
통합구현

MEMBER 테이블 CRUD 프로그램 작업

김이화

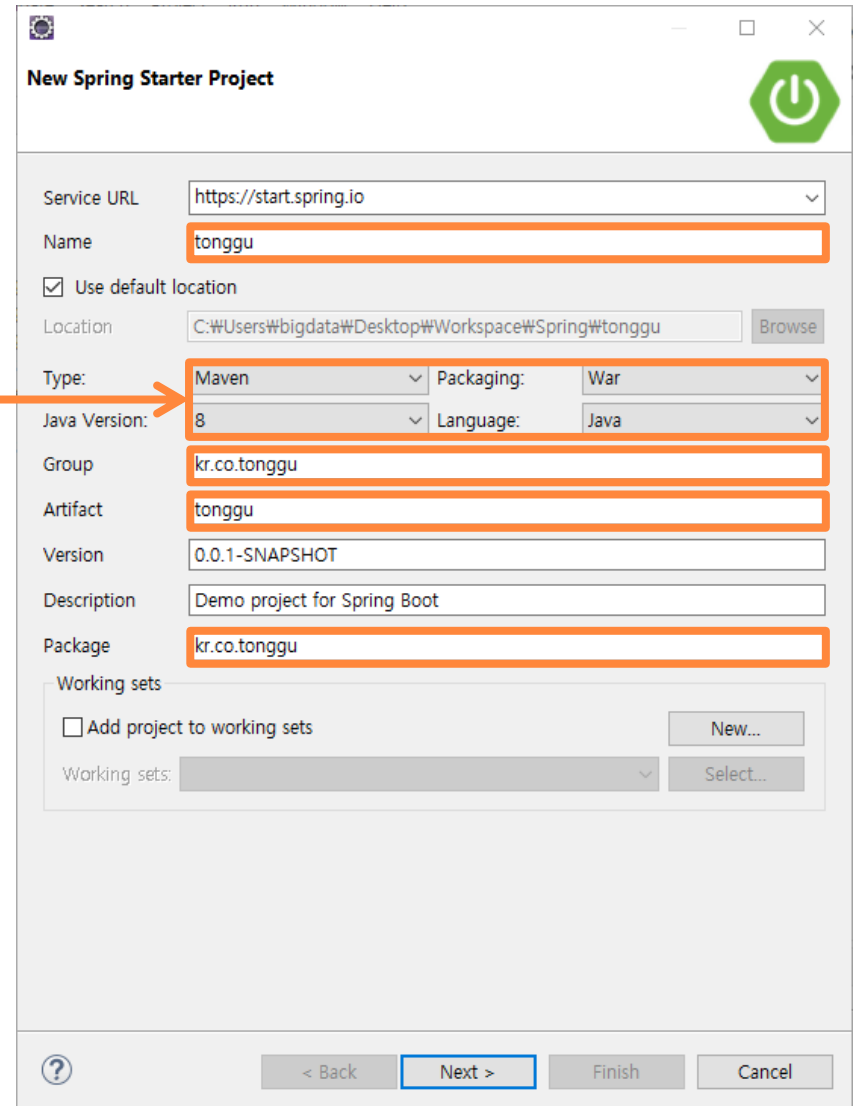
1. Spring Boot 기반 프로젝트 작업

1. Spring Starter Project 생성

- Name, Artifact명 입력

- Type: Maven
- Packaging: War
- Java Version: 8
- Language: Java

- Group, Package명 입력



New Spring Starter Project

Service URL:

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:

Package:

