

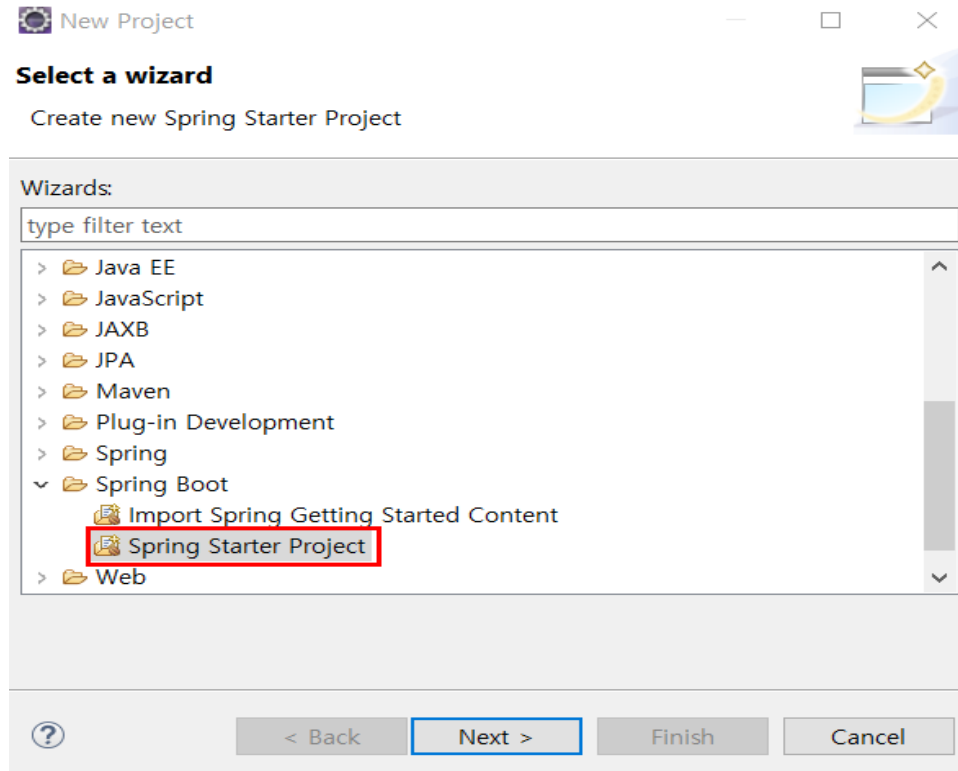
# MyBatis연동

안화수

# myBatis01 project 생성

## ❖ myBatis01 project 생성

[File] – New - Project



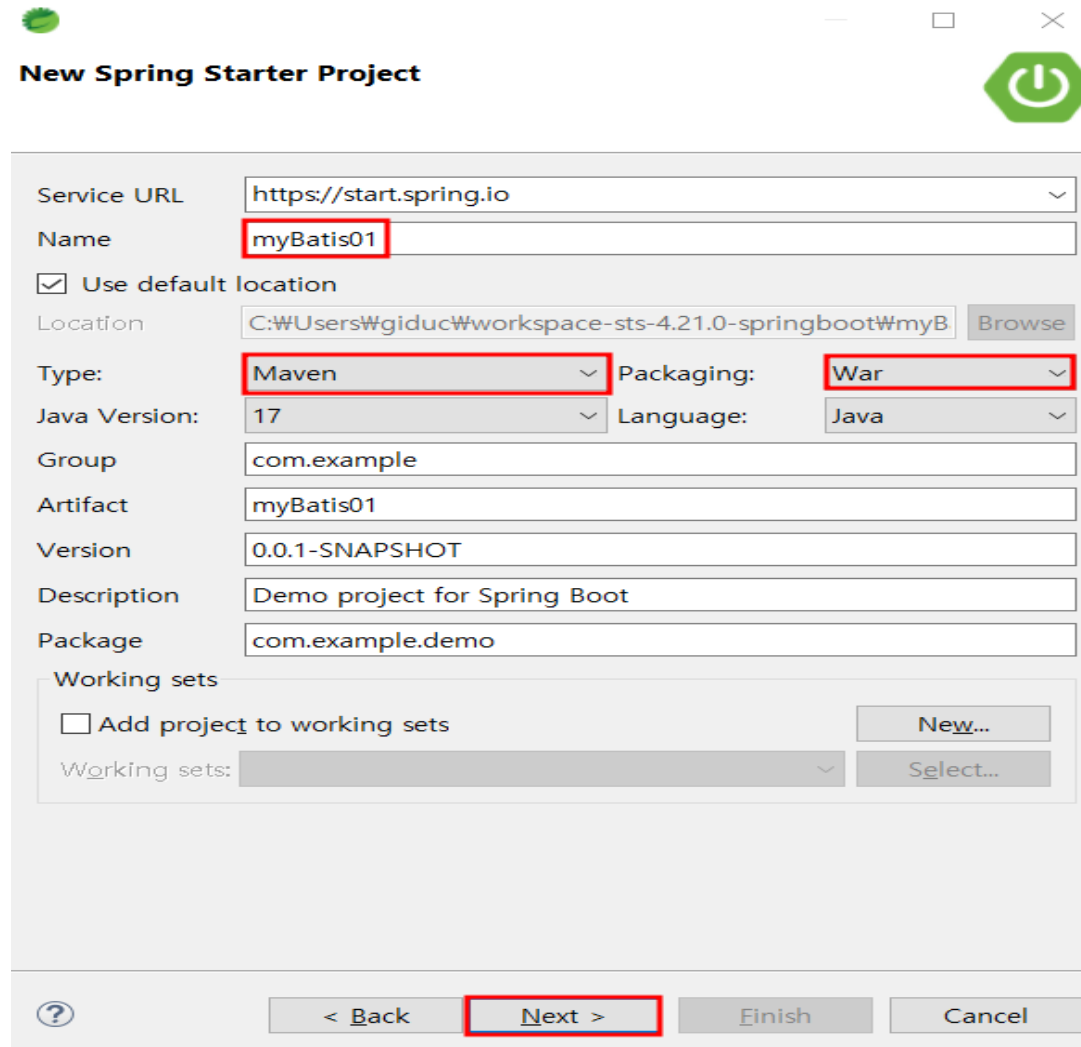
# myBatis01 project 생성

## ❖ myBatis01 project 생성

➤ Name : **myBatis01**

➤ Type : **Maven**, Gradle

➤ Packaging : **War**, Jar



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:

