

SPRING-JDBC(JdbcTemplate)

프로젝트명 : B06JdbcTemplate

준비사항

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

New Spring Starter Project Dependencies

Spring Boot Version: 2.6.3

Frequently Used:

☒ Spring Web

Available: jdbc

Selected:

- X Lombok
- X JDBC API
- X Oracle Driver
- X Spring Web

SQL

- ☒ JDBC API
- ☐ Spring Data JDBC
- ☐ IBM DB2 Driver
- ☐ H2 Database
- ☐ MariaDB Driver
- ☐ MS SQL Server Driver
- ☐ MySQL Driver
- ☒ Oracle Driver
- ☐ PostgreSQL Driver

Make Default Clear Selection

< Back Next > Finish Cancel

SPRING-JDBC(Java Database Connectivity)란..??

- Spring JDBC는 스프링 프레임워크에서 제공하는 JDBC 기반의 데이터 액세스 기술
- JDBC는 자바 언어를 사용하여 관계형 데이터베이스에 접근하고 조작하기 위한 표준 API
- Spring JDBC는 JDBC를 보다 쉽고 효율적으로 사용할 수 있도록 추상화된 기능을 제공
- 이를 통해 개발자는 반복적이고 번거로운 JDBC 작업을 간소화하고 생산성을 향상시킬 수 있다.

JdbcTemplate 주요메서드

queryForObject()	단일 결과를 반환하고 해당 결과를 자바 객체로 매핑할 때 사용 Ex) jdbcTemplate.queryForObject(쿼리문, RowMapper, 인파라미터)
query()	여러행을 반환하는 select 쿼리문 실행시 주로 사용 Ex) jdbcTemplate.query(쿼리문, RowMapper)
update()	INSERT, UPDATE, DELETE와 같은 데이터 변경 쿼리를 실행할 때 사용 Ex) jdbcTemplate.update(쿼리문)

◆ JDBC 작업을 위한 준비

/B04JdbcTemplate/build.gradle

우리는 Oracle21c 를 사용하고 있으므로 ojdbc11로 수정

```
2 dependencies {
3     implementation 'org.springframework.boot:spring-boot-starter-jdbc'
4     implementation 'org.springframework.boot:spring-boot-starter-web'
5     compileOnly 'org.projectlombok:lombok'
6     runtimeOnly 'com.oracle.database.jdbc:ojdbc11'
7     annotationProcessor 'org.projectlombok:lombok'
8     providedRuntime 'org.springframework.boot:spring-boot-starter-tomcat'
9     testImplementation 'org.springframework.boot:spring-boot-starter-test'
10    testRuntimeOnly 'org.junit.platform:junit-platform-launcher'
11
12    //JSP설정
13    implementation 'jakarta.servlet:jakarta.servlet-api'
14    implementation 'jakarta.servlet.jsp.jstl:jakarta.servlet.jsp.jstl-api'
15    implementation 'org.glassfish.web:jakarta.servlet.jsp.jstl'
16    implementation 'org.apache.tomcat.embed:tomcat-embed-jasper'
17}
```

프로퍼티 : resources/application.properties

```
1 # 포트 설정
2 server.port=8586
3
4 # JSP 설정
5 spring.mvc.view.prefix=/WEB-INF/views/
6 spring.mvc.view.suffix=.jsp
7
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
```

DTO : com.edu.springboot.jdbc.MemberDTO.java

```
1 package com.edu.springboot.jdbc;
2
3 import lombok.Data;
4
5 @Data
6 public class MemberDTO {
7     private String id;
8     private String pass;
9     private String name;
10    private String regidate;
11 }
```

서비스 : com.edu.springboot.jdbc.IMemberService.java

```
1 package com.edu.springboot.jdbc;
2
3 import java.util.List;
4
5
6
7 @Service
8 public interface IMemberService {
9
10     public List<MemberDTO> select();
11     public int insert(MemberDTO memberDTO);
12     public MemberDTO selectOne(MemberDTO memberDTO);
13     public int update(MemberDTO memberDTO);
14     public int delete(MemberDTO memberDTO);
15 }
```

여기까지 작성하세요.