Working sets

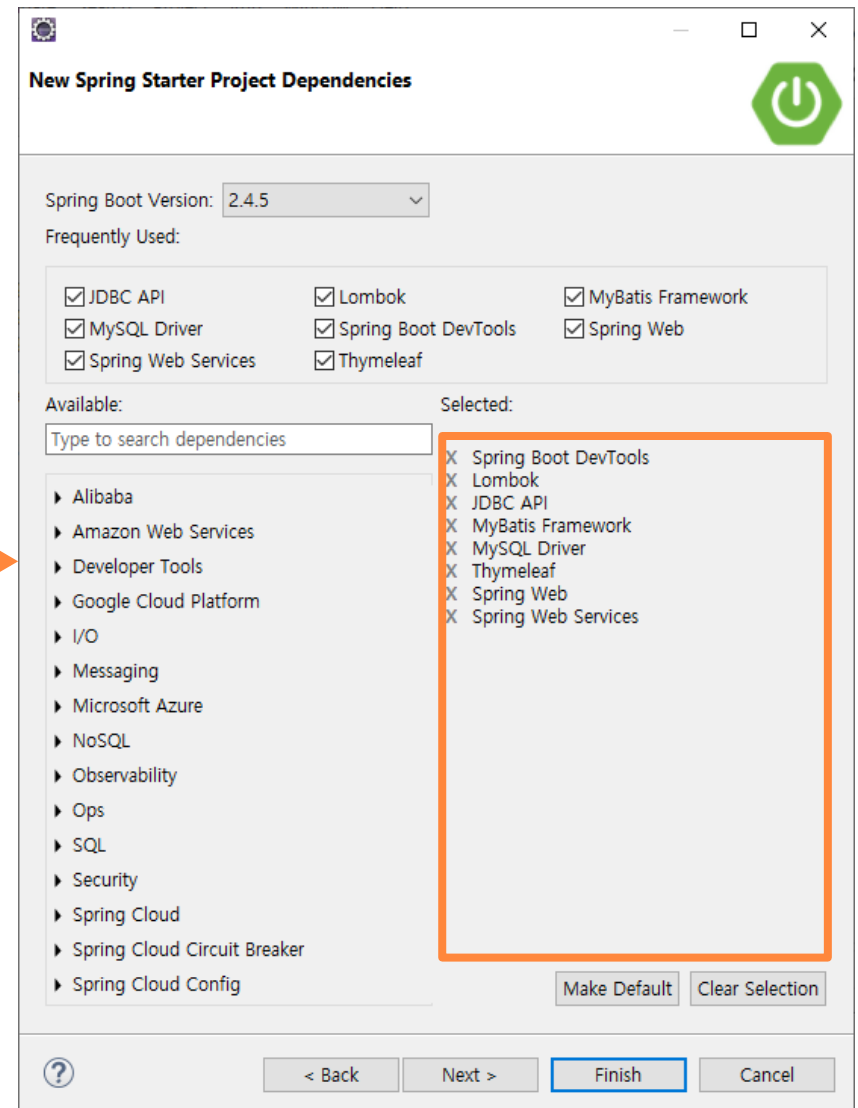
☐ Add project to working sets

Working sets:

1. Spring Boot 기반 프로젝트 작업

2. Dependency 선택

- Spring Boot DevTools
- Lombok
- JDBC API
- Mybatis Framework
- MySQL Driver
- Thymeleaf
- Spring Web
- Spring Web Services



1. Spring Boot 기반 프로젝트 작업

3. src/main/resources

- application.properties 작성

```
#livereload Setting(Must install Chrome livereload plugin First)
spring.devtools.livereload.enabled=true
spring.thymeleaf.cache=false
```

```
#Context root setting
server.servlet.context-path=/tonggu
```

작성하는 프로젝트명 입력

```
#DB Setting
spring.datasource.url=jdbc:mysql://192.168.10.114:3306/keh
spring.datasource.username=keh
spring.datasource.password=1234
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
```

실제로 사용할 데이터베이스 주소,
아이디, 비밀번호 입력

```
#Mybatis Setting(TongguApplication add @MapperScan("kr.co.tonggu.dao))
mybatis.mapper-locations=classpath:mappers/**/*.xml
```

Mybatis 설정 추가 – 매퍼 위치 지정

2. Mybatis ORM 프레임워크 사용

1. src/main/java

- 스프링 부트 메인 어플리케이션(프로젝트명Application.java)에 @MapperScan 입력

```
@SpringBootApplication
@MapperScan("kr.co.tonggu.dao")
public class TongguApplication {
```

2. src/main/resources

- mappers 폴더 생성 후 그 안에 XML Mapper 작성(member.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="kr.co.tonggu.dao.MemberDao">
```

```
<insert id="insertMember">
    INSERT INTO `MEMBER` VALUES (#{uid}, #{name}, #{hp}, #{pos}, #{dep}, NOW());
</insert>
```

매핑된 INSERT 구문

```
<select id="selectMember" resultType="kr.co.tonggu.vo.MemberVo">
    SELECT * FROM `MEMBER` WHERE `uid`=#{uid};
</select>
```

아직 생성 전이지만 작성

매핑된 SELECT 구문 (수정/삭제용)

```
<select id="selectMembers" resultType="kr.co.tonggu.vo.MemberVo">
    SELECT * FROM `MEMBER`;
</select>
```

매핑된 SELECT 구문 (목록 출력용)