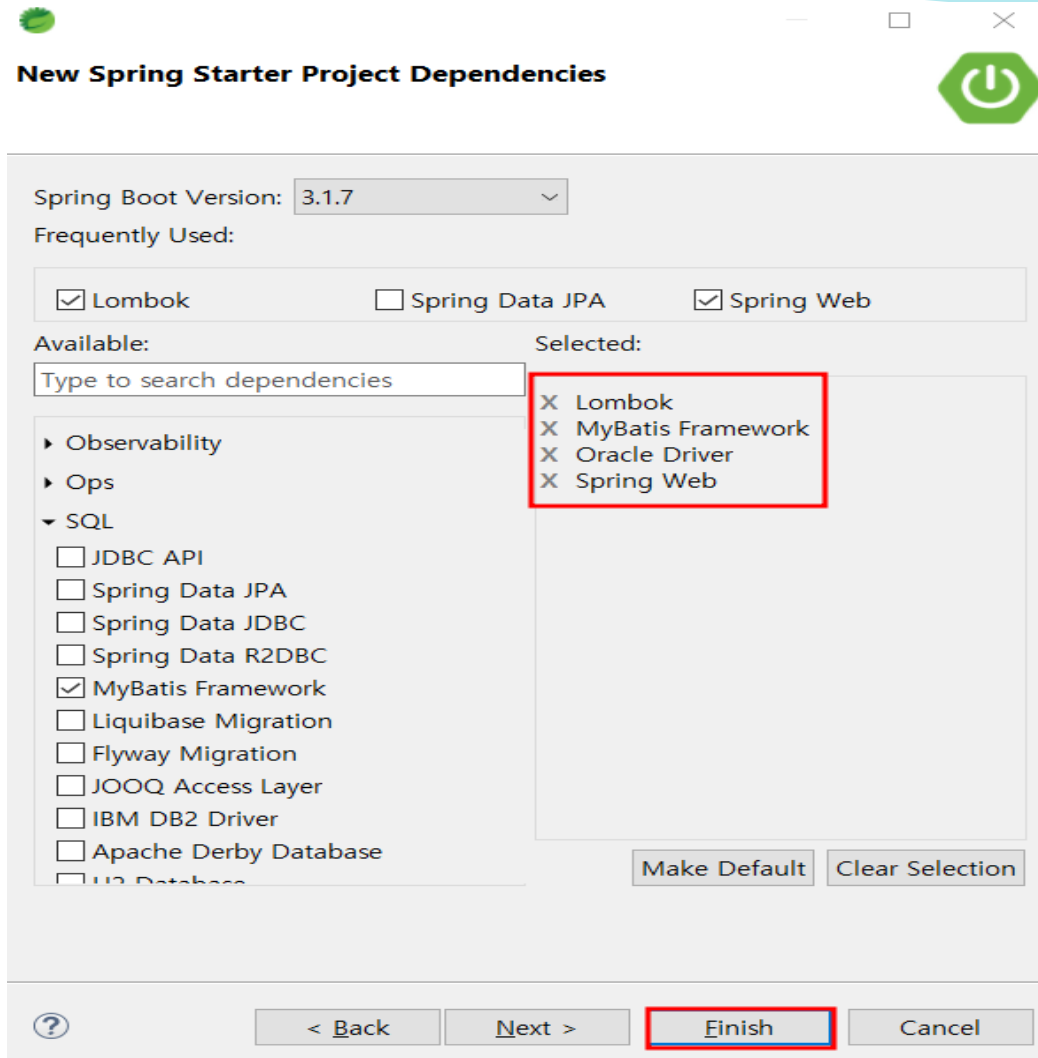
# myBatis01 project 생성

## ❖ myBatis01 project 생성

➤ Web  
Spring Web 체크

➤ SQL  
MyBatis Framework 체크  
Oracle Driver 체크

➤ Developer Tools  
Lombok 체크



# DB설계

## ❖ 테이블 및 시퀀스 생성

```
create table boardtest(  
    no number primary key,  
    name varchar2(20),  
    subject varchar2(50),  
    content varchar2(1000),  
    register date  
);
```

```
create sequence boardtest_seq;
```

# 의존 라이브러리 추가

## ❖ pom.xml

pom.xml 파일에 필요한 의존 라이브러리를 추가한다.

```
<dependencies>
  <!-- 내장 Tomcat실행시 jsp 파일을 사용하기 위한 의존 라이브러리 -->
  <dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <scope>provided</scope>
  </dependency>
  <!-- jakarta jstl -->
  <dependency>
    <groupId>org.glassfish.web</groupId>
    <artifactId>jakarta.servlet.jsp.jstl</artifactId>
    <version>3.0.0</version>
  </dependency>
</dependencies>
```

# 환경 설정파일 수정(1/2)

## ❖ 환경 설정 파일 수정

main / resources / **application.properties**

# port

server.port=80

# prefix and suffix

spring.mvc.view.prefix=/WEB-INF/views/

spring.mvc.view.suffix=.jsp

## ❖ webapp / **WEB-INF** / **views** 폴더 생성

# 환경 설정파일 수정(2/2)

## ❖ 환경 설정 파일 수정

main / resources / **application.properties**

# oracle

spring.datasource.hikari.driver-class-name=oracle.jdbc.OracleDriver

spring.datasource.hikari.jdbc-url=jdbc:oracle:thin:@localhost:1521:xe

spring.datasource.hikari.username=spring

spring.datasource.hikari.password=spring123



# DatabaseConfiguration 설정

## ❖ DatabaseConfiguration 설정 파일 생성

/main/java/com/example/demo/configuration – DatabaseConfiguration.java (1/2)

```
@Configuration
@PropertySource("classpath:/application.properties")
public class DatabaseConfiguration {

    @Autowired
    private ApplicationContext applicationContext;

    @Bean
    @ConfigurationProperties(prefix = "spring.datasource.hikari")
    public HikariConfig hikariConfig() {
        return new HikariConfig();
    }

    @Bean
    @ConfigurationProperties(prefix = "mybatis.configuration")
    public org.apache.ibatis.session.Configuration mybatisConfig() {
        return new org.apache.ibatis.session.Configuration();
    }
}
```

