**ㄴ11/7 김민식 OJT Day 04**

**Spring FrameWork 기본**

스프링 10 – AOP(Aspect Oriented Programming)-II

참고영상 - [신입SW인력을 위한 실전 자바(Java) 스프링(Spring) 동영상과정 제 10강 AOP-II - YouTube](https://www.youtube.com/watch?v=x4qWZpDN444&list=PLPvokKzUkaLeiLUDKpqRdP8jtLU8lLKgR&index=10)

* 목차
  + @Aspect를 이용한 AOP구현
  + AspectJ Pointcut 표현식

10-1. @Aspect를 이용한 AOP구현

* 작업순서

1. 의존 설정(pom.xml)
2. @Aspect 어노테이션을 이용한 Aspect클래스 제작
3. XML파일에 <aop:aspectj-autoproxy /> 설정

10-2. AspectJ Pointcut 표현식

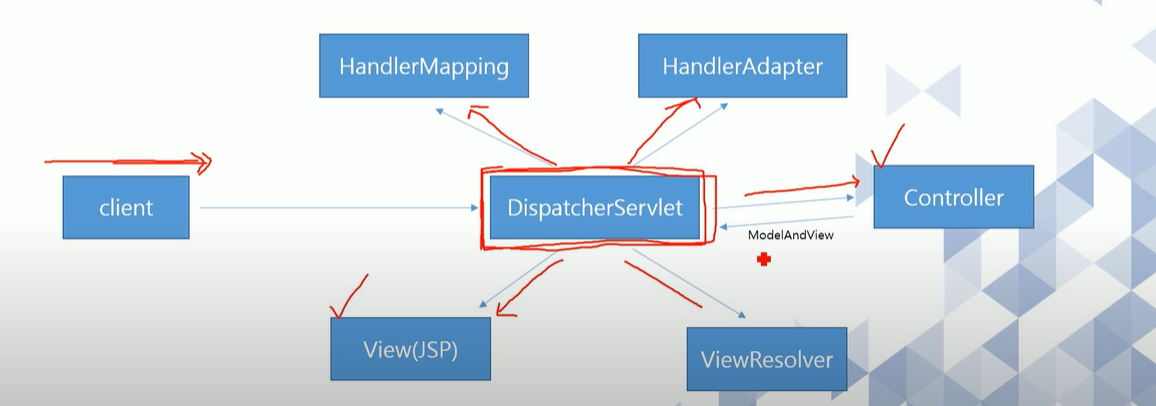
* + \* : 모든
  + . : 현재
  + .. : 0개 이상
* Execution
* Within
* Bean

스프링 11– 스프링 MVC 기초

참고영상 - [신입SW인력을 위한 실전 자바(Java) 스프링(Spring) 동영상과정 제 11강 MVC기초 - YouTube](https://www.youtube.com/watch?v=6P9C3myUsrw&list=PLPvokKzUkaLeiLUDKpqRdP8jtLU8lLKgR&index=11)

* 목차
  + 스프링 MVC 개요
  + 스프링 MVC 구조 살펴보기
  + Resources 폴더

11-1. 스프링 MVC 개요



11-2. 스프링 MVC 구조 살펴보기

* + New -> Spring project -> Spring MVC Project –> 기본 패키지 설정
* 뷰(.jsp)
* 컨트롤러 - Dispatcher에서 전달된 요청을 처리
* Servlet-context.xml – 스프링 컨테이너 설정 파일
* Web.xml - DispatcherServlet 서블릿 맵핑, 스프링 설정 파일 위치 정의
* DispatcherServlet – 클라이언트의 요청을 최초 받아 컨트롤러에게 전달

스프링 12 – 컨트롤러

참고영상 - [신입SW인력을 위한 실전 자바(Java) 스프링(Spring) 동영상과정 제 12강 컨트롤러 - YouTube](https://www.youtube.com/watch?v=MDFEmeuTuv4&list=PLPvokKzUkaLeiLUDKpqRdP8jtLU8lLKgR&index=12)

* 목차
  + 컨트롤러 클래스 제작
  + 요청 처리 메소드 제작
  + 뷰에 데이터 전달
  + 클래스에 @RequestMapping 적용

12-1. 컨트롤러 클래스 제작

* + 컨트롤러 클래스 제작 순서

1\_ @Controller를 이용한 클래스 생성

2\_ @RequestMapping을 이용한 요청 경로 지정

3\_ 요청 처리 메소드 구현

4\_ 뷰 이름 리턴

12-2. 요청 처리 메소드 제작

@RequestMapping(“board/view”) 🡨-------------요청 경로(path)

