|  |
| --- |
|  |
| **Java + Spring Boot 로 게시판 만들기** |
| Spring Boot로 구현한 웹의 동작 원리 및 구조 파악하기 |
|  |
| 인턴 이 다 솔 |
|  |
| 2018-10-15 |

|  |
| --- |
|  |

목차

1**. 개요1**

**2. Spring Boot로 구현한 웹의 동작 원리1**

2.1. Spring framework로 구현한 웹의 동작 원리2

2.2. Java + Spring Boot 게시판 전체 코드 구조3

**3. src/main/java4**

3.1. com.board4

3.2. com.board.controller4

3.3. com.board.domain5

3.4. com.board.mapper5

**4. src/main/resources5**

4.1. mappers5

4.2. static/css6

4.3. application.properties6

**5. src/main/webapp/WEB-INF/views6**

5.1. main.jsp6

5.2. boardList.jsp7

5.3. boardpost.jsp7

5.4. boardView.jsp8

5.5. boardUpdate.jsp8

**6. build.gradle9**

**7. 출처9**

# 1. 개요

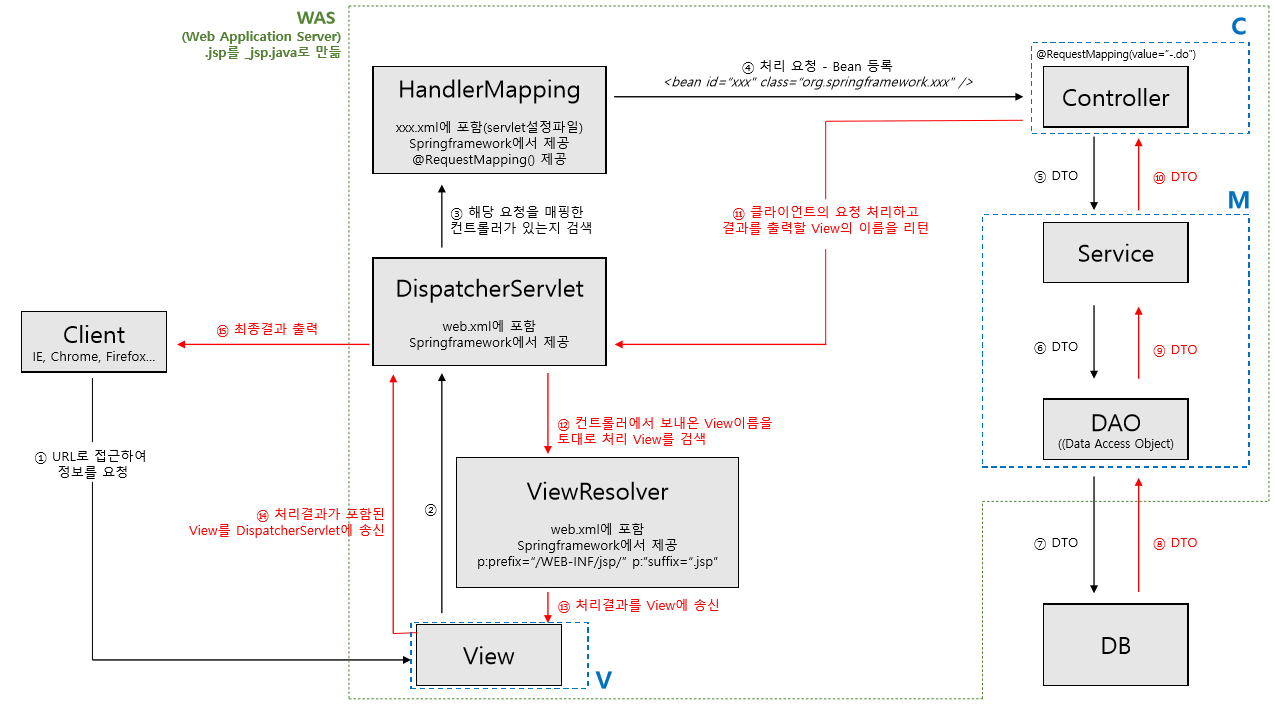
본 문서에서는 Java, Spring Boot, 그리고 빌드 툴인 gradle을 이용하여 게시판 웹 서비스를 구현한 코드에 대해 분석한다. Spring Boot의 동작 원리 및 전체 코드의 구조, 각 코드의 역할, 코드 내에서 사용된 메소드나 어노테이션에 대한 분석을 주로 다룬다.