◆ 회원목록

home : [webapp/WEB-INF/views/home.jsp](#)

```
<body>
  <h2>스프링 부트 프로젝트</h2>
  <ul>
    <li><a href="/">루트</a></li>
  </ul>

  <h2>JdbcTemplate으로 구현한 회원관리</h2>
  <ul>
    <li><a href="/list.do">회원목록</a></li>
  </ul>
</body>
```

컨트롤러 : [java/com.edu.springboot.MainController.java](#)

```
1 package com.edu.springboot;
2
3+ import org.springframework.beans.factory.annotation.Autowired;
11
12 @Controller
13 public class MainController {
14
15     @Autowired
16     IMemberService dao;
17
18     @RequestMapping("/")
19     public String home() {
20         return "home";
21     }
22
23     //회원목록
24     @RequestMapping("/list.do")
25     public String member2(Model model) {
26         model.addAttribute("memberList", dao.select());
27         return "list";
28     }
29 }
```

DAO : com.edu.springboot.jdbc.MemberDAO.java

```
1 package com.edu.springboot.jdbc;
2
3 import java.sql.Connection;
15
16 @Repository
17 public class MemberDAO implements IMemberService {
18
19     @Autowired
20     private JdbcTemplate jdbcTemplate;
21
22     //회원목록
23     @Override
24     public List<MemberDTO> select() {
25         String sql = "select * from member order by regidate desc";
26
27         return jdbcTemplate.query(sql,
28             new BeanPropertyRowMapper<MemberDTO>(MemberDTO.class));
29     }
30
```

뷰 : webapp/WEB-INF/views/list.jsp

```
10 <body>
11     <h2>회원리스트</h2>
12     <table border="1">
13         <tr>
14             <th>아이디</th>
15             <th>패스워드</th>
16             <th>이름</th>
17             <th>가입일</th>
18             <th></th>
19         </tr>
20
21         <c:forEach items="${memberList}" var="row" varStatus="loop">
22             <tr>
23                 <td>${row.id}</td>
24                 <td>${row.pass}</td>
25                 <td>${row.name}</td>
26                 <td>${row.regidate}</td>
27                 <td>
28                     <a href="edit.do?id=${row.id}">수정</a>
29                     <a href="delete.do?id=${row.id}">삭제</a>
30                 </td>
31             </tr>
32         </c:forEach>
33     </table>
34     <a href="regist.do">회원등록</a>
35 </body>
</html>
```

여기까지 작성하세요.

◆ 회원등록

컨트롤러 : com.edu.springboot.MainController.java

```
30 //회원등록
31 @RequestMapping(value="/regist.do", method=RequestMethod.GET)
32 public String member1() {
33     return "regist";
34 }
35 @RequestMapping(value="/regist.do", method=RequestMethod.POST)
36 public String member6(MemberDTO memberDTO) {
37     int result = dao.insert(memberDTO);
38     if(result==1) System.out.println("입력되었습니다.");
39     return "redirect:list.do";
40 }
```

뷰 : webapp/WEB-INF/views/regist.jsp

```
10 <body>
11     <h2>회원등록</h2>
12     <form action="regist.do" method="post">
13     <table border="1">
14         <tr>
15             <th>아이디</th>
16             <td><input type="text" name="id" value="" /></td>
17         </tr>
18         <tr>
19             <th>패스워드</th>
20             <td><input type="text" name="pass" value="" /></td>
21         </tr>
22         <tr>
23             <th>이름</th>
24             <td><input type="text" name="name" value="" /></td>
25         </tr>
26     </table>
27     <input type="submit" value="전송하기" />
28 </form>
29 </body>
```

DAO : com.edu.springboot.jdbc.MemberDAO.java

```
31 //회원등록
32 @Override
33 public int insert(MemberDTO memberDTO) {
34     int result = jdbcTemplate.update(new PreparedStatementCreator() {
35         @Override
36         public PreparedStatement createPreparedStatement(Connection con)
37             throws SQLException {
38             String sql = "insert into member (id, pass, name) "
39                 + " values (?, ?, ?)";
40
41             PreparedStatement psmt = con.prepareStatement(sql);
42
43             psmt.setString(1, memberDTO.getId());
44             psmt.setString(2, memberDTO.getPass());
45             psmt.setString(3, memberDTO.getName());
46
47             return psmt;
48         }
49     });
50     return result;
51 }
```

여기까지 작성하세요.

◆ 회원정보수정1 - 기존 정보 불러오기

컨트롤러 : com.edu.springboot.MainController.java

```
42 //회원수정
43 @RequestMapping(value="/edit.do", method=RequestMethod.GET)
44 public String member3(MemberDTO memberDTO, Model model) {
45     memberDTO = dao.selectOne(memberDTO);
46     model.addAttribute("dto", memberDTO);
47     return "edit";
48 }
```

DAO : com.edu.springboot.jdbc.MemberDAO.java

```
53 //회원조회
54 @Override
55 public MemberDTO selectOne(MemberDTO memberDTO) {
56     String sql = "select * from member where id=?";
57     memberDTO = jdbcTemplate.queryForObject(sql,
58         new BeanPropertyRowMapper<MemberDTO>(MemberDTO.class),
59         new Object[] { memberDTO.getId() });
60     return memberDTO;
61 }
```

