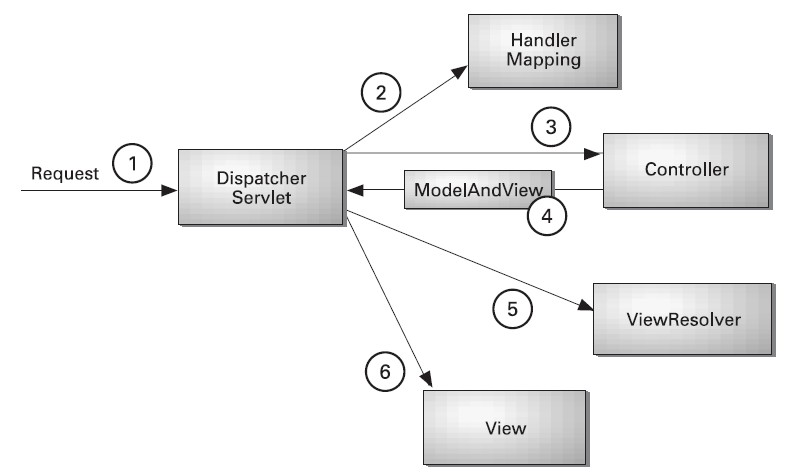


**spring MVC에서 하나의 요청에 대한 Life Cycle**

****

/hello.mvc

요청

/service.mvc

xxxController

web.xml

/ ->

ModelAndView

DispatcherServlet

View 파일

결과 리턴

servlet-context.xml

**mvc**

**@Controller**

**@RequestMapping**

**@RequestParam**

**@ModelAttribute**

**===========================**

**@RestController**

**@ResponseBody**

**- 컨트롤러 메소드 리턴타입**

**ModelAndView: 모델+뷰(명시또는url자동)**

**addObject("m1", 객체)**

**addObject("m2", 객체)**

**//setViewName("a");**

**String : model 없고 뷰(명시)**

**void : model 없고 뷰(url 자동)**

**Map : 모델 (뷰(url 자동))**

**- 컨트롤러 메소드 매개변수타입**

**요청-servlet api-model-view-응답**

**1> HttpServletRequest 등 서블릿 api**

**request.getParameter("id")**

**2> String id , int, double, boolean 타입 + 변수명(=요청파라미터명): 요청파라미터값 자동 저장**

**3> @RequestParam("i") String j , int, double, boolean 타입 + 변수명(=요청파라미터명): 요청파라미터값 자동 저장**

**4> 요청파라미터갯수 많으면 VO 객체 정의하고**

**@ModelAttribute("m") XXXVO v**

**4-1. 요청파라미터명과 xxxVO 멤버변수명 동일 자동 저장**

**4-2. XXXVO 객체는 자동 MODEL – JSP 전달**

**web.xml(annotation, xml 설정)**

**/ (현재 컨텍스트의 모든 url) ==> DispatcherServlet(스프링api)**

**post (한글) utf-8 ===> xxxEncodingFilter(스프링api)**

**servlet-context.xml**

**<annotation-driven />**

**===> mvc annotation 설정 객체 생성**

**@Controller @RequestMapping**

**<context:component-scan base-packages=”” />**

**==>@Component, @Service, @Repository, @Autowired**

**WEB-INF/views/a.jsp**

**@Controller**

**class HelloController**

**@Autowired**

**HelloDAO vo;**

**@RequestMapping("/hello")**

**handleRequest(){**

**MODEL,VIEW**

**}**

**hello.jsp**

**@Repository**

**HelloDAO**

**crud : 프로그램 기능 제공**

**조회/생성/수정/삭제"기본필요기능"**

**board 테이블**

**============================**

**-html**

**-jquery**

**-sql**

**-java**

**- ajax 통신 스프링 방식**

**-ajax**

**-jquery**

**- \*.jsp**

**\*.jsp--> web-inf/views/**

**/ ---> DispatcherServlet – contorller – model – view**

**컨트롤러 매핑 없다**

**webapp\resources\\*.png, \*.mp3, \*mp4, jquery.js**

**<script src="/mvc/resources/jquery.js></srcript>**

**<script >javascript / jquery</srcript>**

**1. http 통신**

**요청 1 –서버 처리중 – 응답 1(view)-요청2**

**동기화**

**비동기화 통신 jquery 방식 신규**

**:asynchronous javascript and xml**

**xml : <user>**

**<name>홍길동 </name>**

**<age> 21 </age>**

**</user>**

**json : javascript object notation**

**var obj = { name : "홍길동", age:21};**

**자바언어:String**

**"{ \"name\" : \"홍길동\", \"age\":21}";**

**"{ \"변수명1\" : \"값1\" , \"변수명2\":\"값2\" }";**

**ajax**

**1.요청1 – 서버 처리 - ......**

**2. 클라이언트 동시 브라우저 화면 구성**

**<div > </div>**

**3. 응답 2번 구성 태그에 포함**

**<div > 응답 </div>**

**4. 요청 – 응답 주고 받는 데이터 형식 : json**

**요청 1 – 요청2-요청3-서버 처리중 – 응답 3,2,1(view)-**

**2/10 – 스프링 종료**

**2/11 – 스프링 시험 / 세미프로젝트**

**나 : BoardController**

**짝굼:MemberController**

|  |  |
| --- | --- |
| **ajax 클라이언트**  **$.ajax(**  **{**  **url: '/mvc/test'**  **data: {"id" : "spring" }**  **type: 'get'**  **dataType:'json' success:function(server){**  **server<=**  **"{\"status\":true}";**  **$("div").html**  **(server.status):**  **}**  **);**  **<script>**  **<div></div>** | **ajax 서버**  **@Controller**  **class A{**  **@RequestMapping("/test")**  **@ResponseBody String a(String id){**  **//처리**  **//결과 응답**  **return "{\"status\":true}"; }** |