# 2. Spring Boot 동작 원리 및 전체 코드 구조

Spring Boot는 Spring framework를 사용하는 프로젝트를 간단하게 세팅할 수 있는 서브 프로젝트이다. 따라서 Spring Boot의 동작 원리를 보기에 앞서 Spring framework의 동작 원리를 분석한다.

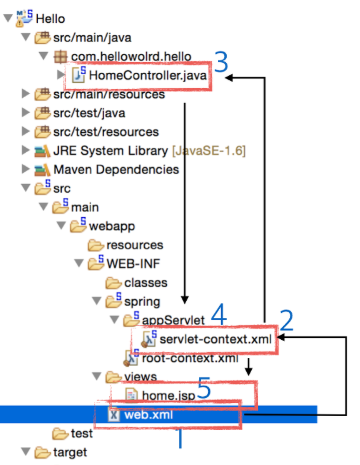
## 2.1. Spring framework로 구현한 웹의 동작원리

Spring framework로 구현한 웹의 동작형태는 간략히 아래의 사진과 같다.



출처) http://server-engineer.tistory.com/253

아래는 간단한 웹 서비스 예제 코드 구조이며, 동작 순서는 아래와 같다.

사용자가 웹페이지를 통해 요청을 보내면,

web.xml 의 Dispatcher Servlet 이 요청을 핸들링한다.

→ HandlerMapping으로 Controller를 검색하여, bean 설정이 있는 Servlet-context.xml내의 해당 요청을 처리하는 Controller에 요청을 객체로 넘겨준다.

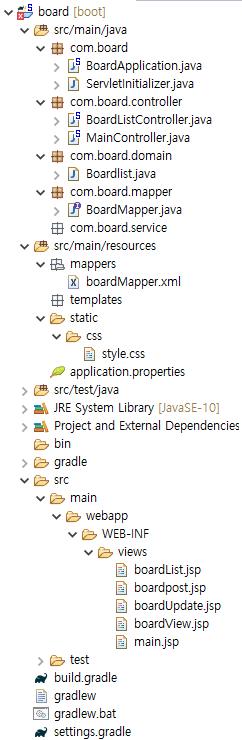
\* bean : 반복적으로 코드를 따로 작성하여 재사용하기 위해 만들어진 클래스

Controller는 요청 처리 후, 결과를 출력할 View의 이름을 리턴한다.

Dispatcher Servlet은 전달받은 View의 이름을 토대로 처리 View를 ViewResolver를 통해 검색하여 처리 결과를 사용자에게 리턴한다.2.2. Java + Spring Boot로 구현한 게시판 전체 코드 구조

아래는 Spring Boot를 사용하여 구현한 게시판 서비스의 전체 코드 구조와 동작 원리이다.

많은 설정을 자동으로 세팅하여 Spring framework와 달리 web.xml, Servlet-context.xml 파일이 필요없다.

메인 클래스인 BoardApplication.java가 실행되면 내장 tomcat이 실행되어 서버가 구동된다.

사용자가 View를 통해 원하는 화면 간에 이동을 하거나, 작업을 수행할 때 Controller로 요청이 넘겨진다.

이 때 Controller에서 사용하는 메소드들이 Mapper 파일에 선언되어 있으며, 해당 메소드가 실행될 때 실행되는 sql 쿼리가 resource의 xml 파일에 나타나있다. 계층간 데이터 교환을 위한 객체인 DTO는 domain 아래에 java 파일로 존재한다.

