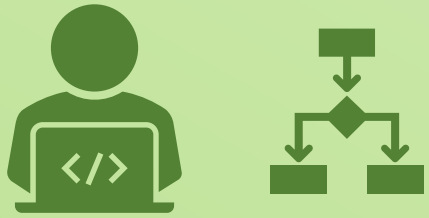
SPRING

MISSION



100

INDEX

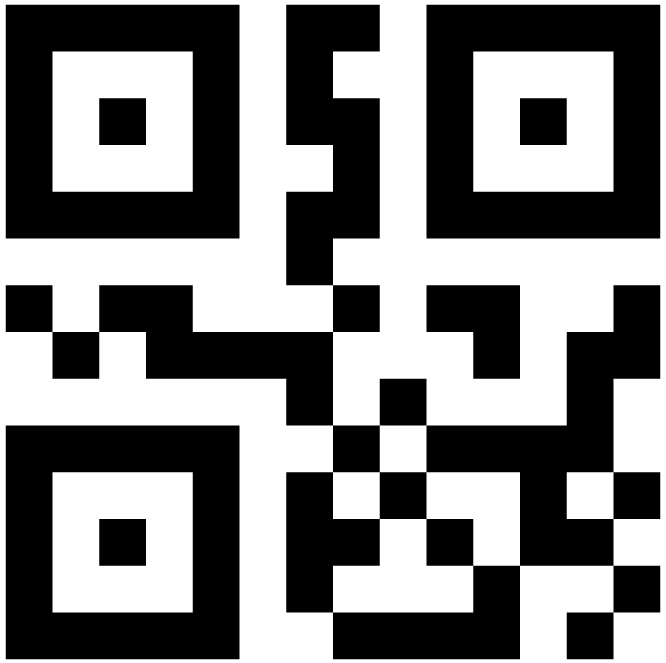
INDEX

INDEX

INDEX

INDEX

아래 코드/링크를 통해 제공소스를 다운로드 받을 수 있습니다.



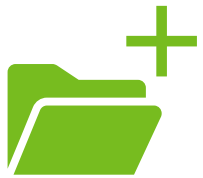
https://github.com/modoomsion/java

1

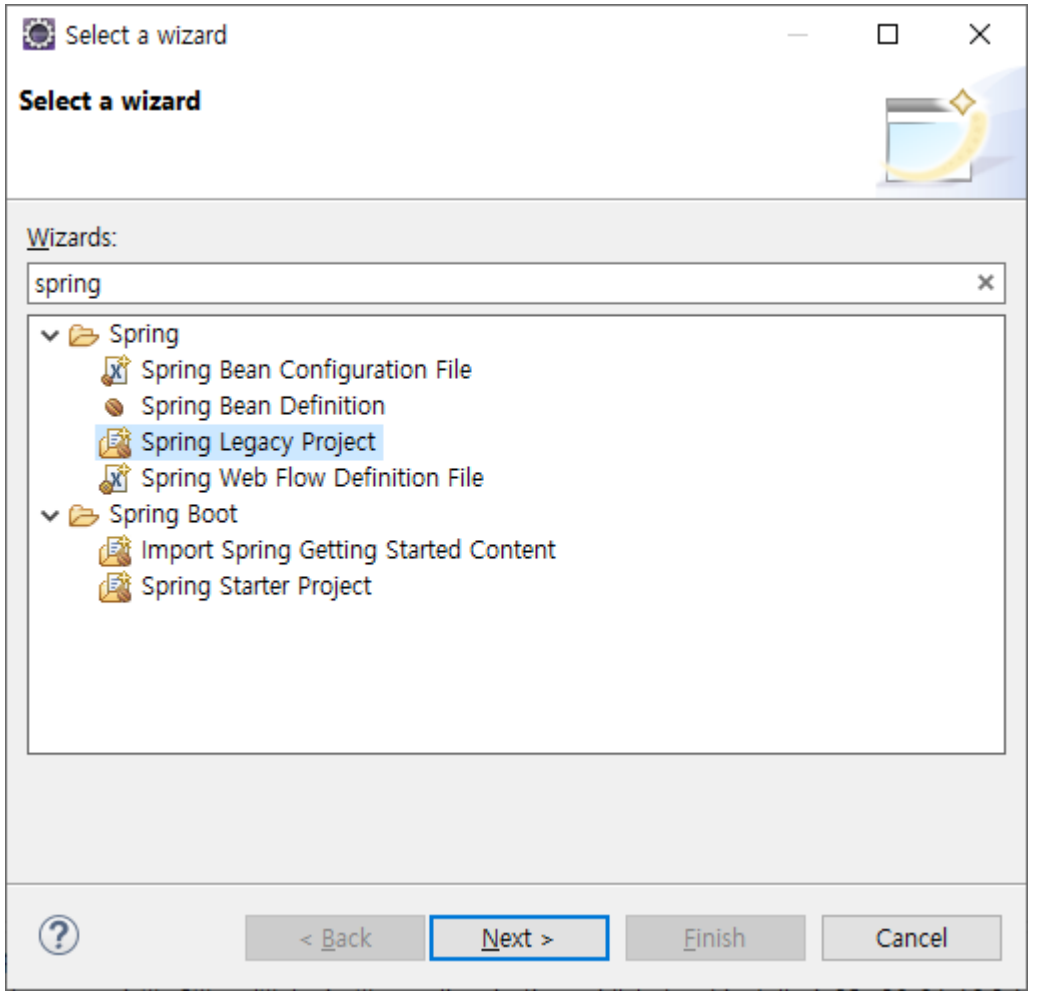
Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청 경로 매핑

1. 프로젝트를 생성하시오.



Spring Lagacy Project



1

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

2. Templates 를 선택하시오.

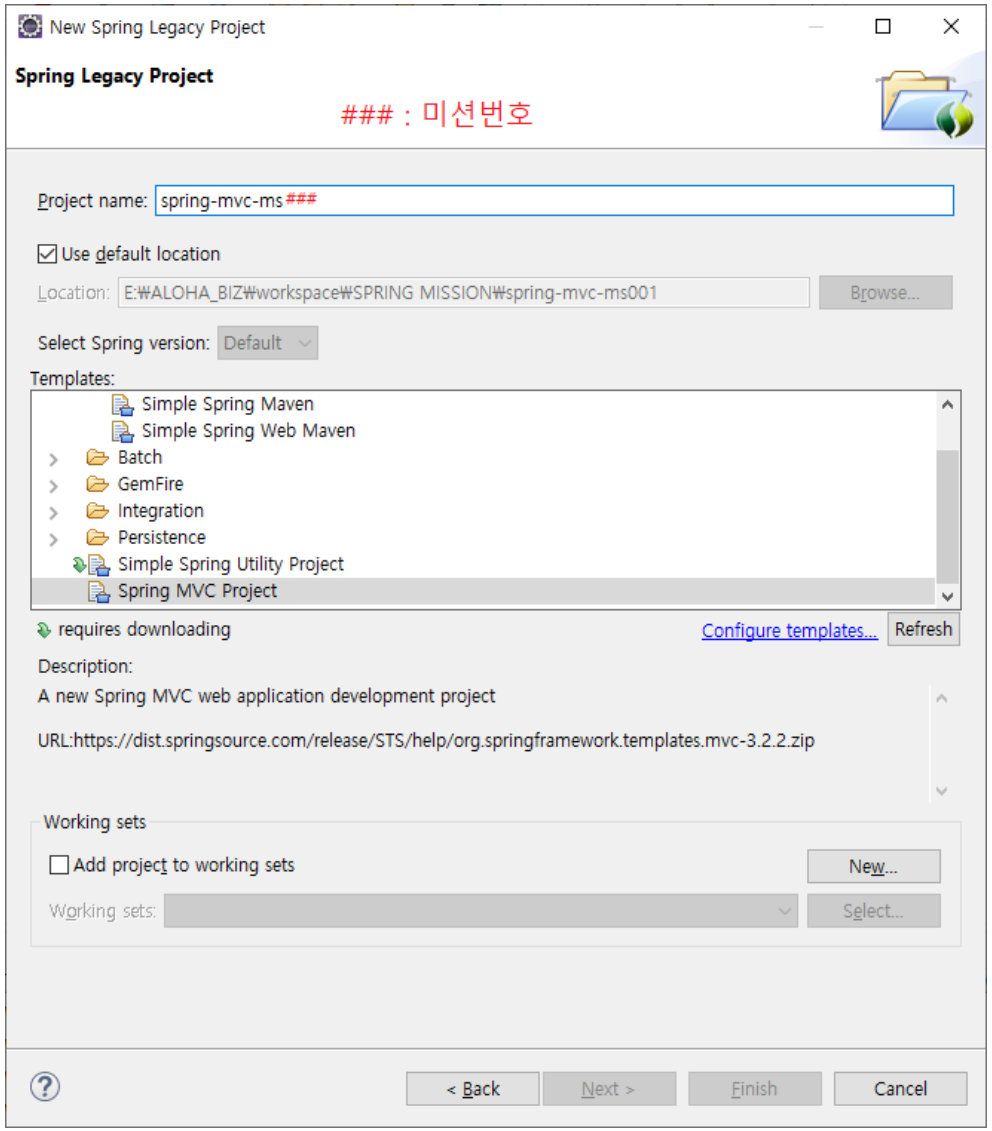
요청 경로 매핑

[Project name]

spring-mvc-ms001

[Templates]

Spring MVC Project



1

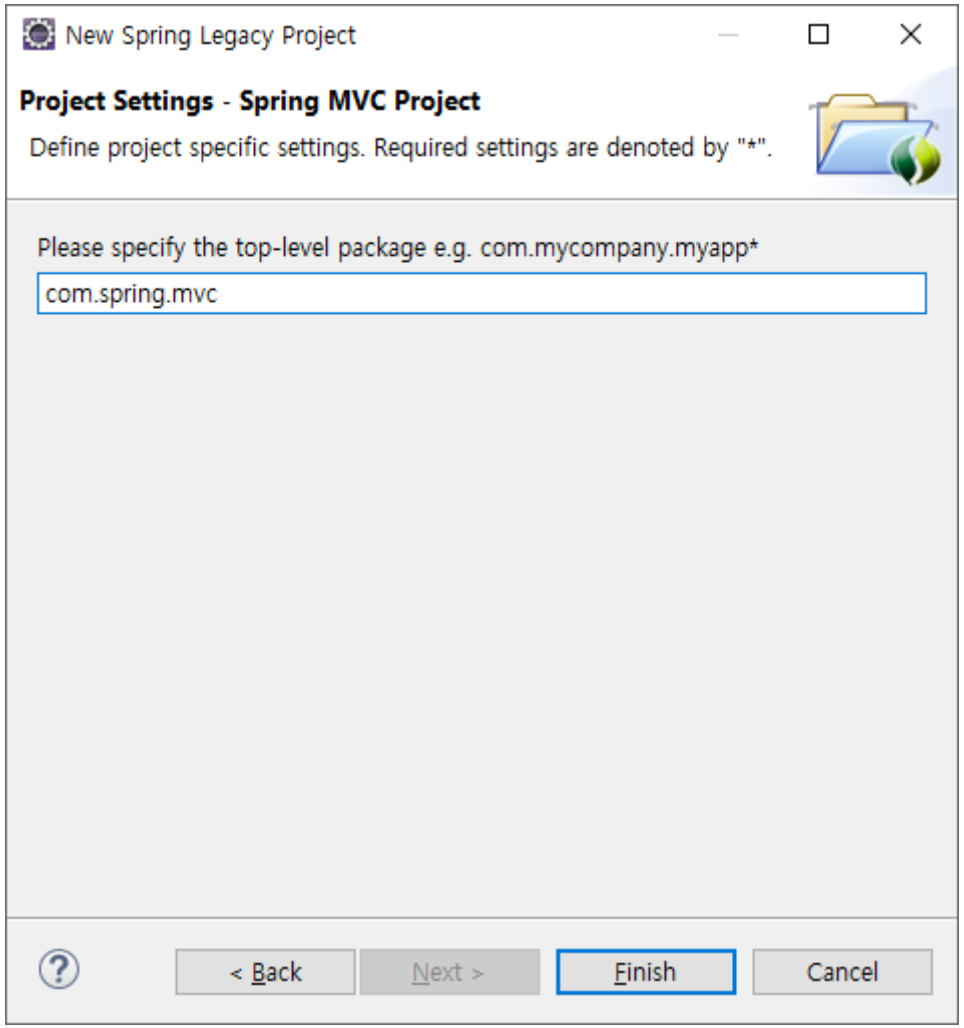
Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

3. Package 경로를 입력하시오.

요청 경로 매핑

[Package]

com.spring.mvc



1

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청 경로 매핑

프로젝트 구조

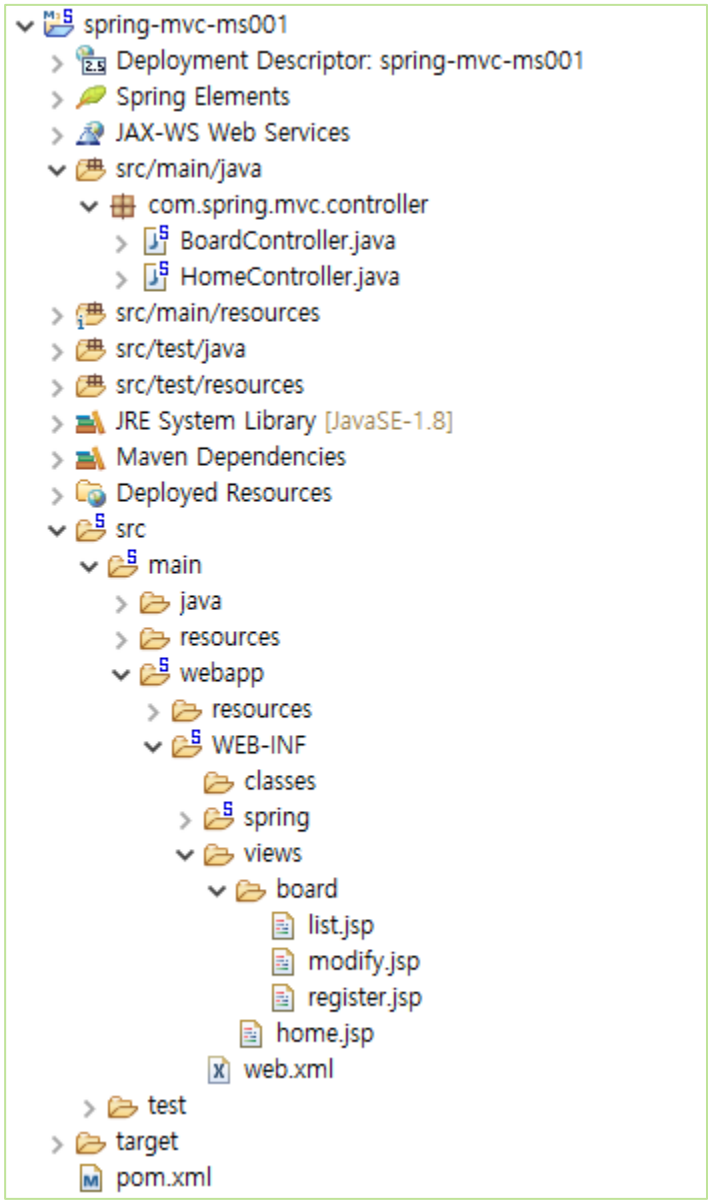
1. com.spring.mvc 패키지 하위 경로에 controller 패키지를 생성하시오.

2. 'Boardcontroller' 클래스를 생성하시오.

3. ~/views 하위 경로에 board 폴더를 생성하시오.

4. ~/views/board 하위 경로에 JSP 파일을 생성하시오.

   - list.jsp
   - register.jsp
   - modify.jsp



1

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청경로 매핑 테이블

Controller	Request Path/Method		Views	기능
BoardController	/board/register	GET	/views/board/register.jsp	등록화면
	registerForm()			
	/board/modify	GET	/views/board/modify.jsp	수정화면
	modifyForm()			
	/board/list	GET	/views/board/list.jsp	목록화면
	list()			

Controller	Request Path/Method		Views	기능
HomeController	/	GET	/views/home.jsp	메인화면
	home(Locale locale, Model model)			

1

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청경로 매핑

"요청경로 매핑 테이블 " 을 참조하여,

1) BoardController.java 파일에 <글 등록화면>을 요청하는 메서드를 작성하시오.

2) BoardController.java 파일에 <글 수정화면>을 요청하는 메서드를 작성하시오.

3) BoardController.java 파일에 <글 목록화면>을 요청하는 메서드를 작성하시오.

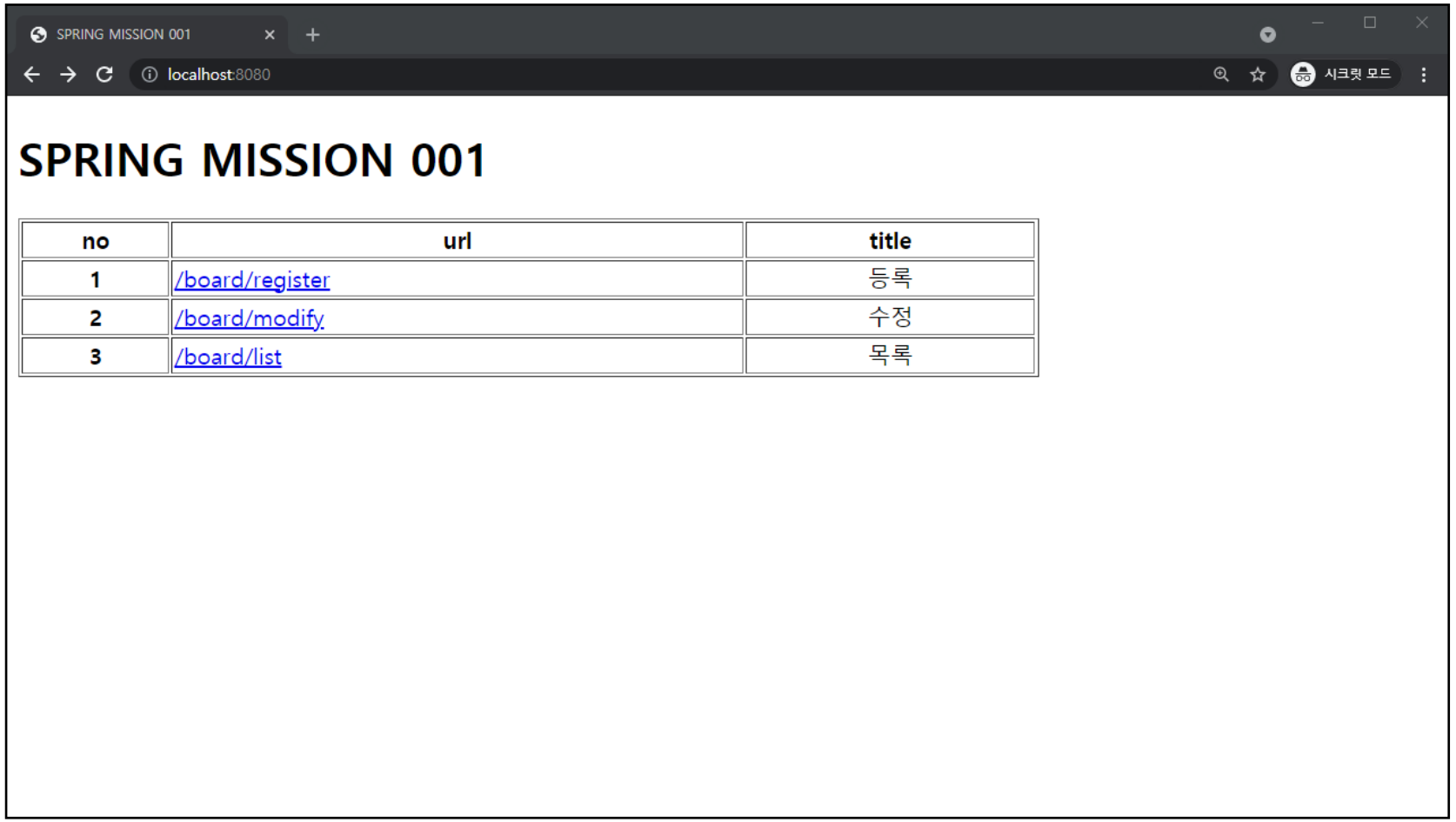
4) HomeController.java 파일에 <메인화면>을 요청하는 메서드를 작성하시오.

1

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청 경로 매핑

localhost:8080/



GUIDE

1. <h1> 태그를 이용하여 제목을 작성하시오.

2. <table> 태그를 이용하여 표를 작성하시오.

3. <a> 태그를 이용하여, 링크를 작성하시오.

- /board/register
- /board/modify
- /board/list

file

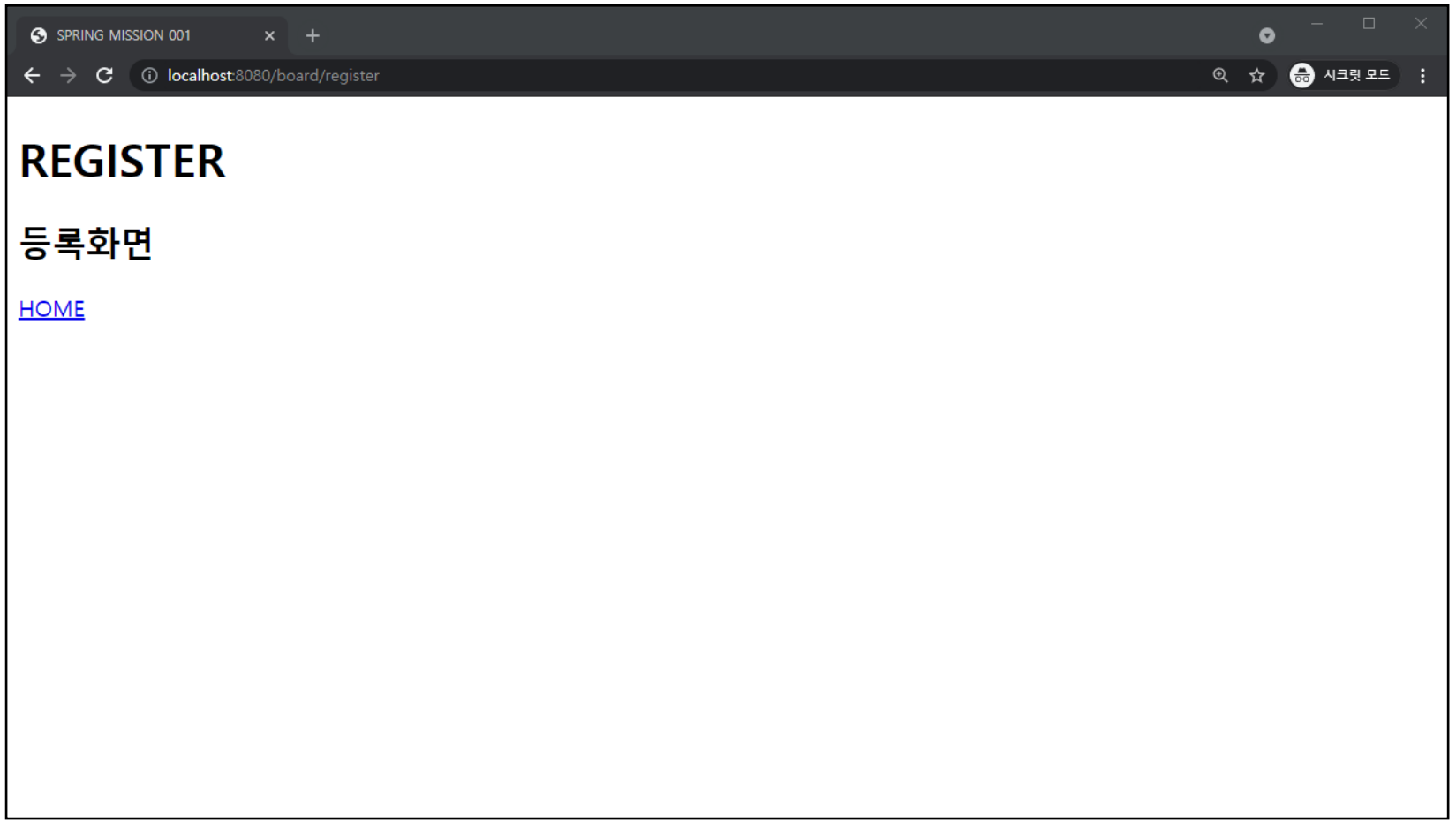
~/views/home.jsp

1

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청 경로 매핑

localhost:8080/board/register



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

file

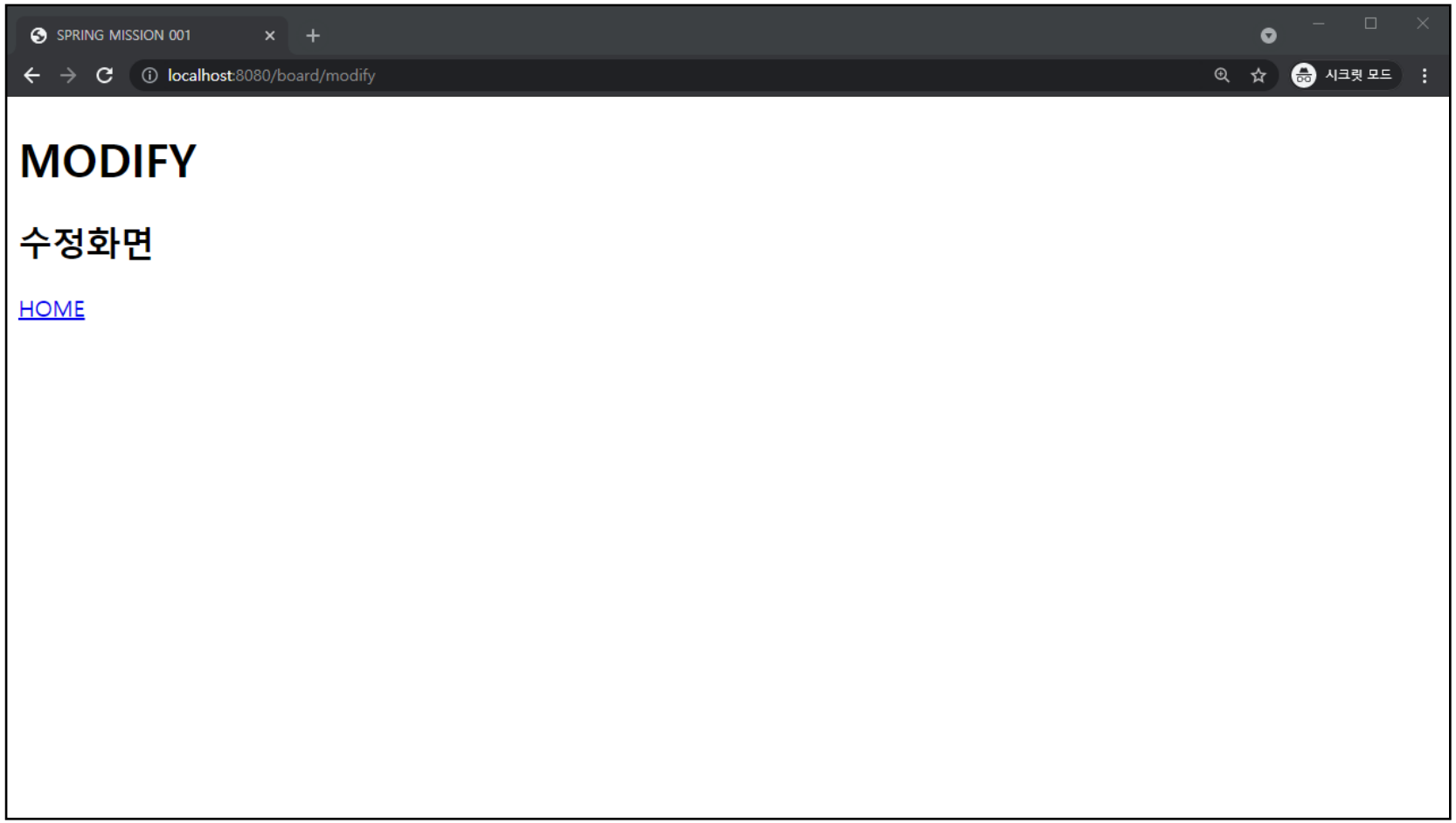
~/views/board/register.jsp

1

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청 경로 매핑

localhost:8080/board/modify



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

file

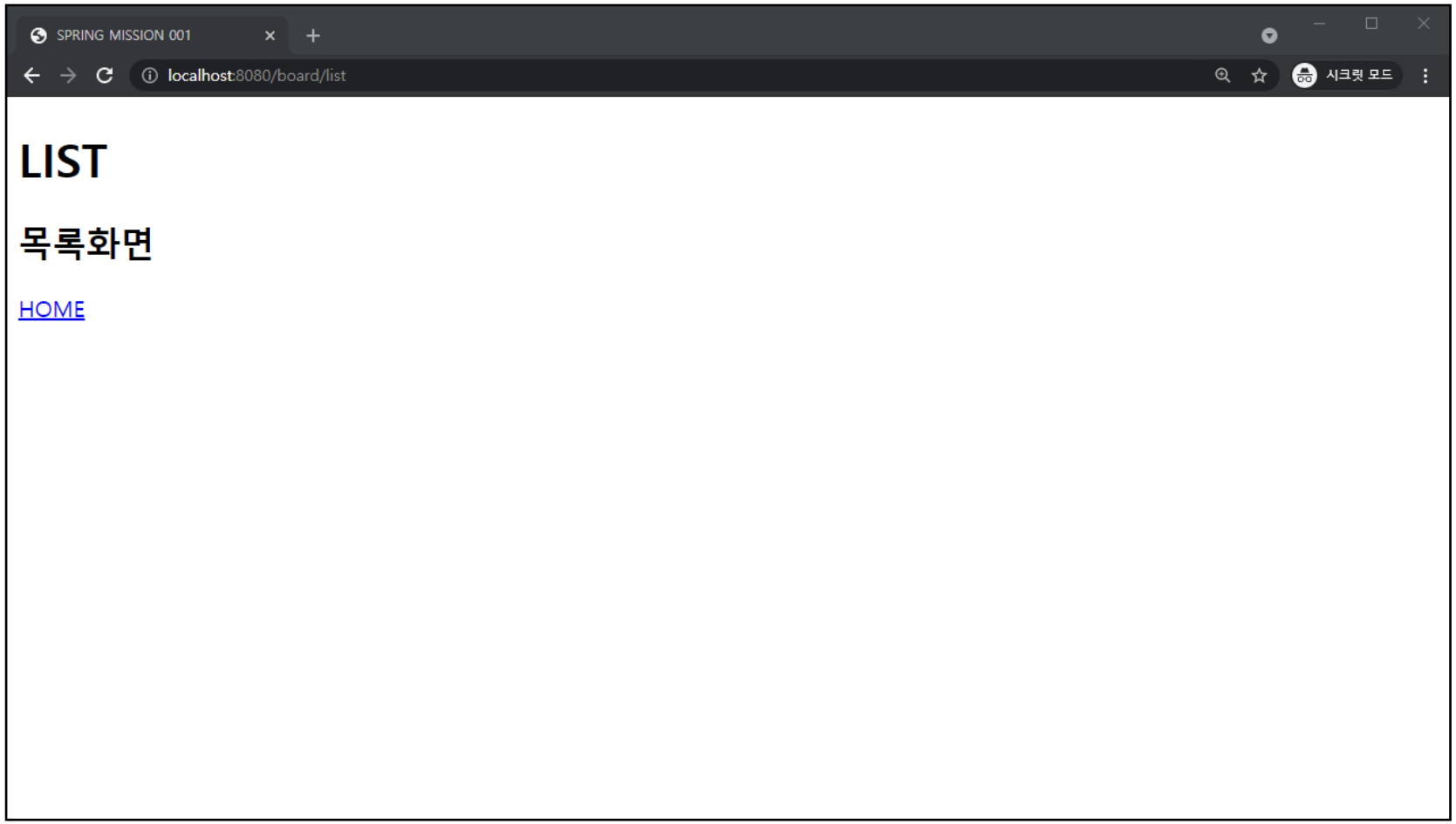
~/views/board/modify.jsp

1

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청 경로 매핑

localhost:8080/board/list



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

file

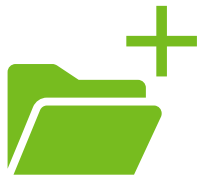
~/views/board/list.jsp

2

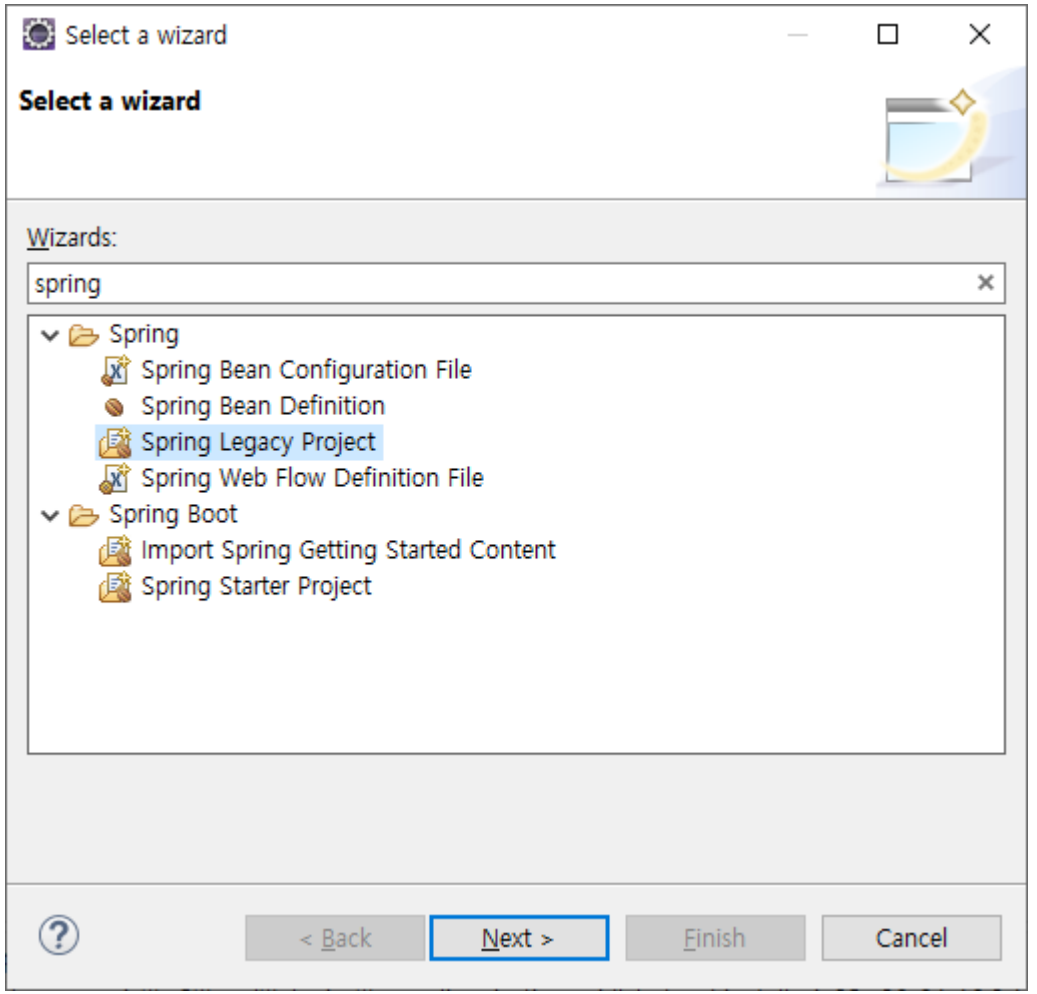
Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

클래스 레벨
요청 경로 매핑

1. 프로젝트를 생성하시오.