Public String view(){

return “board/view”; 🡨---------- 뷰페이지 이름

}

* + 뷰페이지 이름 생성(조합) 방법
    - Servlet-context.xml -> 뷰페이지 이름 = prefix + 요청처리 메소드 반환값 + seffix

12-3. 뷰에 데이터 전달

* Model 클래스를 이용한 데이터 전달

@RequestMapping(“board/content”)

Public String content(Model model){ 🡨------------- Model 객체를 파라미터로 받음

model.addAtrribute(“id”,30); 🡨------------- Model 객체에 데이터를 담음

return “board/view”;

}

* ModelAndView 클래스를 이용한 데이터 전달

@RequestMapping(“board/reply”)

Public ModelAndView reply(){

ModelAndView mv = new ModelAndView(); 🡨----- ModelAndView 객체 생성

mv.adObject(“id,30); 🡨---------------- Model 객체에 데이터를 담음

mv.setViewName(“board/reply”); 🡨------ 뷰이름 설정

return mv;

}

12-4. 클래스에 @RequestMapping적용

@Controller

@RequestMapping(“/board”) 🡨 클래스에 @RequestMapping 적용

Public class HomeController{

+

@RequestMapping(“/write”) 🡨 메소드에 @RequestMapping 적용

public String write(Model model){

// 조합된 요청 경로 : /board/write

스프링 13 – Form 데이터

참고영상 - [신입SW인력을 위한 실전 자바(Java) 스프링(Spring) 동영상과정 제 13강 form 데이터 - YouTube](https://www.youtube.com/watch?v=g4pTP69HrNw&list=PLPvokKzUkaLeiLUDKpqRdP8jtLU8lLKgR&index=13)

* 목차
  + HttpServleRequest 클래스
  + @RequestParam 어노테이션
  + 데이터(커맨드) 객체
  + @PathVariable

13-1. HttpServletRequest 클래스

*@RequestMapping*("/member/memberView")

public String viewMember(HttpServletRequest httpServletRequest ,Model model) {

String id = httpServletRequest.getParameter("id");

String pwd = httpServletRequest.getParameter("pwd");

model.addAttribute("id", id);

model.addAttribute("pwd", pwd);

return "/member/memberView";

}

13-2. @RequestParam 어노테이션

*@RequestMapping*("member/confirm")

public String memberConfirm(*@RequestParam*("id") String id, *@RequestParam*("pw") String pw,Model model) {

model.addAttribute("identify", id);

model.addAttribute("password", pw);

return "member/confirm";

}

13-3. 데이터(커맨드) 객체  
- 데이터(커맨드) 객체를 이용하여 데이터가 많을 경우 간단하게 사용 가능

*@RequestMapping*("/join/formOk")

public String join(Member member) {

return "join/formOk";

}

13-4. @PathVariable

*@RequestMapping*("/student/{studentId}")

public String getStudent(@PathVariable String studentId, Model model) {

model.addAttribute(“studentId”, studentId);

return "join/formOk";

}

* + @PathVariable 어노테이션을 이용하면 경로(path)에 변수를 넣어 요청메소드에서 파라미터로 이용 가능

스프링 14 – @RequestMapping 파라미터

참고영상 - [신입SW인력을 위한 실전 자바(Java) 스프링(Spring) 동영상과정 제 14강 @RequestMapping 파라미터 - YouTube](https://www.youtube.com/watch?v=dlqoM31YPJ8&list=PLPvokKzUkaLeiLUDKpqRdP8jtLU8lLKgR&index=14)

* 목차
  + Get방식과 Post방식
  + @ModelAttribute
  + 리다이렉트(redirect:) 키워드

14-1. @RequestMapping 에서 Get방식과 Post방식

* + Get방식과 Post방식을 구분 가능

*@RequestMapping(method = RequestMethod.GET, value = “/student”)*

*Public String goStudent (HttpServletRequest httpServletRequest, Model model){*

*String id = httpServletRequest.getParameter(“id”);*

*Model.addAttribute(“studentId, id);*

*Return “student/studentId”;*

* Get과 POST가 컨트롤러와 form 태그의 메소드가 동일 해야한다.