\*DTO : 계층간 데이터 교환을 위한 객체로, 속성과 그 속성에 접근하기 위한 getter, setter 메소드만 가진 클래스

# 3. src/main/java

## 3.1. com.board

BoardApplication.java : Spring Boot의 메인 클래스이다. 이 파일을 실행하면 내장 Tomcat이 자동으로 실행된다. 환경 설정, 자동 설정 기능, 컴포넌트 스캔에 관한 어노테이션이 @SpringBootApplication 하나로 선언되어 있다.

MySQL과의 연동을 위해 MapperScan 어노테이션을 이용하여 탐색할 Mapper 인터페이스의 경로를 설정한다. 그리고 SqlSessionFactory을 이용하여 Mapper.xml 파일과 매핑해준다.

- SqlSessionFactory : Datasource 객체를 입력받아 MyBatis의 SqlSessionFactory를 반환하여 스프링 내에서 빈으로 사용가능하게 하는 메소드이다.

- HiddenHttpMethodFilter : <form> 태그에서 ‘\_method’ 의 추가정보를 사용하여 브라우저에서 REST 방식을 사용할 수 있도록 지원해주는 메소드이다.

Servletinitializer.java : .war 파일로 생성했을 때 생성되는 파일이다. Tomcat 같은 Servlet Container 에서 Spring Boot 애플리케이션이 동작 가능하도록 웹 애플리케이션 컨텍스트를 구성한다.

\* Servlet Container : 웹서비스를 위한 인터페이스인 Servlet을 관리하며 네트워크 통신, 생명주기 관리, 스레드 기반의 병렬처리를 대행하는 것

## 3.2. com.board.controller

이 곳의 코드들은 특정 메소드로 특정 페이지를 읽었을 때 표시되는 view를 매칭시킨다.

BoardListController.java : localhost:4309/board로 접속했을 때의 게시판 화면을 관리한다.

- Autowired : Spring이 빈의 요구사항과 매칭되는 애플리케이션 컨텍스트상에서 다른 빈을 찾아 빈 간의 의존성을 자동으로 만족시키도록 하는 수단이다.

- RequestMapping : 어떤 Controller, 어떤 메소드가 요청을 처리할지를 매핑하기 위한 어노테이션이다.

- ResponseEntity : 응답에 대한 데이터(http 응답 헤더와 상태코드 등)를 가지는 객체이다.

MainController.java : localhost:4309로 접속했을 때의 메인 화면을 관리한다.

## 3.3. com.board.domain

Boardlist.java : 게시판 구성에 사용되는 항목을 담은 객체가 선언된 파일이다.

- getter, setter : 클래스의 멤버 필드를 private으로 선언하면 객체에서는 그 값을 변경할 수 없는데, 이를 가능하게 해주는 메소드이다. getter는 get키(), setter는 set키()의 형태를 가진다.

## 3.4. com.board.mapper

BoardMapper.java : mapper 인터페이스 파일이다. boardMapper.xml 의 sql 쿼리와 해당 쿼리에 접근할 수 있는 객체의 매핑을 도와준다.

# 4. src/main/resources

## 4.1. mappers

boardMapper.xml : Controller를 통해 매칭되어 실행되는 sql 쿼리문이 작성된 파일이다.

- 작성 형태 : <매핑된 구문과 인자들> 실행될 sql 쿼리문 </매핑된 구문>

- 예시) <select id=”boardList” parameterType=”int” resultType=”com.board.domain.Boardlist”>

select \* from testboard where board\_id = #{board\_id}

</select>

- 인자 종류

Id : 구문을 찾기 위해 사용될 수 있는 네임스페이스 내 유일한 구분자

parameterType : 구문에 전달될 파라미터의 패키지 경로를 포함한 전체 클래스명 / 별칭

resultType : 구문에 의해 리턴되는 타입의 패키지 경로를 포함한 전체 클래스명 / 별칭

## 4.2. static/css

style.css : 웹의 디자인적 요소가 담긴 파일이다.

