restApi04(){
66     List<PersonVO> lists = new ArrayList<>();
67
68     PersonVO vo1 = new PersonVO("강백호", 21, "파워포워드");
69     PersonVO vo2 = new PersonVO("서태웅", 21, "파워포워드");
70     PersonVO vo3 = new PersonVO("송태섭", 22, "포인트가드");
71     PersonVO vo4 = new PersonVO("정대만", 23, "슈팅가드");
72     PersonVO vo5 = new PersonVO("채치수", 24, "센터");
73

```

```
74         lists.add(vo1);
75         lists.add(vo2);
76         lists.add(vo3);
77         lists.add(vo4);
78         lists.add(vo5);
79
80         return lists;
81     }
82 }
```

여기까지 작성하세요.

게시판 구현을 위한 Rest API 만들기 위한 준비

프로퍼티 파일 : resources/application.properties

```
8 # oracle 설정
9 spring.datasource.driver-class-name=oracle.jdbc.OracleDriver
10 spring.datasource.url=jdbc:oracle:thin:@localhost:1521:xe
11 spring.datasource.username=musthave
12 spring.datasource.password=1234
13
14 # mybatis 매퍼
15 mybatis.mapper-locations=classpath:mybatis/mapper/**/*.xml
```

매퍼 : resources/mybatis/mapper/BoardDAO.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <!-- Mapper파일을 구성하기 위한 Doctype 설정 -->
4 <!DOCTYPE mapper
5     PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
6     "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
7
8 <!-- 서비스를 인터페이스를 namespace로 지정한다. -->
9 <mapper namespace="com.edu.springboot.restboard.IBoardService">
10
```

인터페이스 : com.edu.springboot.restboard.IBoardService.java

```
1 package com.edu.springboot.restboard;
2
3 import java.util.ArrayList;
4
5
6
7 @Mapper
8 public interface IBoardService {
9     //게시물 갯수
10     public int totalCount();
11     //게시물 가져오기
12     public ArrayList<BoardDTO> list(ParameterDTO parameterDTO);
13     //게시물 검색하기
14     public ArrayList<BoardDTO> search(ParameterDTO parameterDTO);
15     //게시물 내용보기
16     public BoardDTO view(ParameterDTO parameterDTO);
17 }
18
```

여기까지 작성하세요.

Rest API 만들어보기1 - 페이지별 레코드 조회

뷰 : webapp/WEB-INF/views/home.jsp

```
20      <h2>Rest API 만들어보기</h2>
21      <style>
22      fieldset{width:400px;}
23      </style>
24      <fieldset>
25          <legend>페이지별 레코드 조회</legend>
26          <form action="restBoardList.do">
27              <select name="pageNum">
28                  <c:forEach begin="1" end="10" var="p">
29                      <option value="{p}">{p}</option>
30                  </c:forEach>
31              </select>
32              <input type="submit" value="요청하기" />
33          </form>
34      </fieldset>
```

DTO1 : com.edu.springboot.restboard.BoardDTO.java

```
1 package com.edu.springboot.restboard;
2
3 import lombok.Data;
4
5 @Data
6 public class BoardDTO {
7     private String num;
8     private String title;
9     private String content;
10    private String id;
11    private java.sql.Date postdate;
12    private String visitcount;
13 }
```

DTO2 : com.edu.springboot.restboard.ParameterDTO.java

```
1 package com.edu.springboot.restboard;
2
3 import java.util.ArrayList;
4
5 @Data
6 public class ParameterDTO {
7     private String num;
8     private String pageNum;
9     private String searchField;
10    private ArrayList<String> searchWord;
11    private int start;
12    private int end;
13 }
```

컨트롤러 : com.edu.springboot.restboard.BoardRestController.java

```
1 package com.edu.springboot.restboard;
2
3 import java.util.List;
11
12 @RestController
13 public class BoardRestController {
14
15     @Autowired
16     IBoardService dao;
17
18     @GetMapping("/restBoardList.do")
19     public List<BoardDTO> restBoardList(ParameterDTO parameterDTO){
20         int pageSize = 10;
21         int pageNum = parameterDTO.getPageNum()==null ? 1 :
22             Integer.parseInt(parameterDTO.getPageNum());
23         int start = (pageNum-1) * pageSize + 1;
24         int end = pageNum * pageSize;
25         parameterDTO.setStart(start);
26         parameterDTO.setEnd(end);
27         List<BoardDTO> boardList = dao.list(parameterDTO);
28         return boardList;
29     }
30 }
```

