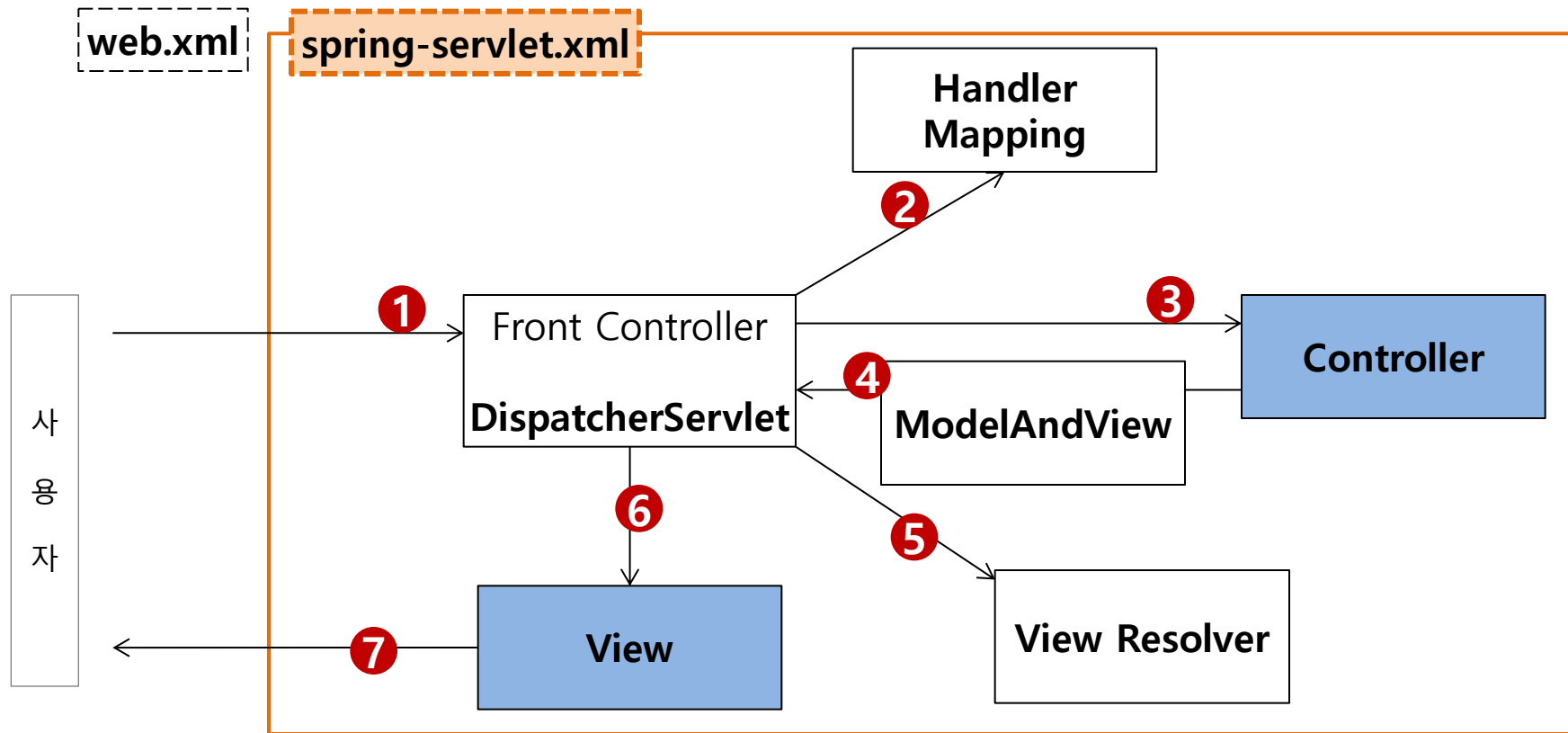


chapter02

주요 어노테이션 과 Context

1. 스프링의 DispatcherServlet 과 MVC
2. DispatcherServlet → controller
(주요 Annotation)
3. controller → DispatcherServlet

■ 스프링의 DispatcherServlet 과 MVC



■ 스프링의 DispatcherServlet 과 MVC

- 사용자의 요청을 DispatcherServlet이 받는다.
- 요청을 처리해야 하는 컨트롤을 찾기위해 HandlerMapping에게 질의를 하고 HandlerMapping은 컨트롤 객체에 매핑되어 있는 URL를 찾아낸다.
- Dispatcher Servlet은 찾은 컨트롤에게 요청을 전달하고 Controller는 서비스 계층의 인터페이스를 호출하여 적절한 비즈니스를 수행한다.
- 컨트롤러는 비즈니스 로직의 수행결과로 받아낸 도메인 모델 객체와 함께 뷰이름을 ModelAndView 객체에 저장하여 리턴한다.
- DispatcherServlet은 응답할 View를 찾기위해 ViewResolver에게 질의를 한다.
- DispatcherServlet은 찾아낸 View 객체에게 요청을 전달한다.