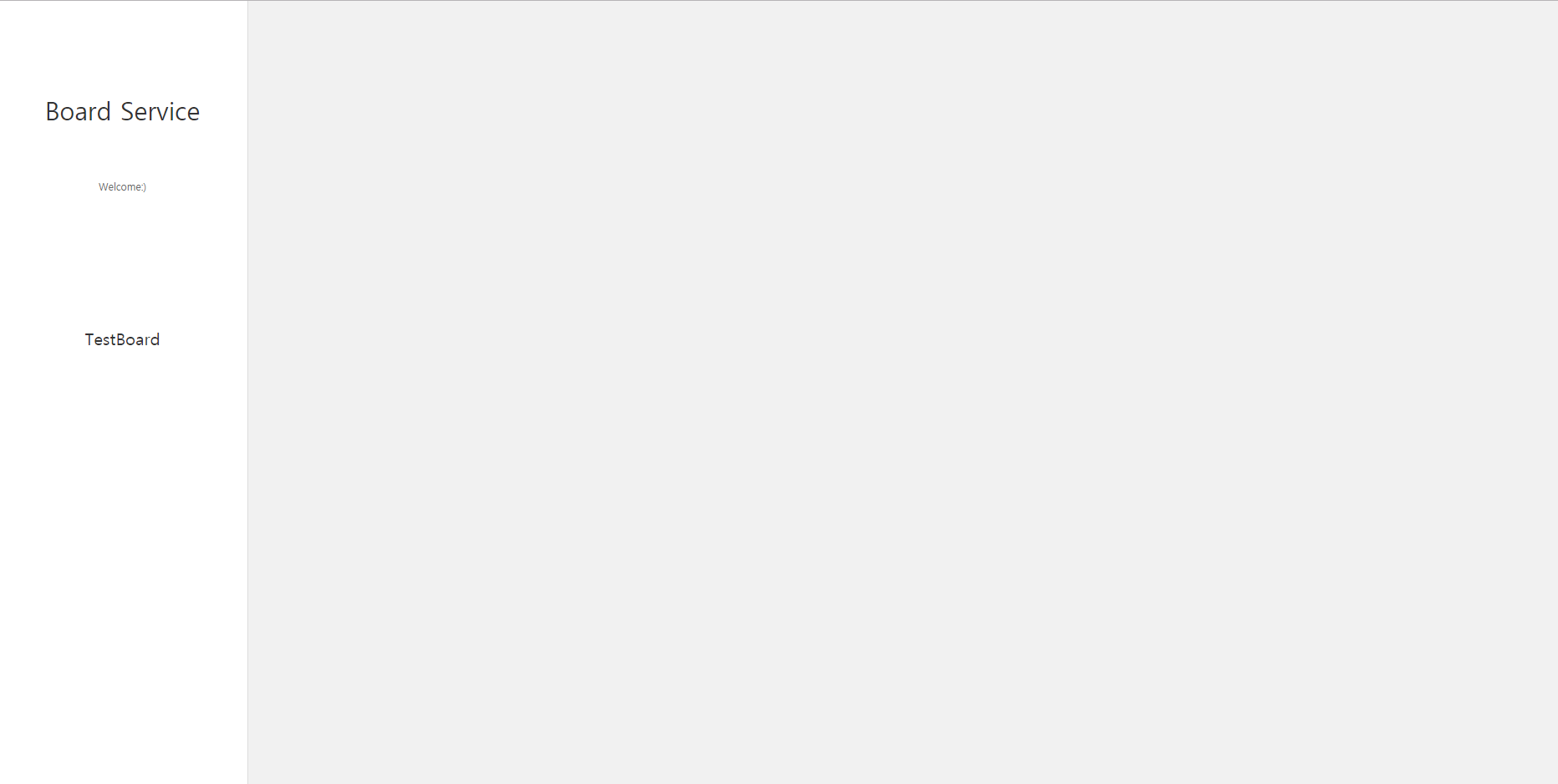
## 4.3. application.properties

Spring Boot에서 환경설정을 하기 위한 파일이다. DB 경로와 계정, static 파일 경로, jsp 페이지 설정(server.servlet.jsp.init-parameters.development)이 담겨있다.