Spring Lagacy Project



2

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

2. Templates 를 선택하시오.

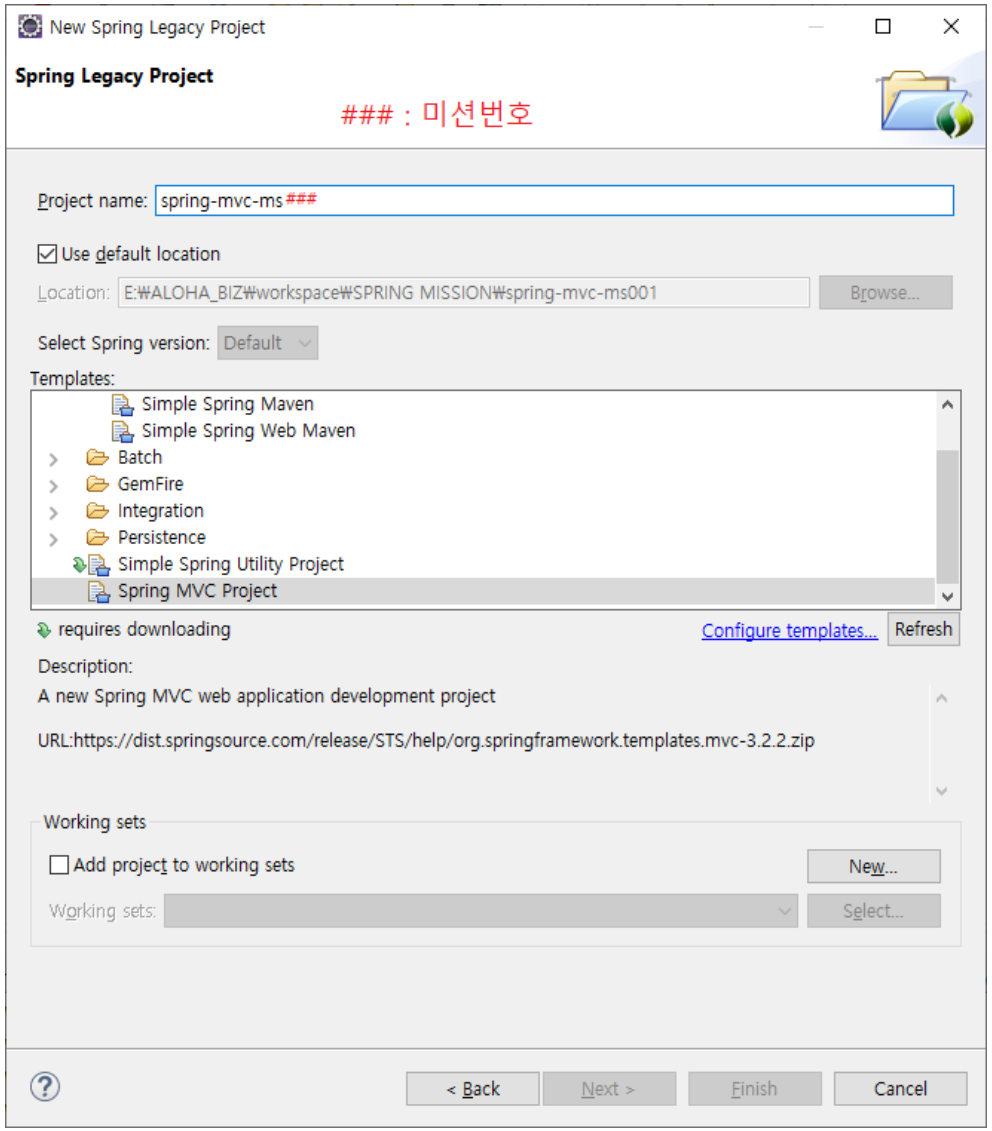
클래스 레벨
요청 경로 매핑

[Project name]

spring-mvc-ms002

[Templates]

Spring MVC Project



2

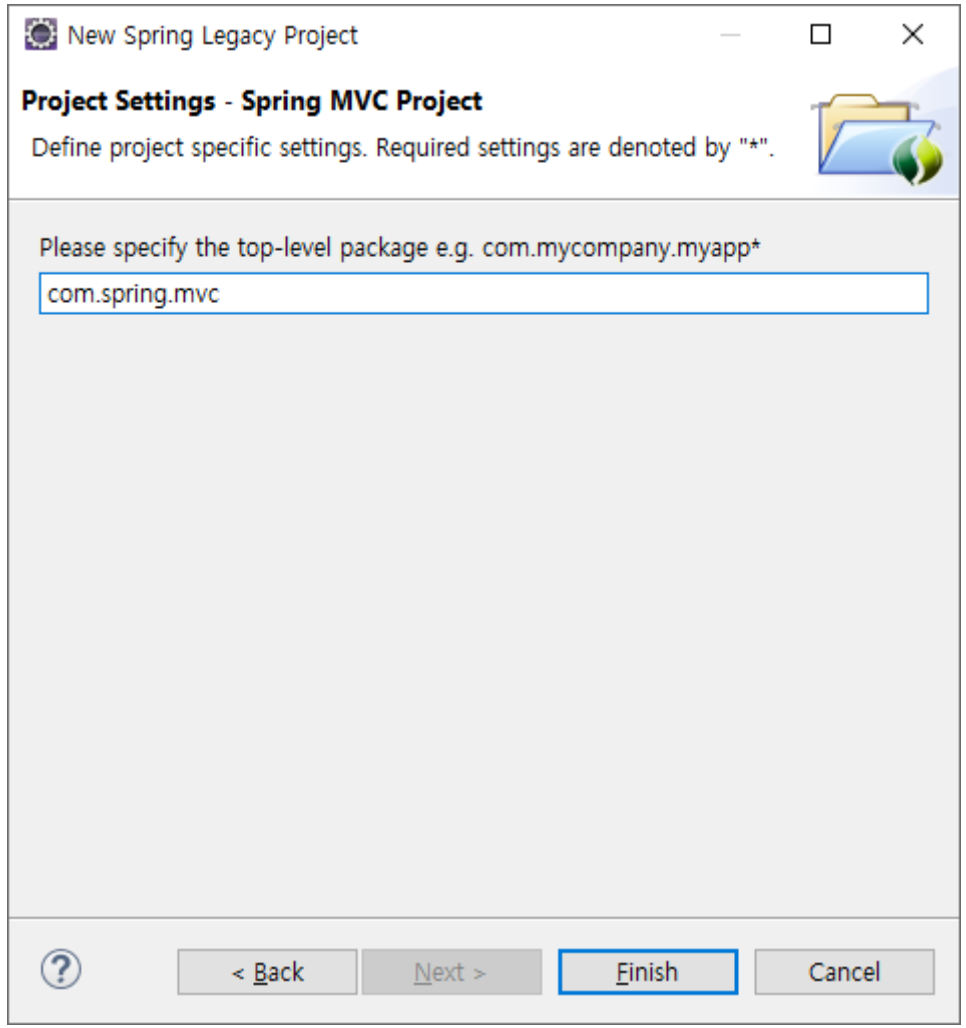
Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

3. Package 경로를 입력하시오.

클래스 레벨
요청 경로 매핑

[Package]

com.spring.mvc



2

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

프로젝트 구조

클래스 레벨
요청 경로 매핑

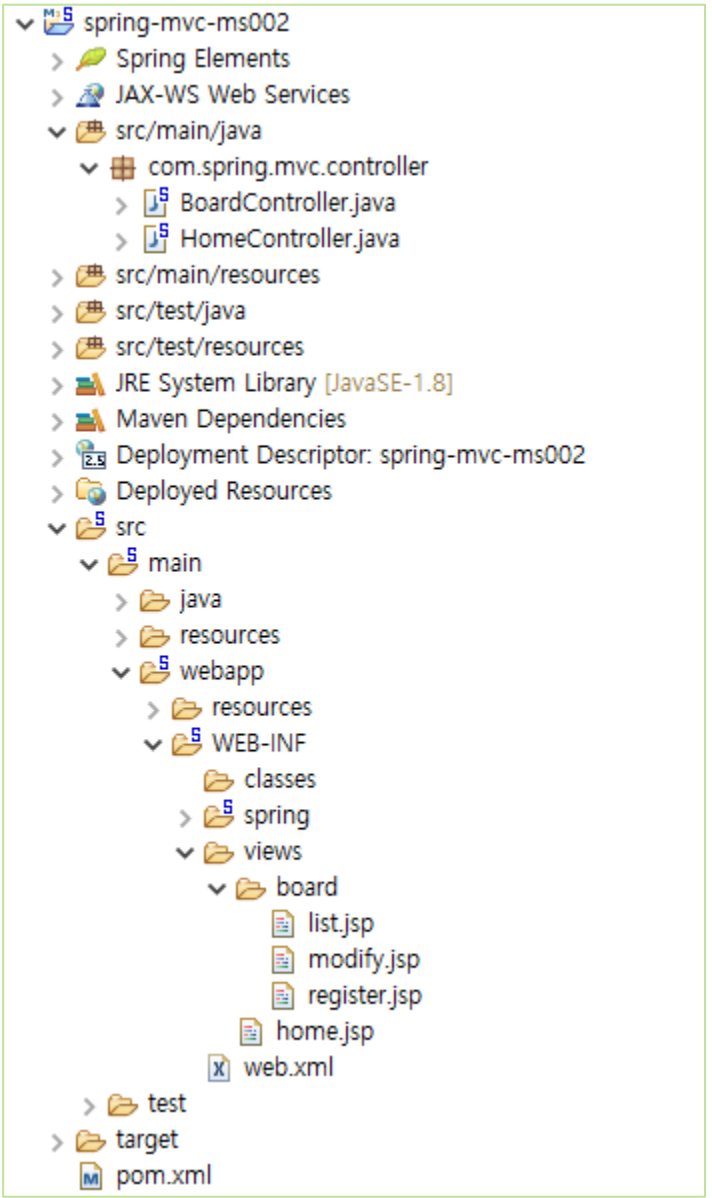
1. com.spring.mvc 패키지 하위 경로에 controller 패키지를 생성하시오.

2. 'Boardcontroller' 클래스를 생성하시오.

3. ~/views 하위 경로에 board 폴더를 생성하시오.

4. ~/views/board 하위 경로에 JSP 파일을 생성하시오.

   - list.jsp
   - register.jsp
   - modify.jsp



2

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

클래스 레벨
요청 경로 매핑

요청경로 매핑 테이블

Controller	Request Path/Method		Views	기능
BoardController	/register	GET	/views/board/register.jsp	등록화면
	registerForm()			
	/modify	GET	/views/board/modify.jsp	수정화면
	modifyForm()			
/board	/list	GET	/views/board/list.jsp	목록화면
	list()			

* 클래스 레벨에 “/board” 경로를 매핑한다.

Controller	Request Path/Method		Views	기능
HomeController	/	GET	/views/home.jsp	메인화면
	home(Locale locale, Model model)			

2

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청경로 매핑

"요청경로 매핑 테이블 " 을 참조하여,

1) BoardController.java 파일에 <글 등록화면>을 요청하는 메서드를 작성하시오.

2) BoardController.java 파일에 <글 수정화면>을 요청하는 메서드를 작성하시오.

3) BoardController.java 파일에 <글 목록화면>을 요청하는 메서드를 작성하시오.

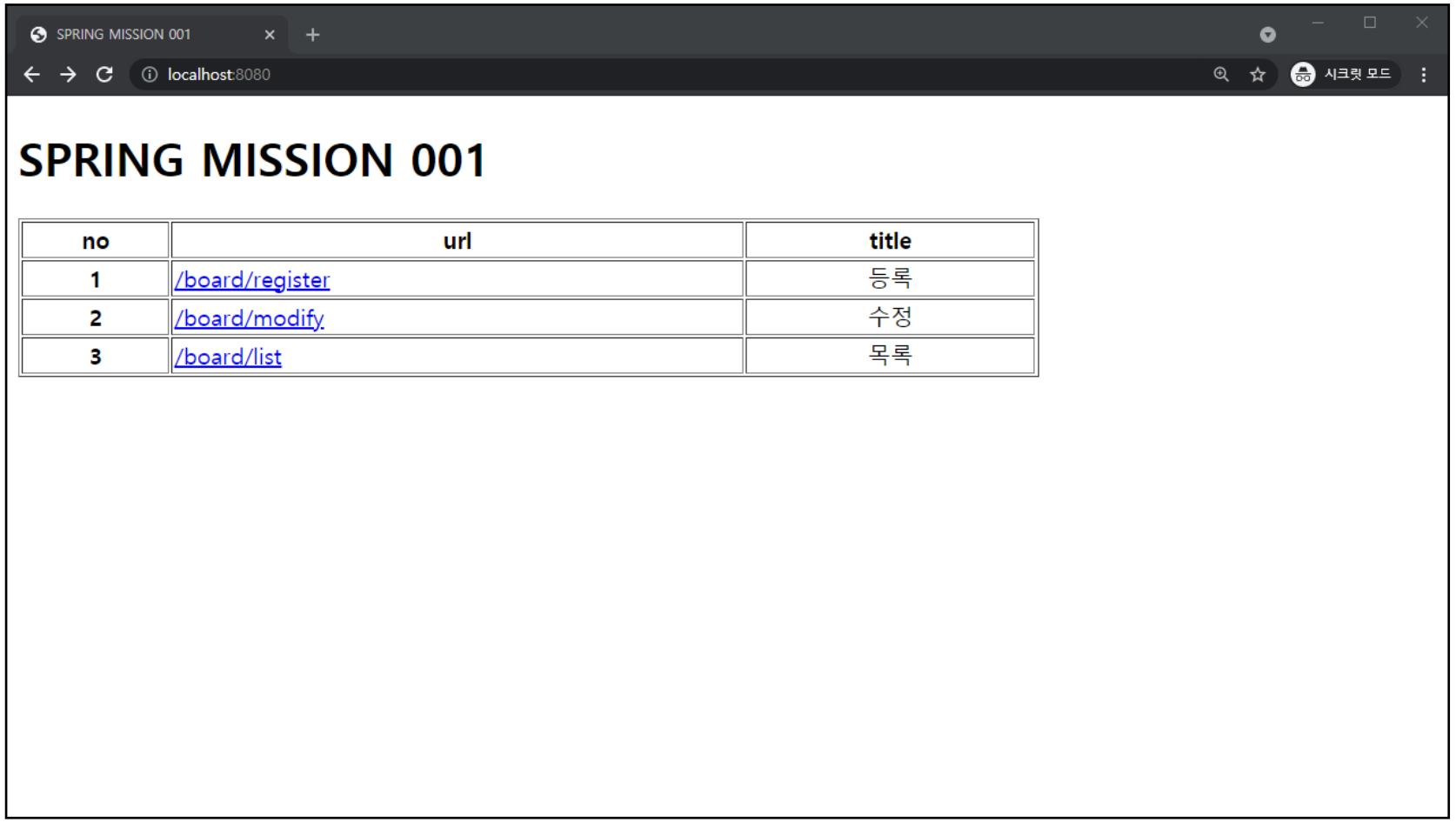
4) HomeController.java 파일에 <메인화면>을 요청하는 메서드를 작성하시오.

2

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

클래스 레벨
요청 경로 매핑

localhost:8080/



GUIDE

1. <h1> 태그를 이용하여 제목을 작성하시오.

2. <table> 태그를 이용하여 표를 작성하시오.

3. <a> 태그를 이용하여, 링크를 작성하시오.

- /board/register
- /board/modify
- /board/list

file

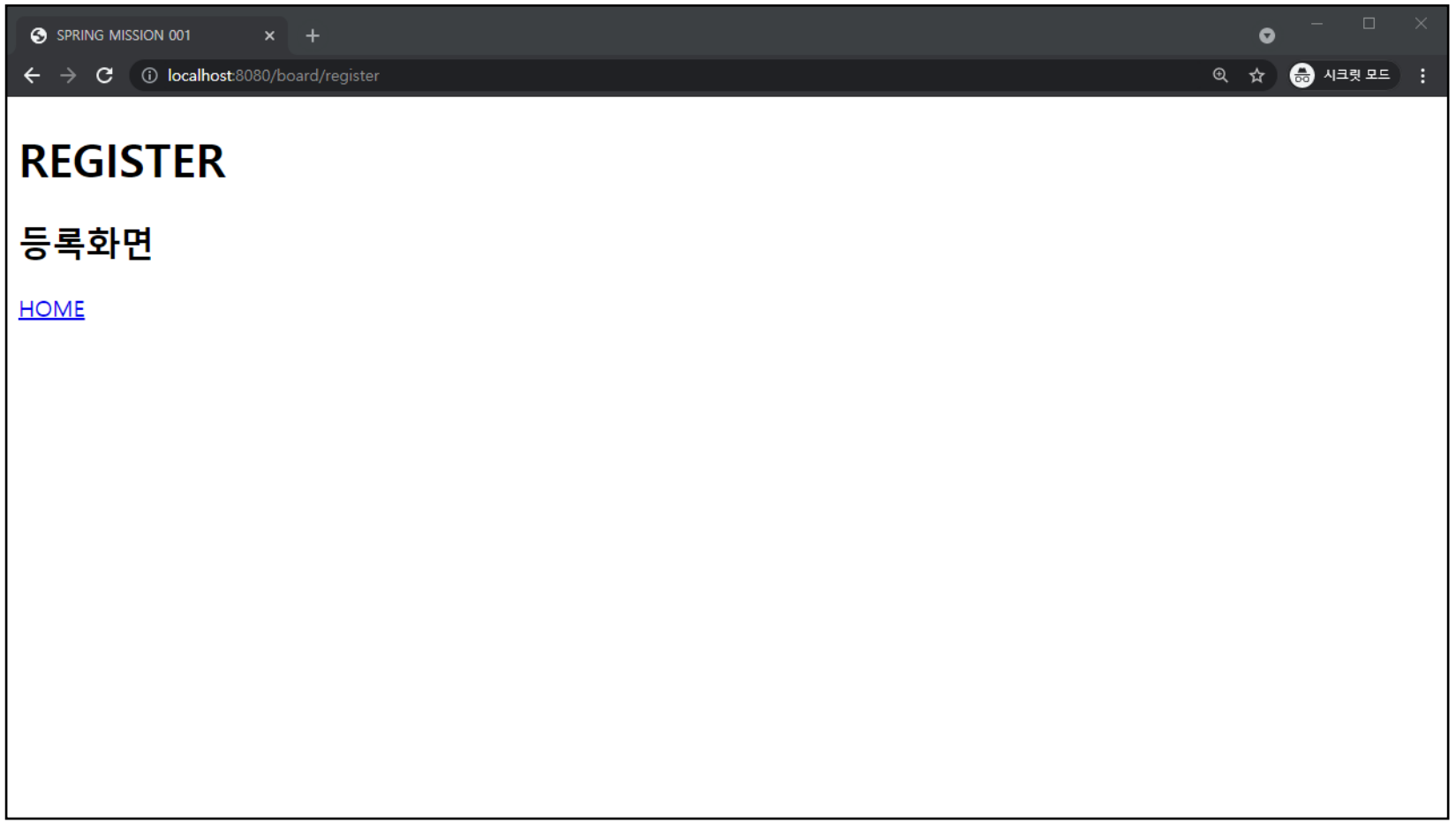
~/views/home.jsp

2

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

클래스 레벨
요청 경로 매핑

localhost:8080/board/register



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

file

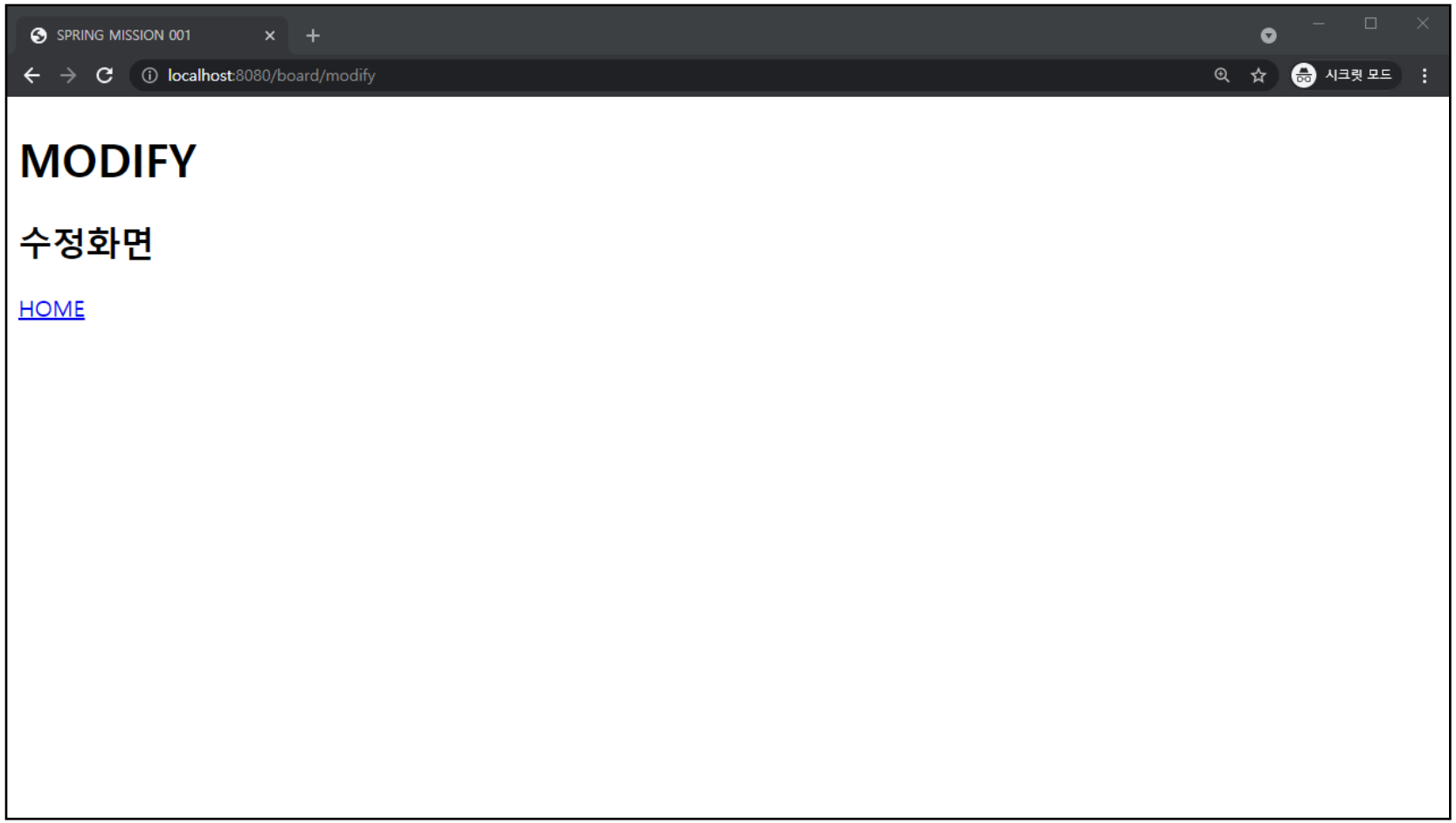
~/views/board/register.jsp

2

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

클래스 레벨
요청 경로 매핑

localhost:8080/board/modify



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

file

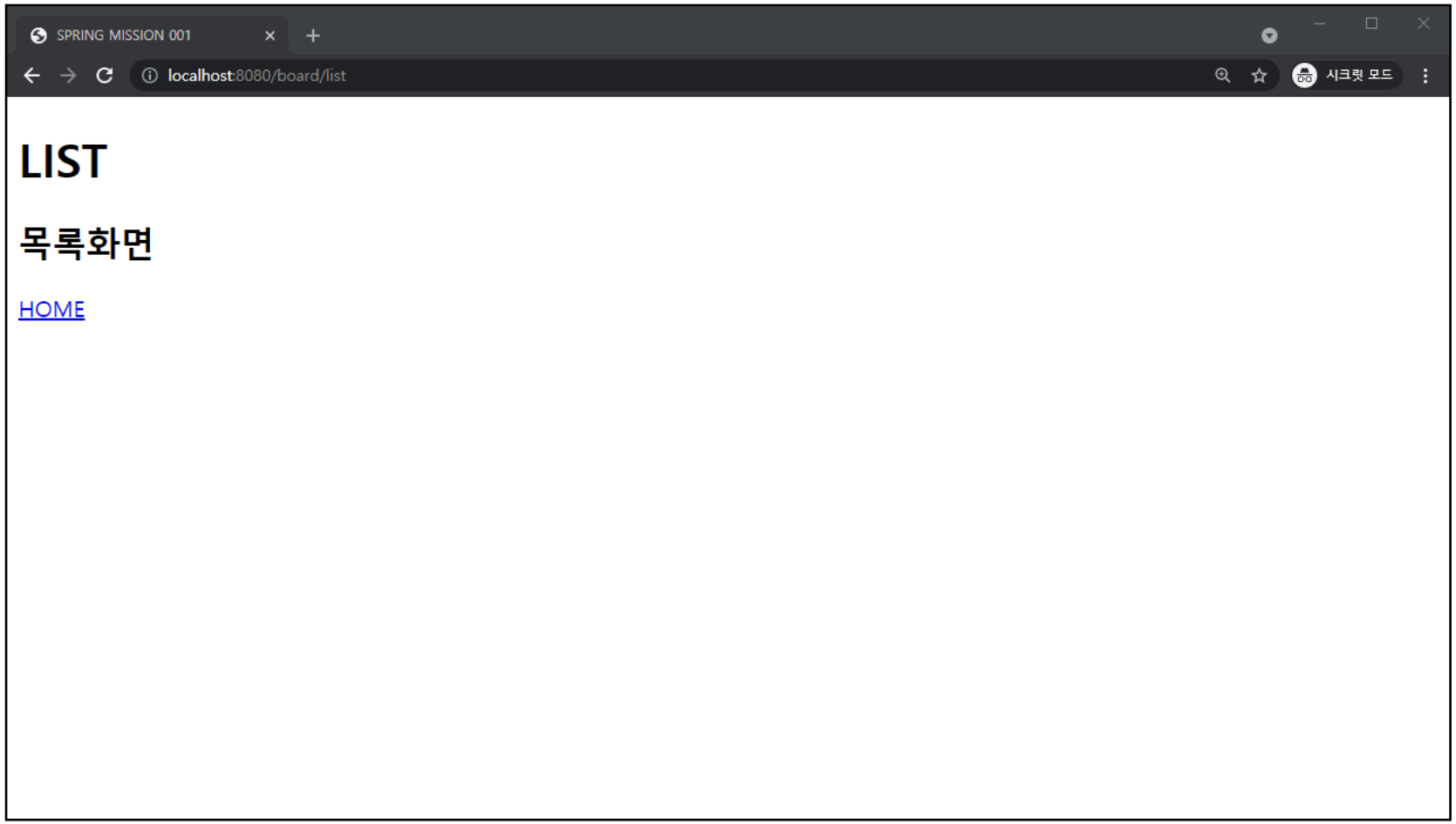
~/views/board/modify.jsp

2

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

클래스 레벨
요청 경로 매핑

localhost:8080/board/list



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

file

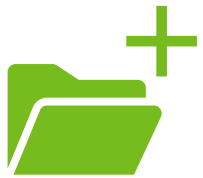
~/views/board/list.jsp

3

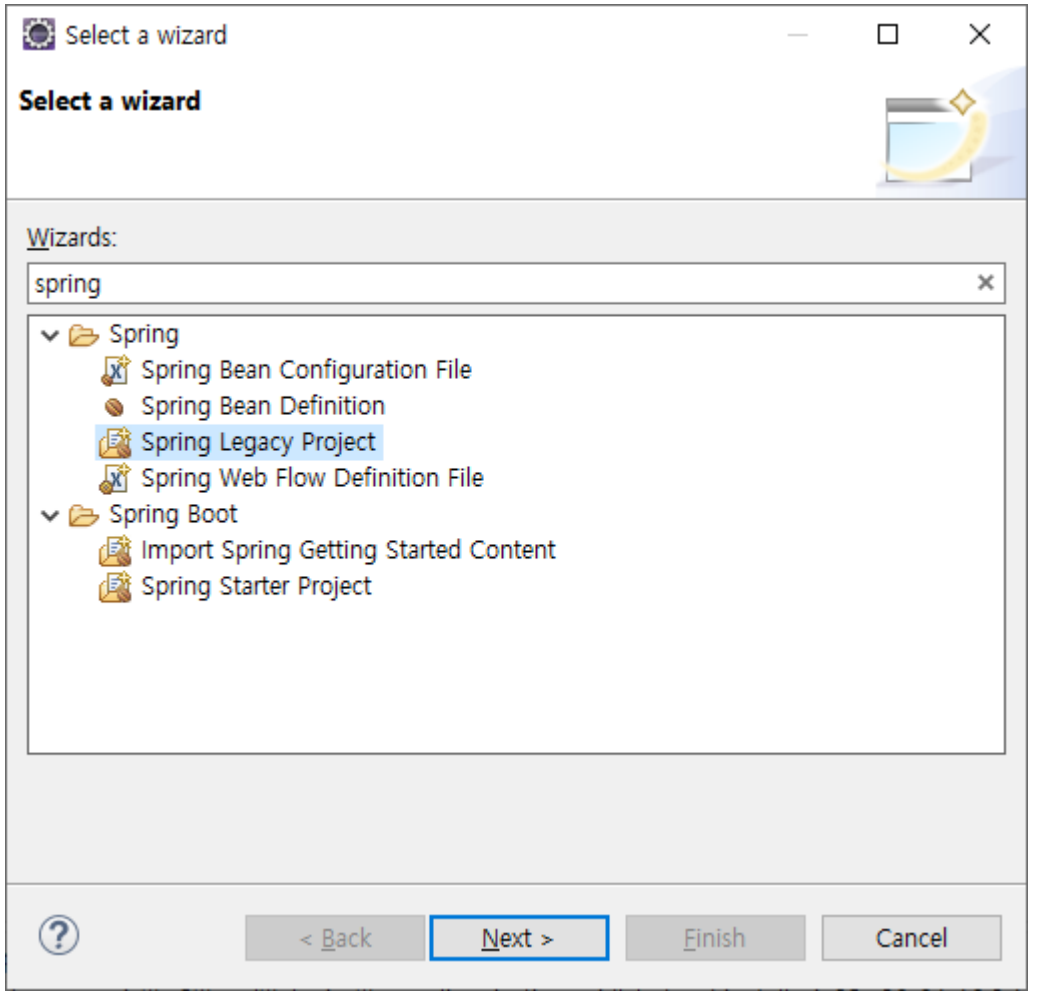
Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

경로 패턴 매핑

1. 프로젝트를 생성하시오.



Spring Lagacy Project



3

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

2. Templates 를 선택하시오.

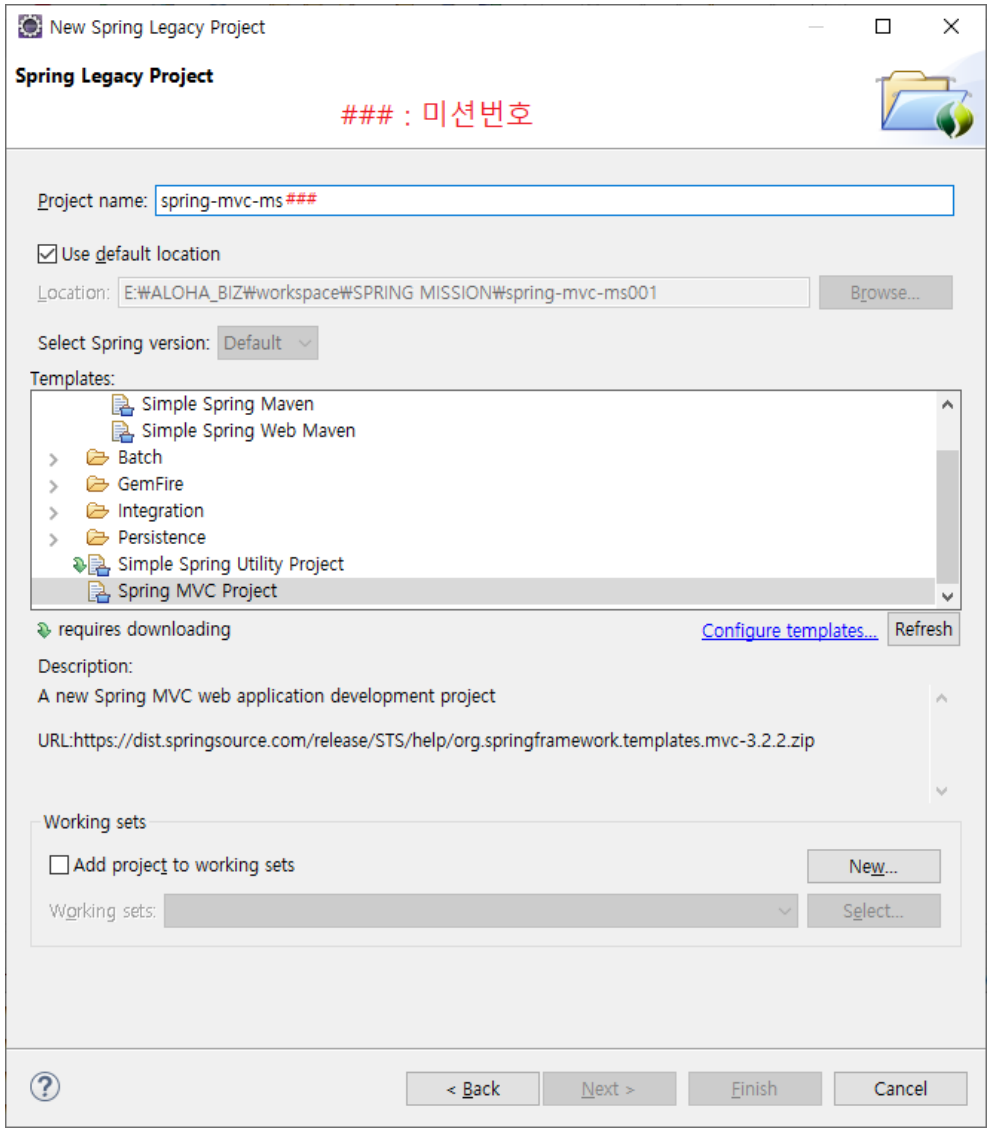
경로 패턴 매핑

[Project name]

spring-mvc-ms003

[Templates]

Spring MVC Project



3

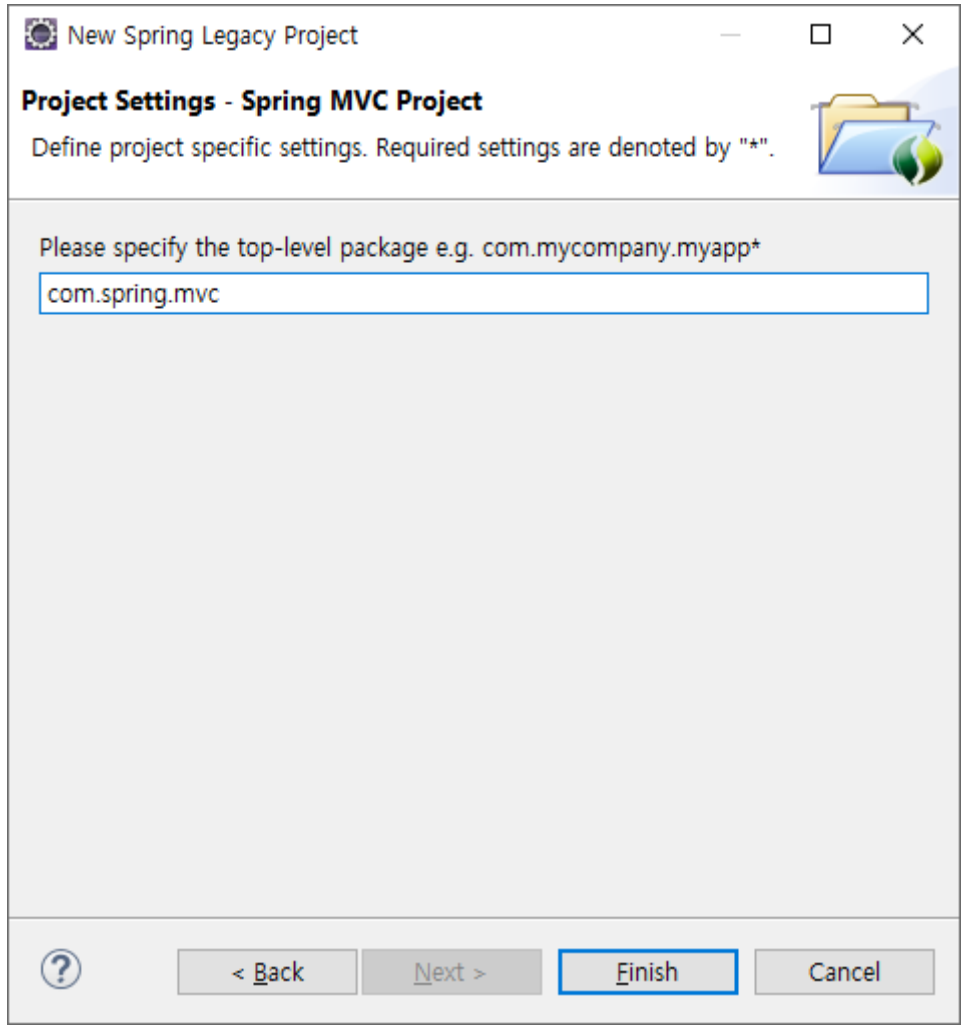
Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

3. Package 경로를 입력하시오.

경로 패턴 매핑

[Package]

com.spring.mvc



3

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

경로 패턴 매핑

프로젝트 구조

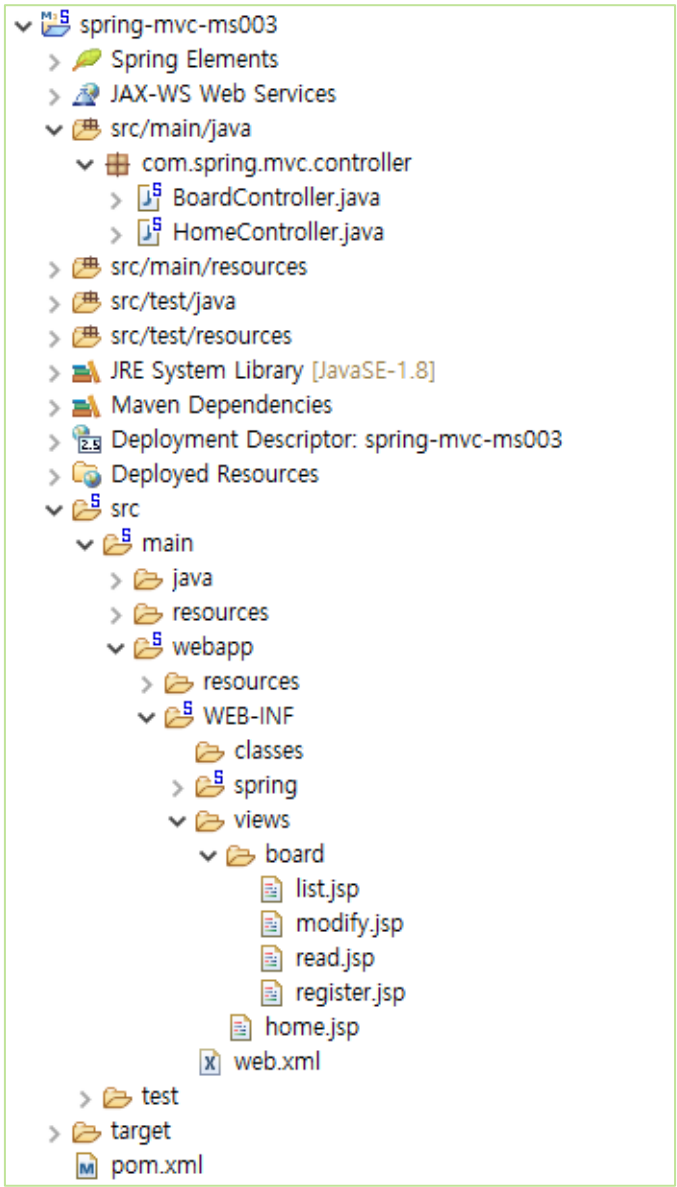
1. com.spring.mvc 패키지 하위 경로에 controller 패키지를 생성하시오.

2. 'Boardcontroller' 클래스를 생성하시오.

3. ~/views 하위 경로에 board 폴더를 생성하시오.

4. ~/views/board 하위 경로에 JSP 파일을 생성하시오.

   - list.jsp
   - register.jsp
   - modify.jsp
   - read.jsp



3

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청경로 매핑 테이블

Controller	Request Path/Method		Views	기능
BoardController	/register	GET	/views/board/register.jsp	등록화면
	registerForm()			
	/modify	GET	/views/board/modify.jsp	수정화면
	modifyForm()			
	/list	GET	/views/board/list.jsp	목록화면
	list()			
	/read/{boardNo}		/views/board/read.jsp	조회화면
	read(int boardNo)			
	/read2/{no}		/views/board/read.jsp	조회화면
/board	read2(int boardNo)			

* 요청 경로에 파라미터 정보를 매핑한다.

Controller	Request Path/Method		Views	기능
HomeController	/	GET	/views/home.jsp	메인화면
	home(Locale locale, Model model)			

3

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청경로 매핑

"요청경로 매핑 테이블 " 을 참조하여,

1) BoardController.java 파일에 <글 등록화면>을 요청하는 메서드를 작성하시오.

2) BoardController.java 파일에 <글 수정화면>을 요청하는 메서드를 작성하시오.

3) BoardController.java 파일에 <글 목록화면>을 요청하는 메서드를 작성하시오.

4) BoardController.java 파일에 <글 조회화면>을 요청하는 메서드를 작성하시오.
   1) 요청경로에 파라미터 정보를 매핑하시오.
   2) URL : /board/read/10 , read() 메서드에 경로변수 boardNo가 파라미터 boardNo에 매핑되어 값 10 이 전달되도록 하시오.
   3) URL : /board/read2/20 , read() 메서드에 경로변수 no가 파라미터 boardNo에 매핑되어 값 20 이 전달되도록 하시오.

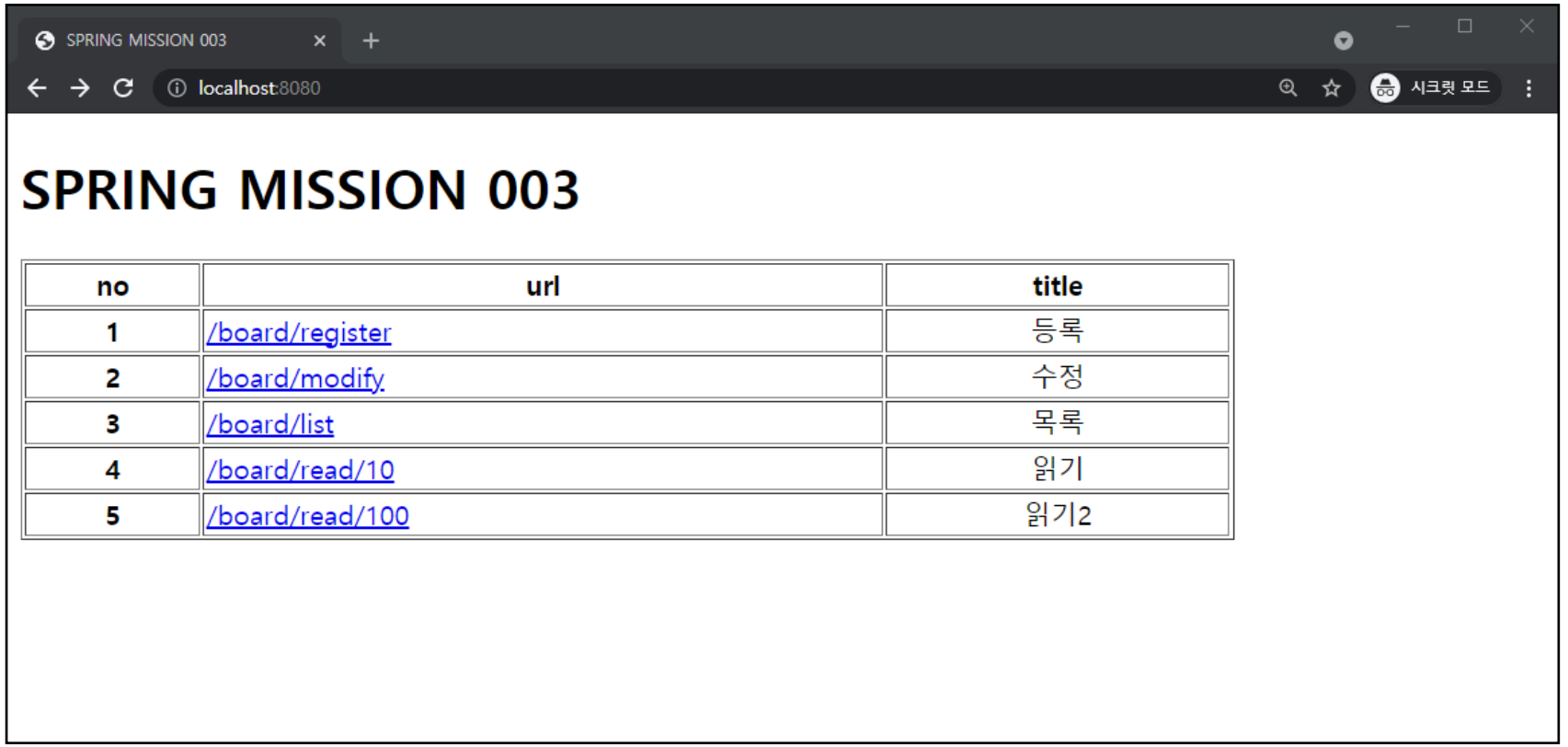
5) HomeController.java 파일에 <메인화면>을 요청하는 메서드를 작성하시오.

