-spring : 사용자 -> 주소 -> 주소메소드(<-RequestMapping) -> view경로 -> java실행 -> html -> 사용자. resoures로 들어올 경우 예외로 주소메소드 사용X(파일 경로 직접 사용 - servelet-context.xml에 정의되어 있음)

-webapp이 웹루트

|  |
| --- |
| <form action="test03" id="actionForm" method="post">  <input type="number" name="num" />  <input type="text" name="a" />  <input type="checkbox" name="b" value="1" />  <input type="checkbox" name="b" value="2" />  <input type="button" value="전송" id="sendBtn" /> </form> |

-넘어오는 데이터 처리  
1) HttpServletRequest

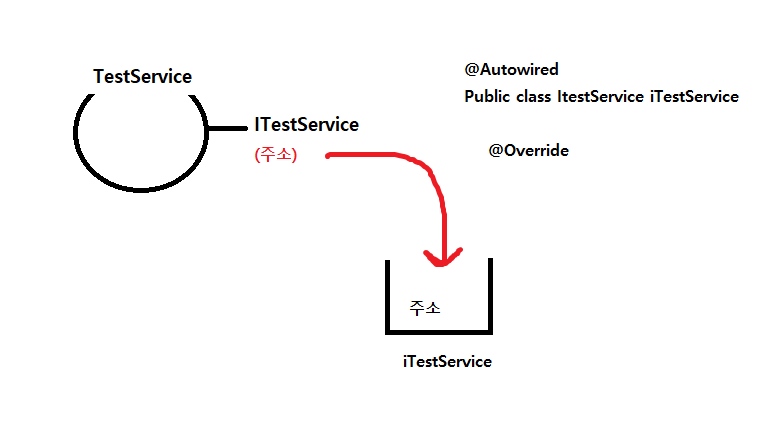
|  |
| --- |
| @RequestMapping(value="/test03") public ModelAndView test03(HttpServletRequest req, ModelAndView mav) {  int num = Integer.parseInt((String) req.getParameter("num"));  -> 외부에서 넘어온 키의 값을 num 변수에 담는 과정 |

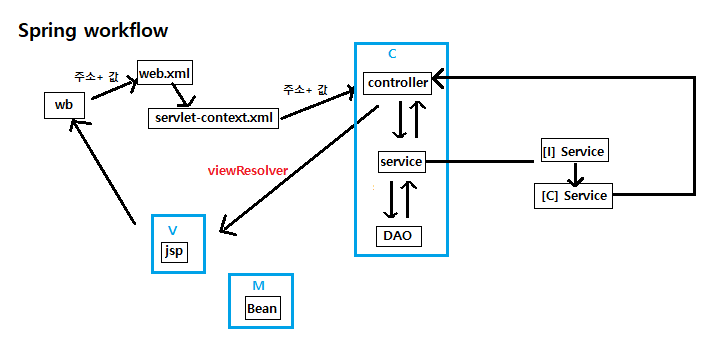
2) @RequestParam : 넘어오는 데이터를 자동으로 담아줌

|  |
| --- |
| @RequestMapping(value="/test03")  public ModelAndView test03(@RequestParam String a,  @RequestParam HashMap<String, String> params  @RequestParam (value="a") String o,  @RequestParam (value="b") List<Integer> check,  ,  ModelAndView mav) { |

**1) @RequestParam 타입 변수명** : 변수명과 동일한 키값을 가진 데이터를 해당 변수에 넣어준다.  
**2) @RequestParam 맵 변수명** : 맵에 들어오는 키와 값을 자동으로 넣어준다. **자주씀.**  
**3) @RequestParam(value=키) 타입 변수명** : 해당 키와 동일한 키를 가진 데이터를 해당 변수에 넣어준다.  
**4) @RequestParam(value=키) 목록형 변수명** : 해당 키와 동일한 키를 가진 데이터들을 해당 목록형에 자동으로 넣어준다. **자주씀.**

-DI(Dependency Injection) : 의존성 주입  
객체는 스프링이 만들어서 가지고 있음 -> 인터페이스를 이용하여 이름이 일치하는 것을 가져다 씀. 이름으로 부른 후 작업이 끝나면 사라짐.(필요할 때 불러서 사용)  
관계를 느슨하게 해서 특정 부분만 문제가 있어도 나머지는 정상 동작(안정성 확보 위해). 인터페이스끼리만 관계를 맺음.

Autowired를 이용해서 연결. ITestService iTestservice;  
@Autowired : 바로 아래 있는 인터페이스형에 맞는 객체를 주입.  




-Maven : 자동 라이브러리 관리 및 배포 지원.