# DatabaseConfiguration 설정

## ❖ DatabaseConfiguration 설정 파일 생성 /main/java/com/example/demo/configuration – DatabaseConfiguration.java (2/2)

```
@Bean
public DataSource dataSource() {
    DataSource dataSource = new HikariDataSource(hikariConfig());
    return dataSource;
}

@Bean
public SqlSessionFactory sqlSessionFactory(DataSource dataSource) throws Exception {
    SqlSessionFactoryBean sqlSessionFactoryBean = new SqlSessionFactoryBean();
    sqlSessionFactoryBean.setDataSource(dataSource);
    sqlSessionFactoryBean.setMapperLocations(
        applicationContext.getResources("classpath:/mapper/*.xml"));
    sqlSessionFactoryBean.setConfiguration(mybatisConfig());
    sqlSessionFactoryBean.setTypeAliasesPackage("com.example.demo.model");
    return sqlSessionFactoryBean.getObject();
}

@Bean
public SqlSessionTemplate sqlSessionTemplate(SqlSessionFactory sqlSessionFactory) {
    return new SqlSessionTemplate(sqlSessionFactory);
}

}
```

# DTO 생성

## ❖ DTO 생성

main / java / com / example / demo / model – Board.java

@Getter

@Setter

@Alias("board") // DTO 클래스 alias 설정

public class Board {

private int no;

private String name;

private String subject;

private String content;

private Date register;

}

# Controller 생성

## ❖ Controller 생성

/main/java/com/example/demo/controller – BoardController.java

@Controller

```
public class BoardController {  
    @Autowired  
    private BoardService service;  
    @RequestMapping("boardform")  
    public String main() {  
        return "boardform";  
    }  
    @RequestMapping("boardInsert")  
    public String boardInsert(Board board) {  
        int result = service.boardInsert(board);  
        return "redirect:boardlist";  
    }  
    @RequestMapping("boardlist")  
    public String boardlist(Model model) {  
        List<Board> list = service.boardlist();  
        model.addAttribute("list", list);  
        return "boardlist";  
    }  
}
```

# Service 생성

## ❖ Service 생성

**/main/java/com/example/demo/service – BoardService.java**

```
@Service
```

```
public class BoardService {
```

```
    @Autowired
```

```
    private BoardDao dao;
```

```
    public int boardInsert(Board board) {  
        return dao.boardInsert(board);  
    }
```

```
    public List<Board> boardlist() {  
        return dao.boardlist();  
    }
```

```
}
```

# Dao 생성

## ❖ Dao 생성

**/main/java/com/example/demo/dao – BoardDao.java**

```
@Repository
public class BoardDao {

    @Autowired
    private SqlSession session;

    public int boardInsert(Board board) {
        return session.insert("insert", board);
    }

    public List<Board> boardlist() {
        return session.selectList("list");
    }
}
```

# mapper 생성

## ❖ mapper 생성

main / resources / mapper – board.xml

```
<mapper namespace="boardns">
```

```
  <insert id="insert" parameterType="board">
```

```
    insert into boardtest values(boardtest_seq.nextval,#{name},#{subject},#{content},sysdate)
```

```
  </insert>
```

```
  <select id="list" resultType="board">
```

```
    select * from boardtest
```

```
  </select>
```

```
</mapper>
```

# View 생성

## ❖ View 생성

### webapp - index.jsp 생성

```
<%@ 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>

<script>
    location.href="boardform";
</script>

</body>
</html>
```