3

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

경로 패턴 매핑

localhost:8080/



SPRING MISSION 003

no	url	title
1	/board/register	등록
2	/board/modify	수정
3	/board/list	목록
4	/board/read/10	읽기
5	/board/read/100	읽기2

GUIDE

1. <h1> 태그를 이용하여 제목을 작성하시오.

2. <table> 태그를 이용하여 표를 작성하시오.

3. <a> 태그를 이용하여, 링크를 작성하시오.

- /board/register
- /board/modify
- /board/list

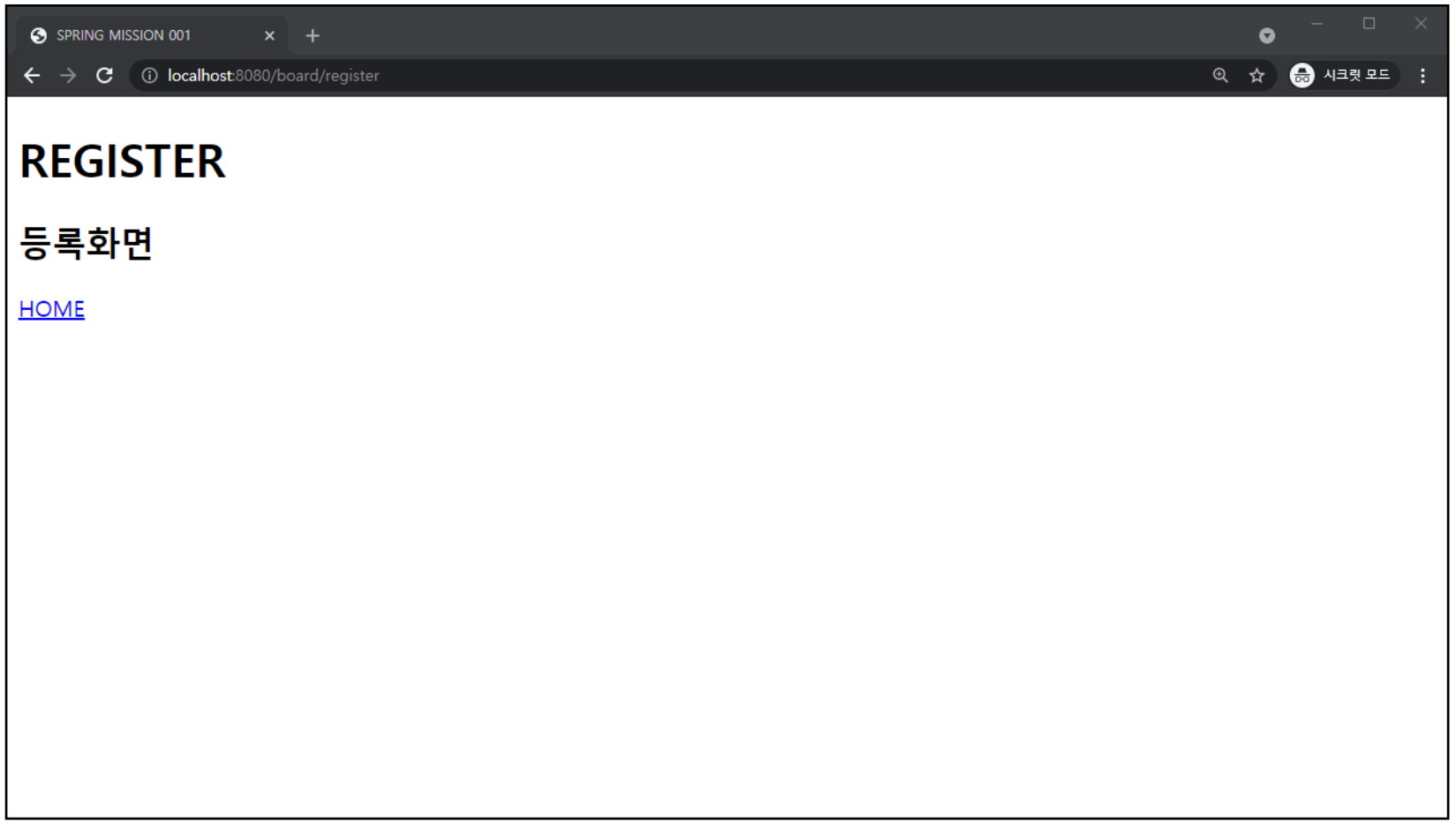
file

~/views/home.jsp

3

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

localhost:8080/board/register



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

file

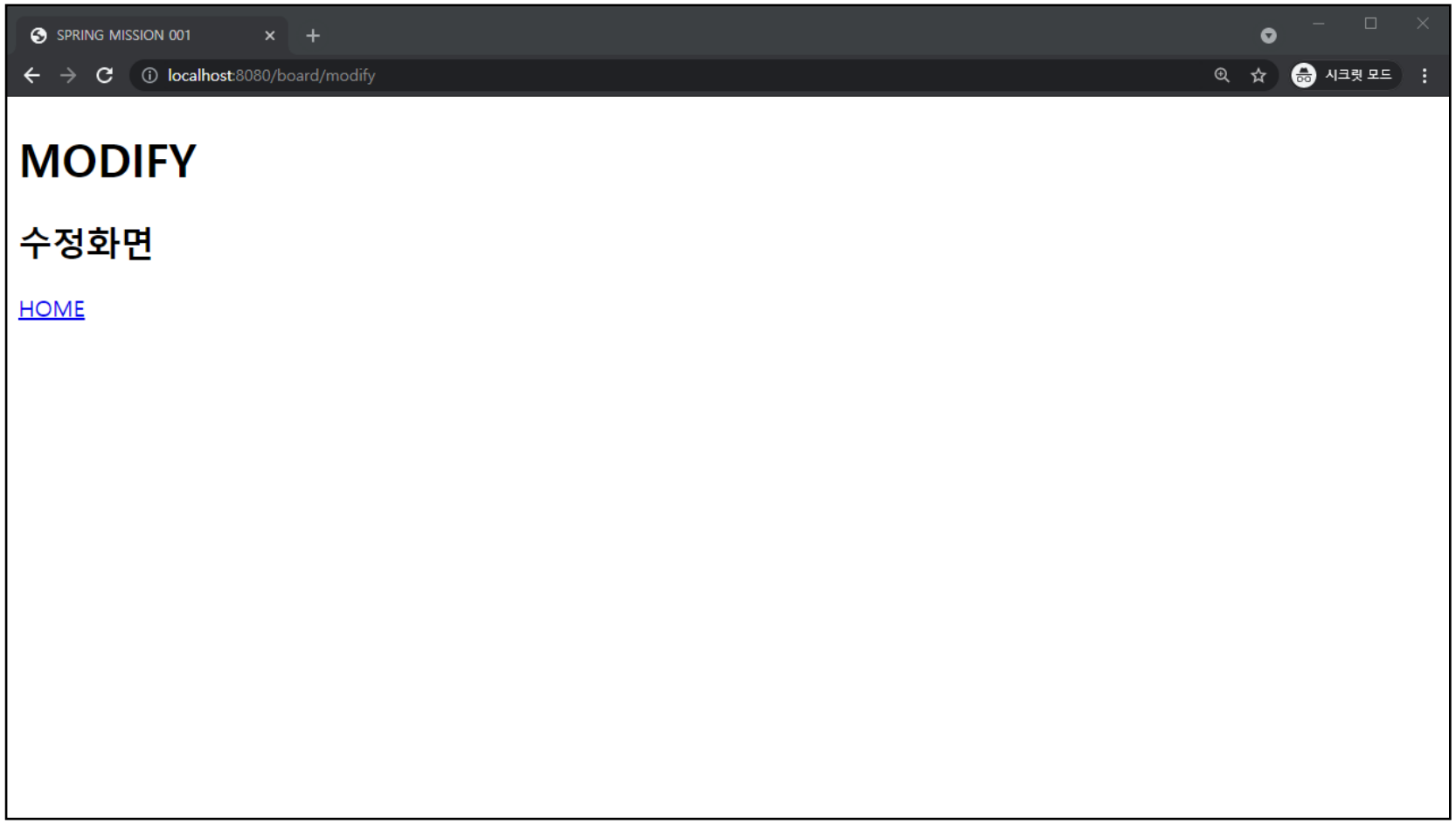
~/views/board/register.jsp

3

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

경로 패턴 매핑

localhost:8080/board/modify



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

file

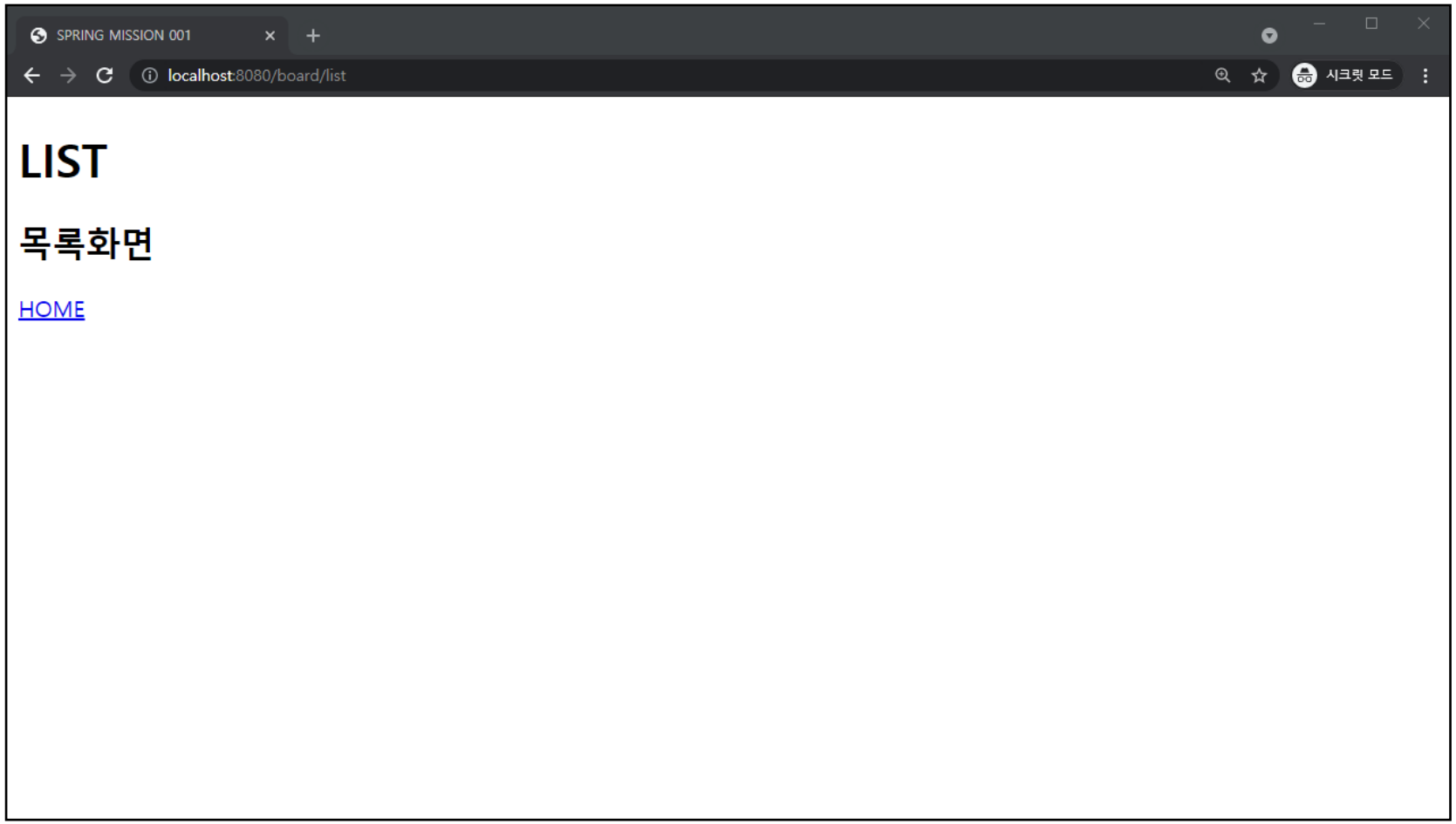
~/views/board/modify.jsp

3

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

경로 패턴 매핑

localhost:8080/board/list



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

file

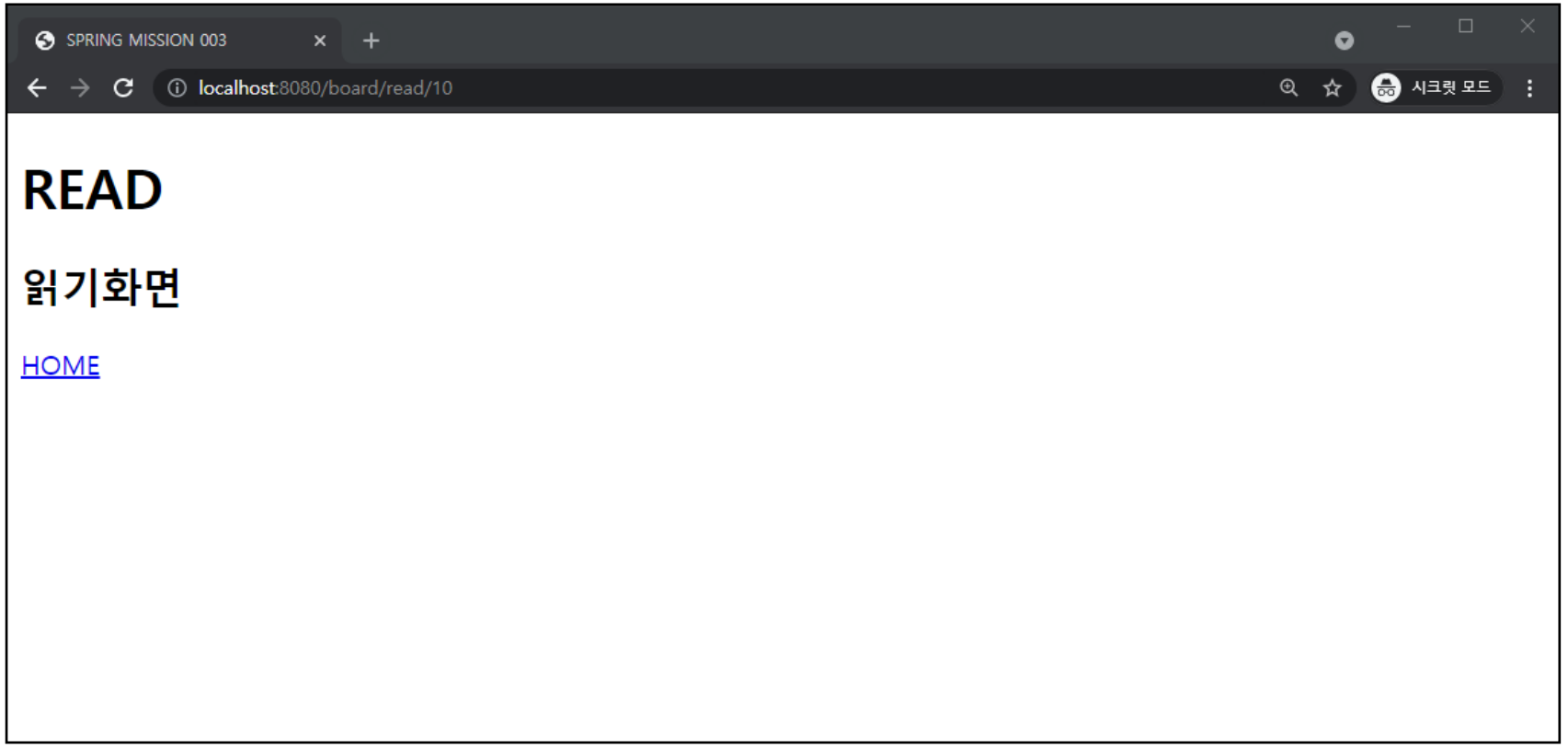
~/views/board/list.jsp

3

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

경로 패턴 매핑

localhost:8080/board/list



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

file

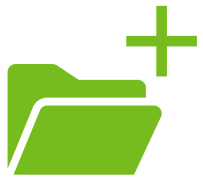
~/views/board/read.jsp

4

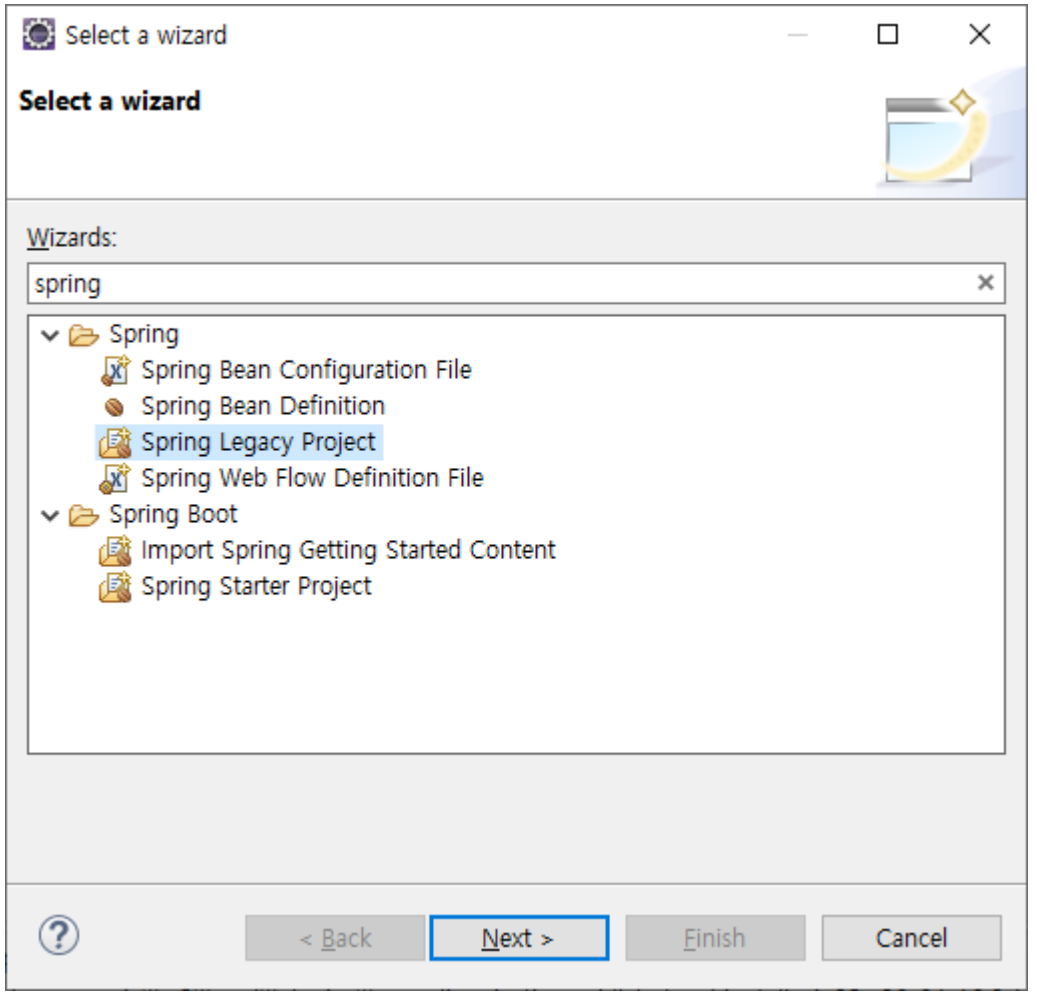
Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

Http 메서드 매핑

1. 프로젝트를 생성하시오.



Spring Lagacy Project



4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

Http 메서드 매핑

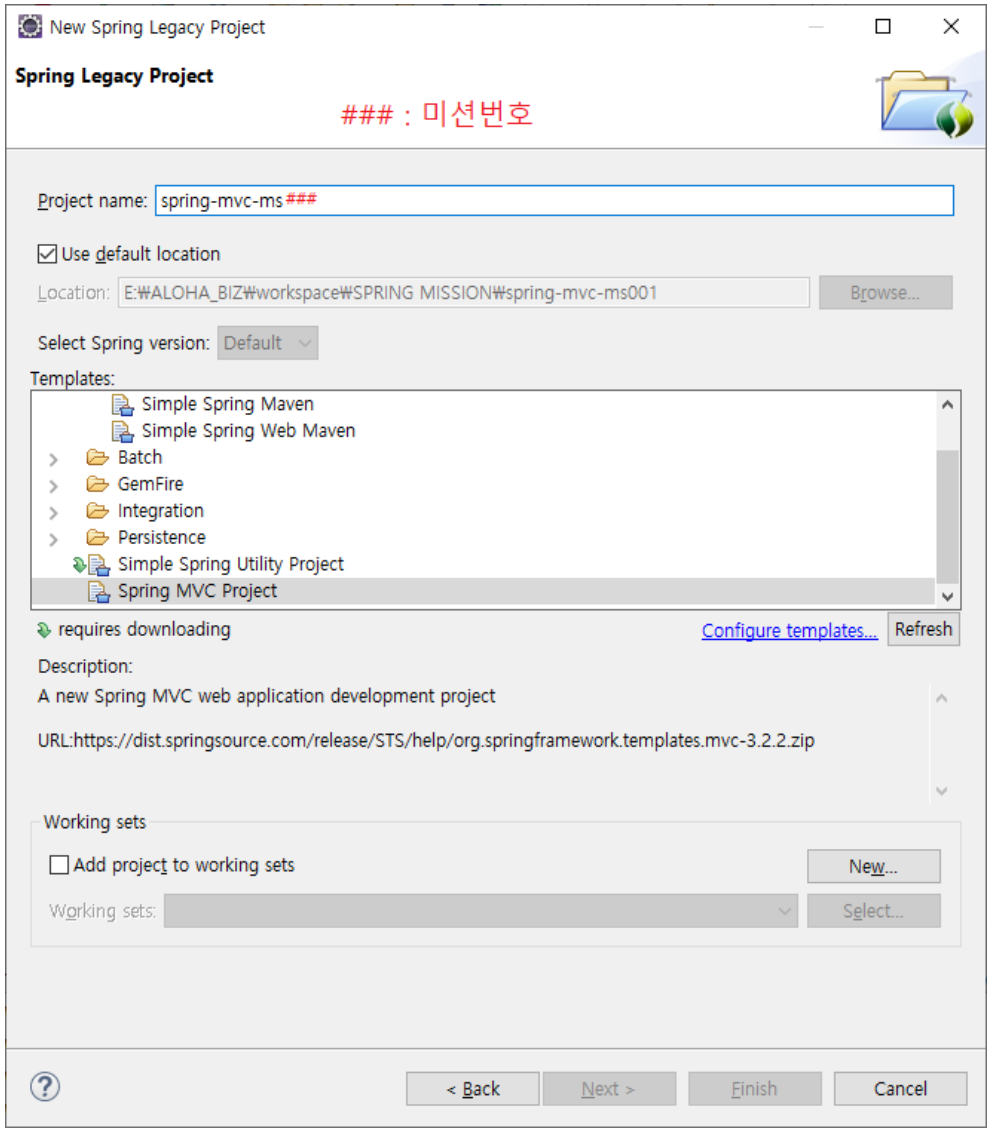
2. Templates 를 선택하시오.

[Project name]

spring-mvc-ms004

[Templates]

Spring MVC Project



4

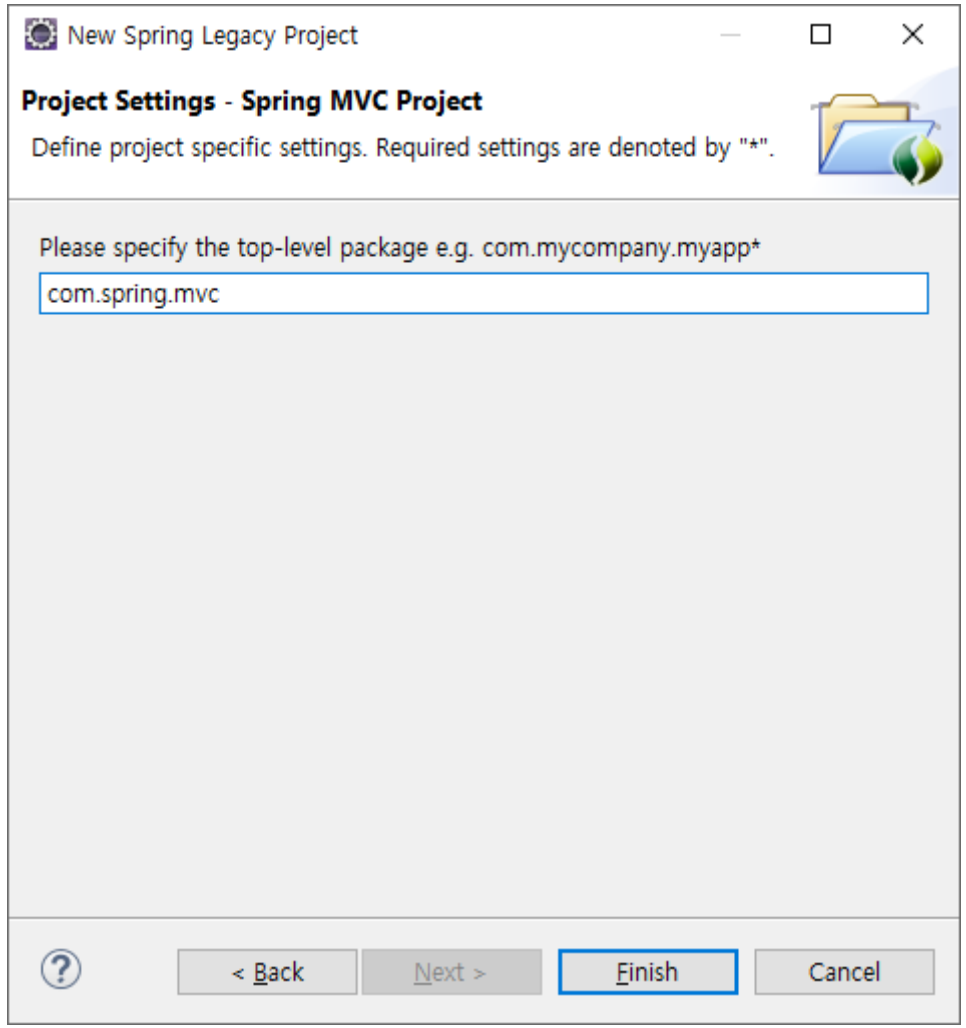
Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

Http 메서드 매핑

3. Package 경로를 입력하시오.

[Package]

com.spring.mvc



4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

Http 메서드 매핑

프로젝트 구조

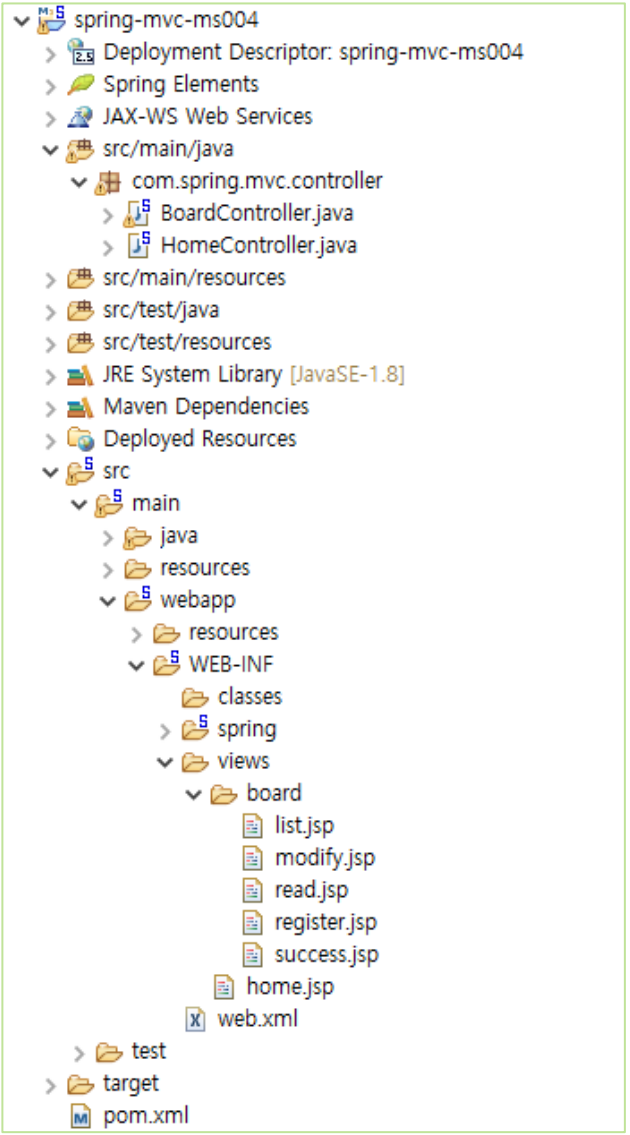
1. com.spring.mvc 패키지 하위 경로에 controller 패키지를 생성하시오.

2. 'Boardcontroller' 클래스를 생성하시오.

3. ~/views 하위 경로에 board 폴더를 생성하시오.

4. ~/views/board 하위 경로에 JSP 파일을 생성하시오.

   - list.jsp
   - register.jsp
   - modify.jsp
   - read.jsp
   - success.jsp



4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청경로 매핑 테이블

Controller	Request Path/Method			Views	기능
BoardController	/register	registerForm()	GET	/views/board/register.jsp	등록
	/register	register()	POST	/views/board/success.jsp	
	/modify	modifyForm()	GET	/views/board/modify.jsp	수정
	/modify	modify()	POST	/views/board/success.jsp	
	/remove	remove()	POST	/views/board/success.jsp	삭제
	/list	list()	GET	/views/board/list.jsp	목록
/board	/read	read(int boardNo)	GET	/views/board/read.jsp	조회

Controller	Request Path/Method		Views	기능
HomeController	/	GET	/views/home.jsp	메인화면
	home(Locale locale, Model model)			

4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청경로 매핑

"요청경로 매핑 테이블 " 을 참조하여,

1) BoardController.java 파일에 <글 등록화면> 을 GET 방식으로 요청하는 메서드를 작성하시오.

2) BoardController.java 파일에 <글 등록처리> 를 POST 방식으로 요청하는 메서드를 작성하시오.

3) BoardController.java 파일에 <글 수정화면> 을 GET 방식으로 요청하는 메서드를 작성하시오.

4) BoardController.java 파일에 <글 수정처리> 를 POST 방식으로 요청하는 메서드를 작성하시오.

5) BoardController.java 파일에 <글 삭제처리> 를 POST 방식으로 요청하는 메서드를 작성하시오.

6) BoardController.java 파일에 <글 목록화면> 을 GET 방식으로 요청하는 메서드를 작성하시오.

7) BoardController.java 파일에 <글 조회화면> 을 GET 방식으로 요청하는 메서드를 작성하시오.

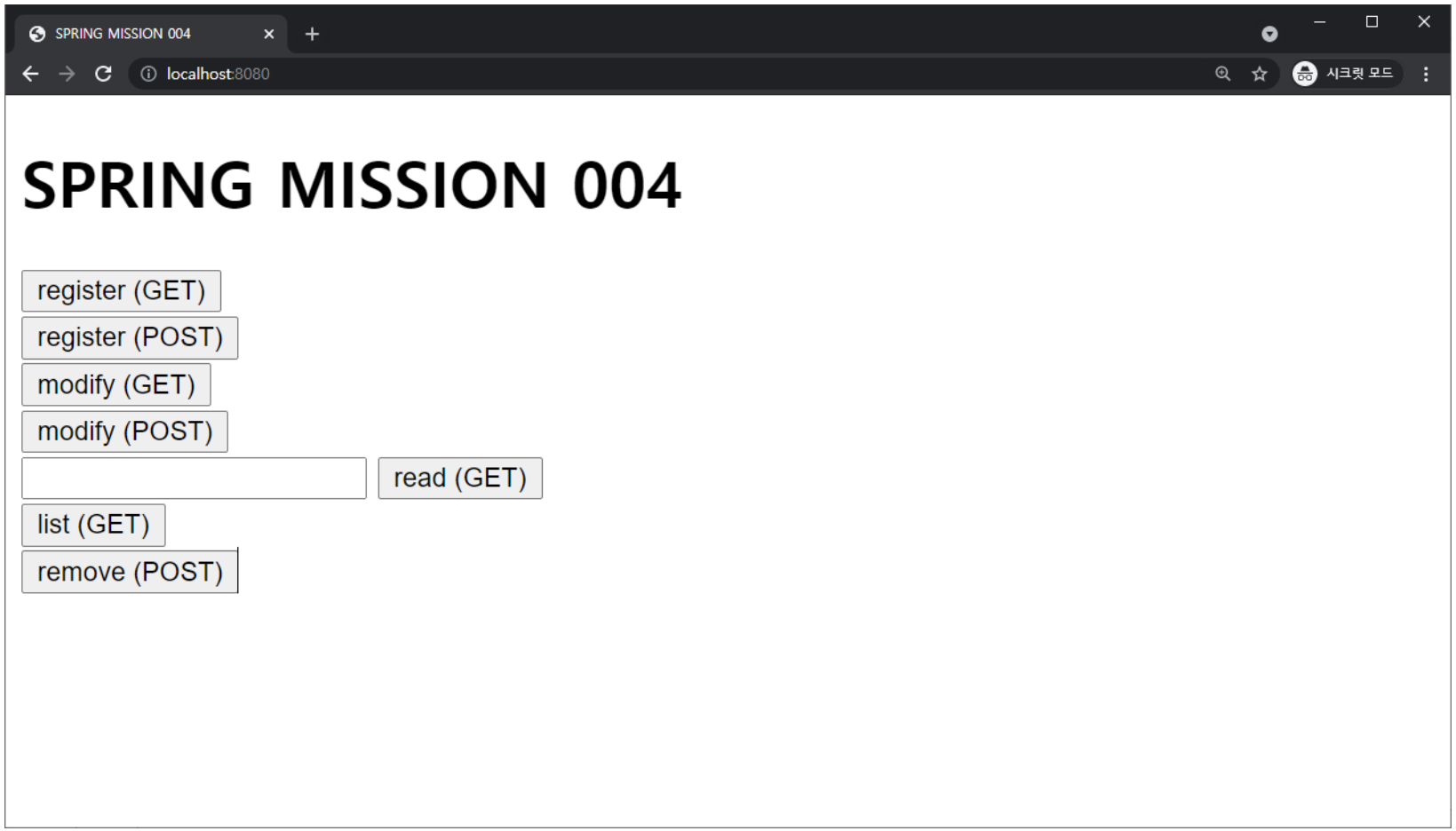
8) HomeController.java 파일에 <메인화면>을 요청하는 메서드를 작성하시오.

4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

경로 패턴 매핑

localhost:8080/



GUIDE

1. <h1> 태그를 이용하여 제목을 작성하시오.

2. <form> 태그를 이용하여 제출양식을 작성하시오.
각 경로 및 메서드 방식을 지정하시오.