뷰 : webapp/WEB-INF/views/edit.jsp

```
10 <body>
11     <h2>회원수정</h2>
12     <form action="edit.do" method="post">
13     <table border="1">
14         <tr>
15             <th>아이디(수정불가)</th>
16             <td><input type="text" name="id" value="${dto.id}" readonly/></td>
17         </tr>
18         <tr>
19             <th>패스워드</th>
20             <td><input type="text" name="pass" value="${dto.pass}" /></td>
21         </tr>
22         <tr>
23             <th>이름</th>
24             <td><input type="text" name="name" value="${dto.name}" /></td>
25         </tr>
26     </table>
27     <input type="submit" value="전송하기" />
28 </form>
29 </body>
```

여기까지 작성하세요.

◆ 회원정보수정2 - 수정 처리하기

컨트롤러 : com.edu.springboot.MainController.java

```
49 @RequestMapping(value="/edit.do", method=RequestMethod.POST)
50 public String member7(MemberDTO memberDTO) {
51     int result = dao.update(memberDTO);
52     if(result==1) System.out.println("수정되었습니다.");
53     return "redirect:list.do";
54 }
55
```

DAO : com.edu.springboot.jdbc.MemberDAO.java

```
63 //회원수정
64 @Override
65 public int update(MemberDTO memberDTO) {
66     String sql = "update member set pass=?, name=? where id=?";
67     int result = jdbcTemplate.update(sql, new PreparedStatementSetter() {
68         @Override
69         public void setValues(PreparedStatement ps) throws SQLException {
70             ps.setString(1, memberDTO.getPass());
71             ps.setString(2, memberDTO.getName());
72             ps.setString(3, memberDTO.getId());
73         }
74     });
75     return result;
76 }
77
```

여기까지 작성하세요.

◆ 회원삭제

컨트롤러 : com.edu.springboot.MainController.java

```
56 //회원삭제
57 @RequestMapping("/delete.do")
58 public String member4(MemberDTO memberDTO) {
59     int result = dao.delete(memberDTO);
60     if(result==1) System.out.println("삭제되었습니다.");
61     return "redirect:list.do";
62 }
63 }
```

DAO : com.edu.springboot.jdbc.MemberDAO.java

```
78 //회원삭제
79 @Override
80 public int delete(MemberDTO memberDTO) {
81     String sql = "delete from member where id=?";
82     int result = jdbcTemplate.update(sql,
83         new Object[] {memberDTO.getId()});
84     return result;
85 }
86
87 }
```

퀴즈1]

기존 프로젝트를 복사한 후 **B06JdbcTemplateQuiz** 로 변경한다.

회원 목록에 다음과 같이 검색 기능을 추가하시오.

회원리스트

아이디 ▼

검색

아이디	패스워드	이름	가입일	
cccc	cccc	cccc	2023-11-01 15:41:40	수정 삭제
ee	ee	ee	2023-10-26 16:05:56	수정 삭제
bbbb	bbbb	bbbb	2023-10-25 12:57:45	수정 삭제
aaaa	aaaa	aaaa	2023-10-25 12:37:32	수정 삭제

검색어 입력 후 Enter를 누르면 다음과 같이 검색된다.

← → ↻ 🏠 ⓘ localhost:8585/list.do?searchField=id&searchKeyword=nakja ☆

회원리스트

아이디 ▼

검색

아이디	패스워드	이름	가입일	
nakja	1234	낙자샘	2023-08-22 10:56:56	수정 삭제

[회원등록](#)

MemberDTO 클래스에 아래 2개의 멤버변수를 추가한 후 진행한다.

```
6 @Data
7 public class MemberDTO {
8     private String id;
9     private String pass;
10    private String name;
11    private String regidate;
12
13    private String searchField;
14    private String searchKeyword;
15 }
16
```

퀴즈2]

현재 회원목록의 검색폼에는 빈값에 대한 검증이 없는 상태이다. 다음과 같이 검색어가 없는 경우 경고창이 뜰수있도록 Javascript 혹은 jQuery로 구현하시오.

← → ↺ 🏠 ⓘ localhost:8585/list.do ☆ 🗨️ 📄 🗑️ 📱 👤 ⋮

회원리스트

아이디 ▼

아이디	패스워드	이	이	이
ee	ee	ee	2023-11-01 17:16:01	수정 삭제
d1	d1	d1	2023-11-01 17:14:38	수정 삭제

localhost:8585 내용:
검색어를 입력하세요

확인