# View 생성

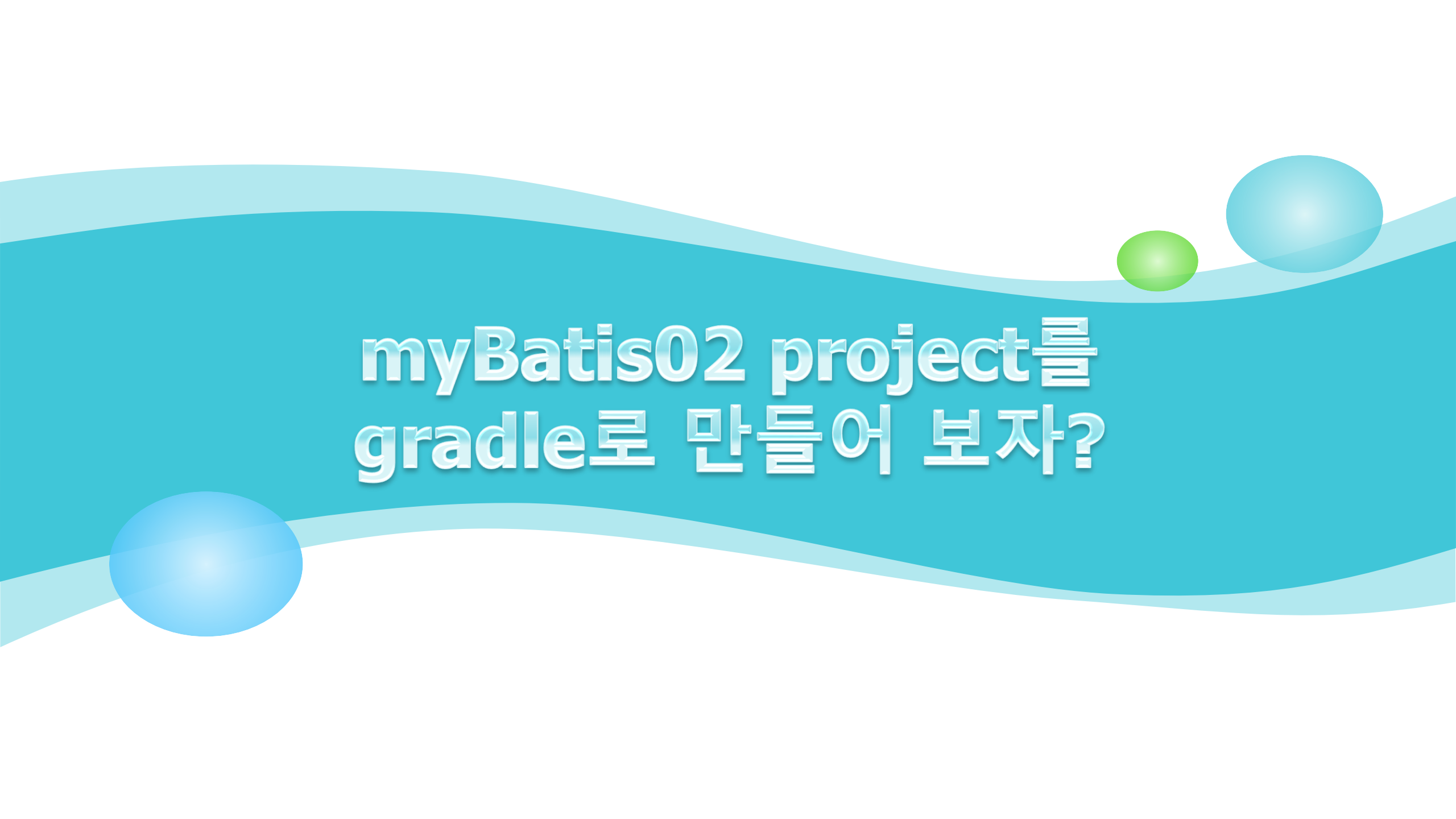
webapp / WEB-INF / views - boardform.jsp 생성