- /board/register (GET)
- /board/register (POST)
- /board/modify (GET)
- /board/modify (POST)
- /board/read (GET)
- /board/list (GET)
- /board/remove (POST)

3. <input> 태그를 이용하여, 입력상자와 제출버튼을 작성하시오.

file

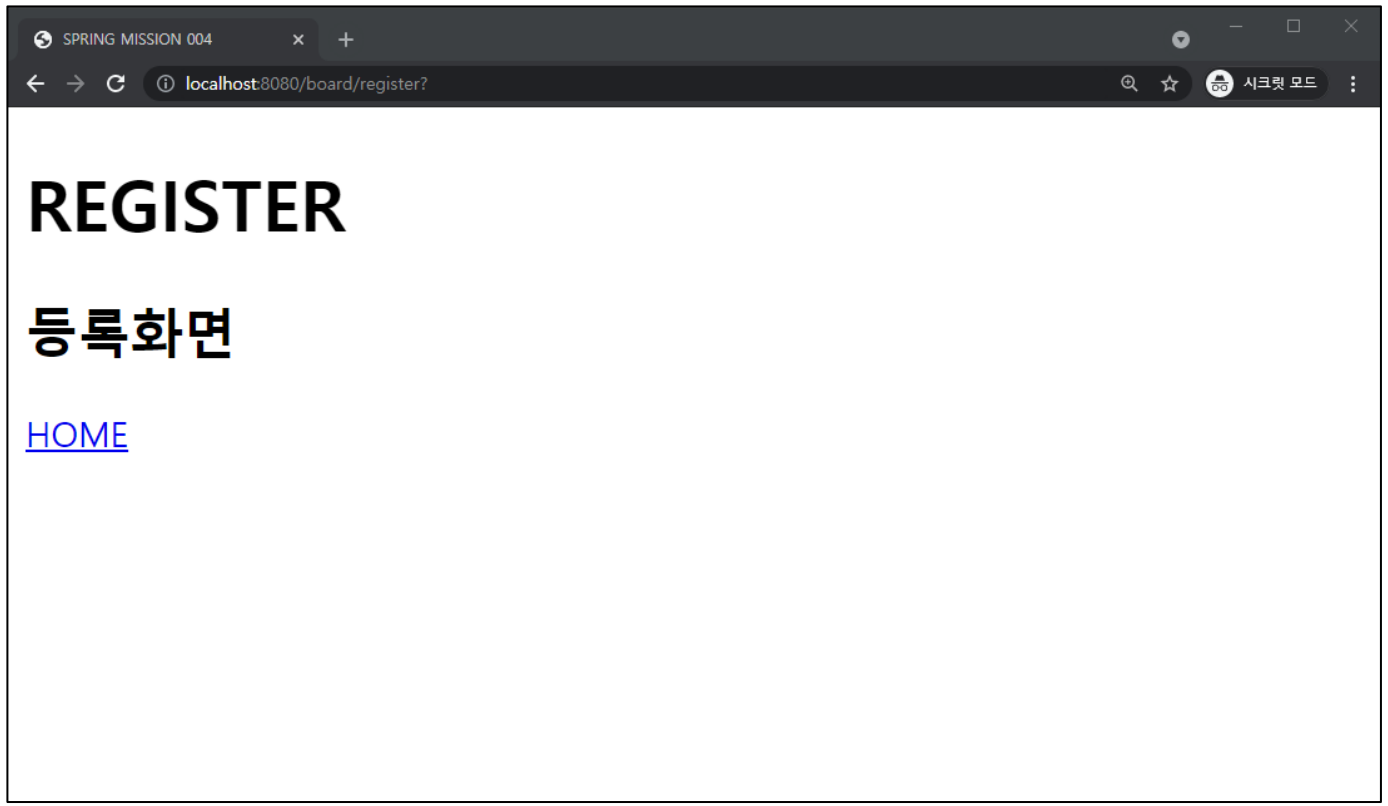
~/views/home.jsp

4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

경로 패턴 매핑

localhost:8080/board/register



register (GET)

GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

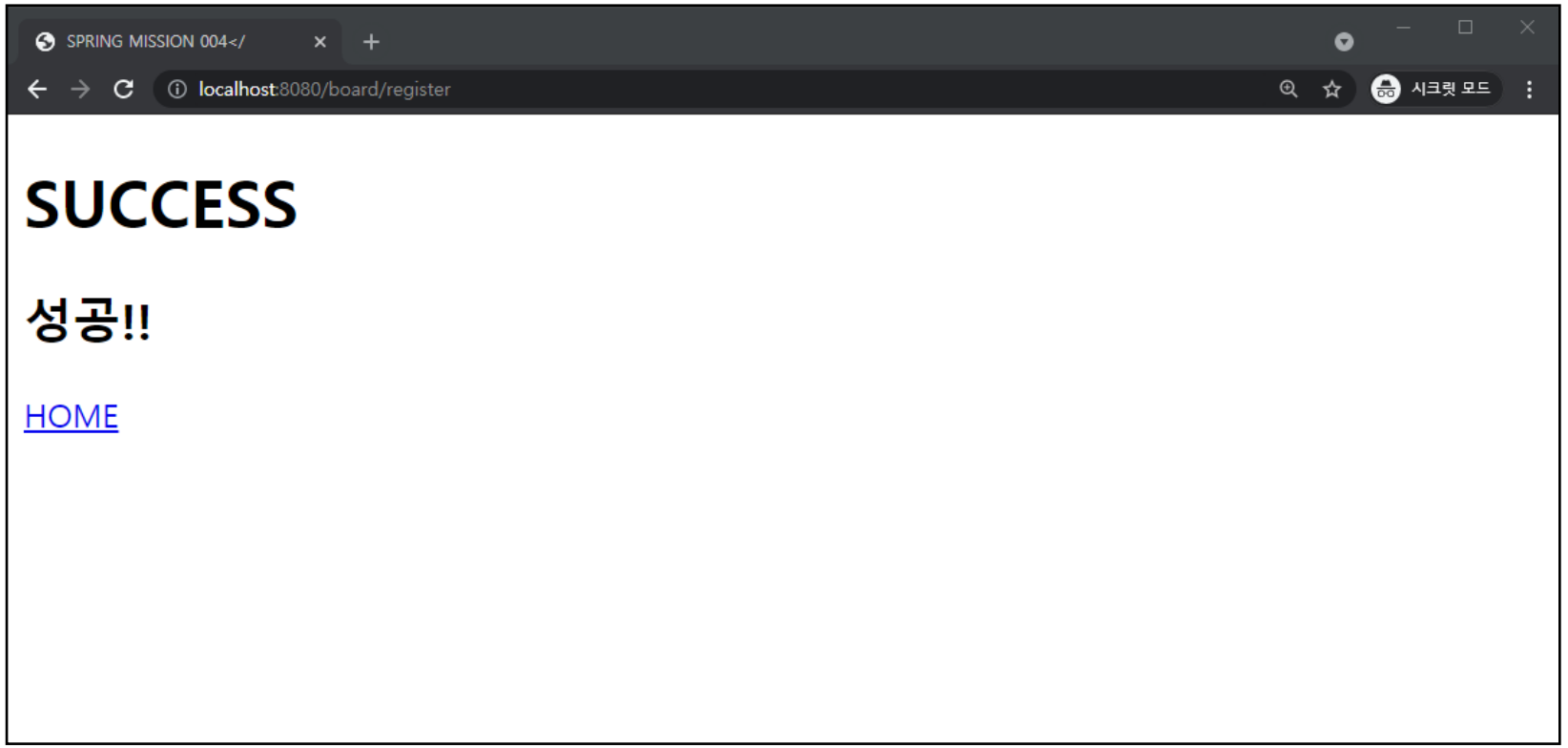
file

~/views/board/register.jsp

4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

localhost:8080/board/register



register (POST)

file

~/views/board/success.jsp

GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

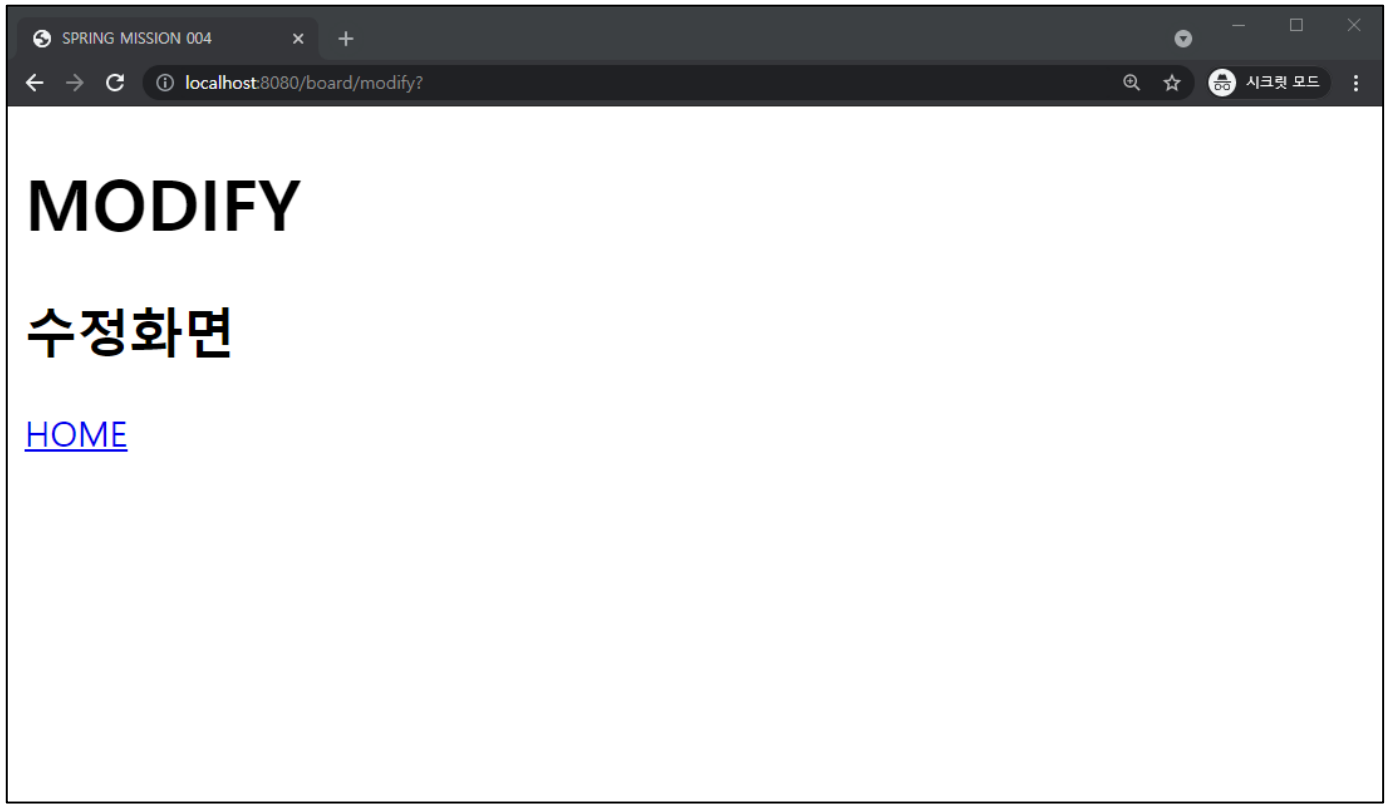
2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

경로 패턴 매핑

localhost:8080/board/modify



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

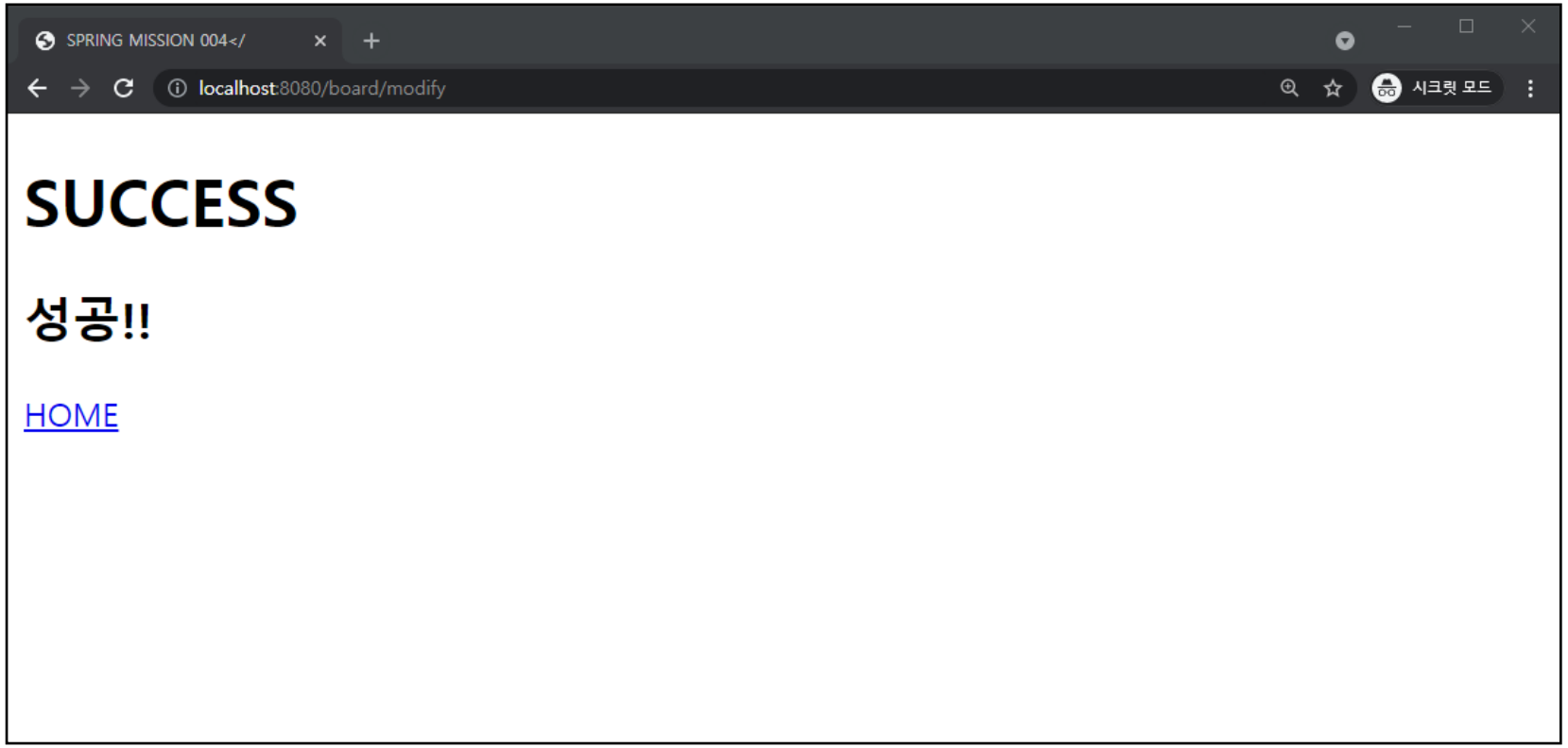
file

~/views/board/modify.jsp

4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

localhost:8080/board/modify



modify (POST)

file

~/views/board/success.jsp

GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

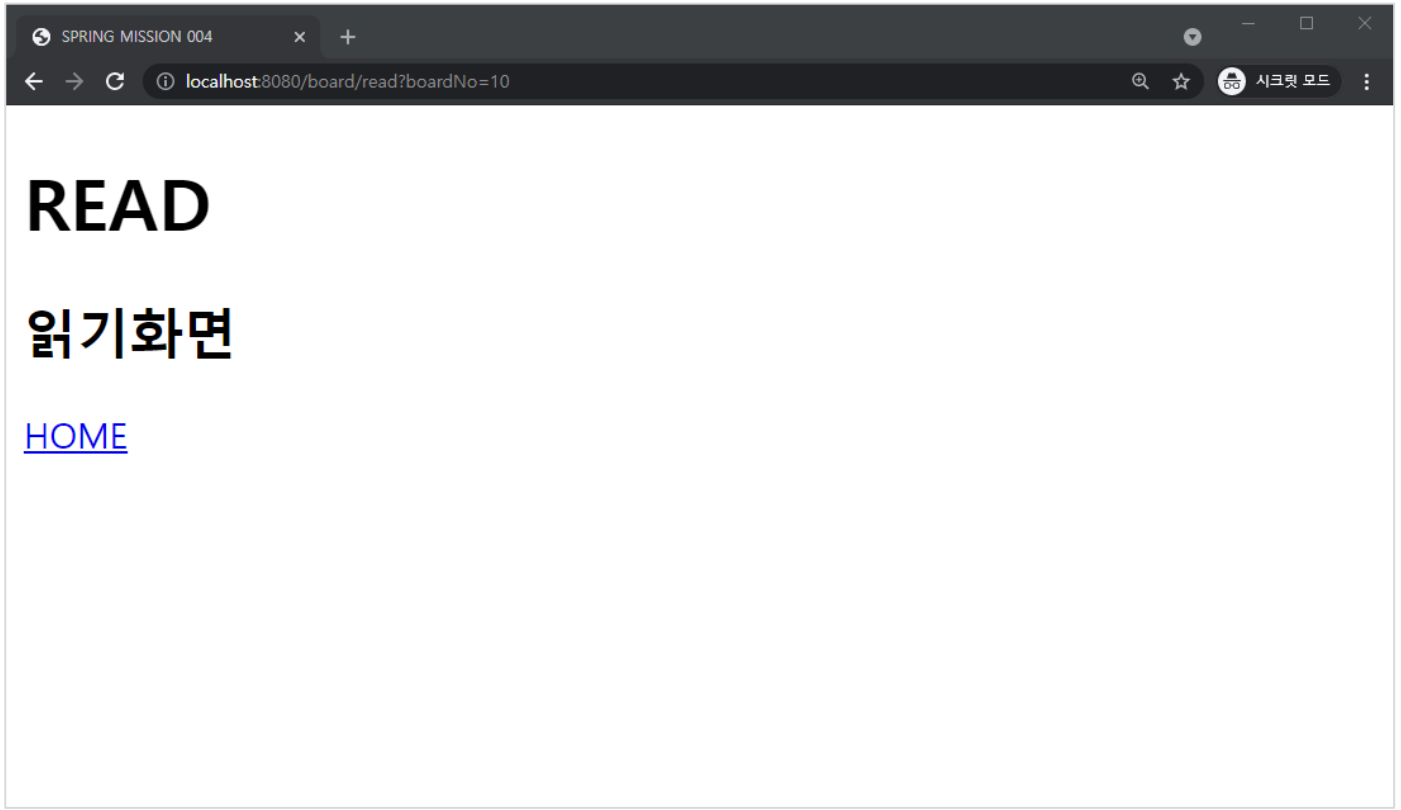
2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

경로 패턴 매핑

localhost:8080/board/read



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

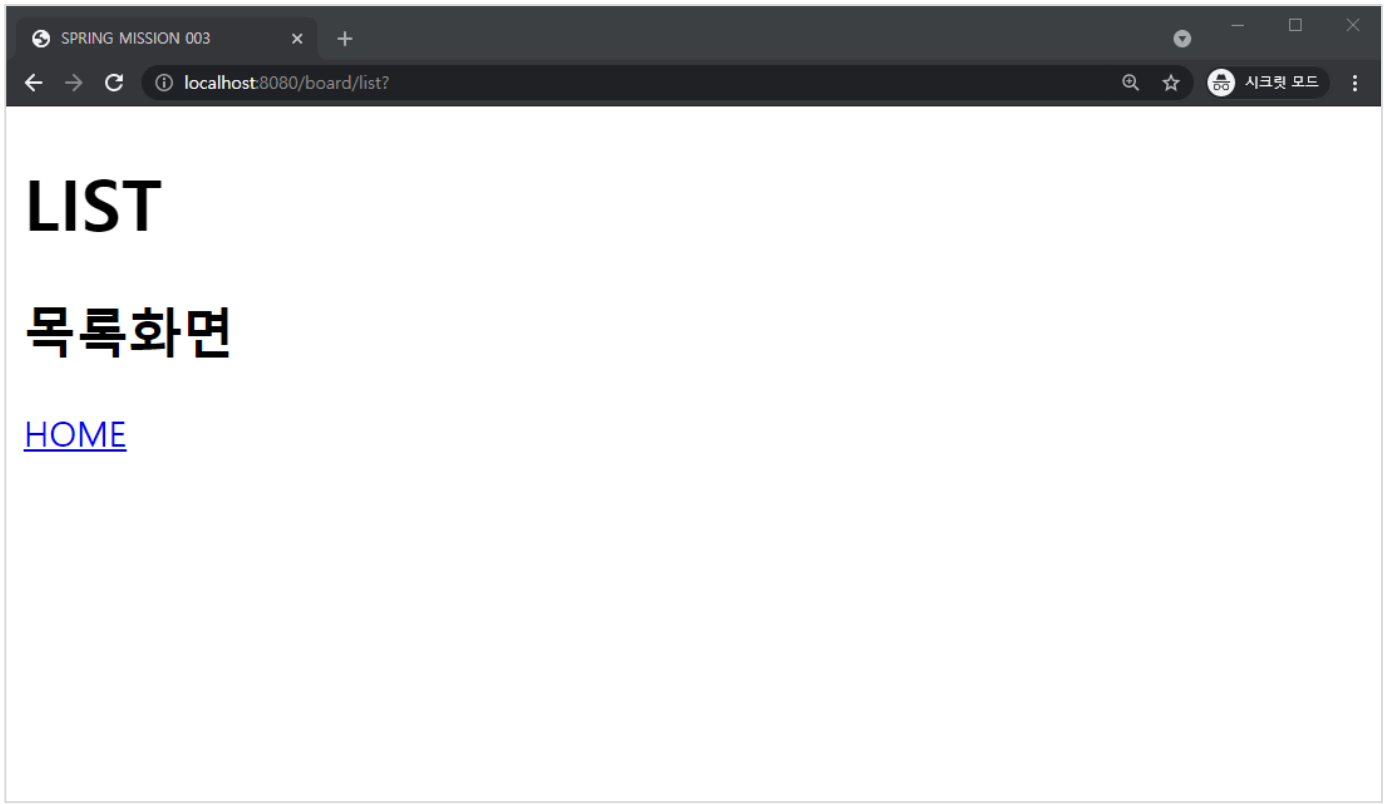
file

~/views/board/read.jsp

4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

localhost:8080/board/list



file

~/views/board/list.jsp

GUIDE

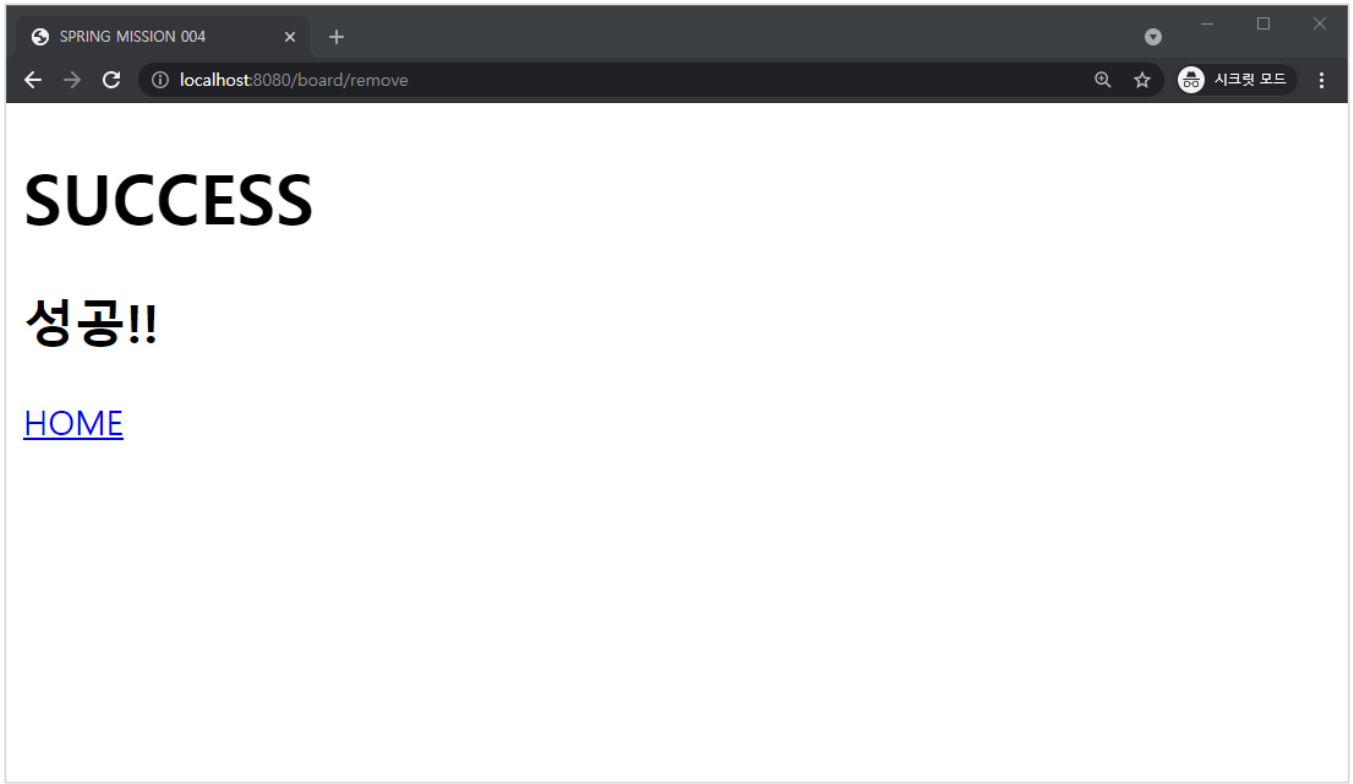
1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

4

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

localhost:8080/board/remove



remove (POST)

file

~/views/board/success.jsp

GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

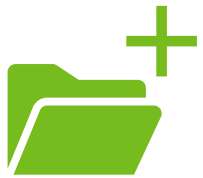
2. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

5

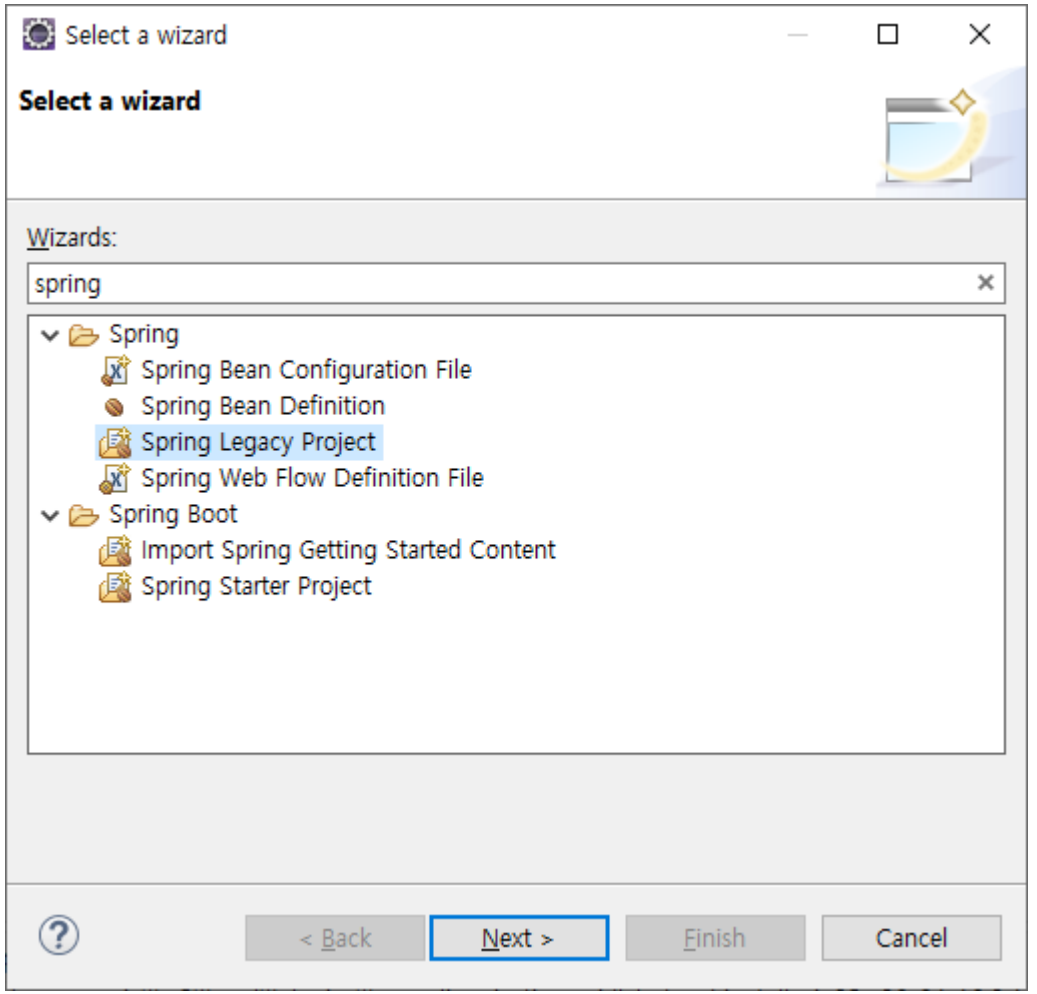
Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

Params 매핑

1. 프로젝트를 생성하시오.



Spring Lagacy Project



5

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

2. Templates 를 선택하시오.

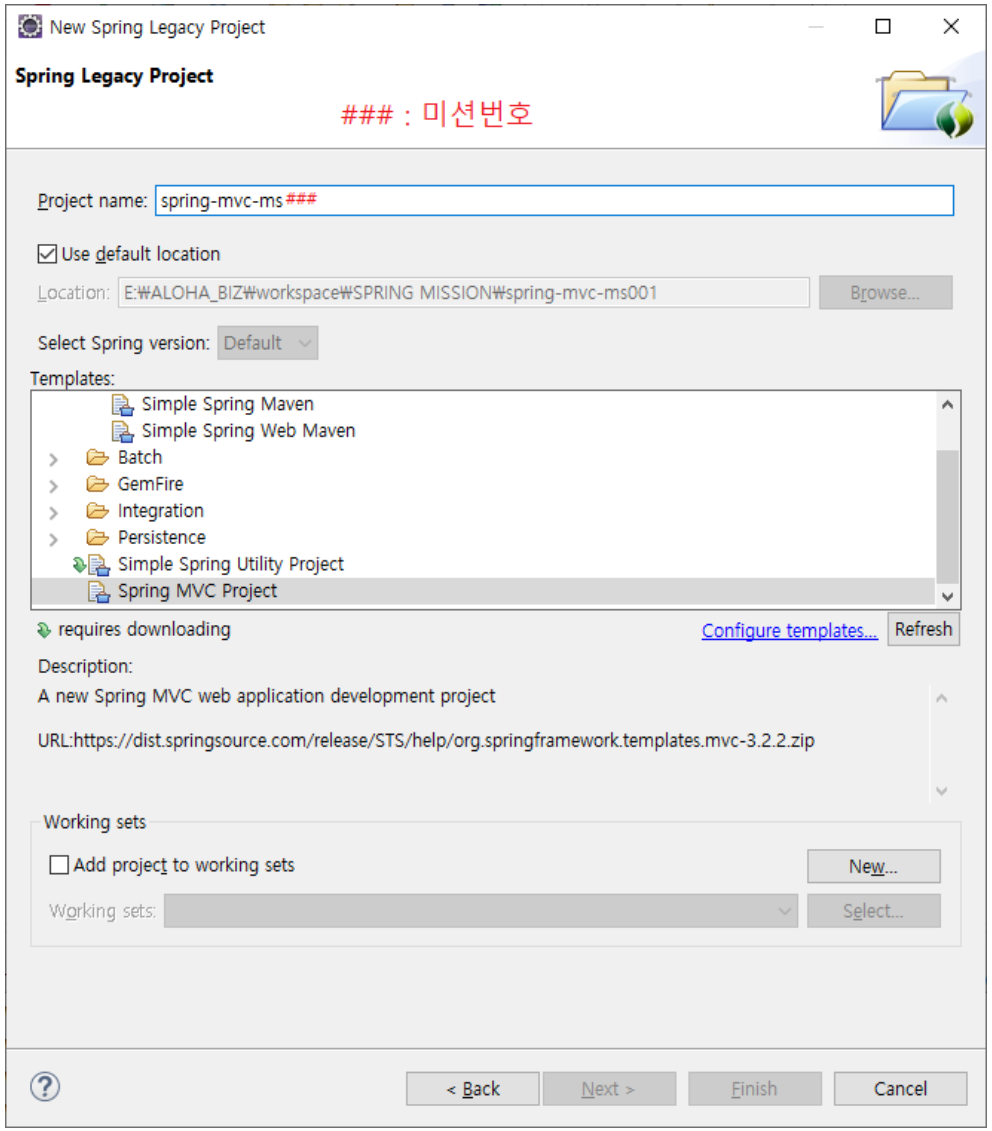
Params 매핑

[Project name]

spring-mvc-ms005

[Templates]

Spring MVC Project



5

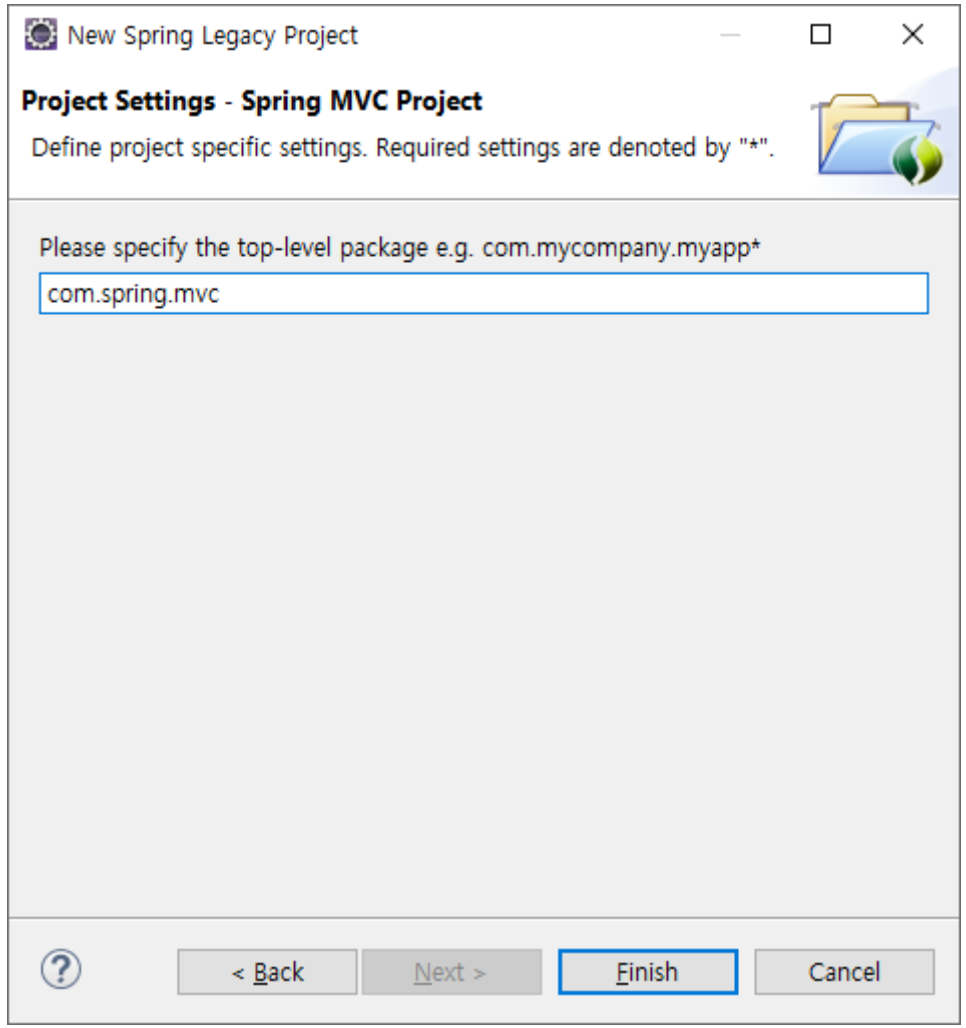
Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

3. Package 경로를 입력하시오.

Params 매핑

[Package]

com.spring.mvc



5

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

Params 매핑

프로젝트 구조

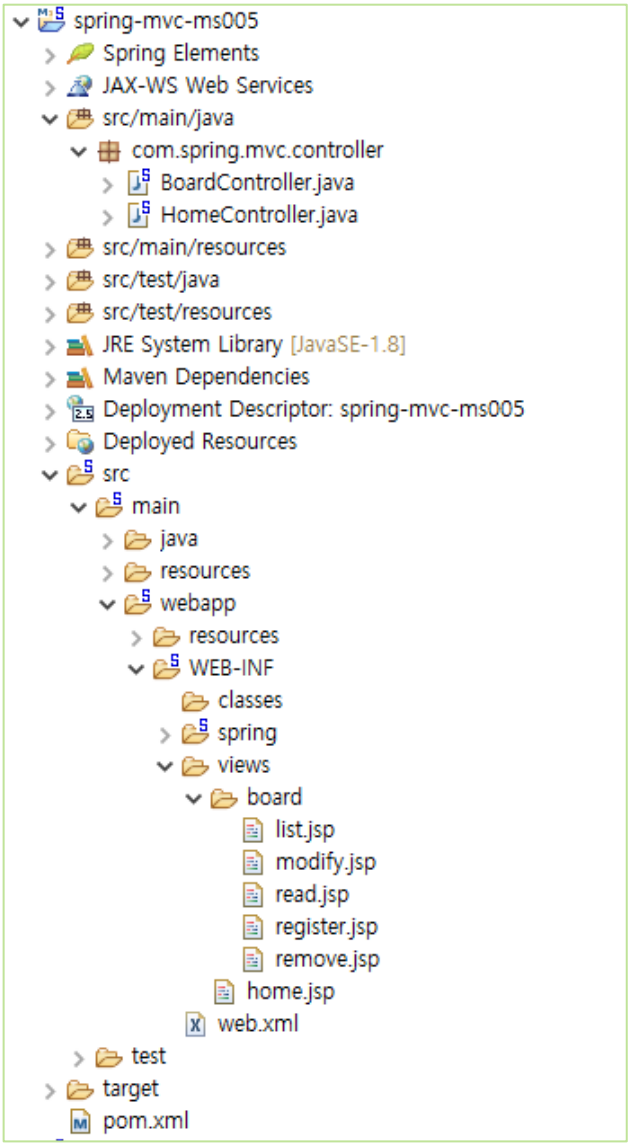
1. com.spring.mvc 패키지 하위 경로에 controller 패키지를 생성하시오.

2. ‘Boardcontroller’ 클래스를 생성하시오.

3. ~/views 하위 경로에 board 폴더를 생성하시오.

4. ~/views/board 하위 경로에 JSP 파일을 생성하시오.

   - list.jsp
   - register.jsp
   - modify.jsp
   - read.jsp
   - remove.jsp



5

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청경로 매핑 테이블

Controller	Request Path/Method			Views	기능
BoardController	/get?register	registerForm()	GET	/views/board/register.jsp	등록
	/post?register	register()	POST	/views/board/list.jsp	
	/get?modify	modifyForm()	GET	/views/board/modify.jsp	수정
	/post?modify	modify()	POST	/views/board/list.jsp	
	/get?remove	removeForm()	GET	/views/board/remove.jsp	삭제
	/post?remove	remove	POST	/views/board/list.jsp	
	/get?list	list()	GET	/views/board/list.jsp	목록
/board	/get?read	read(int boardNo)	GET	/views/board/read.jsp	조회

* 요청 경로 매핑을 param 속성을 이용하여 매핑한다.

Controller	Request Path/Method		Views	기능
HomeController	/	GET	/views/home.jsp	메인화면
	home(Locale locale, Model model)			

5

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

요청경로 매핑

"요청경로 매핑 테이블 " 을 참조하여,

1) BoardController.java 파일에 <글 등록화면> 을 GET 방식으로 요청하는 메서드를 작성하시오.

2) BoardController.java 파일에 <글 등록처리> 를 POST 방식으로 요청하는 메서드를 작성하시오.

3) BoardController.java 파일에 <글 수정화면> 을 GET 방식으로 요청하는 메서드를 작성하시오.

4) BoardController.java 파일에 <글 수정처리> 를 POST 방식으로 요청하는 메서드를 작성하시오.

5) BoardController.java 파일에 <글 삭제화면> 을 POST 방식으로 요청하는 메서드를 작성하시오.

6) BoardController.java 파일에 <글 삭제처리> 를 POST 방식으로 요청하는 메서드를 작성하시오.

7) BoardController.java 파일에 <글 목록화면> 을 GET 방식으로 요청하는 메서드를 작성하시오.

8) BoardController.java 파일에 <글 조회화면> 을 GET 방식으로 요청하는 메서드를 작성하시오.

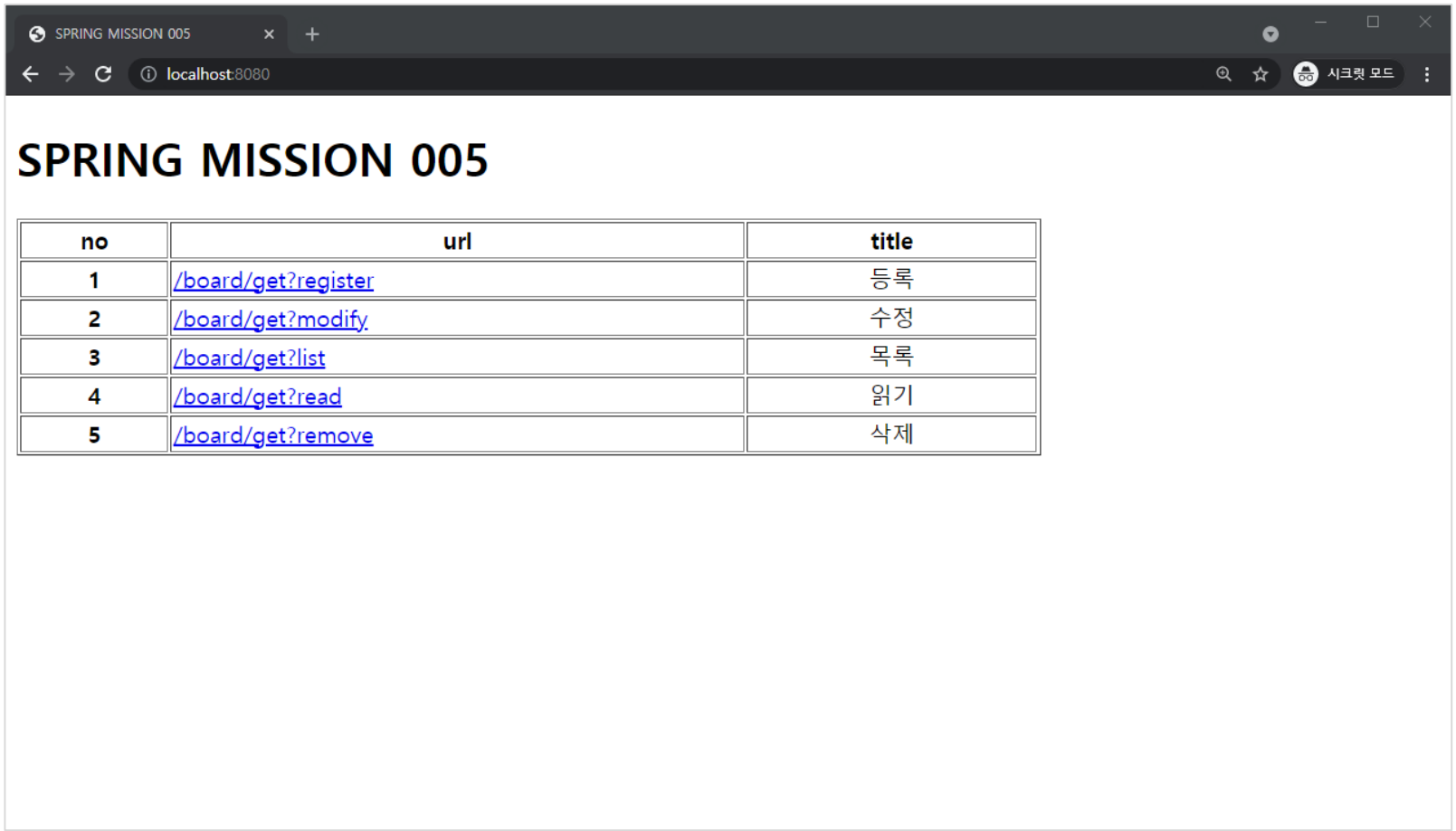
9) HomeController.java 파일에 <메인화면>을 요청하는 메서드를 작성하시오.

5

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

Params 매핑

localhost:8080/



GUIDE

1. <h1> 태그를 이용하여 제목을 작성하시오.

2. <form> 태그를 이용하여 제출양식을 작성하시오.
각 경로 및 메서드 방식을 지정하시오.

- /board/register (GET)
- /board/register (POST)
- /board/modify (GET)
- /board/modify (POST)
- /board/read (GET)
- /board/list (GET)
- /board/remove (POST)

3. <input> 태그를 이용하여, 입력상자와 제출버튼을 작성하시오.