```
<update id="updateMember">
    UPDATE `MEMBER` SET
        `name`=#{name},
        `hp`=#{hp},
        `pos`=#{pos},
        `dep`=#{dep}
    WHERE
        `uid`=#{uid};
</update>
```

매핑된 UPDATE 구문

```
<delete id="deleteMember">
    DELETE FROM `MEMBER` WHERE `uid`=#{uid};
</delete>
```

매핑된 DELETE 구문

```
</mapper>
```

2. Mybatis ORM 프레임워크 사용

3. src/main/java

- 다음과 같이 패키지와 파일을 생성

```
▼ > src/main/java
  ▼ > kr.co.tonggu
    > ServletInitializer.java
    > TongguApplication.java
  ▼ > kr.co.tonggu.controller
    > MemberController.java
  ▼ > kr.co.tonggu.dao
    > MemberDao.java
  ▼ > kr.co.tonggu.service
    > MemberService.java
  ▼ > kr.co.tonggu.vo
    > MemberVo.java
```

4. MemberVo.java 작성

```
@Getter
@Setter
public class MemberVo {
    private String uid;
    private String name;
    private String hp;
    private String pos;
    private int dep;
    private String rdate;
}
```

5. MemberDao.java(인터페이스) 작성

```
@Repository
public interface MemberDao {

    public void insertMember(MemberVo vo);
    public MemberVo selectMember(String uid);
    public List<MemberVo> selectMembers();
    public void updateMember(MemberVo vo);
    public void deleteMember(String uid);

}
```

6. MemberService.java 작성

```
@Service
public class MemberService {

    @Autowired
    private MemberDao dao;

    public void insertMember(MemberVo vo) {
        dao.insertMember(vo);
    }

    public MemberVo selectMember(String uid) {
        return dao.selectMember(uid);
    }

    public List<MemberVo> selectMembers() {
        return dao.selectMembers();
    }

    public void updateMember(MemberVo vo) {
        dao.updateMember(vo);
    }

    public void deleteMember(String uid) {
        dao.deleteMember(uid);
    }

}
```

2. Mybatis ORM 프레임워크 사용

7. MemberController.java 작성

```
@Controller
public class MemberController {

    @Autowired
    private MemberService service;

    @GetMapping(value = {"/", "/member/list"})
    public String list(Model model) {

        List<MemberVo> members = service.selectMembers();

        model.addAttribute("members", members);

        return "/member/list";
    }

    @GetMapping("/member/register")
    public String register() {
        return "/member/register";
    }

    @PostMapping("/member/register")
    public String register(MemberVo vo) {
        service.insertMember(vo);
        return "redirect:/member/list";
    }
}
```

```
    @GetMapping("/member/modify")
    public String modify(String uid, Model model) {

        MemberVo member = service.selectMember(uid);

        model.addAttribute(member);

        return "/member/modify";
    }

    @PostMapping("/member/modify")
    public String modify(MemberVo vo) {

        service.updateMember(vo);

        return "redirect:/member/list";
    }

    @GetMapping("/member/delete")
    public String delete(String uid) {

        service.deleteMember(uid);

        return "redirect:/member/list";
    }
}
```

3. 화면 및 기능 구현 - List

1. src/main/resources - templates - member

- list.html 파일 생성
- modify.html 파일 생성
- register.html 파일 생성

2. list.html 작성

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>list</title>
</head>
<body>
  <h3>Member 목록</h3>
  <a th:href="@{/member/register}">등록하기</a>
  <table border="1">
    <tr>
      <th>아이디</th>
      <th>이름</th>
      <th>휴대폰</th>
      <th>직급</th>
      <th>부서</th>
      <th>입사일</th>
      <th>관리</th>
    </tr>
    <tr th:each="member:${members}">
      <td th:text="${member.uid}">아이디</td>
      <td th:text="${member.name}">이름</td>
      <td th:text="${member.hp}">휴대폰</td>
      <td th:text="${member.pos}">직급</td>
      <td th:text="${member.dep}">부서</td>
      <td th:text="${member.rdate}">입사일</td>
      <td>
        <a th:href="@{/member/modify(uid=${member.uid})}">수정</a>
        <a th:href="@{/member/delete(uid=${member.uid})}">삭제</a>
      </td>
    </tr>
  </table>
</body>
</html>
```

3. 결과 화면

Member 목록

[등록하기](#)