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>게시판 폼</title>
</head>
<body>
<form method="post" action="boardInsert">
<table border="1" align="center" width="500">
    <caption>글작성</caption>
    <tr><td>작성자</td>
        <td><input type="text" name="name"> </td>
    </tr>
    <tr><td>제목</td>
        <td><input type="text" name="subject"> </td>
    </tr>
    <tr><td>내용</td>
        <td><textarea cols="50" rows="5" name="content"></textarea> </td>
    </tr>
    <tr><td colspan="2" align="center">
        <input type="submit" value="글작성">
        <input type="reset" value="취소">
    </td>
    </tr>
</table>
</form>
</body>
</html>
```

# View 생성

webapp / WEB-INF / views - boardlist.jsp 생성

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>게시판 목록</title>
</head>
<body>
<a href="boardform">글작성</a> <br>
<table border=1 align=center width=800>
    <caption>게시판 목록</caption>
    <tr>
        <th>번호</th>
        <th>작성자</th>
        <th>제목</th>
        <th>날짜</th>
        <th>내용</th>
    </tr>
    <c:forEach var="b" items="${list }">
    <tr>
        <td>${b.no}</td>
        <td>${b.name}</td>
        <td>${b.subject}</td>
        <td>
            <fmt:formatDate value="${b.register}" pattern="yyyy-MM-dd HH:mm:ss"/>
        </td>
        <td>${b.content}</td>
    </tr>
    </c:forEach>
</table>
</body>
</html>
```

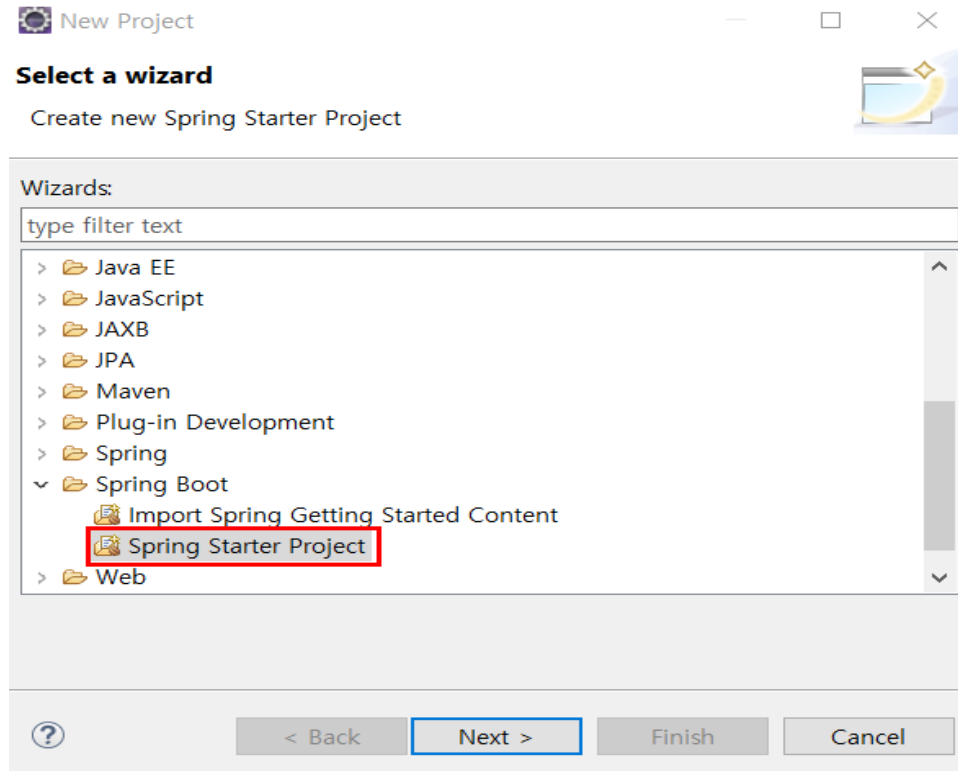


myBatis02 project를  
gradle로 만들어 보자?