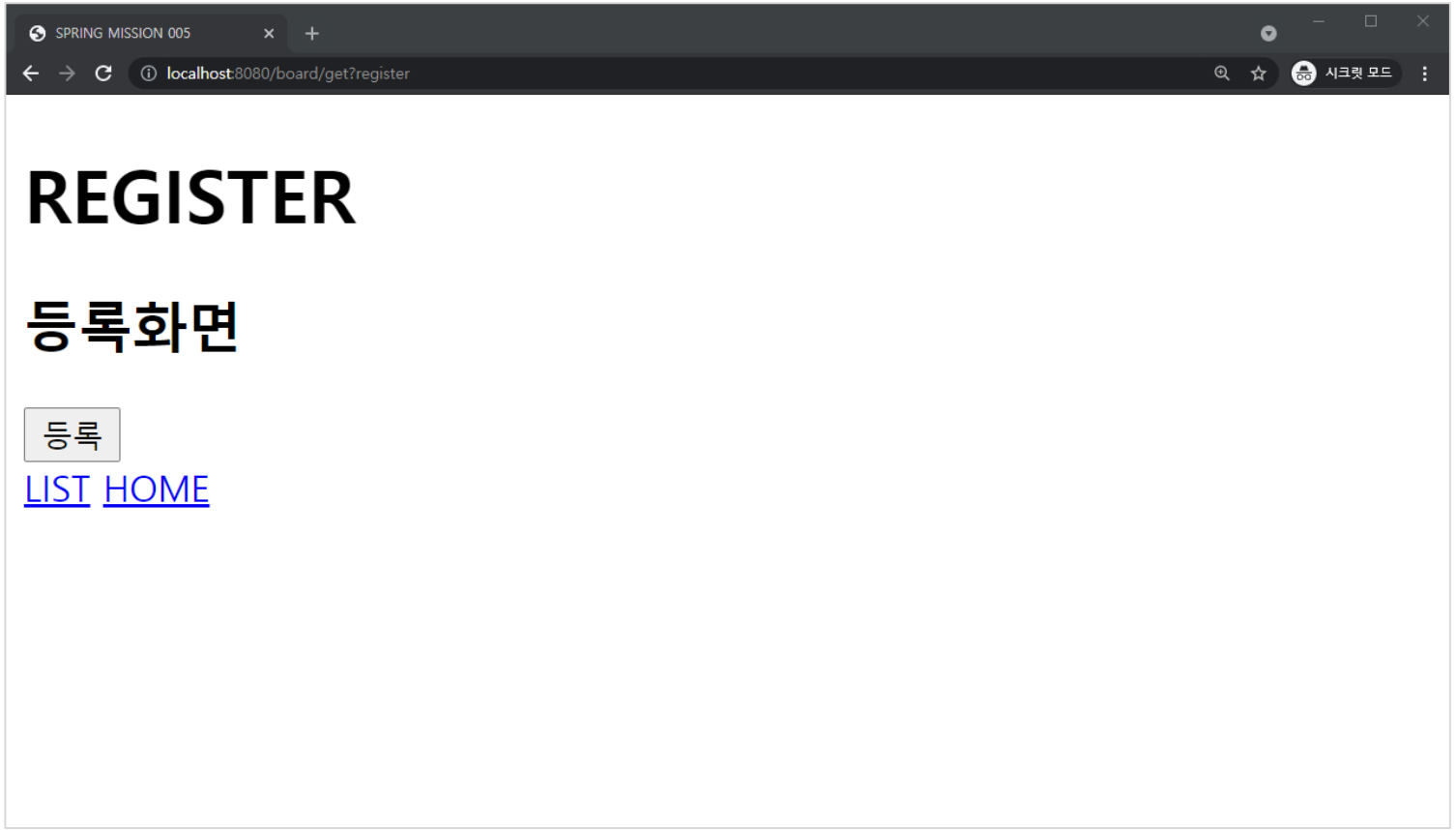
file

~/views/home.jsp

5

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

localhost:8080/board/register



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <form>, <button> 태그를 이용하여 등록버튼 클릭 시,
url: /board/post?register
로 POST 방식으로 요청을 보내는 제출 양식을 작성하시오.

3. <a> 태그를 이용하여, 링크를 작성하시오.

- /board/get?list
- /

file

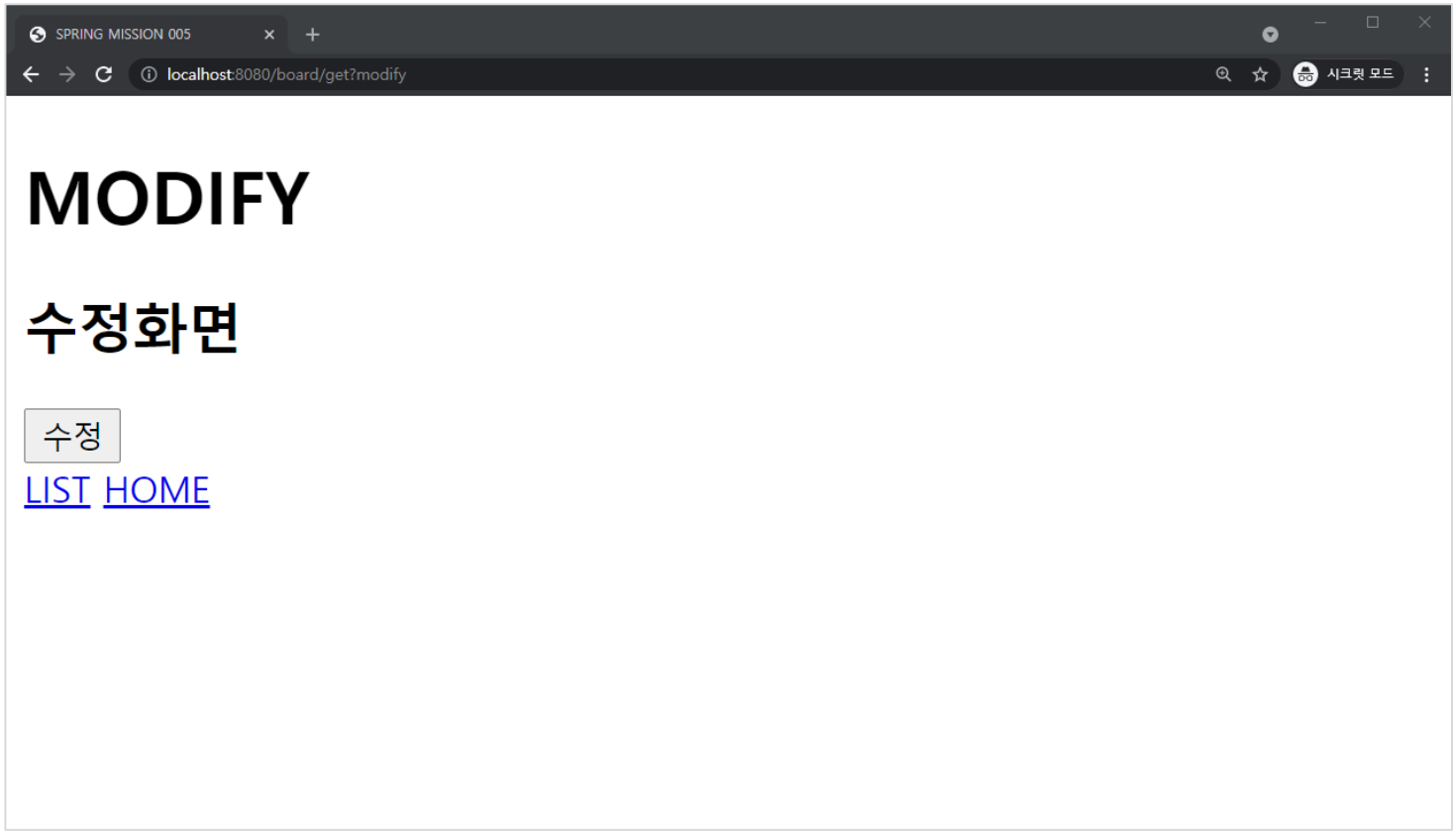
~/views/board/register.jsp

5

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

Params 매핑

localhost:8080/board/modify



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <form>, <button> 태그를 이용하여 등록버튼 클릭 시,
url: /board/post?modify
로 POST 방식으로 요청을 보내는 제출 양식을 작성하시오.

3. <a> 태그를 이용하여, 링크를 작성하시오.

- /board/get?list
- /

file

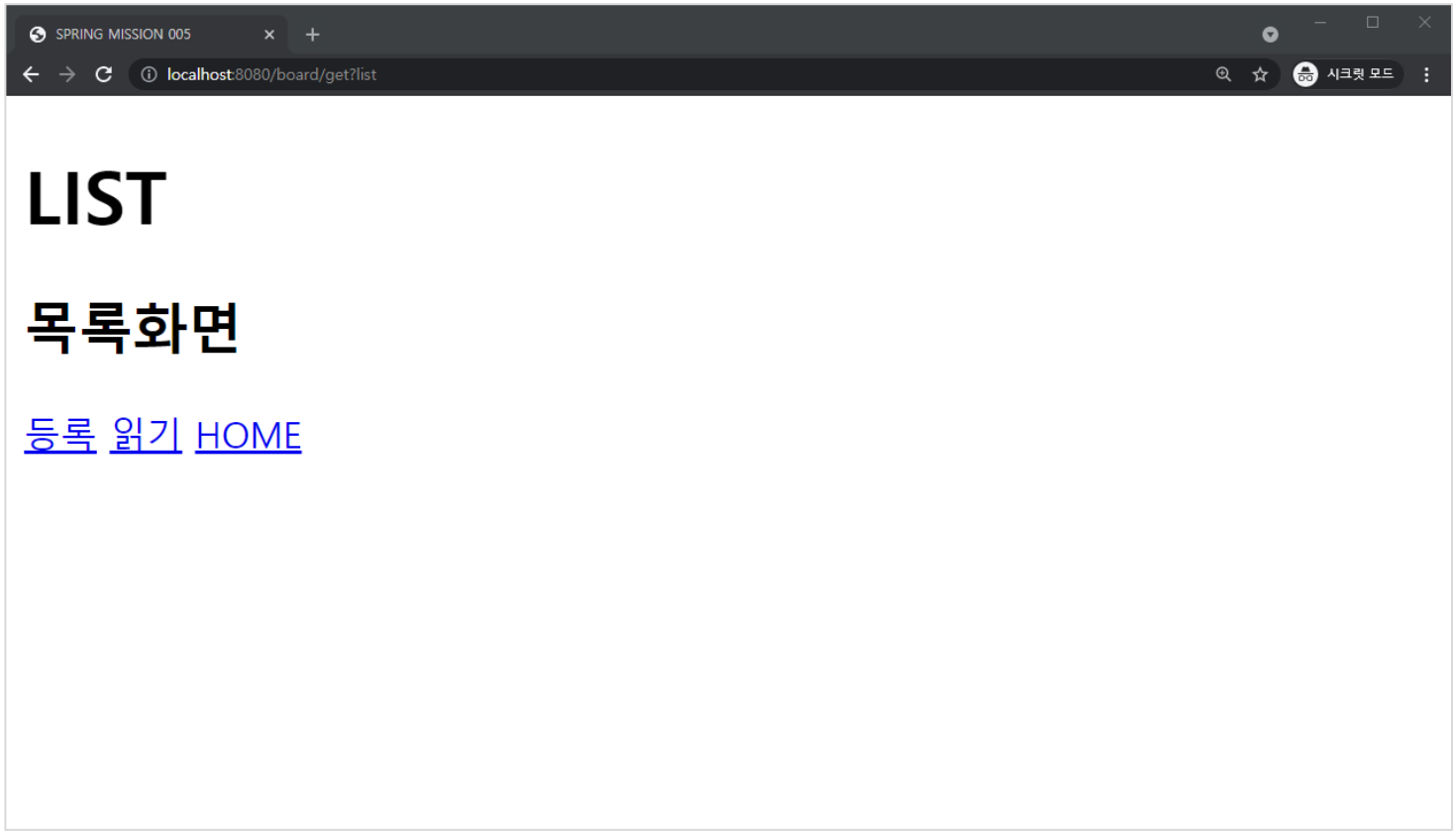
~/views/board/modify.jsp

5

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

Params 매핑

localhost:8080/board/list



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

3. <a> 태그를 이용하여, 링크를 작성하시오.

- /board/get?register
- /board/get?read
- /

file

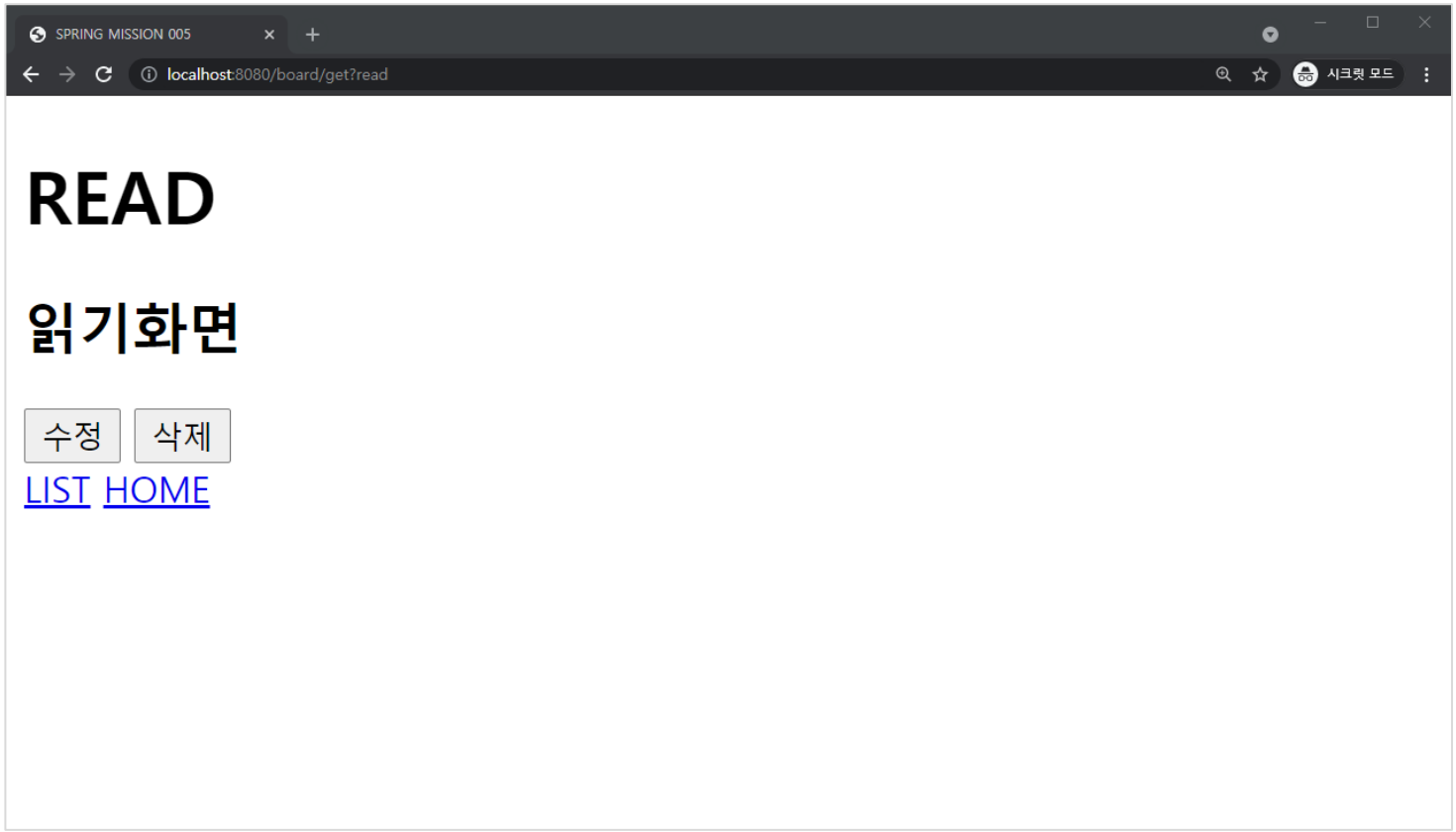
~/views/board/list.jsp

5

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

Params 매핑

localhost:8080/board/read



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <form>, <button> 태그를 이용하여 등록버튼 클릭 시,
(수정)
url: /board/post?modify
(삭제)
url: /board/post?remove
로 POST 방식으로 요청을 보내는 제출 양식을 작성하시오.

3. <a> 태그를 이용하여, 링크를 작성하시오.

- /board/get?list
- /

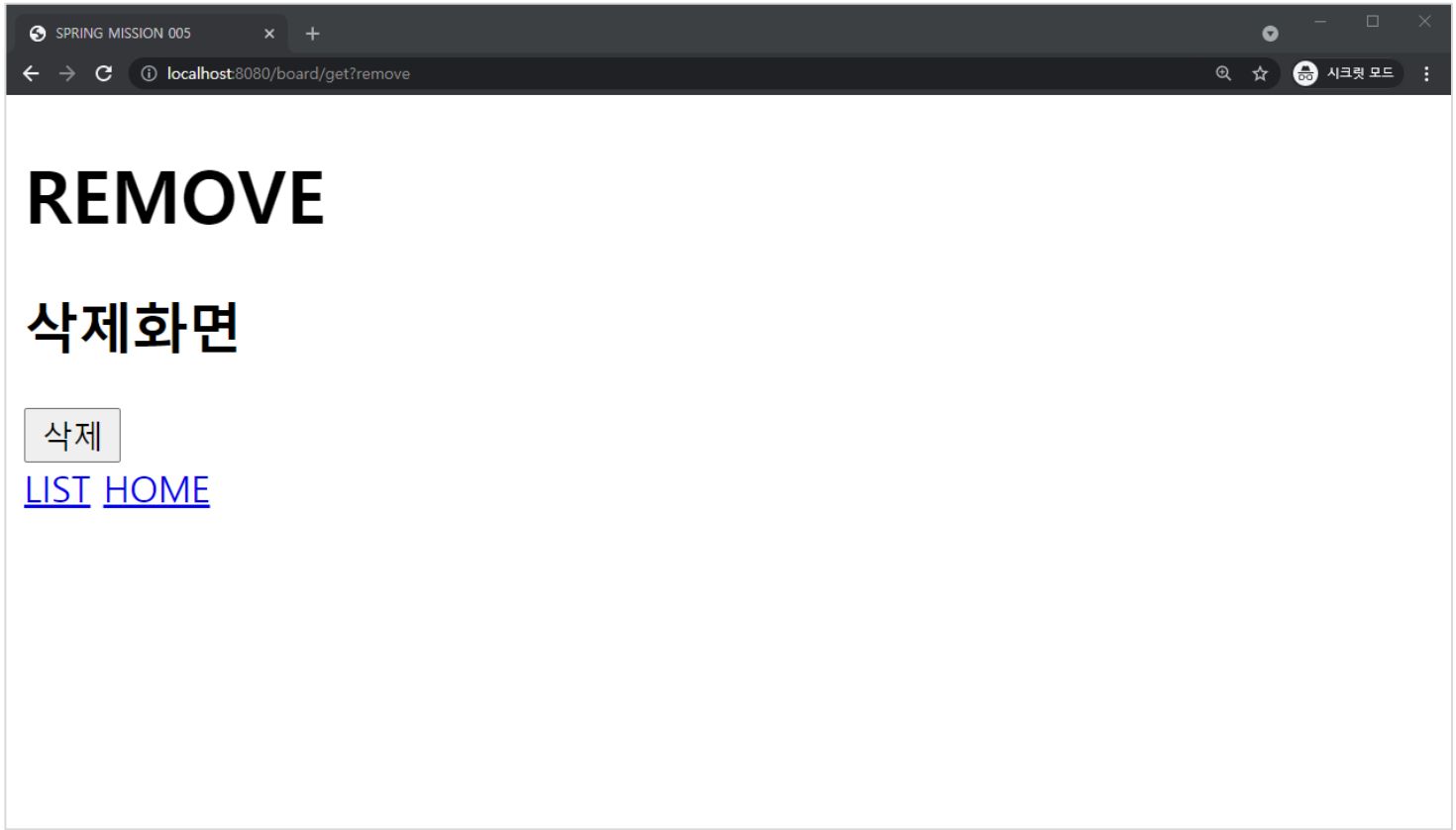
file

~/views/board/read.jsp

5

Controller 를 생성하고, 요청경로를 매핑하시오. 그리고 뷰 파일도 작성하시오.

localhost:8080/board/remove



GUIDE

1. <h1>, <h2> 태그를 이용하여 제목을 작성하시오.

2. <form>, <button> 태그를 이용하여 등록버튼 클릭 시,
url: /board/get?remove
로 POST 방식으로 요청을 보내는 제출 양식을 작성하시오.

3. <a> 태그를 이용하여, 메인화면으로 이동하는 링크를 작성하시오.

file

~/views/board/remove.jsp

6

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/board/{boardNo}

GUIDE

1. GET, PUT 방식으로 매핑하시오.

2. `"X-HTTP-Method-Override" : "PUT"`
   위와 같이 커스텀 헤더가 매핑될 수 있도록 작성하시오.

3. 요청을 확인하기 위한 로그를 작성하시오.

4. 200 OK 응답을 하도록 코드를 작성하시오.

5. ~/views/home.jsp 에 해당 경로로 요청하는 form 양식과 script 를 작성하시오.

file

BoardController.java

7

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/board/{boardNo}

GUIDE

1. 의존 라이브러리를 추가하시오.
    1. jackson-databind
    2. jackson-dataformat

2. 메소드 1
    1. POST 방식으로 매핑하시오.
    2. 요청 확인을 위한 로그를 작성하시오.
    3. 200 OK 응답을 하도록 코드를 작성하시오.

3. 메소드 2
    1. PUT 방식으로 매핑하시오.
    2. contentType : "application/json; charset=utf-8"
       위와 같이 헤더 값을 지정하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. 200 OK 응답을 하도록 코드를 작성하시오.

file

BoardController.java

7

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/board/{boardNo}

GUIDE

4. 메소드 3
    1. PUT 방식으로 매핑하시오.
    2. contentType : "application/xml; charset=utf-8"
       위와 같이 헤더 값을 지정하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. 200 OK 응답을 하도록 코드를 작성하시오.

5. ~/views/home.jsp 에 해당 경로로 요청하는 form 양식과 script 를 작성하시오.

file

BoardController.java

8

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

Accept 매핑

localhost:8080/board/{boardNo}

GUIDE

1. 의존 라이브러리를 추가하시오.
    1. jackson-databind
    2. jackson-dataformat

2. 메소드 1
    1. GET 방식으로 매핑하시오.
    2. 요청 확인을 위한 로그를 작성하시오.
    3. 200 OK 응답을 하도록 코드를 작성하시오.

3. 메소드 2
    1. GET 방식으로 매핑하시오.
    2. "Accept" : "application/json"
       위와 같이 헤더 값을 지정하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. 200 OK 응답을 하도록 코드를 작성하시오.

file

BoardController.java

8

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

Accept 매핑

localhost:8080/board/{boardNo}

GUIDE

4. 메소드 3
    1. GET 방식으로 매핑하시오.
    2. "Accept" : "application/json"
       위와 같이 헤더 값을 지정하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. 200 OK 응답을 하도록 코드를 작성하시오.

5. ~/views/home.jsp 에 해당 경로로 요청하는 form 양식과 script 를 작성하시오.

file

BoardController.java

9

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

void 타입

localhost:8080/test0101

GUIDE

1. 메소드 1
    1. 리턴 타입을 void 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.

1. View
    1. ~/views/test0101.jsp 파일을 생성하시오.
    2. <title> : test0101
       <h1> : test0101
       위와 같이 확인할 수 있는 태그를 작성하시오.

file

HomeController.java

9

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

void 타입

localhost:8080/sub/test0101

GUIDE

1. 메소드 1
    1. 리턴 타입을 void 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.

1. View
    1. ~/views/sub/test0101.jsp 파일을 생성하시오.
    2. <title> : sub/test0101
       <h1> : sub/test0101
       위와 같이 확인할 수 있는 태그를 작성하시오.

file

HomeController.java

10

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

String 타입

localhost:8080/test0201

GUIDE

1. 메소드 1
    1. 리턴 타입을 string 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. ~/views/test0201.jsp 뷰가 응답되도록 리턴하시오.

1. View
    1. ~/views/test0201.jsp 파일을 생성하시오.
    2. <title> : test0201
       <h1> : test0201
       위와 같이 확인할 수 있는 태그를 작성하시오.

file

HomeController.java

10

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

String 타입

localhost:8080/test0202

GUIDE

1. 메소드 2
    1. 리턴 타입을 string 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. ~/views/test0202.jsp 뷰가 응답되도록 리턴하시오.

1. View
    1. ~/views/test0202.jsp 파일을 생성하시오.
    2. <title> : test0202
       <h1> : test0202
       위와 같이 확인할 수 있는 태그를 작성하시오.

file

HomeController.java

10

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

String 타입

localhost:8080/sub/test0203

GUIDE

1. 메소드 3
    1. 리턴 타입을 string 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. ~/views/sub/test0203.jsp 뷰가 응답되도록 리턴하시오.

1. View
    1. ~/views/sub/test0203.jsp 파일을 생성하시오.
    2. <title> : sub/test0203
       <h1> : sub/test0203
       위와 같이 확인할 수 있는 태그를 작성하시오.

file

HomeController.java

10

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

String 타입

localhost:8080/sub/test0204

GUIDE

1. 메소드 4
    1. 리턴 타입을 string 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. redirect 방식으로
       ~/views/sub/test0205.jsp 뷰가 응답되도록 리턴하시오.

2. View
    1. ~/views/sub/test0205.jsp 파일을 생성하시오.
    2. <title> : sub/test0205
       <h1> : sub/test0205
       위와 같이 확인할 수 있는 태그를 작성하시오.

file

HomeController.java

10

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

자바빈즈 타입

localhost:8080/sub/test0205

GUIDE

1. 메소드 5
    1. 리턴 타입을 string 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. ~/views/sub/test0205.jsp 뷰가 응답되도록 리턴하시오.

2. View
    1. ~/views/sub/test0205.jsp 파일을 생성하시오.
    2. <title> : sub/test0205
       <h1> : sub/test0205
       위와 같이 확인할 수 있는 태그를 작성하시오.

file

HomeController.java

11

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/test0301

1. 의존 라이브러리를 추가하시오.
    1. jackson-databind
    2. jackson-dataformat

2. 자바 클래스를 생성하시오.
    1. {top-level-package}/domain 패키지를 생성하시오.
    2. {top-level-package}/domain/Member.java 클래스를 생성하시오.
    3. userId (String), password (String) 필드를 정의하시오.
    4. Getter & Setter 메소드를 정의하시오.

3. 메소드 1
    1. 리턴 타입을 Member 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. Member 객체를 생성하시오. (userId : “test”, password : “123456”)
    5. Member 객체 데이터가 JSON 타입으로 응답되도록 리턴하시오.

GUIDE

@ResponseBody

- 서버에서 클라이언트로 응답 데이터를 전송하기 위해서 사용하는 어노테이션
- 자바 객체를 HTTP 응답 본문의 객체로 변환한다.

file

HomeController.java

11

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/test0302

1. 의존 라이브러리를 추가하시오.
    1. jackson-databind
    2. jackson-dataformat

2. 자바 클래스를 생성하시오. (Member)

3. 메소드 2
    1. 리턴 타입을 Member 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. Member 객체를 생성하시오. (userId : "test", password : "123456")
    5. Member 객체 데이터가 XML 타입으로 응답되도록 리턴하시오.

GUIDE

@ResponseBody

- 서버에서 클라이언트로 응답 데이터를 전송하기 위해서 사용하는 어노테이션
- 자바 객체를 HTTP 응답 본문의 객체로 변환한다.

file

HomeController.java

12

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/test04

1. 의존 라이브러리를 추가하시오. (미션 11)
2. 자바 클래스를 생성하시오. (Member)

3. 메소드 1
    1. 리턴 타입을 List<Member> 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. Member 객체를 생성하시오.
       member 1 - (userId : "test1", password : "123456")
       member 2 -(userId : "test2", password : "123456")
       member 3 -(userId : "test3", password : "123456")
    5. List<Member> 를 생성하여, member 1~3 을 추가하시오.
    6. List<Member> 데이터가 응답되도록 리턴하시오.

file

HomeController.java

GUIDE

@ResponseBody

- 서버에서 클라이언트로 응답 데이터를 전송하기 위해서 사용하는 어노테이션
- 자바 객체를 HTTP 응답 본문의 객체로 변환한다.

13

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/test05

1. 의존 라이브러리를 추가하시오. (미션 11)
2. 자바 클래스를 생성하시오. (Member)

3. 메소드 1
    1. 리턴 타입을 Map<String, Member> 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. Member 객체를 생성하시오.
       member 1 - (userId : "test1", password : "123456")
       member 2 -(userId : "test2", password : "123456")
       member 3 -(userId : "test3", password : "123456")
    5. Map<String, Member> 를 생성하여, member 1~3 을 추가하시오.
    6. Map<String, Member> 데이터가 응답되도록 리턴하시오.
       "key1" : member1
       "key2" : member2
       "key3" : member3

GUIDE

@ResponseBody

- 서버에서 클라이언트로 응답 데이터를 전송하기 위해서 사용하는 어노테이션
- 자바 객체를 HTTP 응답 본문의 객체로 변환한다.

file

HomeController.java

14

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

ResponsEntity<Void>

localhost:8080/test06

1. 의존 라이브러리를 추가하시오. (미션 11)
2. 자바 클래스를 생성하시오. (Member)

3. 메소드 1
    1. 리턴 타입을 ResponseEntity<Void> 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. Http 상태코드 "200 OK" 가 응답되도록 리턴하시오.

GUIDE

@ResponseBody

- 서버에서 클라이언트로 응답 데이터를 전송하기 위해서 사용하는 어노테이션
- 자바 객체를 HTTP 응답 본문의 객체로 변환한다.

file

HomeController.java

15

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

ResponsEntity <String>

localhost:8080/test07

1. 의존 라이브러리를 추가하시오. (미션 11)
2. 자바 클래스를 생성하시오. (Member)

3. 메소드 1
    1. 리턴 타입을 ResponseEntity<String> 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. "SUCCESS" 메시지와 Http 상태코드 "200 OK" 가 응답되도록 리턴하시오.

GUIDE

@ResponseBody

- 서버에서 클라이언트로 응답 데이터를 전송하기 위해서 사용하는 어노테이션
- 자바 객체를 HTTP 응답 본문의 객체로 변환한다.

file

HomeController.java

16

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

ResponsEntity
<자바빈즈>

localhost:8080/test08

1. 의존 라이브러리를 추가하시오. (미션 11)
2. 자바 클래스를 생성하시오. (Member)

3. 메소드 1
    1. 리턴 타입을 ResponseEntity<Member> 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. Member 객체를 생성하시오.
       member - (userId : "test1", password : "123456")
    5. member 객체의 데이터와 Http 상태코드 "200 OK" 가 응답되도록 리턴하시오.

GUIDE

@ResponseBody

- 서버에서 클라이언트로 응답 데이터를 전송하기 위해서 사용하는 어노테이션
- 자바 객체를 HTTP 응답 본문의 객체로 변환한다.

file

HomeController.java

17

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

ResponsEntity <List>

localhost:8080/test09

1. 의존 라이브러리를 추가하시오. (미션 11)
2. 자바 클래스를 생성하시오. (Member)

3. 메소드 1
    1. 리턴 타입을 ResponseEntity<List<Member>> 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. Member 객체를 생성하시오.
       member 1 - (userId : "test1", password : "123456")
       member 2 -(userId : "test2", password : "123456")
       member 3 -(userId : "test3", password : "123456")
    5. List<Member> 를 생성하여, member 1~3 을 추가하시오.
    6. List<Member> 데이터와 Http 상태코드 "200 OK" 가 응답되도록 리턴하시오.

GUIDE

@ResponseBody

- 서버에서 클라이언트로 응답 데이터를 전송하기 위해서 사용하는 어노테이션
- 자바 객체를 HTTP 응답 본문의 객체로 변환한다.

file

HomeController.java

18

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

ResponsEntity <Map>

localhost:8080/test10

1. 의존 라이브러리를 추가하시오. (미션 11)
2. 자바 클래스를 생성하시오. (Member)

3. 메소드 1
   1. 리턴 타입을 ResponseEntity<Map<String, Member>> 타입으로 정의하시오.
   2. GET 방식으로 매핑하시오.
   3. 요청 확인을 위한 로그를 작성하시오.
   4. Member 객체를 생성하시오.
      member 1 - (userId : "test1", password : "123456")
      member 2 -(userId : "test2", password : "123456")
      member 3 -(userId : "test3", password : "123456")
   5. Map<String, Member> 를 생성하여, member 1~3 을 추가하시오.
   6. Map<String, Member> 데이터와 Http 상태코드 "200 OK" 가 응답되도록 리턴하시오.

GUIDE

@ResponseBody

- 서버에서 클라이언트로 응답 데이터를 전송하기 위해서 사용하는 어노테이션
- 자바 객체를 HTTP 응답 본문의 객체로 변환한다.

file

HomeController.java

19

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

ResponsEntity <byte>

localhost:8080/test1101

1. 의존 라이브러리를 추가하시오.
    1. commons-io
2. 메소드
    1. 리턴 타입을 ResponseEntity<byte[]> 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. 이미지 파일을 입력받아 바이트 배열로 파일 데이터를 전송하시오.
        1. 파일 경로 : C:₩upload₩test.png
        2. 파일 경로의 이미지 파일을 읽어온다.
        3. HttpHeaders 객체에
            1. ContentType 을 IMAGE_PNG 타입으로 지정한다.
        4. ResponseEntity 에 파일 데이터를 바이트 배열로 지정한다.
    5. ResponseEntity<byte[]> 에 파일 데이터와 Http 상태코드 "201 CREATED" 가 응답되도록 리턴하시오.
    6. 예외 발생 시,
       ResponseEntity<byte[]> Http 상태코드 "400 BAD_REQUEST" 가 응답되도록 리턴하시오.

GUIDE

@ResponseBody

- 서버에서 클라이언트로 응답 데이터를 전송하기 위해서 사용하는 어노테이션
- 자바 객체를 HTTP 응답 본문의 객체로 변환한다.

file

HomeController.java

19

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

ResponsEntity <byte>

localhost:8080/test1102

1. 의존 라이브러리를 추가하시오.
    1. commons-io
2. 메소드 2
    1. 리턴 타입을 ResponseEntity<byte[]> 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. 압축 파일을 입력받아 바이트 배열로 파일 데이터를 전송하시오.
        1. 파일 경로 : C:₩upload₩test.zip (테스트할 이미지 파일을 준비해둔다.)
        2. 파일 경로의 압축 파일을 읽어온다.
        3. HttpHeaders 객체 지정
            1. ContentType 을 APPLICATION_OCTET_STREAM 타입으로 지정한다.
            2. Content-Disposition 헤더를 추가한다. ("attachment; filename="{filename}" ")
        4. ResponseEntity 에 파일 데이터를 바이트 배열로 지정한다.
    5. ResponseEntity<byte[]> 에 파일 데이터와 Http 상태코드 "201 CREATED" 가 응답되도록 리턴하시오.
    6. 예외 발생 시,
       ResponseEntity<byte[]> Http 상태코드 "400 BAD_REQUEST" 가 응답되도록 리턴하시오.

GUIDE

@ResponseBody

- 서버에서 클라이언트로 응답 데이터를 전송하기 위해서 사용하는 어노테이션
- 자바 객체를 HTTP 응답 본문의 객체로 변환한다.

file

HomeController.java

20

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/registerForm

GUIDE

1. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. ~/views/registerForm.jsp 뷰가 응답되도록 리턴하시오.

2. View
    1. ~/views/registerForm.jsp 파일을 생성하시오.
    2. <title> : registerForm
       <h1> : registerForm
       위와 같이 확인할 수 있는 태그를 작성하시오.

file

~/controller/MemberController.java

20

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register

GUIDE

1. 메소드 2
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) userId
        2. (String) password
    4. 요청 확인을 위한 로그를 작성하시오.
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.
    2. <title> : success
       <h1> : success
       위와 같이 확인할 수 있는 태그를 작성하시오.

3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <a> 태그를 작성하시오.
        1. url : /register?userId=test&password=123456

file

~/controller/MemberController.java

20

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register/{userId}

1. 메소드 3
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) userId
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "userId = " + userId
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <a> 태그를 작성하시오.
        1. url : /register/test

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

20

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register01

GUIDE

1. 메소드 4
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) userId
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "userId = " + userId
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register01 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 3 : 전송 (submit)

file

~/controller/MemberController.java

20

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register02

GUIDE

1. 메소드 5
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) userId
        2. (String) password
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "userId = " + userId
        2. "password = " + password
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register02 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 3 : 전송 (submit)

file

~/controller/MemberController.java

20

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register03

GUIDE

1. 메소드 6
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) password
        2. (String) userId
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "userId = " + userId
        2. "password = " + password
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register03 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 3 : 전송 (submit)

file

~/controller/MemberController.java

20

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register04

GUIDE

1. 메소드 7
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) userId
        2. (String) password
        3. (String) coin
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "userId = " + userId
        2. "password = " + password
        3. "coin = " + coin
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.
2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.
3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register04 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 2 : coin (text)
        5. 필드 3 : 전송 (submit)

file

~/controller/MemberController.java

20

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register05

GUIDE

1. 메소드 7
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) userId
        2. (String) password
        3. (int) coin
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "userId = " + userId
        2. "password = " + password
        3. "coin = " + coin
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.
2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.
3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register04 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 2 : coin (text)
        5. 필드 3 : 전송 (submit)

file

~/controller/MemberController.java

21

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register/{userId}

GUIDE

1. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) userId
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "userId = " + userId
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.
    2. <title> : success
       <h1> : success
       위와 같이 확인할 수 있는 태그를 작성하시오.

3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <a> 태그를 작성하시오.
        1. url : /register/test

file

~/controller/MemberController.java

21

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register/{userId}/{coin}

1. 메소드 2
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) userId
        2. (String) coing
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "userId = " + userId
        2. "coint = " + coin
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.
2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.
3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <a> 태그를 작성하시오.
        1. url : /register/test/100

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

21

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register01

1. 메소드 3
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) userId
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "userId = " + userId
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register01 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 2 : coin (text)
        5. 필드 3 : 전송 (submit)

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

21

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register0201

GUIDE

1. 메소드 4
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) username
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "username = " +username
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register0201 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 2 : coin (text)
        5. 필드 3 : 전송 (submit)

file

~/controller/MemberController.java

21

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register0202

1. 메소드 5
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) username - @PathVariable("userId")
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "username = " +username
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register0201 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 2 : coin (text)
        5. 필드 3 : 전송 (submit)

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

21

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register0301

GUIDE

1. 메소드 6
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) username
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "username = " +username
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register0301 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 2 : coin (text)
        5. 필드 3 : 전송 (submit)

file

~/controller/MemberController.java

21

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register0302

1. 메소드 7
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오
        1. (String) memberId - @PathVariable("userId")
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "username = " +username
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

3. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register0301 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 2 : coin (text)
        5. 필드 3 : 전송 (submit)

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

22

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register01

1. 자바 클래스를 생성하시오.
    1. ~/domain 패키지를 생성하시오.
    2. ~/domain/Member.java 클래스를 생성하시오.
    3. 필드를 정의하시오.
        1. (String) userId
        2. (String) password
        3. (int) coin
    4. Getter & Setter 메소드를 정의하시오.
    5. toString() 메소드를 오버라이딩 하시오.

2. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (Member) member
    4. 요청 확인을 위한 로그를 작성하시오.
        1. member.toString();
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

22

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register01

1. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

2. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register01 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 2 : coin (text)
        5. 필드 3 : 전송 (submit)

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

22

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register02

1. 자바 클래스를 생성하시오.
    1. ~/domain 패키지를 생성하시오.
    2. ~/domain/Member.java 클래스를 생성하시오.
    3. 필드를 정의하시오.
        1. (String) userId
        2. (String) password
        3. (int) coin
    4. Getter & Setter 메소드를 정의하시오.
    5. toString() 메소드를 오버라이딩 하시오.

2. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (Member) member
        2. int coing

    4. 요청 확인을 위한 로그를 작성하시오.
        1. member.toString();

    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

22

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register02

1. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

2. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register02 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 2 : coin (text)
        5. 필드 3 : 전송 (submit)

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

22

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register03

1. 자바 클래스를 생성하시오. (Member)

2. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (int) uid
        2. (Member) member
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "uid = " + uid
        2. member.toString();
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

22

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register03

1. View 1
    1. ~/views/success.jsp 파일을 생성하시오.

2. View 2
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register03 , method="post"
        2. 필드 1 : uid (text)
        3. 필드 2 : userId (text)
        4. 필드 3 : password (password)
        5. 필드 4 : coin (text)
        6. 필드 5 : 전송 (submit)

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

23

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register01

1. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (String) userId
        2. (Date) dateOfBirth
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "userId = " + userId
        2. "dateOfBirth = " + dateOfBirth
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.
2. View 1
    1. ~/views/registerForm.jsp 파일을 생성하고 GET method 로 요청처리하시오.
    2. <a> 태그를 작성하시오.
        1. register01?userId=aloha&dateOfBirth=2022/10/24
3. View 2
    1. ~/views/success.jsp 파일을 생성하시오.