# 5. src/main/webapp/WEB-INF/views

## 5.1. main.jsp

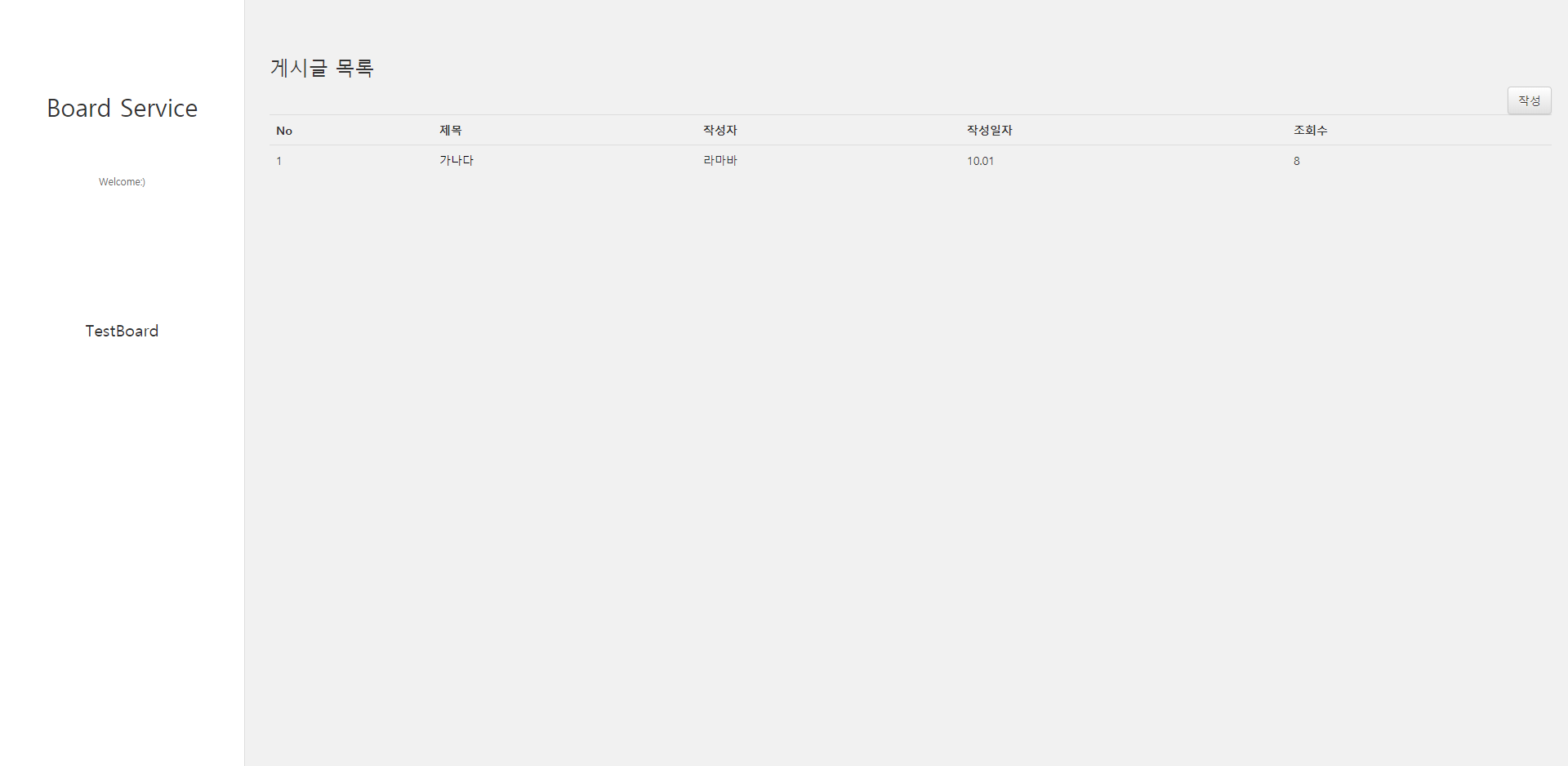
localhost:4309로 접속했을 때의 메인 화면 View이다.