매퍼 : resources/mybatis/mapper/BoardDAO.xml

```
11 <select id="list"
12     parameterType="com.edu.springboot.restboard.ParameterDTO"
13     resultType="com.edu.springboot.restboard.BoardDTO">
14     SELECT * FROM (
15         SELECT Tb.*, rownum rNum FROM (
16             SELECT * FROM board ORDER BY num DESC
17         ) Tb
18     )
19     WHERE rNum BETWEEN #{start} AND #{end}
20 </select>
21
```

여기까지 작성하세요.

Rest API 만들어보기2 - 2개 이상의 단어 검색

뷰 : webapp/WEB-INF/views/home.jsp

```
36<fieldset>
37    <legend>2개 이상의 단어 검색</legend>
38    <form action="restBoardSearch.do">
39        <select name="searchField">
40            <option value="title">제목</option>
41            <option value="content">내용</option>
42        </select>
43        <input type="text" name="searchWord" />
44        <input type="submit" value="요청하기" />
45    </form>
46</fieldset>
47
```

컨트롤러 : com.edu.springboot.restboard.BoardRestController.java

```
33@GetMapping("/restBoardSearch.do")
34public List<BoardDTO> restBoardSearch(HttpServletRequest req,
35    ParameterDTO parameterDTO){
36    //searchField는 parameterDTO가 받음
37    //searchWord는 별도로 저장해야함
38    if(req.getParameter("searchWord")!=null) {
39        String[] sTxtArray = req.getParameter("searchWord")
40            .split(" ");
41        parameterDTO.getSearchWord().clear();
42        for(String str : sTxtArray) {
43            System.out.println(str);
44            parameterDTO.getSearchWord().add(str);
45        }
46    }
47    List<BoardDTO> searchList = dao.search(parameterDTO);
48    return searchList;
49}
```

매퍼 : resources/mybatis/mapper/BoardDAO.xml

```
22<select id="search"
23    parameterType="com.edu.springboot.restboard.ParameterDTO"
24    resultType="com.edu.springboot.restboard.BoardDTO">
25    SELECT * FROM board
26    <if test="searchWord!=null">
27        WHERE
28        <foreach collection="searchWord" item="str" open="("
29            close=")" separator="or">
30            ${searchField} like '%|#{str}|%'
31        </foreach>
32    </if>
33    ORDER BY num DESC
34</select>
```


여기까지 작성하세요.

Rest API 만들어보기3 - 상세 내용 조회

뷰 : webapp/WEB-INF/views/home.jsp

```
48<=    <fieldset>
49        <legend>상세 내용 조회</legend>
50<=    <form action="restBoardView.do">
51        일련번호:<input type="number" name="num" />
52        <input type="submit" value="요청하기" />
53    </form>
54</fieldset>
55
```

컨트롤러 : com.edu.springboot.restboard.BoardRestController.java

```
54<=    @GetMapping("/restBoardView.do")
55    public BoardDTO restBoardView(ParameterDTO parameterDTO){
56        BoardDTO boardDTO = dao.view(parameterDTO);
57        return boardDTO;
58    }
59 }
```

매퍼 : resources/mybatis/mapper/BoardDAO.xml

```
36<=    <select id="view"
37        parameterType="com.edu.springboot.restboard.ParameterDTO"
38        resultType="com.edu.springboot.restboard.BoardDTO">
39        SELECT * FROM board WHERE num=#{num}
40    </select>
41
```

여기까지 작성하세요.

퀴즈]

주어진 작성품을 이용해서 글쓰기를 구현하시오.

```
<fieldset>
  <legend>[퀴즈]작성하기</legend>
  <form method="post" action="restBoardWrite.do">
    아이디:<input type="text" name="id" />
    제목:<input type="text" name="title" />
    내용: <textarea name="content" cols="30" rows="10"></textarea>
    <input type="submit" value="요청하기" />
  </form>
</fieldset>
```

- home.jsp 하단에 붙여넣기 한다.
- com.edu.springboot.restboard.BoardRestController 에서 작성한다.
- 글쓰기에 대한 결과는 아래와 같이 반환한다.
 - 성공 : {'result':1}
 - 실패 : {'result':0}

해답 [바로가기]