GUIDE

@PathVariable

- URL 경로 변수 값을 가져오기 위한 어노테이션

file

~/controller/MemberController.java

24

아래 조건에 만족하도록 코드를 작성하시오.

@DateTimeFormat 애너테이션

1. ~/domain 패키지를 생성하시오.

2. ~/domain/Member.java 클래스 파일을 생성하시오.
    1. 필드를 정의하시오.
        1. (String) userId
        2. (String) password
        3. (Date) dateOfBirth
            1. @DateTimeFormat(patter="yyMMdd")
        4. Getter & Setter 메소드를 정의하시오.
        5. toString() 메소드를 오버라이딩 하시오.

file

~/domain/Member.java

24

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

@DateTimeFormat 애너테이션

localhost:8080/register02

1. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (Member) member
    4. 요청 확인을 위한 로그를 작성하시오.
        1. member.toString();
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

2. View 1
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /register02 , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 3 : dateOfBirth (text)
        5. 필드 4 : 전송 (submit)

3. View 2
    1. ~/views/success.jsp 파일을 생성하시오.

file

~/controller/MemberController.java

GUIDE

@DateTimeFormat

- pattern 속성에 날짜형식을 지정하여 포맷에 맞춰 Date 를 전달 받게 해주는 애너테이션

25

아래 조건에 만족하도록 코드를 작성하시오.

1. ~/domain 패키지를 생성하시오.

2. ~/domain/Address.java 클래스 파일을 생성하시오.
    1. 필드를 정의하시오.
        1. (String) postcode
        2. (String) location
    2. Getter & Setter 메소드를 정의하시오.
    3. toString() 메소드를 오버라이딩 하시오.

file

~/domain/Address.java

25

아래 조건에 만족하도록 코드를 작성하시오.

1. ~/domain 패키지를 생성하시오.

2. ~/domain/Card.java 클래스 파일을 생성하시오.
    1. 필드를 정의하시오.
        1. (String) no
        2. (String) validMonth
            1. @DateTimeFormat(patter="yyMMdd")

    2. Getter & Setter 메소드를 정의하시오.

    3. toString() 메소드를 오버라이딩 하시오.

file

~/domain/Card.java

25

아래 조건에 만족하도록 코드를 작성하시오.

1. ~/domain 패키지를 생성하시오.

2. ~/domain/Member.java 클래스 파일을 생성하시오.
    1. 필드를 정의하시오.
        1. (String) userId
        2. (String) password
        3. (String) email
        4. (String) gender
        5. (String) hobby
        6. (String[]) hobbyArray
        7. (List<String>) hobbyList
        8. (boolean) foreigner
        9. (String) developer
        10. (String) nationality
        11. (Address) address
        12. (List<Card>) cardList
        13. (String) cars
        14. (String[]) carsArray
        15. (List<String>) carList
        16. (String) introduction
    2. Getter & Setter 메소드를 정의하시오.
    3. toString() 메소드를 오버라이딩 하시오.

file

~/domain/Member.java

25

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/registerAllForm

GUIDE

1. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
    4. 요청 확인을 위한 로그를 작성하시오.
    5. ~/views/registerAllForm.jsp 뷰가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

25

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/registerAllForm

GUIDE

1. View 1
    1. ~/views/registerAllForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 요청경로 : /registerUser , method="post"
        2. 필드 1 : userId (text)
        3. 필드 2 : password (password)
        4. 필드 3 : userName (text)
        5. 필드 4 : email (text)
        6. 필드 5 : dateOfBirth (text)
        7. 필드 6 : gender (radio) value : {male, female}
        8. 필드 7 : developer (checkbox) value : Y
        9. 필드 8 : foreigner (checkbox) value : false
        10. 필드 9 : nationality (select) option : {Korea, Japan, China}
        11. 필드 10 : cars (select) option : {volvo, audi, k5}
        12. 필드 11 : carArray (select) option : {volvo, audi, k5}
        13. 필드 12 : carList (select) option : {volvo, audi, k5}
        14. 필드 13 : hobby (checkbox) option : {Sports, Music, Movie}
        15. 필드 14 : hobbyArray (checkbox) option : {Sports, Music, Movie}
        16. 필드 15 : hobbyList (checkbox) option : {Sports, Music, Movie}

file

~/views/registerAllForm.jsp

25

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

1. View 1
    1. ~/views/registerAllForm.jsp 파일에 코드를 추가하시오.
    2. <form> 태그를 작성하시오.
        1. 필드 16 : address.postCode (text)
        2. 필드 17 : address.location (text)
        3. 필드 18 : cardList[0].no (text)
        4. 필드 19 : cardList[0].validMonth (text)
        5. 필드 20 : cardList[1].no (text)
        6. 필드 21 : cardList[1].validMonth (text)
        7. 필드 22 : introduction (textarea)
        8. 필드 23 : 전송 (submit)
        9. 필드 24 : 리셋 (reset)

file

~/view/registerForm.jsp

25

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/registerUser

GUIDE

1. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (Member)member
    4. 요청 확인을 위한 로그를 작성하시오.
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

26

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

파일업로드
폼 방식 요청

1. View
    1. ~/views/registerAllForm.jsp 파일에 코드를 추가하시오.

    2. <form> 태그를 작성하시오.
        1. 요청경로 : /registerFile01 , method="post" enctype="multipart/form-data"
        2. 필드 1 : picture (file)
        3. 필드 2 : 전송 (submit)

~/view/registerForm.jsp

26

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

파일업로드
폼 방식 요청

localhost:8080/registerForm

GUIDE

1. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. ~/views/registerForm.jsp 뷰가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

26

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

파일업로드
폼 방식 요청

localhost:8080/registerFile01

GUIDE

1. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (MultipartFile) picture
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "originalName : " + picture.getOriginalFilename()
    5. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

27

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

1. View 1
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.

    2. <button> 태그를 작성하시오.
        1. id="registerBtn01" : registerBtn01
        2. 이벤트 : click
            1. Ajax 비동기 요청 : [GET], /register/test

    3. <button> 태그를 작성하시오.
        1. id="registerBtn02" : registerBtn02
        2. 이벤트 : click
            1. Ajax 비동기 요청
                - [POST], /register02
                - data : { userId : "test", password : "123456" }

file

~/view/registerForm.jsp

27

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register01/{userId}

GUIDE

1. 메소드 1
    1. 리턴 타입을 ResponseEntity<String> 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. ResponseEntity<String> 객체를 생성하시오.
        1. entity : "SUCCESS", 200 OK
    5. entity 객체가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

27

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register02

GUIDE

1. 메소드 2
    1. 리턴 타입을 ResponseEntity<String> 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 매개변수를 정의하시오.
    4. 요청 확인을 위한 로그를 작성하시오.
    5. ResponseEntity<String> 객체를 생성하시오.
        1. entity : "SUCCESS", 200 OK
    6. entity 객체가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

28

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

1. View
    1. ~/views/registerForm.jsp 파일에 코드를 추가하시오.

    2. <input> 태그를 작성하시오.
        1. file (file), id="inputFile"
        2. 이벤트 : change
            1. 파일을 읽어 Ajax 요청을 보낸다.
            2. [POST], /uploadAjax

~/view/registerForm.jsp

28

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/uploadAjax

GUIDE

1. 메소드
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (MultipartFile) picture
    4. 요청 확인을 위한 로그를 작성하시오.
        1. "originalName : " + picture.getOriginalFilename()
    5. ResponseEntity<String> 객체를 생성하시오.
        1. entity : "UPLOAD SUCCESS" + originalFilename, 200 OK
    6. entity 객체가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

29

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

1. View
    1. ~/views/home.jsp 파일에 코드를 추가하시오.

    2. <h1> 태그를 작성하시오.
        1. Hello Spring~!

    3. <p> 태그를 작성하시오.
        1. Server Time : ${serverTime}

~/view/home.jsp

29

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/

GUIDE

1. 메소드
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (Locale) locale,
        2. (Model) model
    4. 요청 확인을 위한 로그를 작성하시오.
    5. 현재 시스템의 시간을 생성하시오.
    6. 현재 시스템의 시간을 문자열로 변환하여, 모델의 속성에 등록하시오.
        1. "serverTime" : serverTime
    7. ~/views/home.jsp 뷰가 응답되도록 리턴하시오.

file

~/controller/HomeController.java

30

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

1. View
    1. ~/views/read01.jsp 파일을 생성하시오.
    2. <h3> 태그를 작성하시오.
        1. Result
    3. 모델의 속성을 출력하시오.
        1. userId : ${userId}
        2. password : ${password}
        3. userName : ${userName}
        4. email : ${email}

~/view/read01.jsp

30

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/read01

GUIDE

1. 메소드
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (Model) model
    4. 요청 확인을 위한 로그를 작성하시오.
    5. 모델의 속성을 등록하시오.
        1. userId : "test"
        2. Password : "123456"
        3. userName : "홍길동"
        4. Email : "test@naver.com"
    6. ~/views/read01.jsp 뷰가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

30

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

1. View
    1. ~/views/read01.jsp 파일을 생성하시오.
    2. <h3> 태그를 작성하시오.
        1. Result
    3. 모델의 속성을 출력하시오.
        1. userId : ${userId}
        2. password : ${password}
        3. userName : ${userName}
        4. email : ${email}

~/views/read01.jsp

30

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

1. Model
    1. Member 클래스 파일을 생성하시오.
    2. 필드를 정의하시오.
        1. (String) userId
        2. (String) password
        3. (String) username
        4. (String) email
    3. Getter & Setter 메소드를 정의하시오.
    4. toString() 메소드를 오버라이딩 하시오.

file

~/domain/Member.java

30

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

모델을 통한
데이터 전달

localhost:8080/read02

GUIDE

1. 메소드
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (Model) model
    4. 요청 확인을 위한 로그를 작성하시오.
    5. Member 객체를 생성하시오. (member)
        1. userId : "test"
        2. Password : "123456"
        3. userName : "홍길동"
        4. Email : "test@naver.com"
    6. 모델의 속성을 등록하시오.
        1. member
    7. ~/views/read02.jsp 뷰가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

31

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

@ModelAttribute
애너테이션

1. View 1
    1. ~/views/registerForm.jsp 파일을 생성하시오.
    2. <h1> 태그를 작성하시오.
        1. Register Form
    3. <form> 태그를 작성하시오.
        1. 요청경로 : /register, method="post"
        2. 필드1 : userId (text)
        3. 필드2 : password (text)
        4. 필드3 : 전송 (submit)
2. View 2
    1. ~/views/result.jsp 파일을 생성하시오.
    2. <h1> 태그를 작성하시오.
        1. Result
    3. 모델의 속성을 출력하시오.
        1. userId : ${userId}
        2. password : ${password}
        3. userName : ${userName}
        4. email : ${email}

file

~/view/registerForm.jsp

31

아래 경로와 조건에 부합하는 파일의 코드를 작성하시오.

모델을 통한
데이터 전달

1. Model
    1. Member 클래스 파일을 생성하시오.
    2. 필드를 정의하시오.
        1. (String) userId
        2. (String) password
        3. (String) userName
        4. (String) email
    3. Getter & Setter 메소드를 정의하시오.
    4. toString() 메소드를 오버라이딩 하시오.

file

~/domain/Member.java

31

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

@ModelAttribute 애너테이션

localhost:8080/registerForm

1. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. ~/views/registerForm.jsp 뷰가 응답되도록 리턴하시오.

GUIDE

@ModelAttribute

- 전달받은 매개변수를 Model 등록하여 전달하는 애너테이션

file

~/controller/MemberController.java

31

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

@ModelAttribute 애너테이션

localhost:8080/register

1. 메소드 2
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (String) userId - @ModelAttribute
        2. (String) password - @ModelAttribute
        3. (String) userName - @ModelAttribute
        4. (String) email - @ModelAttribute
    4. 요청 확인을 위한 로그를 작성하시오.
    5. ~/views/result.jsp 뷰가 응답되도록 리턴하시오.

GUIDE

@ModelAttribute

- 전달받은 매개변수를 Model 등록하여 전달하는 애너테이션

file

~/controller/MemberController.java

31

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

@ModelAttribute 애너테이션

localhost:8080/register

GUIDE

1. 메소드 3
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (Member) member
    1. 모델의 속성을 등록하시오.
        1. member
    2. 요청 확인을 위한 로그를 작성하시오.
    3. ~/views/result.jsp 뷰가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

32

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

RedirectAttribute 타입

1. View 1
    1. ~/views/registerForm.jsp 파일을 생성하시오.
    2. <h1> 태그를 작성하시오.
        1. Register Form
    3. <form> 태그를 작성하시오.
        1. 요청경로 : /register, method="post"
        2. 필드1 : userId (text)
        3. 필드2 : password (text)
        4. 필드3 : 전송 (submit)
2. View 2
    1. ~/views/result.jsp 파일을 생성하시오.
    2. <h1> 태그를 작성하시오.
        1. Result
    3. 모델의 속성을 출력하시오.
        1. Msg : ${msg}

file

~/view/registerForm.jsp

32

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

RedirectAttribute 타입

localhost:8080/registerForm

GUIDE

1. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. ~/views/registerForm.jsp 뷰가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

32

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

RedirectAttribute 타입

localhost:8080/result

GUIDE

1. 메소드 2
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. 모델에 속성을 등록하시오.
        1. "msg" : "success"
    5. ~/views/result.jsp 뷰가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

32

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

RedirectAttribute 타입

localhost:8080/register

GUIDE

1. 메소드 3
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 요청 확인을 위한 로그를 작성하시오.
    4. ~/views/result.jsp 뷰가 리다이렉트 되도록 리턴하시오.

file

~/controller/MemberController.java

33

아래 경로와 조건에 부합하는 파일의 코드를 작성하시오.

입력값 검증

1. 의존 라이브러리를 추가하시오.
    1. hibernate-validator

2. Model
    1. Member 클래스 파일을 생성하시오.
    2. 필드를 정의하시오.
        1. (String) userId - @NotBlank
        2. (String) password
        3. (String) username - @NotBlank, @Size(max=3)
        4. (String) email
        5. (String) birthDay
        6. (String) gender
    3. Getter & Setter 메소드를 정의하시오.
    4. toString() 메소드를 오버라이딩 하시오.

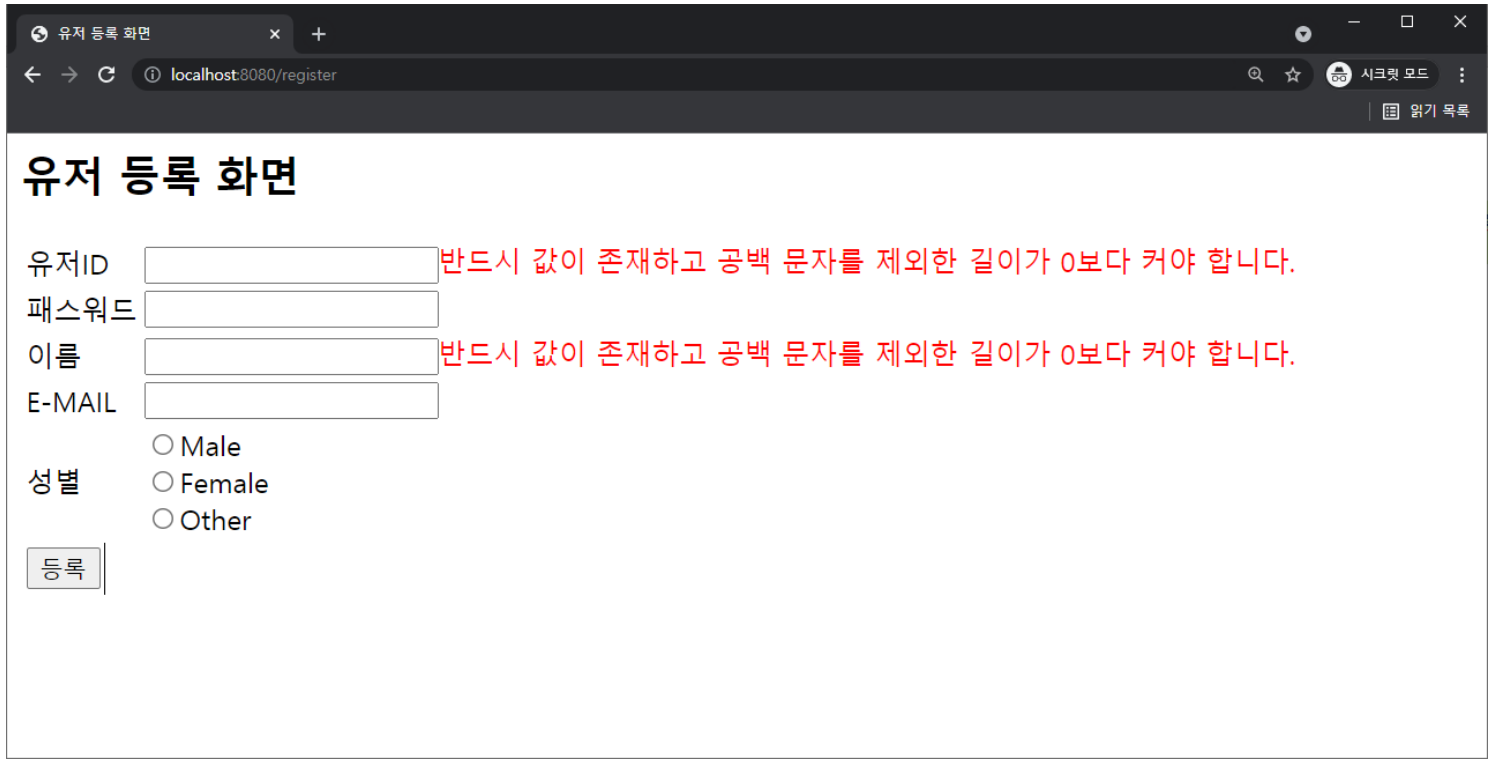
file

~/domain/Member.java

33

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

1. View 1
    1. ~/views/registerForm.jsp 파일을 생성하시오.



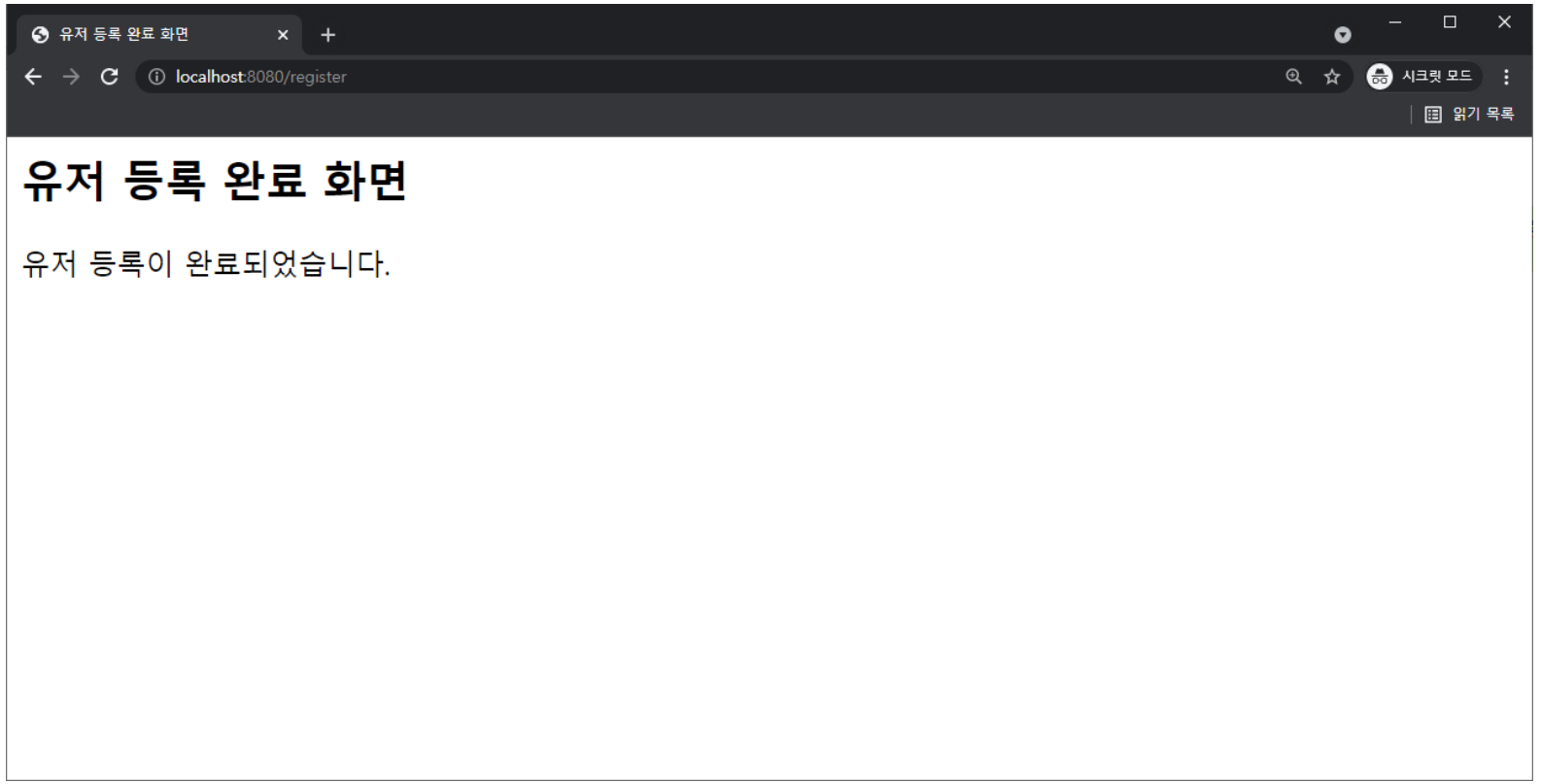
file

~/view/registerForm.jsp

33

아래 경로와 조건에 부합하는 뷰 파일의 코드를 작성하시오.

1. View 2
    1. ~/views/success.jsp 파일을 생성하시오.



file

~/view/success.jsp

33

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

입력값 검증

localhost:8080/registerForm01

GUIDE

1. 메소드 1
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. GET 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (Model) model
    4. 요청 확인을 위한 로그를 작성하시오.
    5. 모델에 속성을 등록하시오.
        1. "member" : new Member();
    6. ~/views/registerForm.jsp 뷰가 응답되도록 리턴하시오.

file

~/controller/MemberController.java

33

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

localhost:8080/register

1. 메소드 2
    1. 리턴 타입을 String 타입으로 정의하시오.
    2. POST 방식으로 매핑하시오.
    3. 파라미터를 정의하시오.
        1. (Member) member - @Validated
        2. (BindingResult) result
    4. 요청 확인을 위한 로그를 작성하시오.
    5. 유효성 검증 실패 시,
       ~/views/registerForm.jsp 뷰가 응답되도록 리턴하시오.
    6. ~/views/success.jsp 뷰가 응답되도록 리턴하시오.

GUIDE

@Validated

- 입력값 검증기능을 활성화 하는 애너테이션

file

~/controller/MemberController.java

34

아래 경로와 조건에 부합하는 요청경로를 매핑하는 코드를 작성하시오.

임력값 검증 결과

localhost:8080/register

※ [미션33] 의 /register 로 매핑한 메소드에 입력값 검증 결과를 확인하는 코드를 추가하시오.

1. 메소드 (미션 33)

2. 아래 BindingResult 의 메소드를 이용하여, 입력값 검증 결과를 확인한다.
    1. hasErrors()
    2. hasGlobalErrors()
    3. hasFieldErrors()
    4. hasFieldErrors(String)

GUIDE

@Validated

- 입력값 검증기능을 활성화 하는 애너테이션

BindingResult

- 요청 데이터의 바인딩 에러와 입력값 검증 에러 정보가 저장된 객체

file

~/controller/MemberController.java

35

아래 경로와 조건에 부합하는 파일의 코드를 작성하시오.

※ [미션33] 의 모델 Member 에 입력값 검증 규칙을 추가하시오.

입력값 검증 규칙

1. 입력값 검증 규칙
    1. @NotNull
    2. @NotBlank
    3. @Size
    4. @Email
    5. @Past
    6. @Future
    7. @DateTimeFormat

2. Member
    1. userId : null 이 아니고 길이가 0보다 큰지 검증
    2. password : null 이 아니고 길이가 0보다 큰지 검증
    3. userName : null 이 아니고 길이가 3보다 작거나 같은지 검증
    4. email : 이메일 주소 형식인지 검증
    5. dateOfBirth : 과거 날짜인지 검증

file

36

아래 조건에 만족하도록 코드를 작성하시오.

1. ~/domain 패키지를 생성하시오.

2. ~/domain/Address.java 클래스 파일을 생성하시오.
    1. 필드를 정의하시오.
        1. (String) postcode
        2. (String) location
    2. Getter & Setter 메소드를 정의하시오.
    3. toString() 메소드를 오버라이딩 하시오.

~/domain/Address.java

36

아래 조건에 만족하도록 코드를 작성하시오.

1. ~/domain 패키지를 생성하시오.

2. ~/domain/Card.java 클래스 파일을 생성하시오.
    1. 필드를 정의하시오.
        1. (String) no
        2. (String) validMonth
            1. @DateTimeFormat(patter="yyMMdd")

    2. Getter & Setter 메소드를 정의하시오.

    3. toString() 메소드를 오버라이딩 하시오.

~/domain/Card.java

36

아래 경로와 조건에 부합하는 파일의 코드를 작성하시오.

중첩된 자바빈즈 입력값 검증

※ [미션35] 의 모델 Member 에 입력값 검증 규칙을 추가하시오.

1. Member
    1. (Addrss) address : 중첩된 자바빈즈의 입력값 검증
    2. (List<Card>) cardList : 자바빈즈 컬렉션의 입력된 검증을 지정한다.

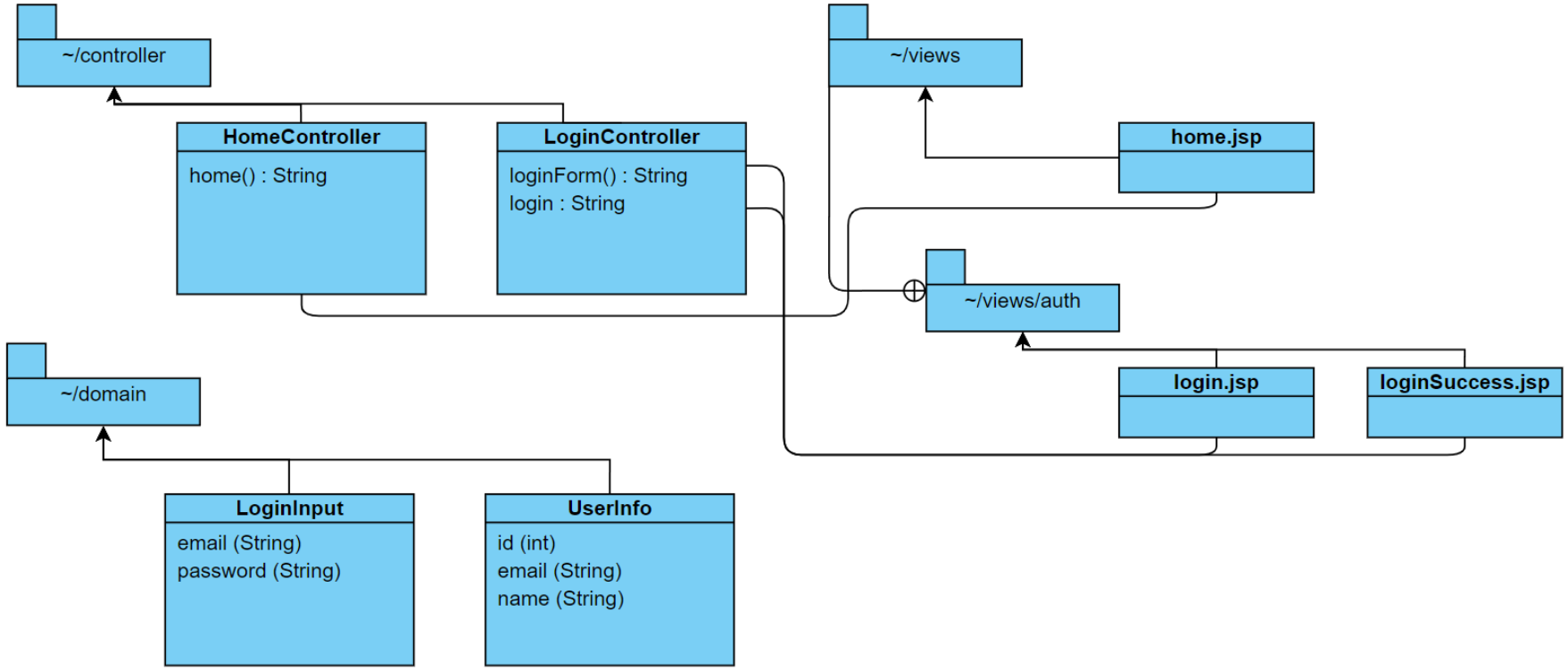
file

~/domain/Member.java

37

아래 다이어그램을 참고하여 폼 방식 로그인 처리를 수행하는 프로젝트를 완성하시오.

로그인 처리



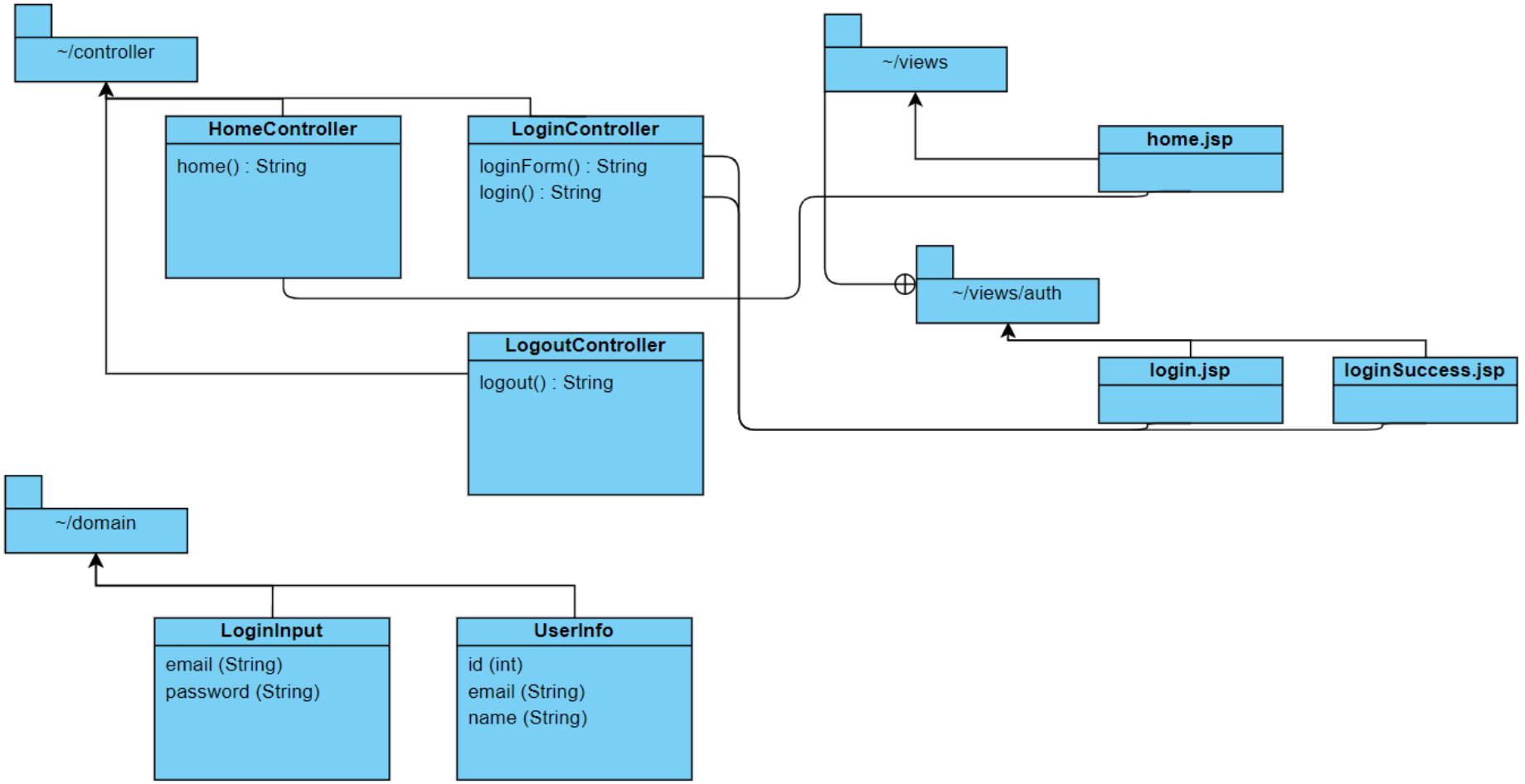
file

Project : spring-mvc-ms037

38

아래 다이어그램을 참고하여 폼 방식 로그아웃 처리를 수행하는 프로젝트를 완성하시오.

로그아웃 처리



file

Project : spring-mvc-ms038

39

아래에서 제시하는 조건을 참조하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.

쇼핑 카트 구현

1. 세션을 이용하여, 로그인 기능을 구현하시오.
    1. DB 와 연동하지는 않는다.
    2. 모든 임의의 email, password 에 대하여 인증을 허용한다.
    3. 세션에 임의의 사용자 명을 등록한다.

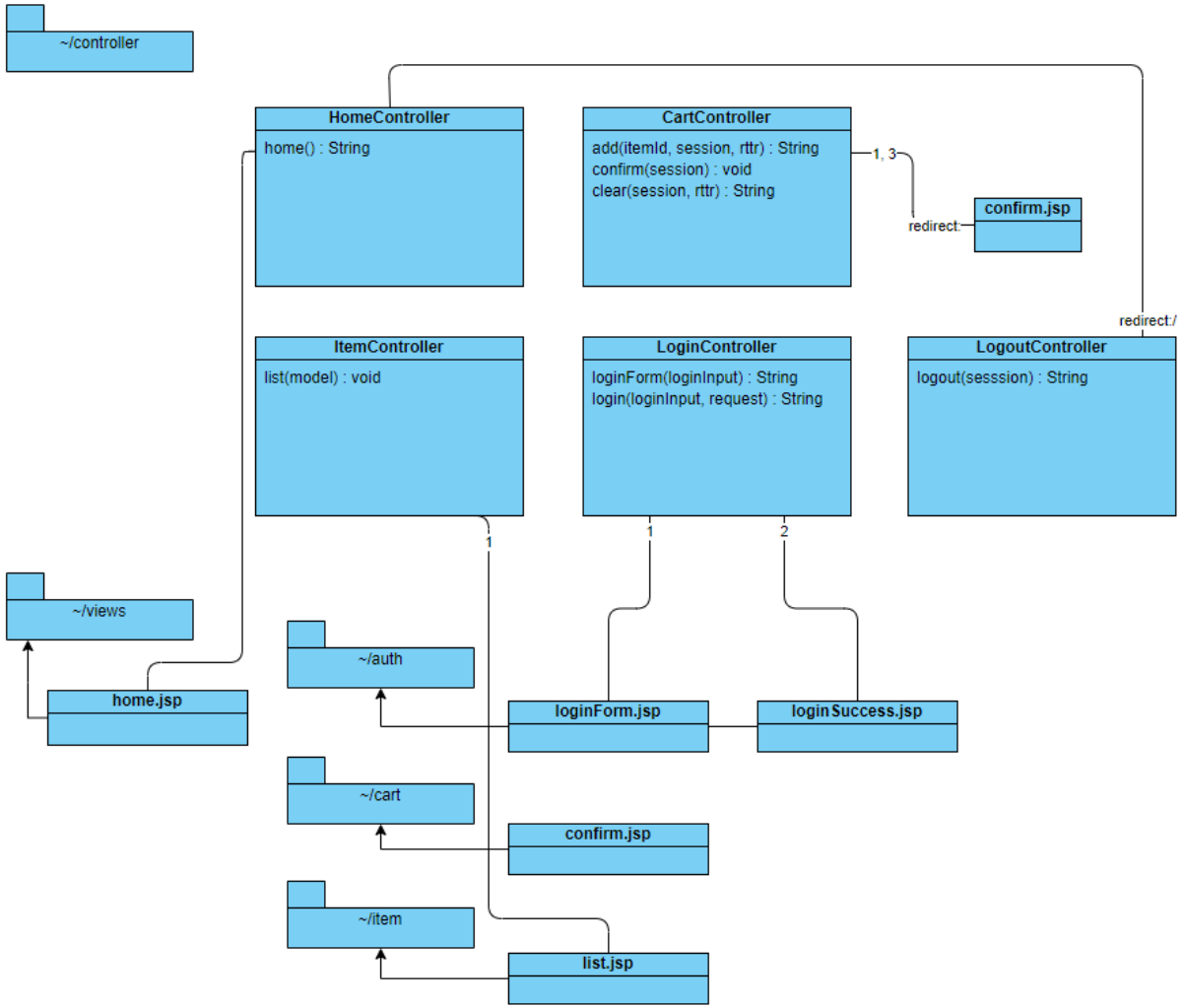
2. 세션을 이용하여, 쇼핑 카트 (장바구니) 기능을 구현하시오.
    1. DB 와 연동하지 않는다.
    2. 임의의 상품 목록을 조회한다.
    3. 상품 목록을 조회하고, 쇼핑카트에 추가하는 UI 를 구성한다.
    4. 쇼핑 카트에 담긴 상품목록을 세션에 등록한다.
    5. 쇼핑 카트에 담긴 상품목록을 조회하는 UI 를 구성한다.

file

Project : spring-mvc-ms039

39

아래 다이어그램 및 실행화면을 참고하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.