왼쪽의 사이드바는 다른 페이지로 이동해도 유지된다. Board Service를 선택하면 localhost:4309로 연결되며, TestBoard를 선택하면 localhost:4309/board로 연결된다.

## 5.2. boardList.jsp

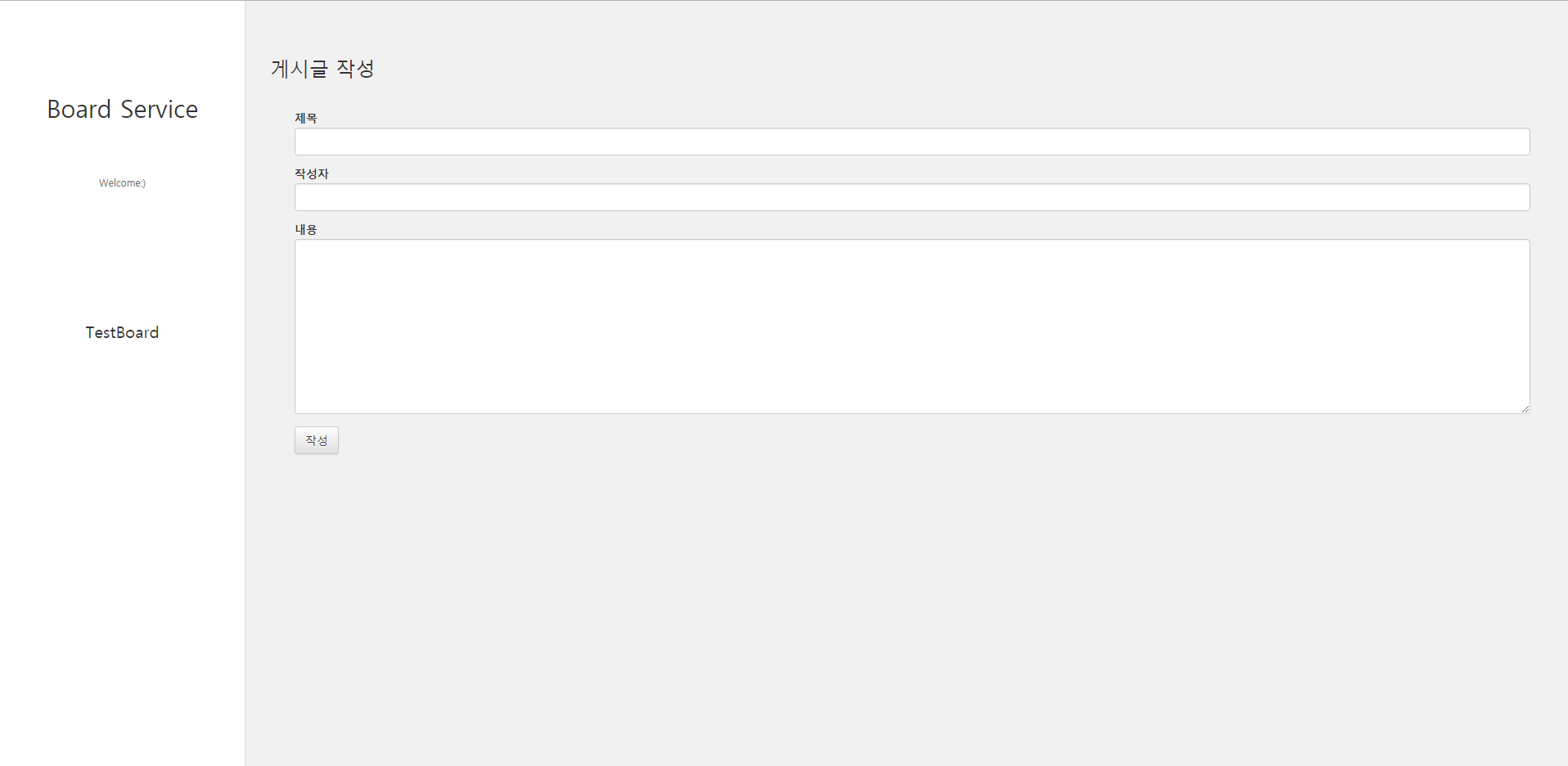
게시글 목록을 나타내는 View이다.