아이디	이름	휴대폰	직급	부서	입사일	관리
a100	단군	010-1234-1000	부장	105	2021-04-19 18:13:30	수정 삭제
a101	박혁거세	010-1234-1001	부장	101	2021-03-25 14:05:05	수정 삭제
a102	김유신	010-1234-1002	차장	101	2021-03-24 14:45:33	수정 삭제
a103	김춘추	010-1234-1003	사원	101	2021-03-25 14:05:58	수정 삭제
a104	장보고	010-1234-1004	대리	102	2021-03-24 14:45:33	수정 삭제
a105	강감찬	010-1234-1005	과장	102	2021-03-24 14:45:33	수정 삭제
a106	이성계	010-1234-1006	차장	103	2021-03-24 14:45:33	수정 삭제
a107	정철	010-1234-1007	차장	103	2021-03-24 14:45:33	수정 삭제
a108	이순신	010-1234-1008	부장	104	2021-03-24 14:45:33	수정 삭제
a109	허균	010-1234-1009	부장	104	2021-03-24 14:45:33	수정 삭제
a110	정약용	010-1234-1010	사원	105	2021-03-24 14:45:33	수정 삭제
a111	박지원	010-1234-1011	사원	105	2021-03-24 14:45:33	수정 삭제

3. 화면 및 기능 구현 – Register

1. register.html 작성

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>register</title>
</head>
<body>
    <h3>Member 등록</h3>
    <a th:href="@{/}">목록으로</a>
    <form th:action="@{/member/register}" method="post">
        <table border="1">
            <tr>
                <td>아이디</td>
                <td><input type="text" name="uid"/></td>
            </tr>
            <tr>
                <td>이름</td>
                <td><input type="text" name="name"/></td>
            </tr>
            <tr>
                <td>휴대폰</td>
                <td><input type="text" name="hp"/></td>
            </tr>
            <tr>
                <td>직급</td>
                <td>
                    <select name="pos">
                        <option>사원</option>
                        <option>대리</option>
                        <option>과장</option>
                        <option>차장</option>
                        <option>부장</option>
                    </select>
                </td>
            </tr>
        </table>
    </form>
</body>
</html>
```

```
<tr>
    <td>부서</td>
    <td>
        <select name="dep">
            <option value="101">영업1부</option>
            <option value="102">영업2부</option>
            <option value="103">영업3부</option>
            <option value="104">영업지원부</option>
            <option value="105">인사부</option>
        </select>
    </td>
</tr>
<tr>
    <td colspan="2" align="right">
        <input type="submit" value="등록"/>
    </td>
</tr>
</table>
</form>
</body>
</html>
```

2. 결과 화면

Member 등록

[목록으로](#)

아이디	<input type="text"/>
이름	<input type="text"/>
휴대폰	<input type="text"/>
직급	<input type="text" value="사원"/>
부서	<input type="text" value="영업1부"/>
<input type="button" value="등록"/>	

3. 화면 및 기능 구현 – Modify

1. modify.html 작성

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>modify</title>
</head>
<body>
  <h3>Member 수정</h3>
  <a th:href="@{/}">목록으로</a>
  <form th:action="@{/member/modify}" method="post">
    <table border="1">
      <tr>
        <td>아이디</td>
        <td>
          <input type="text" name="uid" readonly th:value="${memberVo.uid}"/>
        </td>
      </tr>
      <tr>
        <td>이름</td>
        <td>
          <input type="text" name="name" th:value="${memberVo.name}"/>
        </td>
      </tr>
      <tr>
        <td>휴대폰</td>
        <td>
          <input type="text" name="hp" th:value="${memberVo.hp}"/>
        </td>
      </tr>
      <tr>
        <td>직급</td>
        <td>
          <select name="pos">
            <option>사원</option>
            <option>대리</option>
            <option>과장</option>
            <option>차장</option>
            <option>부장</option>
          </select>
        </td>
      </tr>
    </table>
  </form>
```

```
<tr>
  <td>부서</td>
  <td>
    <select name="dep">
      <option value="101">영업1부</option>
      <option value="102">영업2부</option>
      <option value="103">영업3부</option>
      <option value="104">영업지원부</option>
      <option value="105">인사부</option>
    </select>
  </td>
</tr>
<tr>
  <td colspan="2" align="right">
    <input type="submit" value="수정"/>
  </td>
</tr>
</table>
</form>
</body>
</html>
```

2. 결과 화면 Member 수정

목록으로

아이디	a100
이름	단군
휴대폰	010-1234-1000
직급	사원 ▼
부서	영업1부 ▼
<div>수정</div>	