chapter02

주요 어노테이션 과 Context

1. 스프링의 DispatchedServlet 과 MVC
2. **DispatcherServlet → controller**
(주요 Annotation)
3. controller → DispatcherServlet

02 DispatcherServlet → controller

- @Controller
- @Service
- @Repository

- @Component

- @Autowired

■ @RequestMapping : 메소드 단독 매핑

```
@Controller
```

```
public class Hello {
```

```
    @RequestMapping( value="/list", method=RequestMethod.GET )
```

```
    public String listGet( ) {...}
```

```
    @RequestMapping( value="/list", method=RequestMethod.POST )
```

```
    public String listPost( ) {...}
```

```
    @RequestMapping( value="/list", method={RequestMethod.GET, RequestMethod.POST} )
```

```
    public String list( ) {...}
```

```
    @RequestMapping( "/list" ) //위와같음
```

```
    public String list( ) {...}
```

```
}
```

■ @RequestMapping : 타입 + 메소드 매핑

@Controller

public class UserController {**@RequestMapping("/user/addform")**

public String addFrom() {...}

@RequestMapping("/user/add")

public String add() {...}

@RequestMapping("/user/delete")

public String delete() {...}

@RequestMapping("/user/list")

public String list() {...}

}

@Controller

@RequestMapping("/user")**public class UserController {****@RequestMapping("/addform")**

public String addFrom() {...}

@RequestMapping("/add")

public String add() {...}

@RequestMapping("/delete")

public String delete() {...}

@RequestMapping("/list")

public String list() {...}

}

■ @RequestParam : 파라미터 매핑

- Http 요청 파라미터를 메소드 파라미터에 넣어주는 어노테이션

```
@Controller
public class Hello {

    @RequestMapping( value="/add", method=RequestMethod.GET )
    public String add( @RequestParam("age") int age, @RequestParam("name") String name ) {

        ....

    }
}
```

■ @RequestParam를 사용했을때는 파라미터가 반드시 있어야 한다

- 없으면 HTTP 400 - Bad Request (age가 있을수도, 없을수도 있을경우)

```
public String add( @RequestParam( value = "age", request=false, defaultValue="-1") int age,
                   @RequestParam("name") String name ) {

    ....

}
```


■ @ModelAttribute

- Http 요청 파라미터를 객체에 담을때 사용

```
public class UserVo {  
    private age;  
    private name;  
    //getter setter    → 생략  
}
```

```
@Controller  
public class User {  
  
    @RequestMapping( value="/user/join", method=RequestMethod.POST)  
    public String add( @ModelAttribute UserVo userVo ) {  
  
        ....  
    }  
}
```

■ @PathVariable

- URL에 쿼리스트링 대신 URL패스로 풀어쓰는 방식

```
@Controller
public class User {

    @RequestMapping( value="/board/view/{no}" )
    public String view( @PathVariable("no") int no ) {
        ....
    }
}
```

cf)
value="/board/view/10"

- ex)
/board/view?no=10
/board/view/10

- ex)
blog.naver.com/aaa
blog.naver.com/bbb

02 DispatcherServlet → controller

■ 메소드 파라미터

- HttpServletRequest, HttpServletResponse
- HttpSession

chapter02

주요 어노테이션 과 Context

1. 스프링의 DispatchedServlet 과 MVC
2. DispatcherServlet → controller
(주요 Annotation)
3. **controller → DispatcherServlet**

■ Model model

- 모델정보를 담을 수 있는 오브젝트 전달

```
@RequestMapping( value="/model")  
public String model(Model model) {  
    String str = "model에서 보낸 문자열";  
    model.addAttribute("str", str);  
    return "/WEB-INF/views/index.jsp";  
}
```

■ view

```
@RequestMapping( value="/model")
public String model(Model model) {
    String str = "model에서 보낸 문자열";
    model.addAttribute("str", str);
    return "/WEB-INF/views/index.jsp";
}
```

■ redirect

```
@RequestMapping( "/board/insert")
public String model( @ModelAttribute UserVo userVo) {

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