상단의 ‘작성’ 버튼을 선택하면 boardpost.jsp 로 이동한다. 게시글의 목록에서 자세히 보고자 하는 글의 ‘제목’ 을 선택하면, 해당 글의 상세보기 화면인 boardView.jsp 로 이동한다.

## 5.3. boardpost.jsp

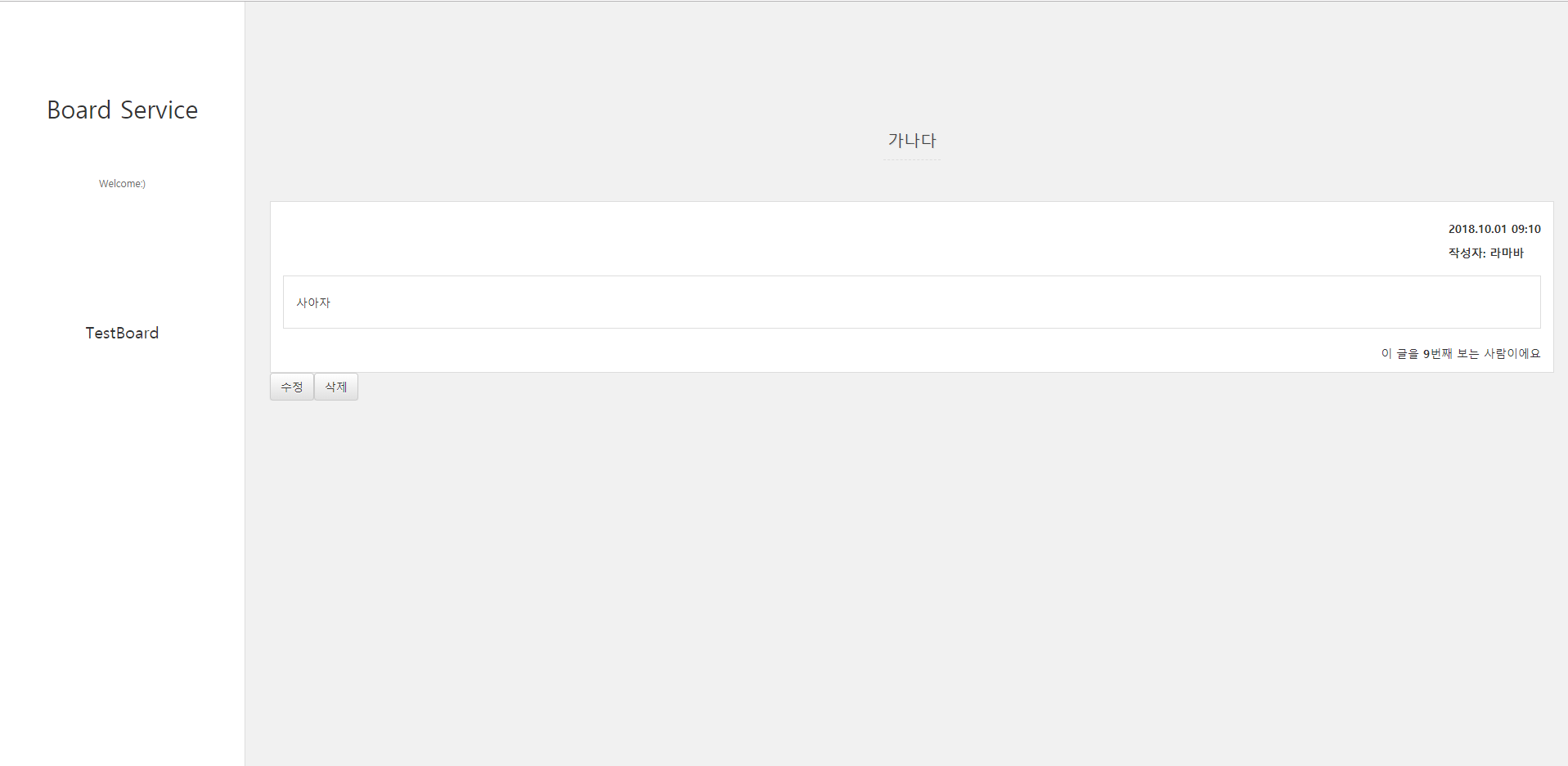
게시글 작성 화면을 나타내는 View이다.