14-2. @ModelAttribute

*@RequestMapping(“/student”)*

*public String studentView(@ModelAttribute(“studentInfo) StudentInformation studentInformation){*

*return “studentView”;*

*}*

* @ModelAttribute 어노테이션을 이용하면 커맨드 객체의 이름을 개발자가 변경 가능

14-3. 리다이렉트 (redirect: 키워드)

*@RequestMapping(“/studentConfirm”)*

*Public String studentRedirect(HttpServletREquest httpServletRequest, Model model){*

*String id = httpServletRequest.getParameter(“id”)*

*If(id.equals(“abc”)){*

*Return “redirect:studentOk”;*

*}*

*Return “redirect:studentNg”;*

*}*

* 다른 페이지로 이동할 때 사용

스프링 15 – 폼 데이터 값 검증

참고영상 - [신입SW인력을 위한 실전 자바(Java) 스프링(Spring) 동영상과정 제 15강 폼 데이터 값 검증 - YouTube](https://www.youtube.com/watch?v=h0yhYSjA9JI&list=PLPvokKzUkaLeiLUDKpqRdP8jtLU8lLKgR&index=15)

* 목차
  + Validator를 이용한 검증
  + ValidationUtils 클래스
  + @Valid와 @IntiBinder

15-1. Validator를 이용한 검증

* 폼에서 전달 되는 데이터를 커맨드 객체에 담아 컨트롤 객체에 전달 할 때 커맨드 객체의 유효성 검사 가능
* Javascript를 이용하는 건 클라이언트에서 검사, Validator 이용은 서버에서 검사

15-2. ValidationUtils 클래스

* Validate()메소드를 좀더 편리하게 사용할 수 있도록 고안된 클래스
* Ex) ValidationUtils.rejectIfEmptyOrWhitespace(erros, “name”, “trouble”);

15-3. @Valid와 @InitBinder

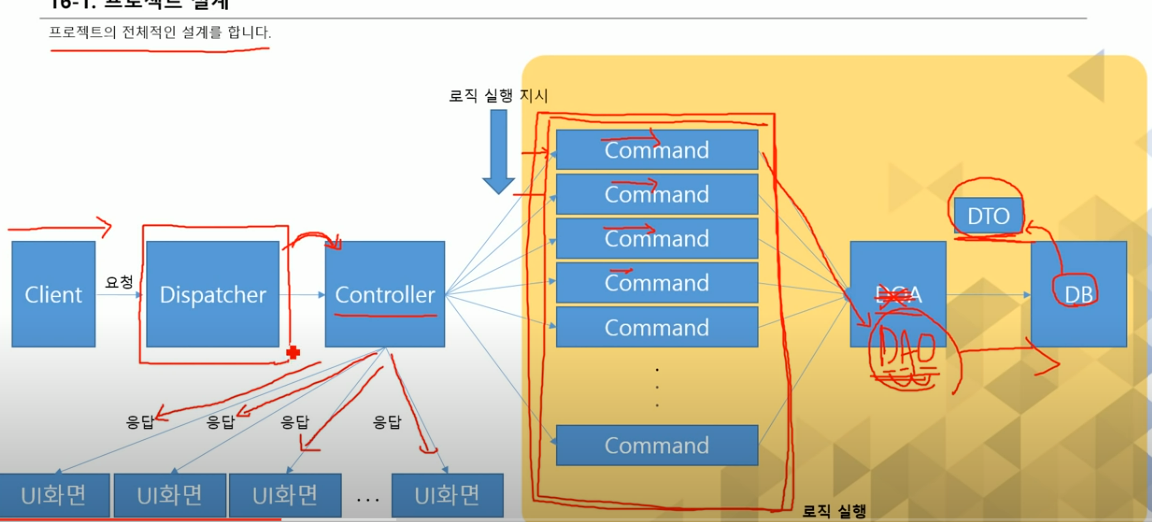
* validate()메소드를 직접 호출하지 않고 스프링 프레임워크에서 자동으로 호출
* 의존 추가
* @InitBinder 추가
* @Valid 추가

스프링 16 – 스프링MVC 게시판-I

참고영상 - [신입SW인력을 위한 실전 자바(Java) 스프링(Spring) 동영상과정 제 16강 스프링MVC 게시판-I - YouTube](https://www.youtube.com/watch?v=RYqlq1F9Jb8&list=PLPvokKzUkaLeiLUDKpqRdP8jtLU8lLKgR&index=16)

* 목차
  + 프로젝트 설계
  + Database 구축
  + 프로젝트 생성

16-1. 프로젝트 설계



16-2. DataBase 구축

질문사항

1. 코드 리뷰