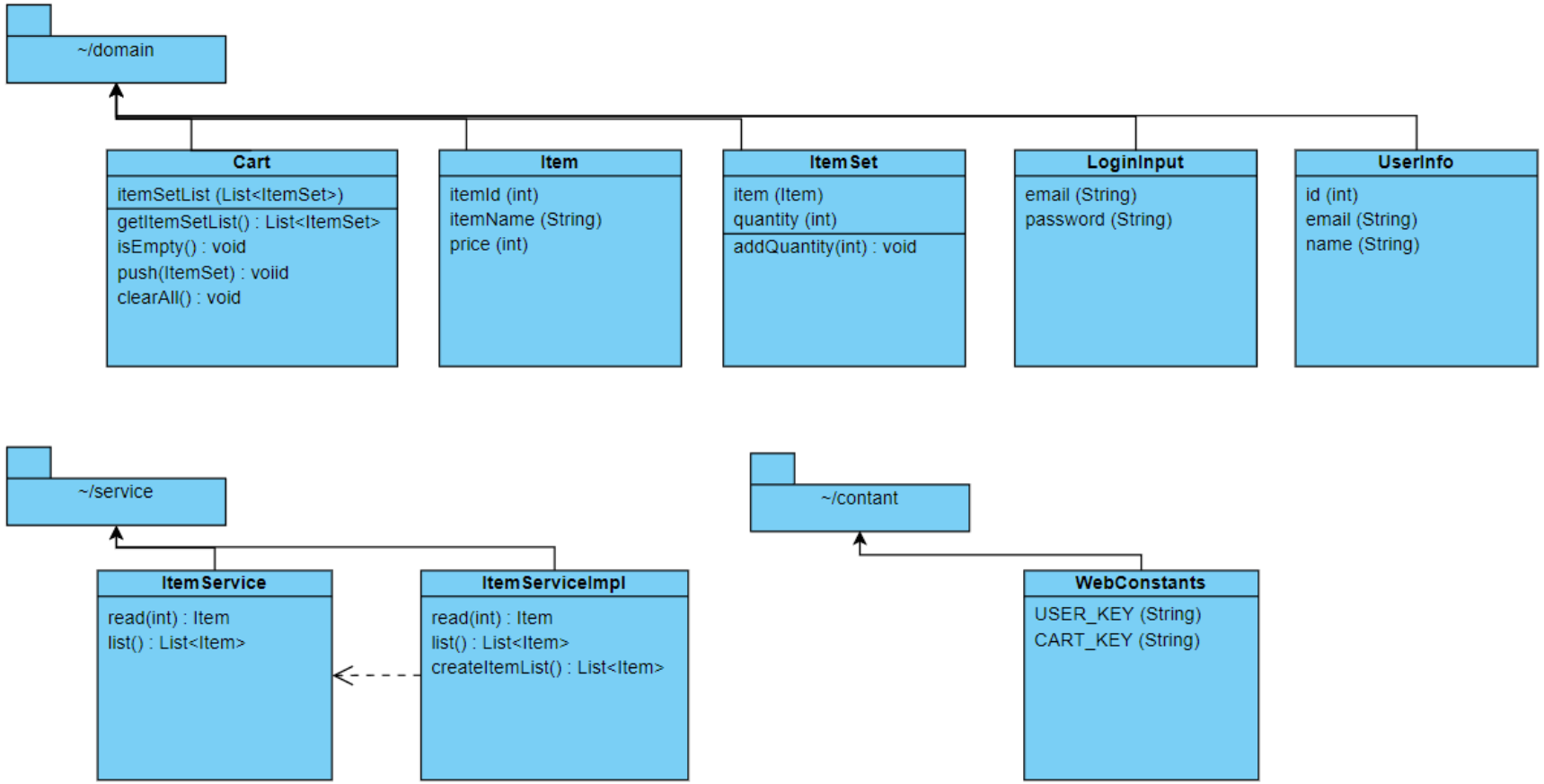


file

Project : spring-mvc-ms039

39

아래 다이어그램 및 실행화면을 참고하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.



file

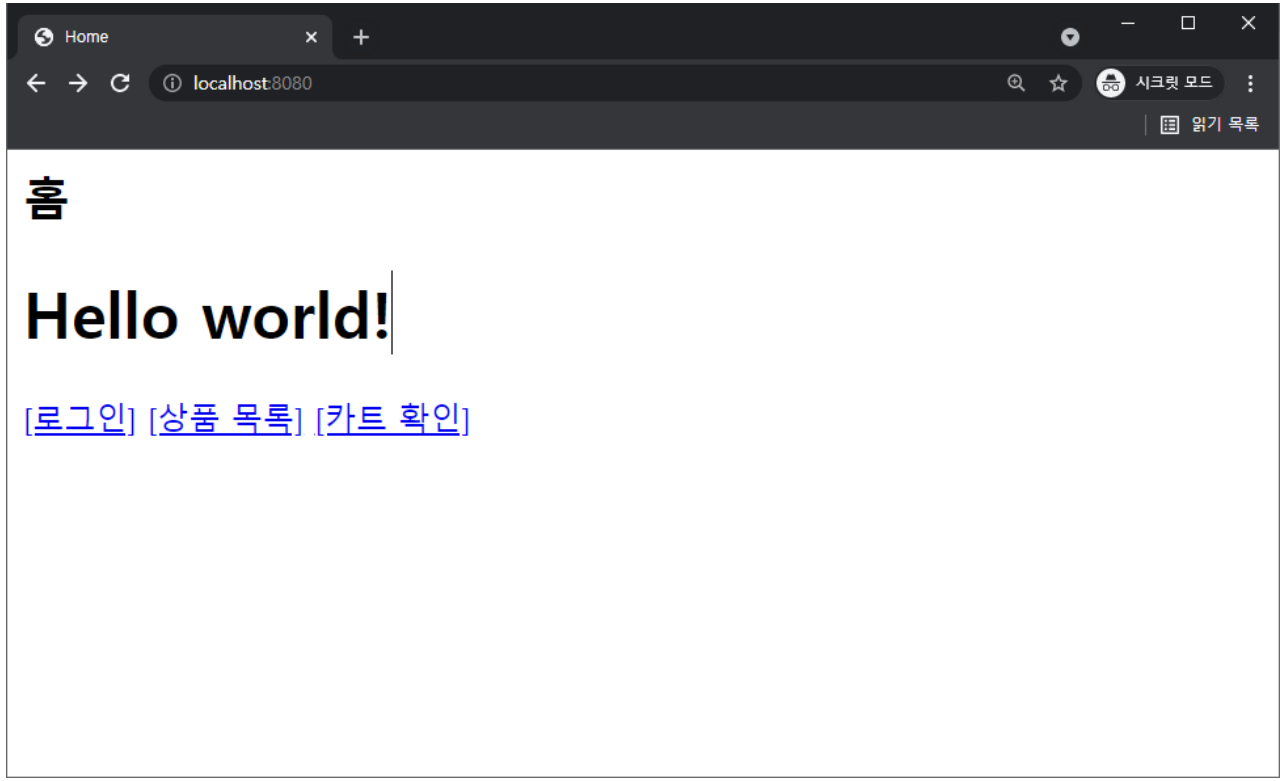
Project : spring-mvc-ms039

39

아래 다이어그램 및 실행화면을 참고하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.

localhost:8080/

GUIDE



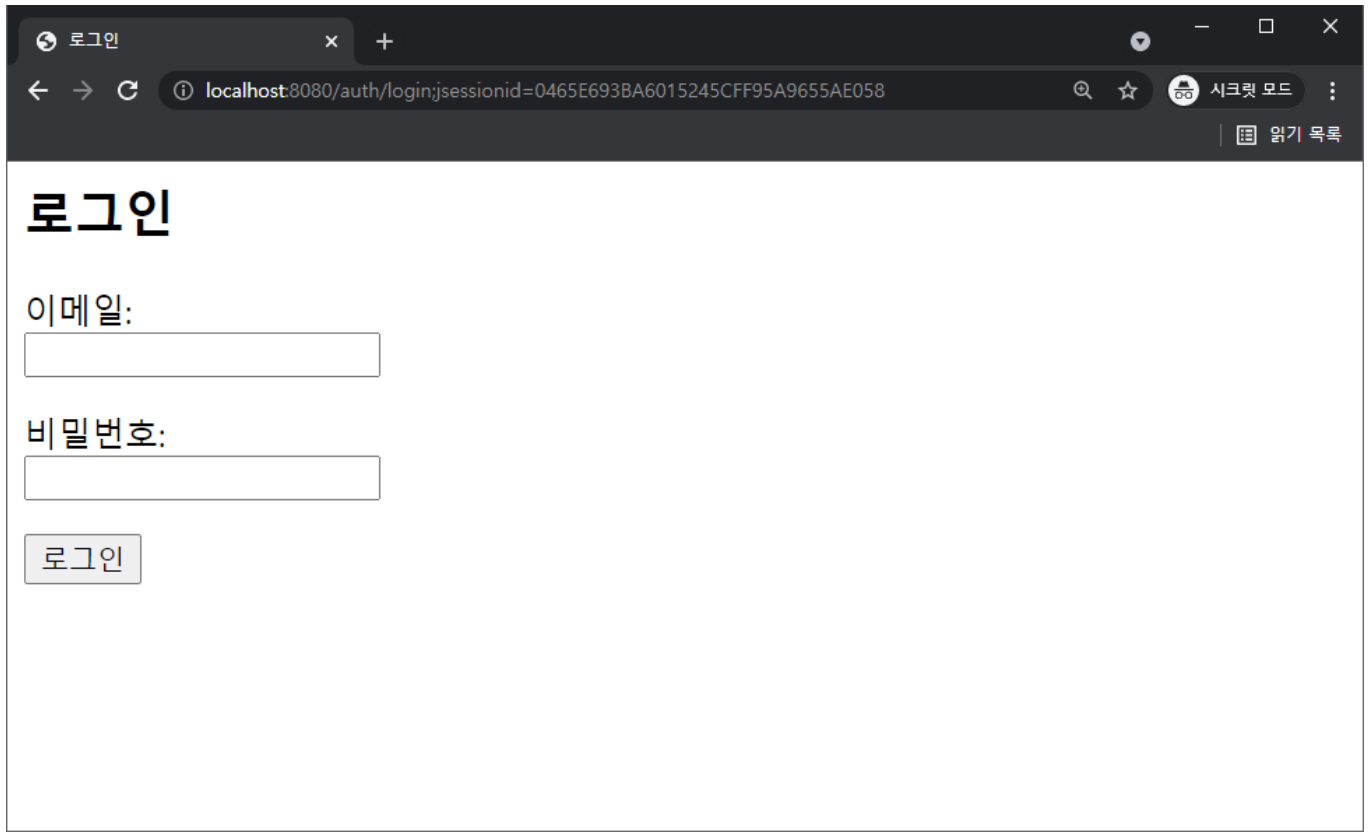
file

~/views/home.jsp

39

아래 다이어그램 및 실행화면을 참고하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.

localhost:8080/auth/login



GUIDE

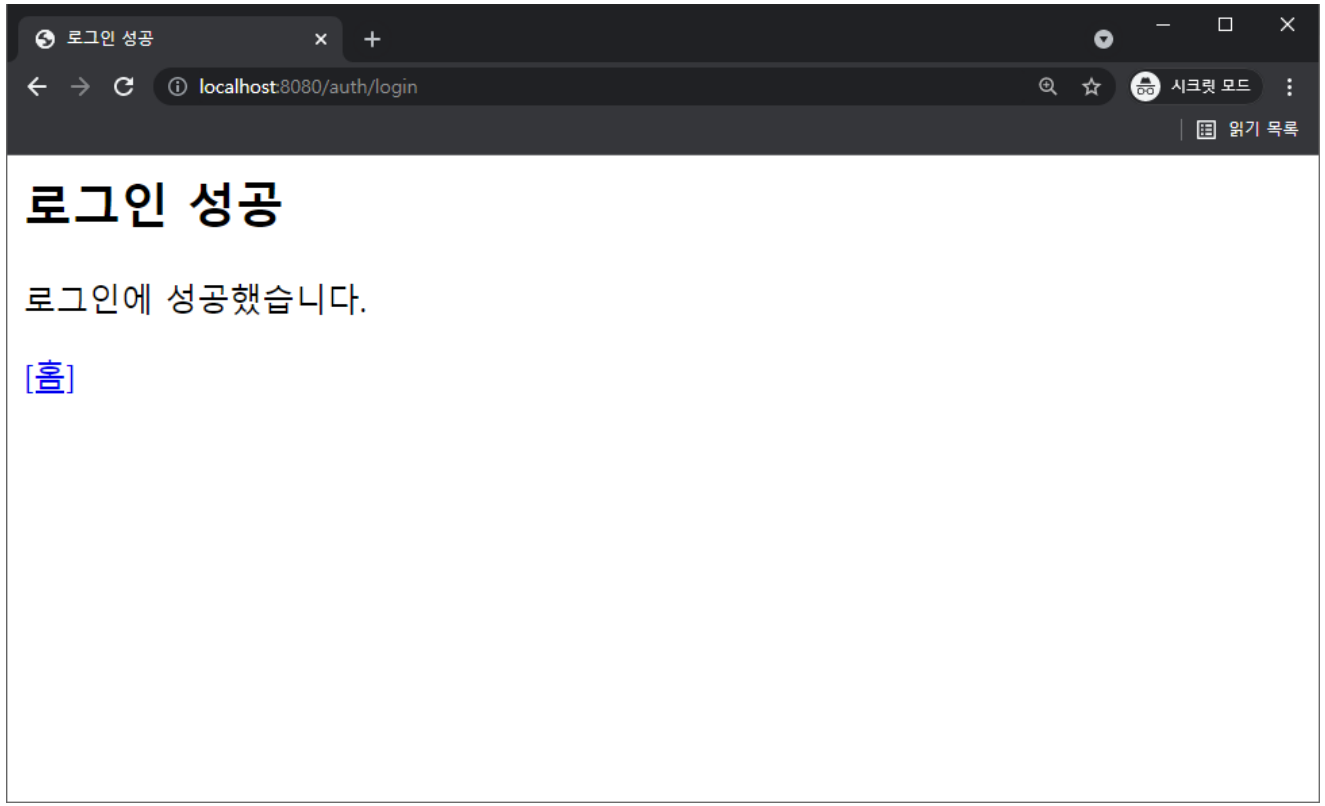
file

~/views/auth/loginForm.jsp

39

아래 다이어그램 및 실행화면을 참고하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.

localhost:8080/auth/login



GUIDE

file

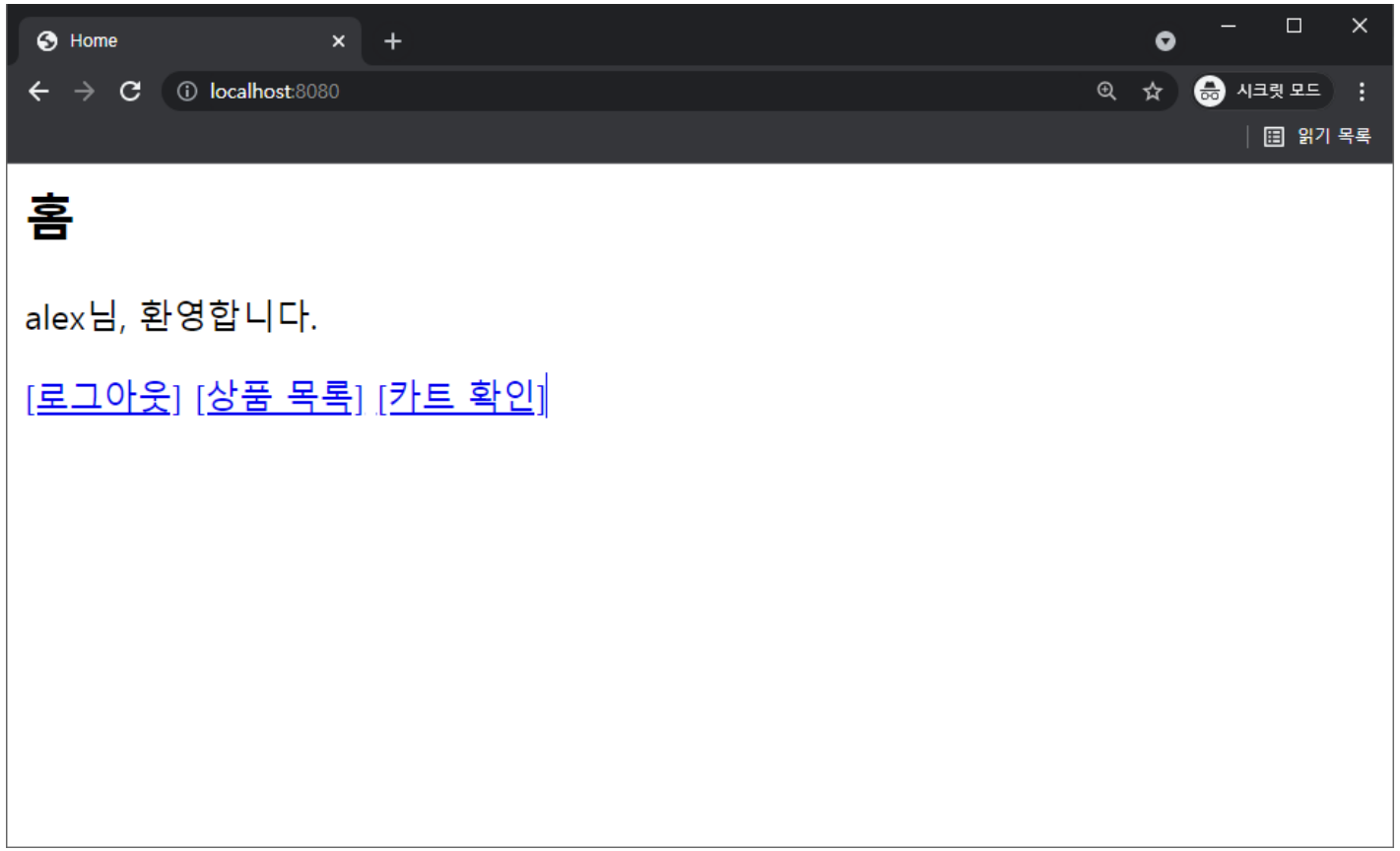
~/views/auth/loginSuccess.jsp

39

아래 다이어그램 및 실행화면을 참고하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.

localhost:8080/

GUIDE



file

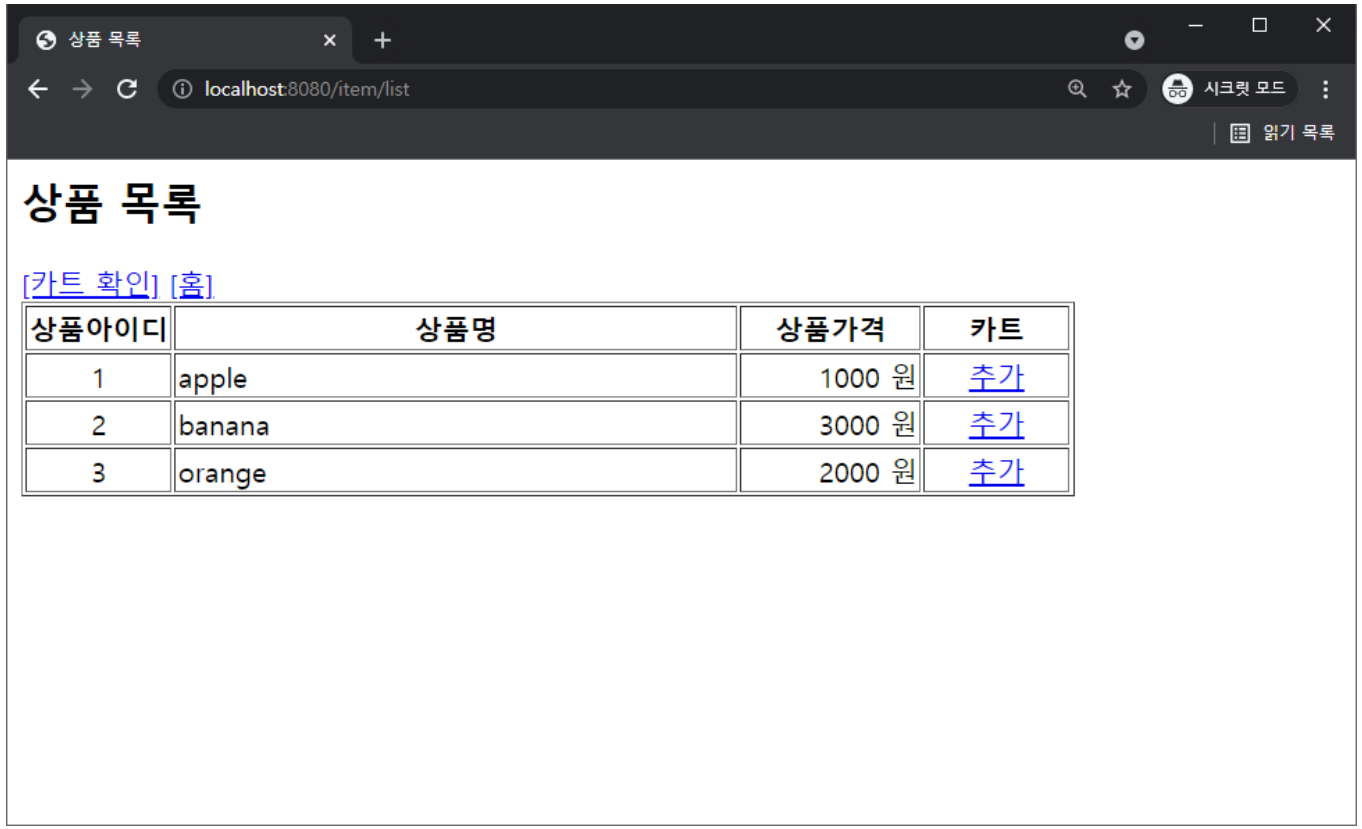
~/views/home.jsp

39

아래 다이어그램 및 실행화면을 참고하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.

localhost:8080/Item/list

GUIDE



file

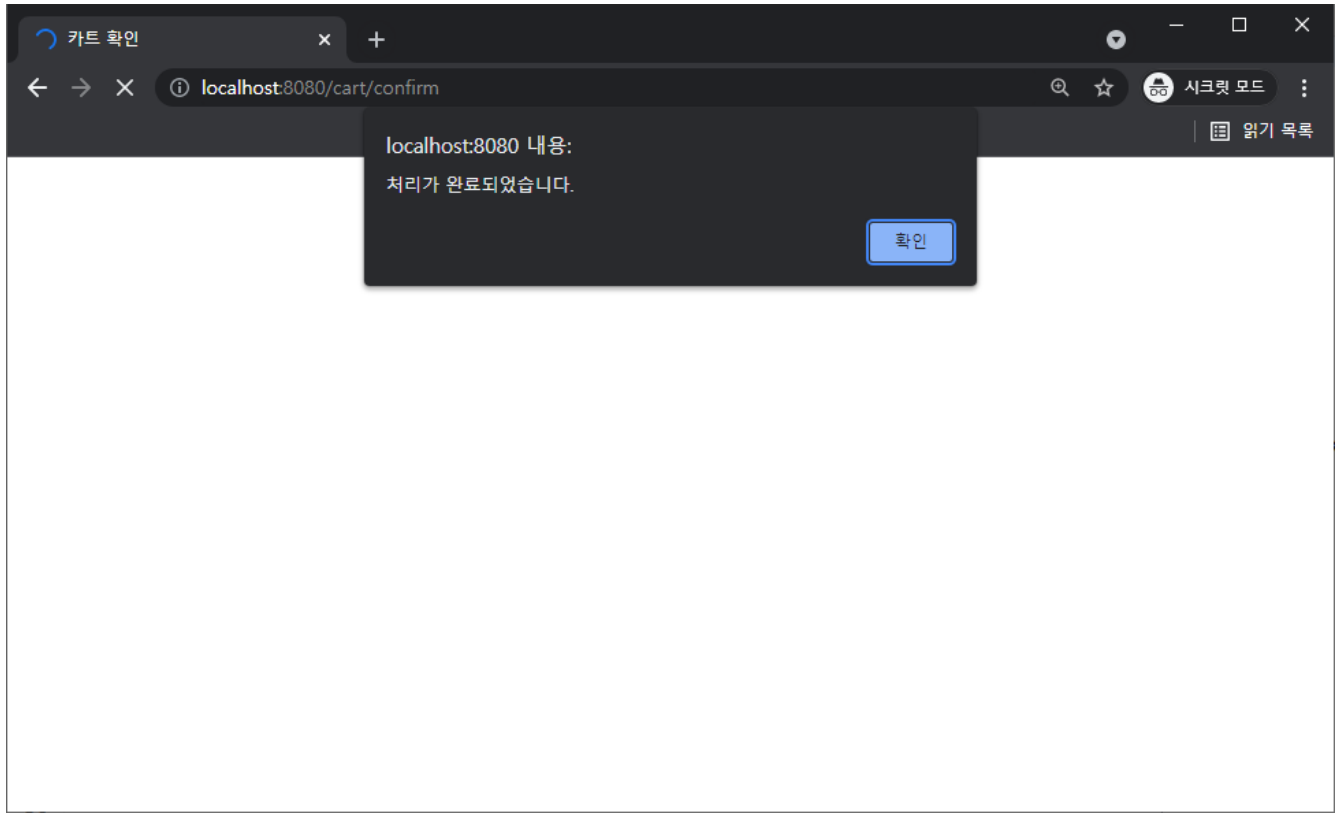
~/views/item/list.jsp

39

아래 다이어그램 및 실행화면을 참고하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.

localhost:8080/cart/add

GUIDE



file

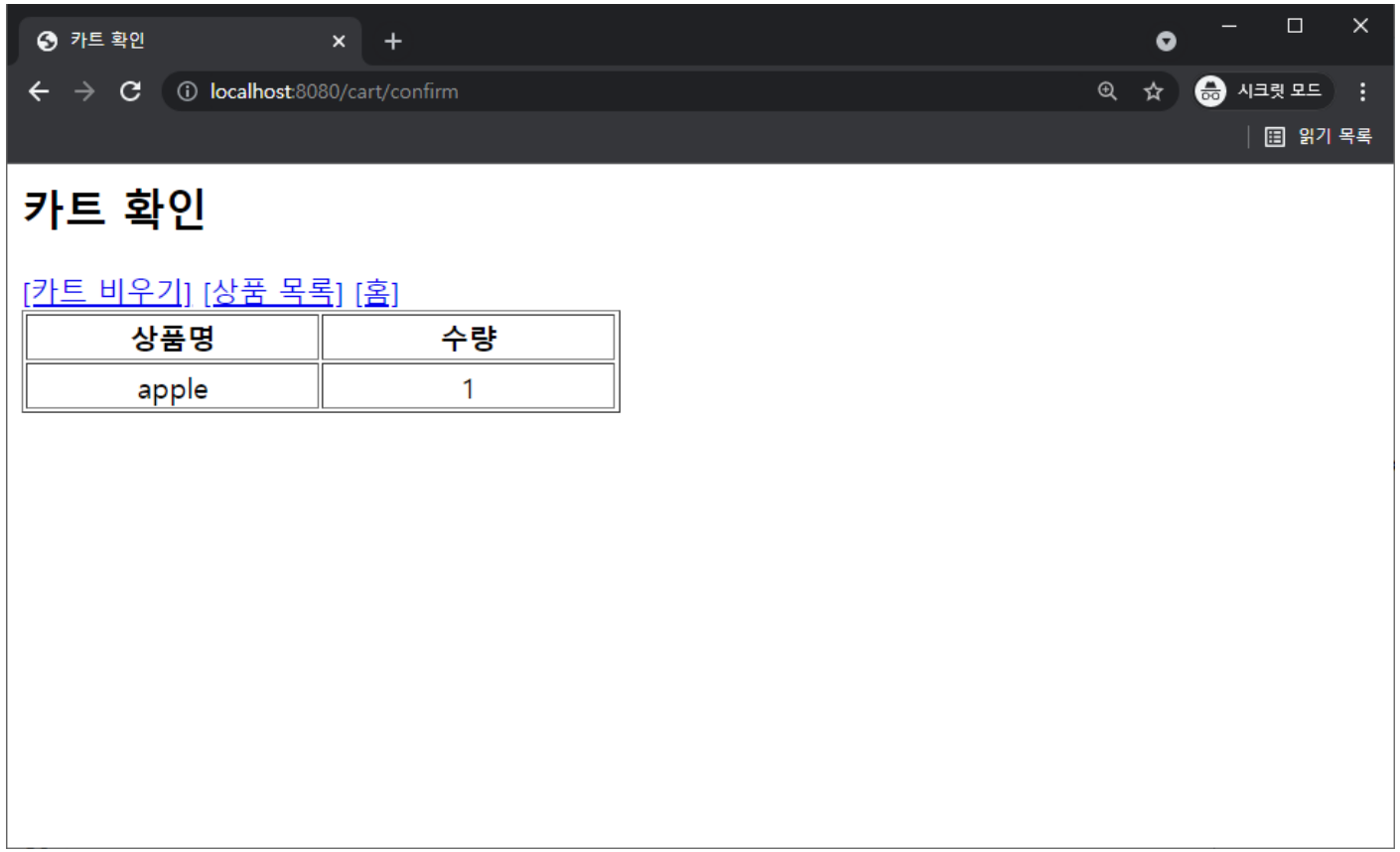
~/views/cart/confirm.jsp

39

아래 다이어그램 및 실행화면을 참고하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.

localhost:8080/cart/confirm

GUIDE



file

~/views/cart/confirm.jsp

40

아래 다이어그램 및 실행화면을 참고하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.

@SessionAttribute

※ [미션 39] 에 이어서 아래 조건에 맞는 코드를 추가하시오.

1. 세션에 등록된 쇼핑카트가 없어도 에러가 발생하지 않도록 애너테이션을 지정하시오.

    1. 메소드 수준에 적용하시오.

file

~/controller/CartController.java

41

@SessionAttributes

아래 다이어그램 및 실행화면을 참고하여 쇼핑 카트 기능을 구현한 프로젝트를 완성하시오.

※ [미션 39] 에 이어서 아래 조건에 맞는 코드를 추가하시오.

1. 세션에 등록된 쇼핑카트가 없어도 에러가 발생하지 않도록 애너테이션을 지정하시오.
    1. 클래스 수준에 적용하시오.

2. Cart 객체가 모델에 생성되도록 파라미터에 명시적으로 지정하시오.

3. SessionStatus 객체를 이용하여, 세션을 초기화하도록 코드를 수정하시오.

file

~/controller/CartController.java

42

아래 조건을 참고하여 파일 업로드 기능을 구현한 프로젝트를 완성하시오.

- 파일 업로드 뷰 페이지를 작성하시오.
- 업로드된 파일을 프로젝트 내 ~/upload 폴더에 저장하시오.

43

아래 조건을 참고하여 파일 다운로드 기능을 구현한 프로젝트를 완성하시오.

- 업로드된 파일을 다운로드 하시오.

44

아래 조건을 참고하여 파일 업로드 기능을 구현한 프로젝트를 완성하시오.

- 비동기 통신을 이용하여 파일을 업로드 하시오.

45

아래 조건을 참고하여 팝업 기능을 구현한 프로젝트를 완성하시오.

1. 쿠키를 활용하여 팝업 기능을 구현하다.

2. 메인 페이지에서 팝업 창을 띄워 출력한다.

3. 오늘 하루 보지 않음 체크박스를 클릭하면, 하루 동안 팝업창이 출력되지 않도로 구현한다.

file

46

아래 조건을 참고하여 아이디 저장 기능을 구현한 프로젝트를 완성하시오.

쿠키 활용
- 아이디 저장

※ [미션37~38] 프로젝트에 이어서, 아이디(email) 저장 기능을 추가하시오.

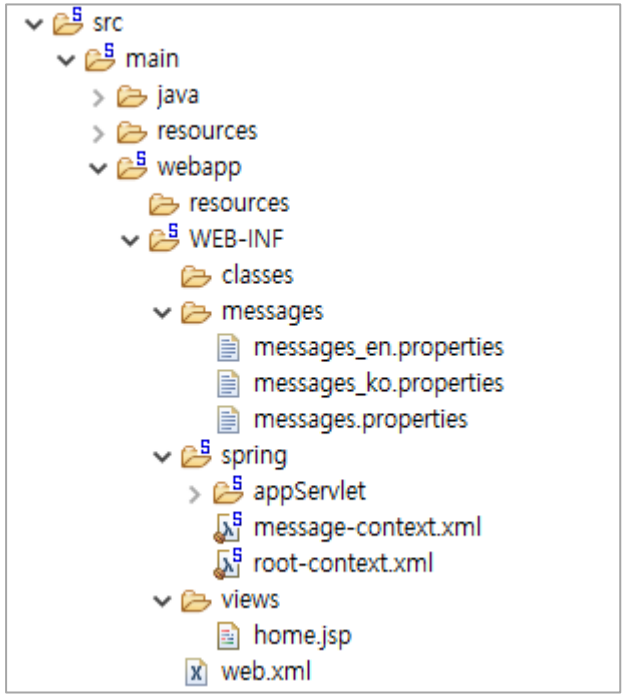
1. 아이디 저장 체크박스 체크 후, 로그인 시 쿠키에 아이디를 등록한다.

2. 아이디 저장 체크박스 해제 후, 로그인 시 등록된 쿠기 정보를 제거한다.

47

아래 조건을 참고하여 메시지를 사용하기 위한 설정을 완성하시오.

1. 그림을 참고하여, 메시지 설정을 하시오.
    1. message-context.xml 파일을 생성하시오.
        1. 메시지 파일의 위치를 지정하시오
        2. 파일의 기본 인코딩을 지정하시오.
        3. 파일이 변경되었는지 확인하는 주기를 설정하시오.
    2. messages.properties 파일을 생성하시오.
        1. 샘플 메시지를 작성하시오.



file

48

아래 조건을 참고하여 메시지를 사용하기 위한 프로젝트를 완성하시오.

Controller에서 메시지 소스 사용

1. 메시지 파일에 작성한 메시지를 가져온다.
    1. 가져온 메시지를 로그에 출력한다.

    2. 가져온 메시지를 화면에 출력한다.

file

~/controller/HomeController.java

49

아래 조건을 참고하여 메시지를 사용하기 위한 프로젝트를 완성하시오.

※ [미션48] 에 이어서, 메시지를 국제화 하는 프로젝트를 완성하시오.

1. 클라이언트의 locale 에 따라서 다른 메시지를 출력하시오.

국제화

file

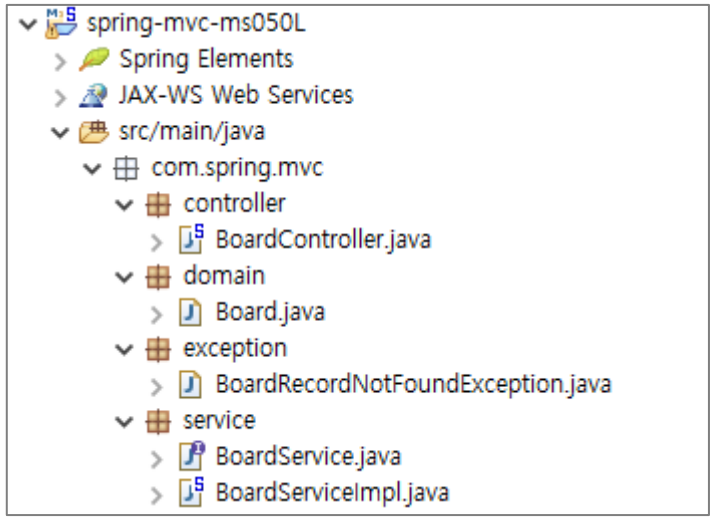
50

아래 조건을 참고하여, 예외 처리를 하기 위한 프로젝트를 완성하시오.

1. 그림을 참고하여, 클래스 파일을 생성하시오.

2. 게시글 쓰기, 읽기 기능을 구현하시오.

3. 게시글 읽기 기능 수행 시, 레코드가 존재하지 않으면, BoardRecordNotFoundException.java 에 정의한 예외 발생 메시지를 출력하는 메소드가 호출되도록 구현하시오.



file

51

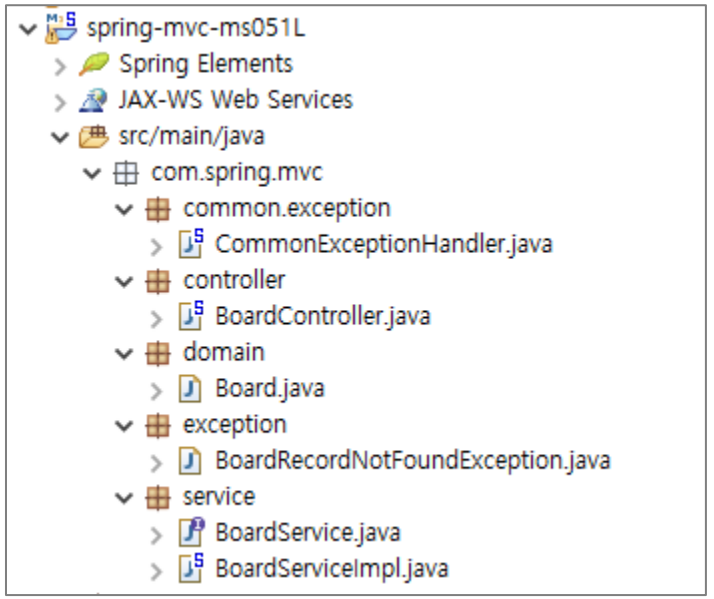
아래 조건을 참고하여, 예외 처리를 하기 위한 프로젝트를 완성하시오.

1. 그림을 참고하여, 클래스 파일을 생성하시오.

2. 게시글 쓰기, 읽기 기능을 구현하시오.

3. 게시글 읽기 기능 수행 시, 레코드가 존재하지 않으면, BoardRecordNotFoundException.java 에 정의한 예외 발생 메시지를 출력하는 메소드가 호출되도록 구현하시오.

4. 예외 처리 애너테이션을 이용하여, 예외 발생 핸들러 CommonExceptionHandler.java 클래스를 완성하시오.



file

52

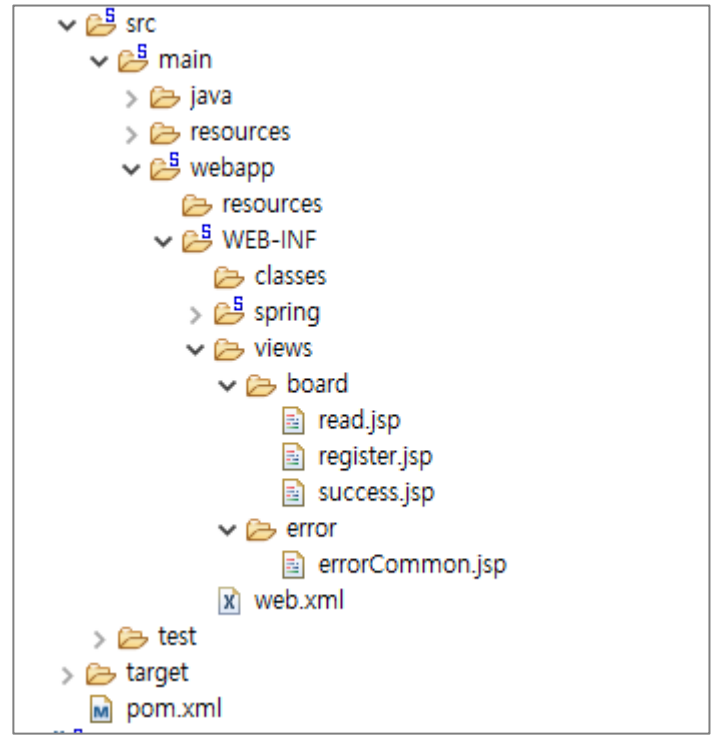
아래 조건을 참고하여, 예외 처리를 하기 위한 프로젝트를 완성하시오.

예외 정보 출력

- ※ [미션50] 에 이어서, 아래 조건에 맞게 프로젝트를 완성하시오.

1. 예외 메시지를 출력하는 뷰 파일을 생성하시오.
    1. ~/views/error/errorCommon.jsp

2. 예외 메시지를 화면에 출력하시오.



file

~/views/error/errorCommon.jsp

53

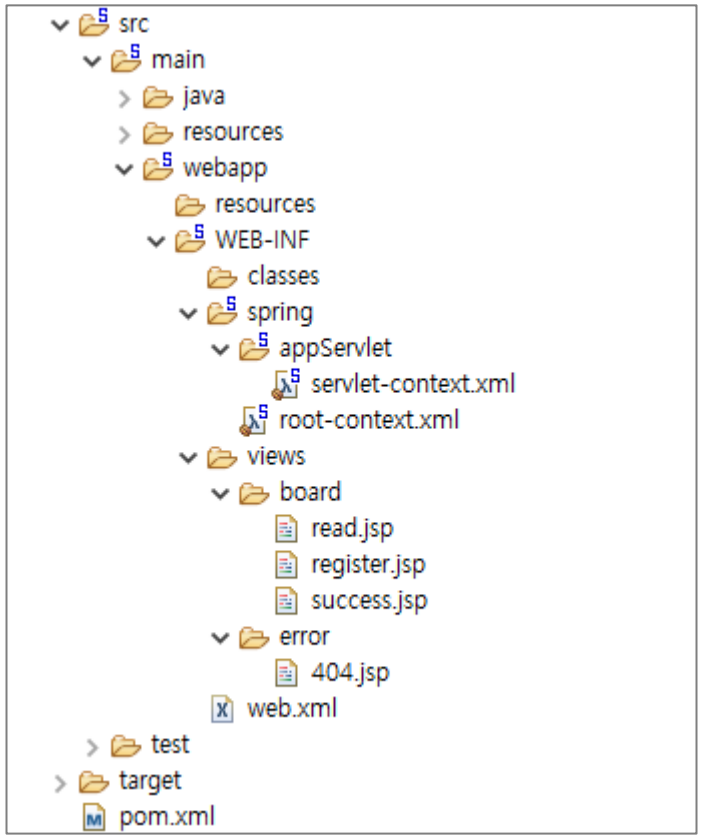
404 에러 페이지 처리

아래 조건을 참고하여, 에러 페이지를 프로젝트에 추가하시오.

※ [미션50] 에 이어서, 아래 조건에 맞게 프로젝트를 완성하시오.

1. 그림을 참고하여, 뷰 파일을 생성하시오.

2. 404 에러 발생 시, 응답될 에러 페이지를 생성하시오.



file

~/views/error/404.jsp

54

아래 조건을 참고하여, 예외 처리를 하기 위한 프로젝트를 완성하시오.

※ [미션50] 에 이어서, 아래 조건에 맞게 프로젝트를 완성하시오.

입력값 검증 에러 처리

1. 글쓰기 요청 시,
   게시글 정보의 제목이 비어있는지 검증하는 입력값 검증 로직을 추가하시오.