내용 작성 후 ‘작성’ 버튼을 선택하면 boardList.jsp 로 이동한다.

## 5.4. boardView.jsp

게시글을 선택했을 때 상세화면을 나타내는 View이다.



하단의 ‘수정’ 버튼을 선택하면 boardUpdate.jsp 로 이동하며, ‘삭제’ 버튼을 선택하면 해당 게시글이 삭제된다.

## 5.5. boardUpdate.jsp

게시글을 수정하는 화면을 나타나는 View이다.

# 

하단의 ‘수정’ 버튼을 선택하면 boardView.jsp 로 이동한다.

# 6. build.gradle

빌드 배포 도구인 gradle의 설정 파일이다. 구성은 아래와 같다.

- buildscript 메소드 : 빌드 스크립트를 빌드하는 과정에서 필요한 외부 라이브러리를 클래스패스에 추가하는 기능을 한다. Spring boot 플러그인이 클래스패스에 선언되어 있다.

- 플러그인 적용 : 다양한 플러그인을 ‘apply plugin : ‘ 의 형태로 선언하여 적용한다.

- dependencies 메소드 : 프로젝트의 의존성 설정을 담당한다.

org.apache.tomcat.embed:tomcat-embed-jasper, javax.servlet:jstl

- JSP 파일의 구동을 위해 선언한다.

org.mybatis.spring.boot:mybatis-spring-boot-starter:1.1.1

org.mybatis:mybatis-spring

org.mybatis:mybatis

- mybatis 연동을 위해 선언한다.

mysql:mysql-connector-java

- mysql DB 사용을 위해 선언한다.

org.springframework.boot:spring-boot-starter-jdbc

org.springframework.boot:spring-boot-starter-web

org.springframework.boot:spring-boot-starter-tomcat

- Spring Boot를 사용하기 위해 기본적으로 선언한다.

# 7. 출처

Tistory 블로그 Innovative Thinking IT, <http://jobc.tistory.com/61>, hiddenhttpmethodfilter

Tistory 블로그 공부해서 남 주자, <http://private.tistory.com/52>, sqlsessionfactory

Tistory 블로그 T없이맑은I, <http://tbang.tistory.com/87>, Autowired

블로그 삵(sarc.io), <https://sarc.io/index.php/development/1139-requestmapping>, RequestMapping

Tistory 블로그 HeeStory, <http://heeestorys.tistory.com/566>, ResponseEntity

Tistory 블로그 무른모 공방, <http://devist.tistory.com/128>, getter와 setter

Mybatis 페이지, <http://www.mybatis.org/mybatis-3/ko/sqlmap-xml.html>, Mapper XML 파일