# myBatis02 project 생성

## ❖ myBatis02 project 생성

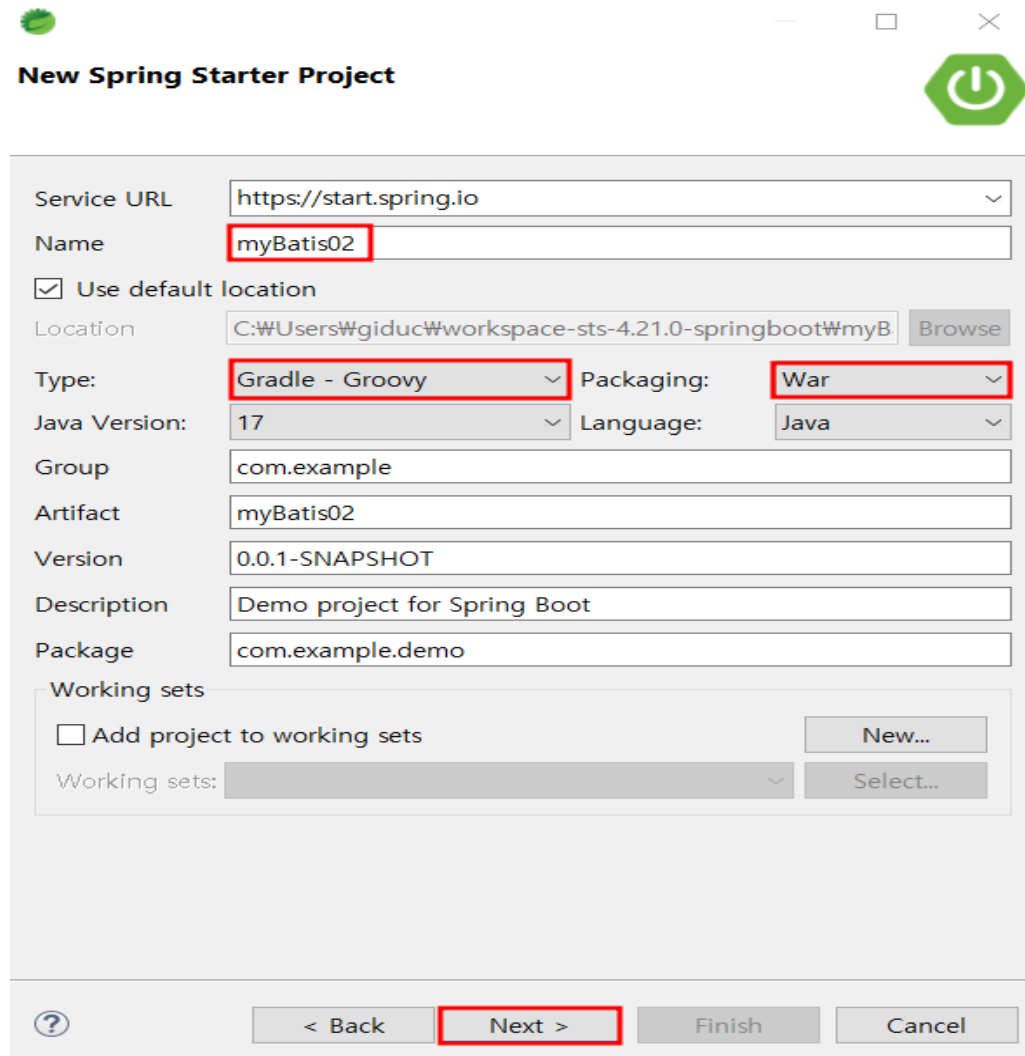
[File] – New - Project



# myBatis02 project 생성

## ❖ myBatis02 project 생성

- Name : **myBatis02**
- Type : Maven, **Gradle**
- Packaging : **War**, Jar



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:

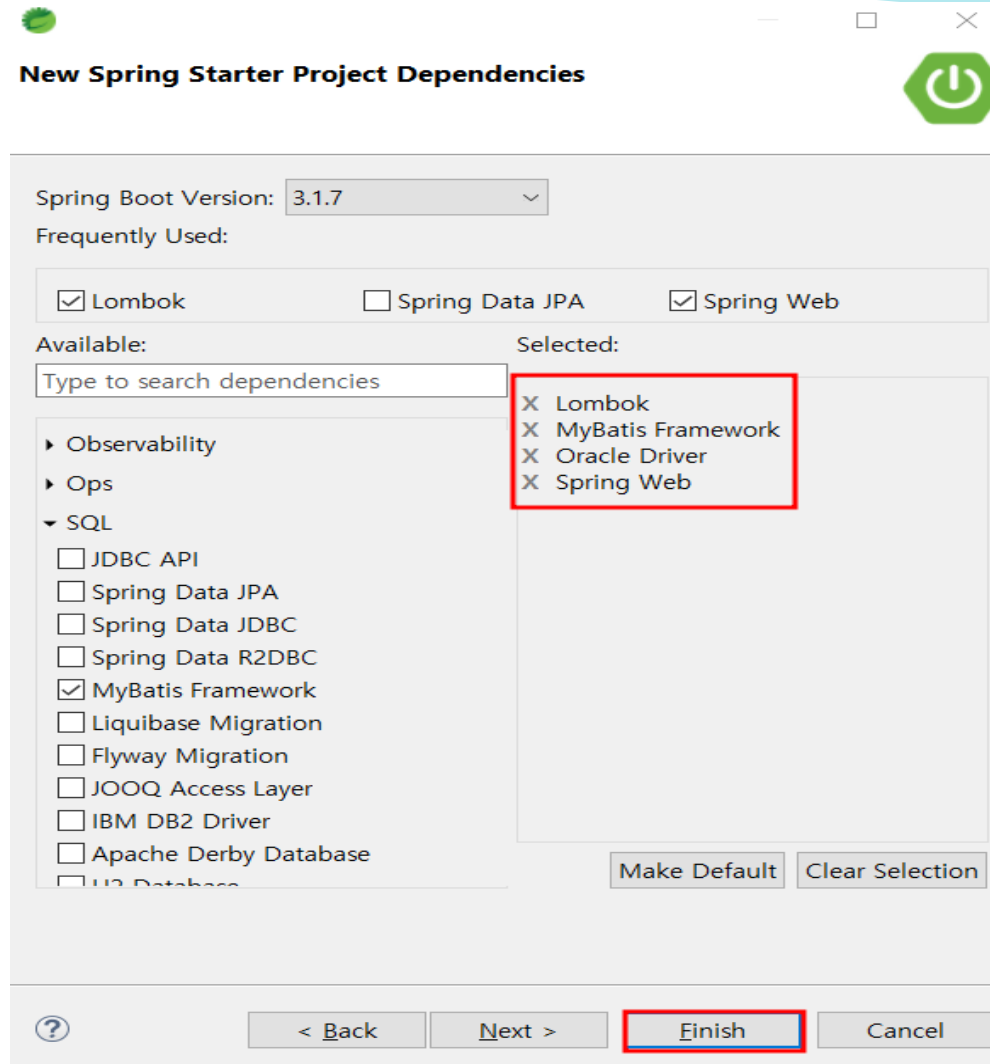
# myBatis02 project 생성

## ❖ myBatis02 project 생성

➤ Web  
Spring Web 체크

➤ SQL  
MyBatis Framework 체크  
Oracle Driver 체크

➤ Developer Tools  
Lombok 체크



# Gradle 환경설정

- ❖ Gradle 의 환경설정 파일에 의존 라이브러리 추가  
myBatis02 – build.gradle

```
dependencies {  
    // jsp  
    implementation 'org.apache.tomcat.embed:tomcat-embed-jasper'  
  
    // jakarta jstl  
    implementation 'org.glassfish.web:jakarta.servlet.jsp.jstl:3.0.0'  
  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    implementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter:3.0.3'  
    compileOnly 'org.projectlombok:lombok'  
    runtimeOnly 'com.oracle.database.jdbc:ojdbc11'  
    annotationProcessor 'org.projectlombok:lombok'  
    providedRuntime 'org.springframework.boot:spring-boot-starter-tomcat'  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
    testImplementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter-test:3.0.3'  
}
```

# myBatis02 실행

## ❖ myBatis02 실행

1. myBatis01 project 내용을 모두 myBatis02로 복사한다.
2. myBatis02 project를 내부 Tomcat으로 실행한다.
3. 웹브라우저로 접속한다.

<http://localhost/boardform>

<http://localhost/boardlist>