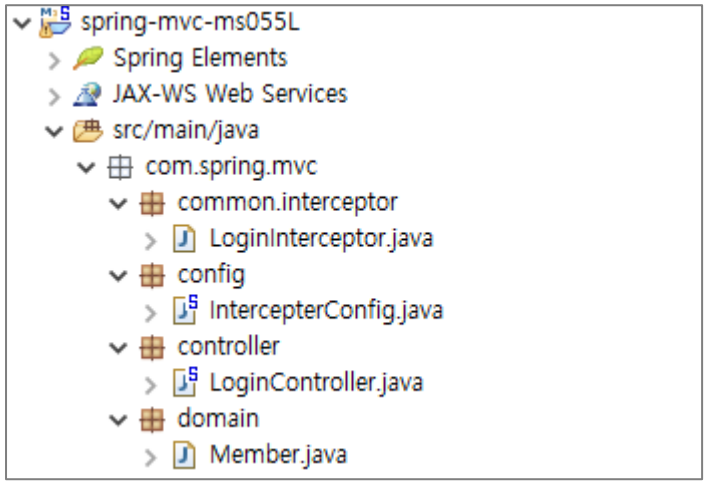
file

~/controller/BoardController.java

55

아래 조건을 참고하여, 인터셉터 설정을 하기 위한 프로젝트를 완성하시오.

1. 그림을 참고하여, 로그인 처리 기능을 구현하시오.



2. 인터셉터 설정 클래스 파일을 생성하시오.
   1. 로그인 요청 시 동작하는 인터셉터를 추가하시오.

3. 로그인 인터셉터 클래스 파일을 생성하시오.
   1. preHandle(), postHandle(), afterCompletion() 메소드를 정의하시오.
   2. 메소드 호출 확인을 위한 로그를 작성하시오.

file

56

아래 조건을 참고하여, 인터셉터 활용 하기 위한 프로젝트를 완성하시오.

※ [미션55] 에 이어서, 인터셉터를 활용하는 프로젝트를 완성하시오.

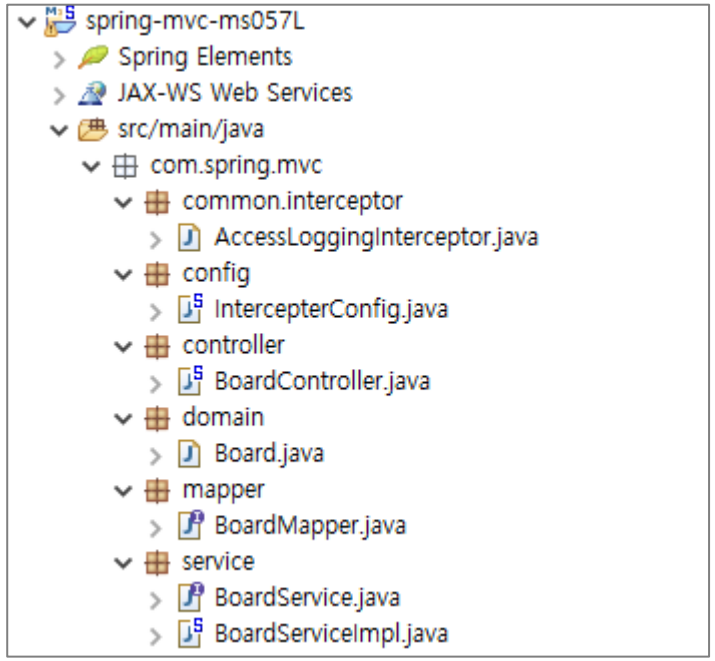
1. 로그인 인터셉터를 이용하여,
   로그인에 대한 세션 처리 로직을 LoginController.java 에서 인터셉터 클래스로 이전하시오.

~/common/interceptor/LoginInterceptor.java

57

아래 조건을 참고하여, 인터셉터 활용 하기 위한 프로젝트를 완성하시오.

1. 그림을 참고하여, 모든 경로에 대해 접근 로깅 인터셉터가 로그를 저장하도록 코드를 완성하시오.



file

~/common/interceptor/AccessLoggingInterceptor.java

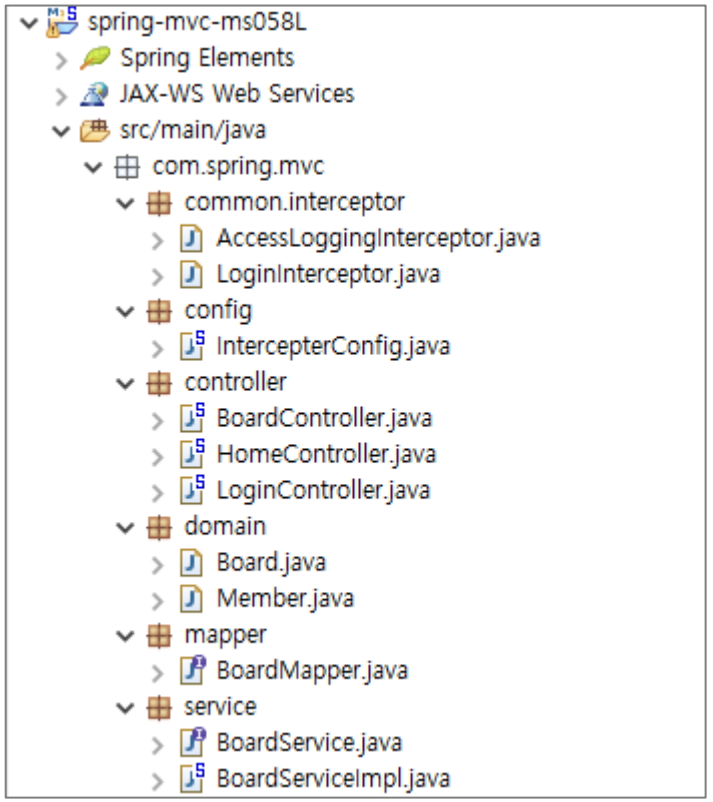
58

아래 조건을 참고하여, 인터셉터 활용 하기 위한 프로젝트를 완성하시오.

※ [미션55~57] 에서 작성한 인터셉터를 통합하시오.

여러 개의 인터셉터 지정

1. 그림을 참고하여,
   로그인 인터셉터, 접근 로깅 인터셉터를 통합하시오.



file

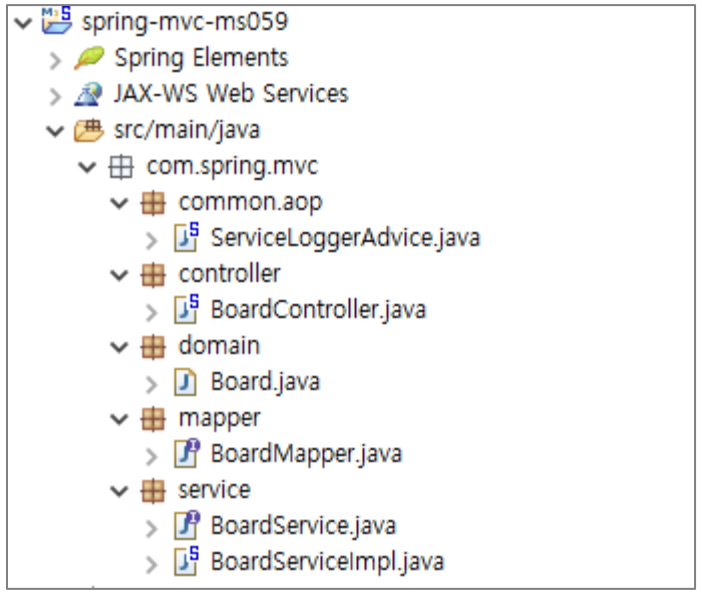
59

아래 조건을 참고하여, AOP 기능를 활용하기 위한 프로젝트를 완성하시오.

Before 어드바이스

1. 그림을 참고하여,
   MVC 패턴의 간단한 게시판 기능을 완성하시오.

2. BoardService 가 실행되기 전,
   AOP 기능을 이용하여 확인 로그를 출력하시오.



file

~/common/aop/ServiceLoggerAdvice.java

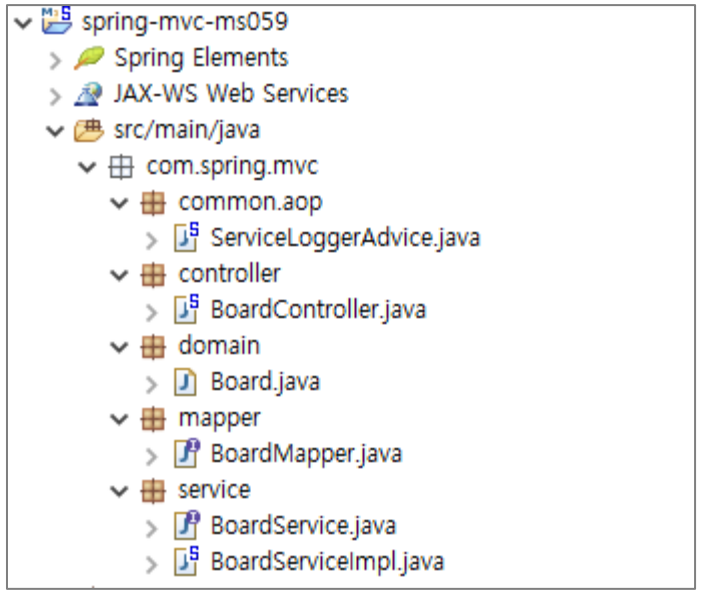
60

아래 조건을 참고하여, AOP 기능를 활용하기 위한 프로젝트를 완성하시오.

After Returning
어드바이스

1. 그림을 참고하여,
MVC 패턴의 간단한 게시판 기능을 완성하시오.

2. BoardService 가 리턴된 후,
AOP 기능을 이용하여 확인 로그를 출력하시오.



file

~/common/aop/ServiceLoggerAdvice.java

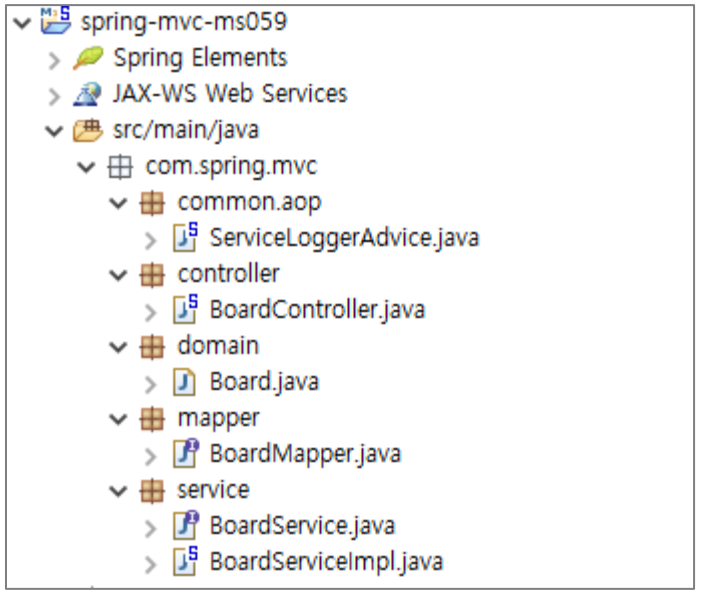
61

아래 조건을 참고하여, AOP 기능를 활용하기 위한 프로젝트를 완성하시오.

After Throwing
어드바이스

1. 그림을 참고하여,
MVC 패턴의 간단한 게시판 기능을 완성하시오.

2. BoardService 에서 예외발생 후,
AOP 기능을 이용하여 확인 로그를 출력하시오.



file

~/common/aop/ServiceLoggerAdvice.java

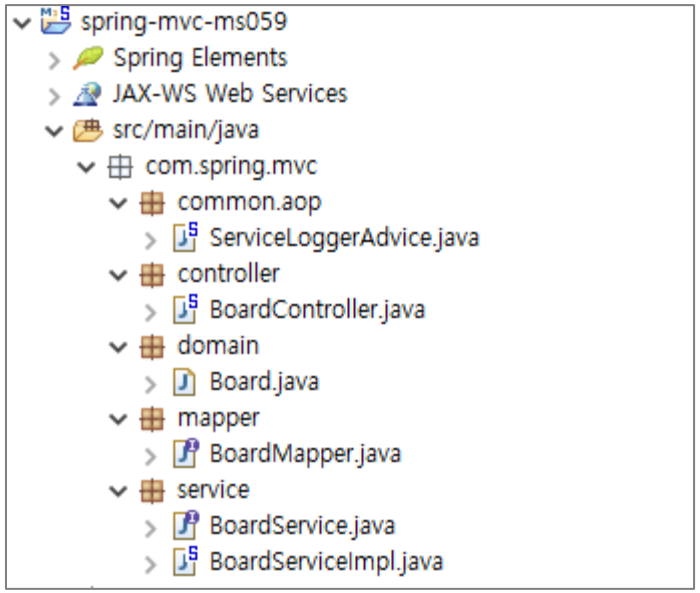
62

아래 조건을 참고하여, AOP 기능를 활용하기 위한 프로젝트를 완성하시오.

After 어드바이스

1. 그림을 참고하여,
MVC 패턴의 간단한 게시판 기능을 완성하시오.

2. BoardService 가 실행된 후,
AOP 기능을 이용하여 확인 로그를 출력하시오.



file

~/common/aop/ServiceLoggerAdvice.java

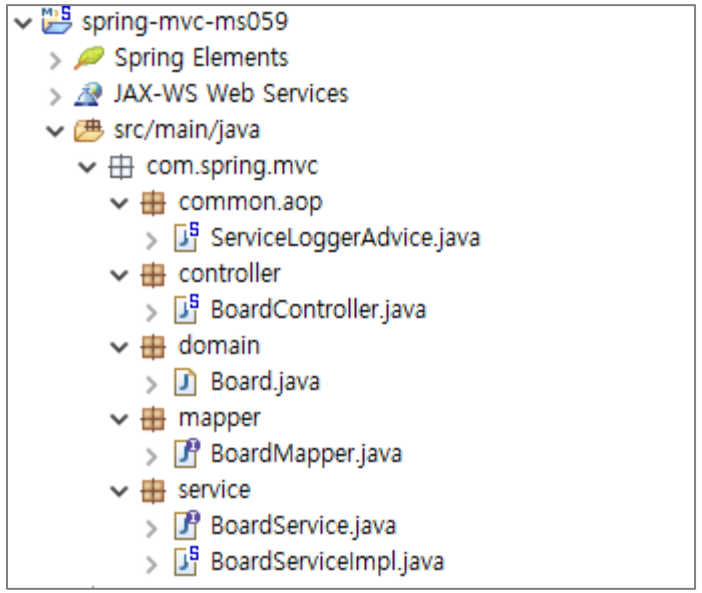
63

아래 조건을 참고하여, AOP 기능를 활용하기 위한 프로젝트를 완성하시오.

Around 어드바이스

1. 그림을 참고하여,
MVC 패턴의 간단한 게시판 기능을 완성하시오.

2. BoardService 가 실행 전과 후,
AOP 기능을 이용하여 확인 로그를 출력하시오.



file

~/common/aop/ServiceLoggerAdvice.java

64

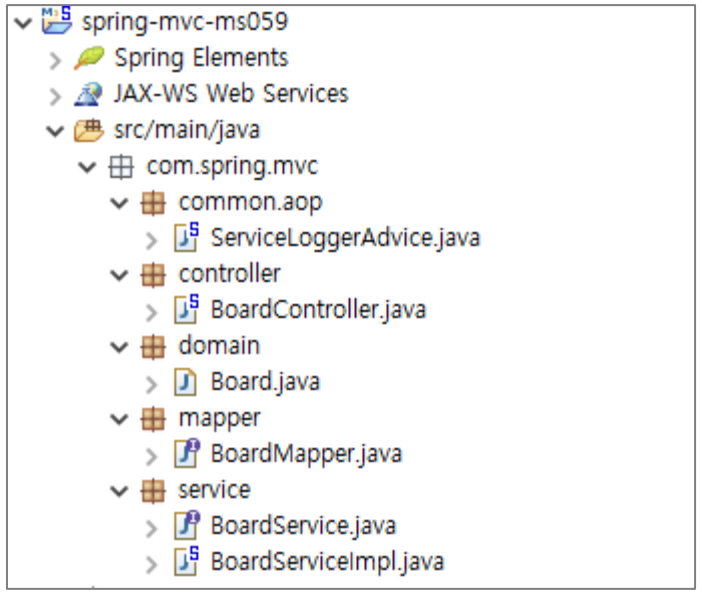
아래 조건을 참고하여, AOP 기능를 활용하기 위한 프로젝트를 완성하시오.

메소드 정보 얻기

1. 그림을 참고하여,
   MVC 패턴의 간단한 게시판 기능을 완성하시오.

2. BoardService 가 실행 전,
   AOP 기능을 이용하여 확인 로그를 출력하시오.

3. 메소드 정보를 가져와 로그에 출력하시오.



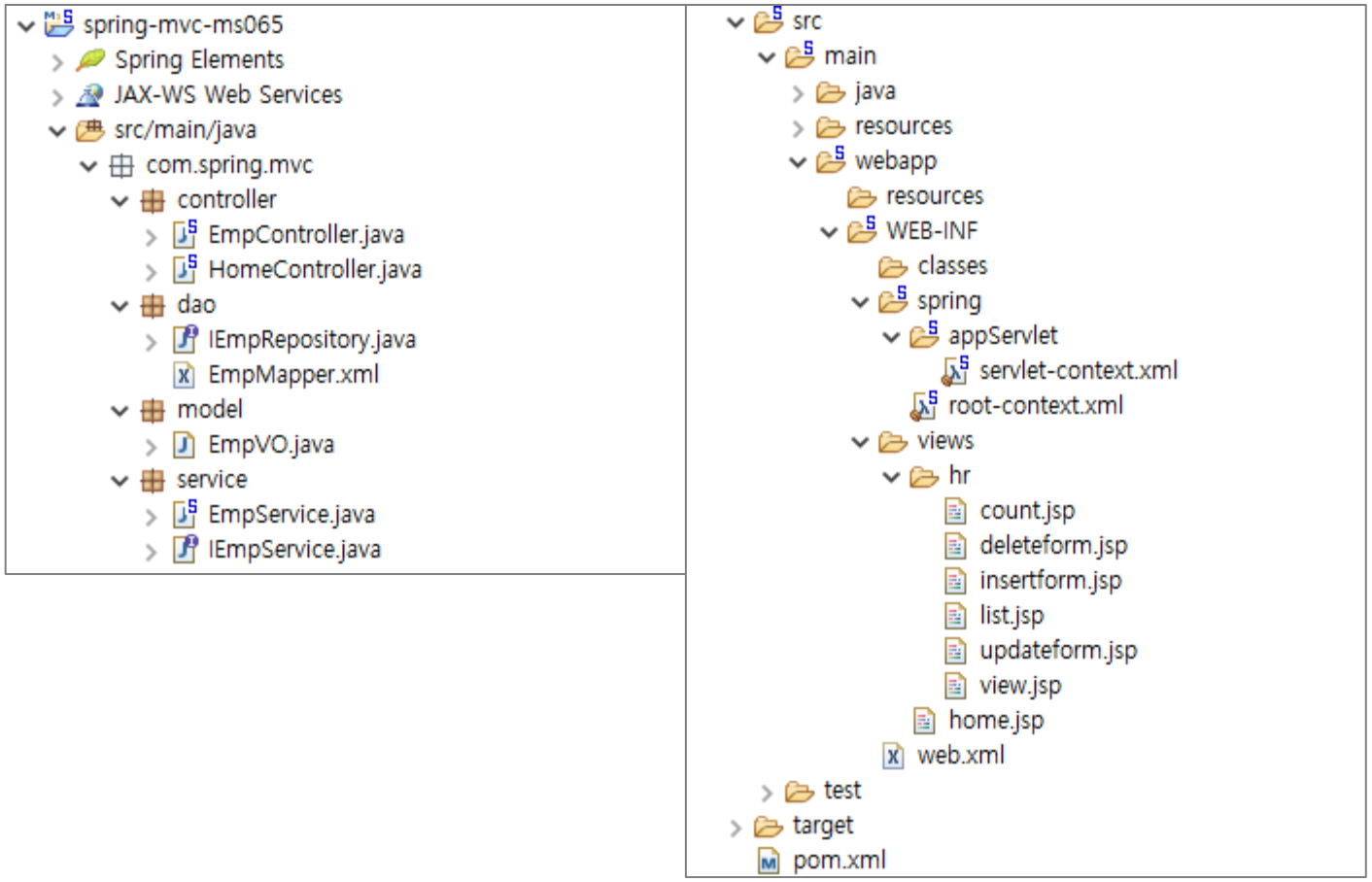
file

~/common/aop/ServiceLoggerAdvice.java

65

아래 조건을 참고하여, Mybatis를 연동하기 위한 프로젝트를 완성하시오.

- Mybatis 를 연동하고, Oracle 샘플 테이블 [hr] 에 대해 C.R.U.D 기능을 완성하시오.



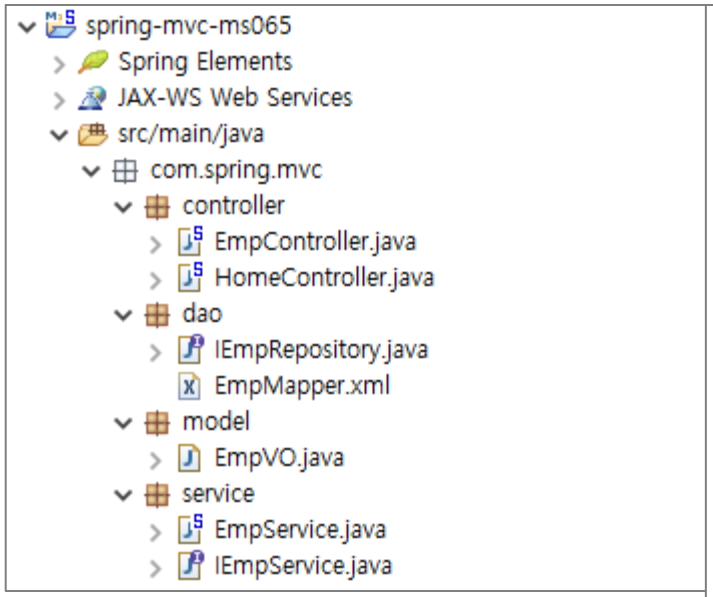
file

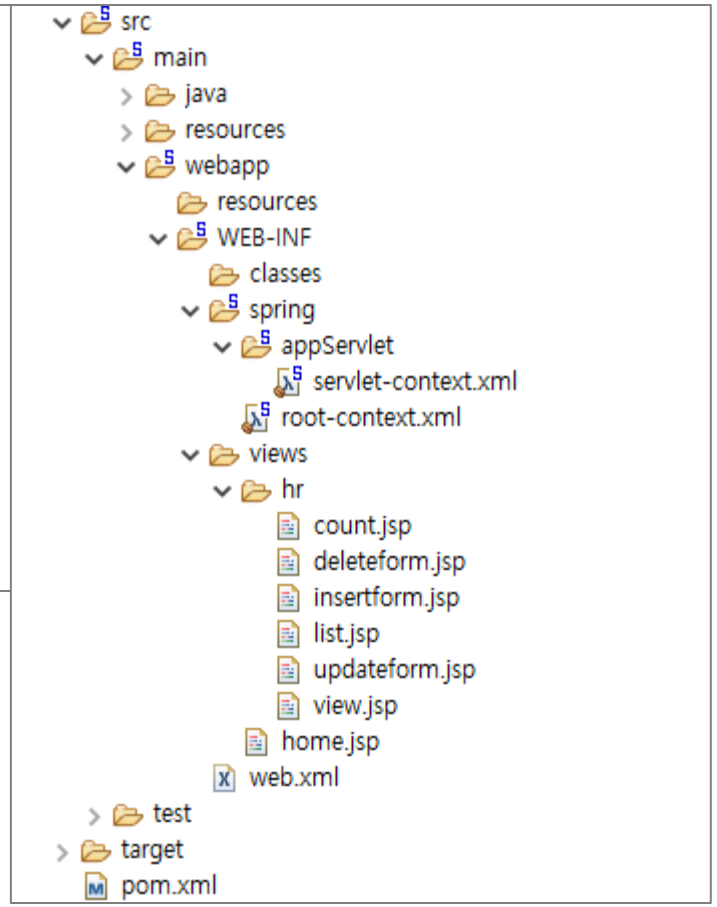
66

아래 조건을 참고하여, Mybatis를 연동하기 위한 프로젝트를 완성하시오.

HR 사원목록 조회

- Mybatis 를 연동하고, Oracle 샘플 테이블 [hr] 에 대해 C.R.U.D 기능을 완성하시오.
  - HR 사원 목록을 조회하는 기능을 완성하시오.





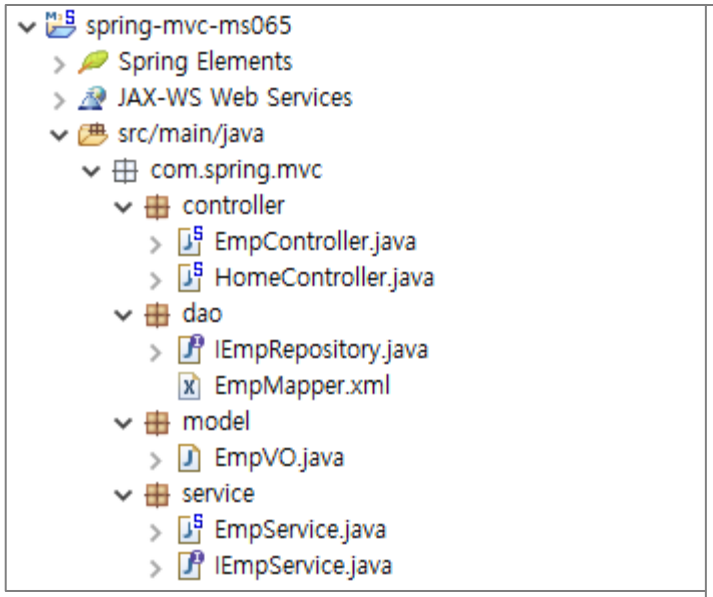
file

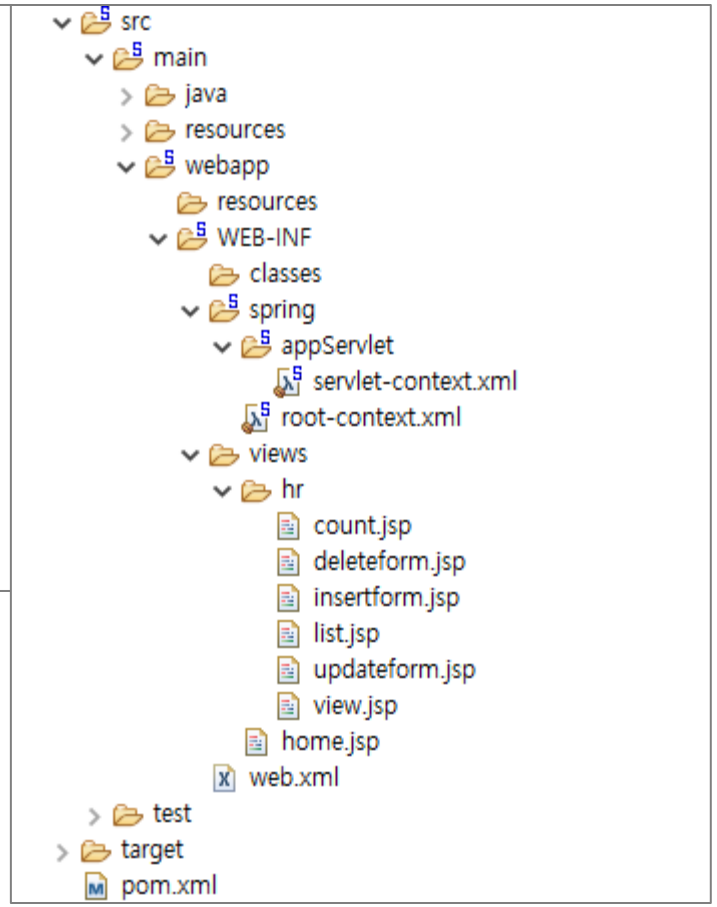
67

아래 조건을 참고하여, Mybatis를 연동하기 위한 프로젝트를 완성하시오.

HR 사원정보 입력

- Mybatis 를 연동하고, Oracle 샘플 테이블 [hr] 에 대해 C.R.U.D 기능을 완성하시오.
  - HR 사원 정보를 입력하는 기능을 완성하시오.





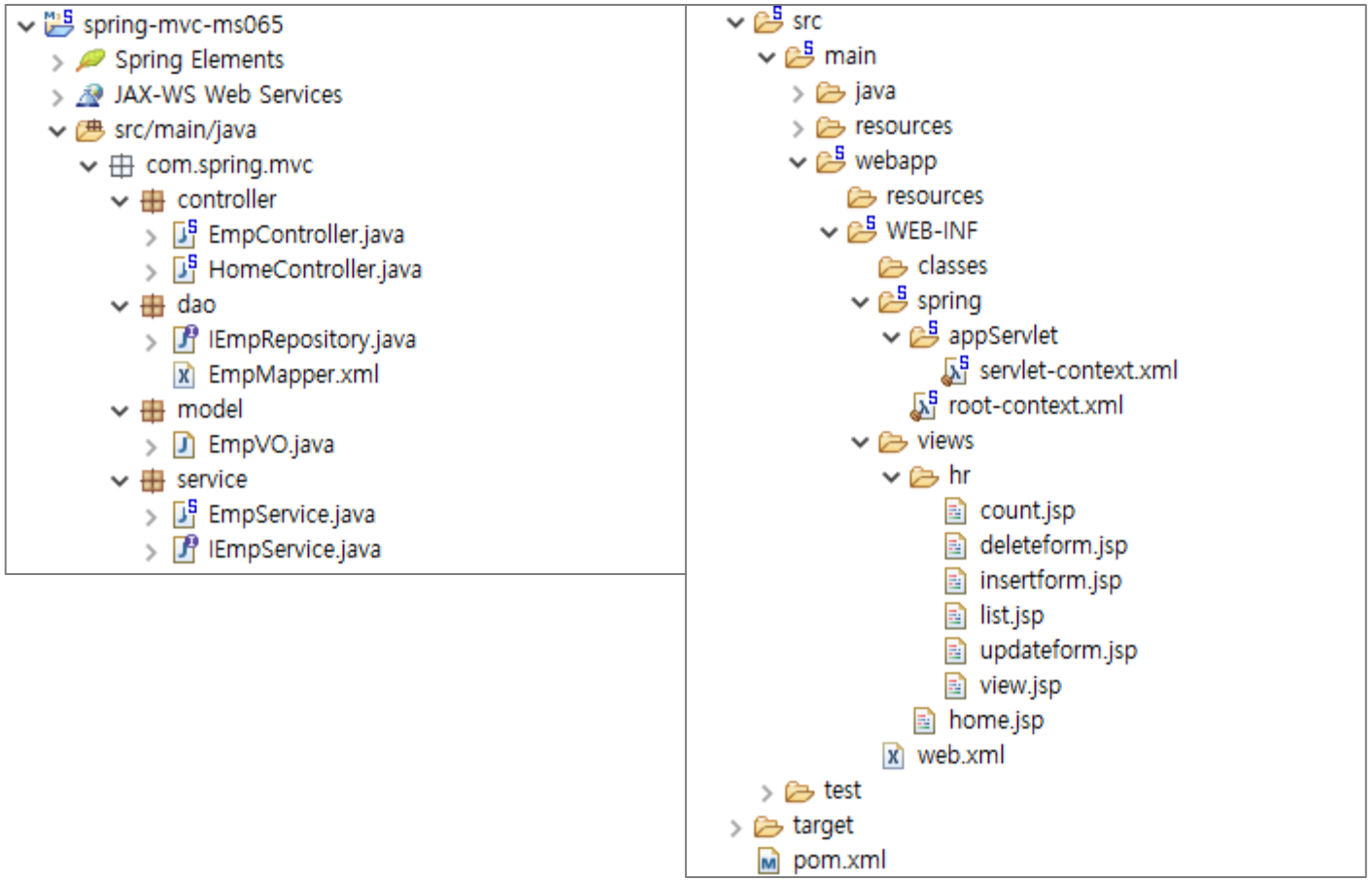
file

68

아래 조건을 참고하여, Mybatis를 연동하기 위한 프로젝트를 완성하시오.

HR 사원정보 수정

- Mybatis 를 연동하고, Oracle 샘플 테이블 [hr] 에 대해 C.R.U.D 기능을 완성하시오.
  - HR 사원 정보를 수정하는 기능을 완성하시오.



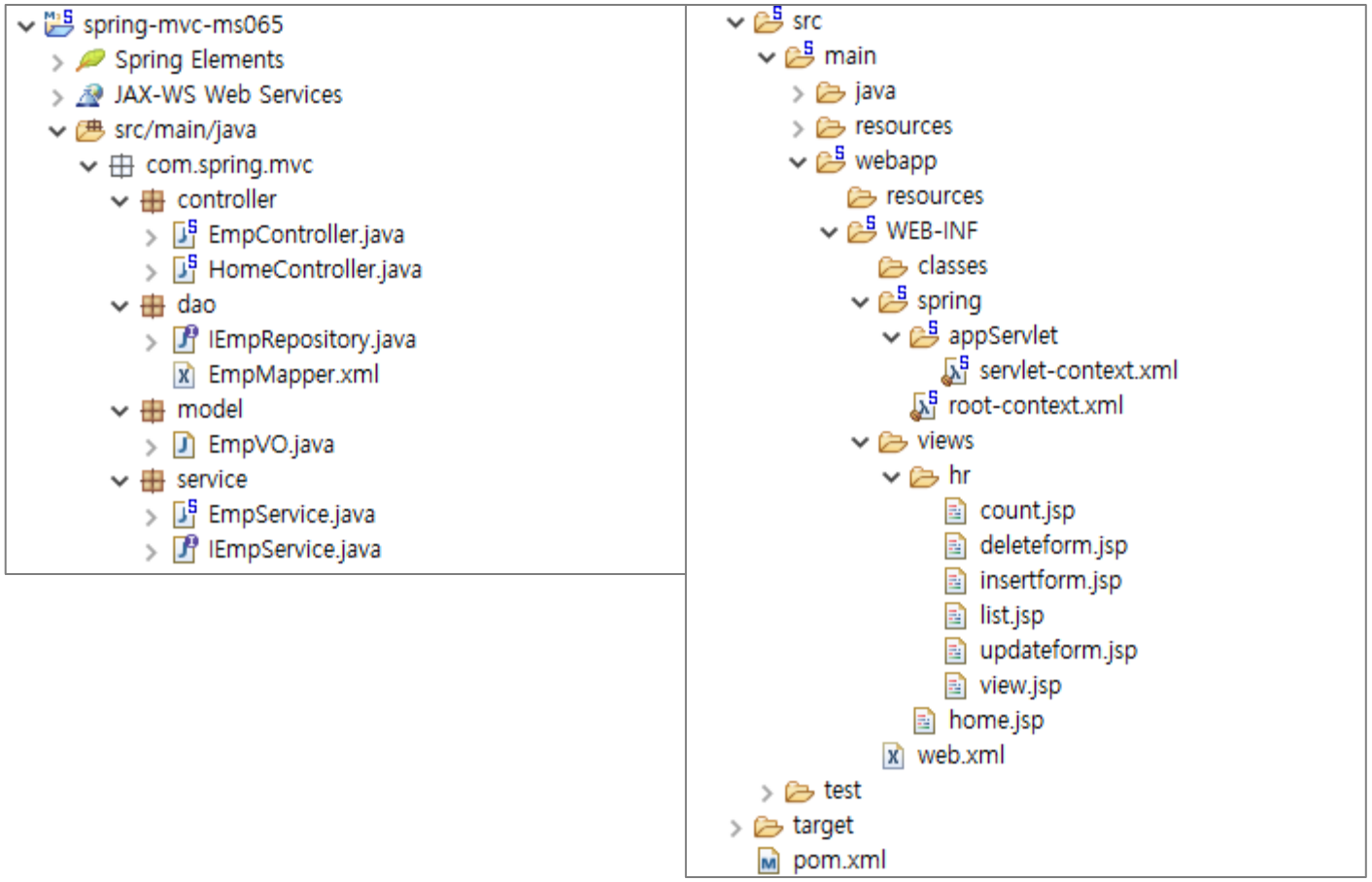
file

69

아래 조건을 참고하여, Mybatis를 연동하기 위한 프로젝트를 완성하시오.

HR 사원정보 삭제

- Mybatis 를 연동하고, Oracle 샘플 테이블 [hr] 에 대해 C.R.U.D 기능을 완성하시오.
  - HR 사원 정보를 삭제하는 기능을 완성하시오.



file

70

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

테이블 생성

- DB 에 멀티게시판 에 필요한 테이블 및 시퀀스 등을 생성하시오.

71

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

환경설정

- 필요한 의존 라이브러리를 추가하시오.
- 빈 설정을 하시오.
- 데이터 소스를 설정하시오.

72

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

Value Object

- 데이터 처리에 필요한 Value Object 클래스들을 생성하시오.

73

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판 카테고리 - @RequestMapping

- 게시판 카테고리에 대한 요청경로 매핑을 하고 메소드를 완성하시오.

74

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판 카테고리 -
Mapper 인터페이스

- 게시판 카테고리에 대한 Mapper 인터페이스를 완성하시오.

75

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판 카테고리 - Mapper XML

- 게시판 카테고리에 대한 Mapper XML 파일을 완성하시오.

76

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판 카테고리 -
Service 인터페이스

- 게시판 카테고리에 대한 Service 인터페이스를 완성하시오.

77

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판 카테고리 -
Service 클래스

- 게시판 카테고리에 대한 Service 클래스를 완성하시오.

78

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판 카테고리 - Controller

- 게시판 카테고리에 대한 Controller 클래스를 완성하시오.

79

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판 카테고리 - View

- 게시판 카테고리에 대한 View 를 완성하시오.

80

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판 - @RequestMapping

- 게시판에 대한 요청경로 매핑을 하고 메소드를 완성하시오.

file

81

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판
- Mapper 인터페이스

- 게시판에 대한 Mapper 인터페이스를 완성하시오.

82

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판
- Mapper XML

- 게시판에 대한 Mapper XML 파일을 완성하시오.

83

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판
- Service 인터페이스

- 게시판에 대한 Service 인터페이스를 완성하시오.

84

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판
- Service 클래스

- 게시판에 대한 Service 클래스를 완성하시오.

85

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판
- Controller

- 게시판에 대한 Controller 클래스를 완성하시오.

86

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

게시판
- View

- 게시판에 대한 View 를 완성하시오.

87

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

- 게시판
  - 페이징 기능을 완성하시오.

file

88

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

- 게시판
  - 검색기능을 완성하시오.

89

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

- 게시판
  - 검색기능을 완성하시오.

90

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

회원 - @RequestMapping

- 회원에 대한 요청경로 매핑을 하고 메소드를 완성하시오.

91

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

회원
- Mapper 인터페이스

- 회원에 대한 Mapper 인터페이스를 완성하시오.

92

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

회원
- Mapper XML

- 회원에 대한 Mapper XML 파일을 완성하시오.

93

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

회원
- Service 인터페이스

- 회원에 대한 Service 인터페이스를 완성하시오.

94

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

회원
- Service 클래스

- 회원에 대한 Service 클래스를 완성하시오.

95

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

회원
- Controller

- 회원에 대한 Controller 클래스를 완성하시오.

file

96

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

회원
- View

- 회원에 대한 View 를 완성하시오.

97

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

로그인 인터셉터

- 인터셉터를 이용하여,
  모든 경로 요청에 대하여 세션에 아이디(email)이 등록되어 있지 않으면
  로그인 페이지로 리다이렉트 하도록 LoginInterceptor.java 를 완성하시오.

98

아래 조건을 참고하여, 멀티 게시판 기능을 위한 프로젝트를 완성하시오.

XXS 공격 대응 – jsoup lib

- Jsoup lib 를 추가하고, XXS 공격 대응을 하는 보안 로직을 추가하시오.

99

GIT, GITHub 연동을 위한 작업을 아래 조건에 따라 수행하시오.

GIT 연동

- 현재 가지 미션을 통해서 작업한 프로젝트를 Git에 연동하시오.
- COMMIT & PUSH 하시오.
- 다른 PC 또는 다른 사용자로 해당 프로젝트를 PULL 하시오.

100

작업 프로젝트를 아마존 웹 서비스를 이용하여 배포하시오.

AWS 배포

- https://aws.amazon.com/ko/
- ECS2, RDS 서비스를 이용한다.
- Putty 등의 프로그램을 이용하여, 서버에 접속한다.
- JDK 등 웹 어플리케이션 실행에 필요한 드라이버 등을 설치한다.
- 서버 실행 스크립트를 작성한다.
- 스크립트 실행 시, git repository 에 업로드된 프로젝트를 내려받아 실행하도록 한다.

모두의 코딩

수강생의 학습목표 달성에
최선을 다하겠습니다.

02) 000 - 0000