Spring MVC

| 1 | 1.다음 설명 중 틀린 것은?  (1)스프링으로 구현한 MVC 프레임웍을 Spring MVC라 한다.  (2)모든 요청을 먼저 받는 front 컨트롤러 패턴이 적용된 서블릿을 스프링 MVC에서는 DispatcherServlet이라  한다  (3)WebApplicationContext 객체는 DispatcherServlet에 의해 생성된다.  (4)HandlerMapping 에 의해 컨트롤러의 메서드가 실행된다. |
| --- | --- |
| 2 | 다음 중 틀린 것은?  (1)HandlerMapping 에 의해 컨트롤러가 DispatcherServlet 에게 반환된다.  (2)DispatcherServlet은 검색 결과로 반환된 컨트롤러의 메서드를 직접 호출하여 컨트롤러로 하여금 요청을  수행하게 한다.  (3)스프링MVC에서는 컨트롤러가 될 수 있는 클래스에 제한이 없으므로, 일반 POJO 로도 컨트롤러를 정의  할 수 있다.  (4)위(3)번 처럼 POJO로 컨트롤러를 정의하면 DispatcherServlet이 컨트롤러가 보유한 메서드명을 알 수 없  기 때문에 메서드 호출은 불가능하다. |
| 3 | 다음 설명 중 틀린 것은?  (1)스프링MVC에서는 컨트롤러 역할을 수행하는 객체를 가리켜 Handler라 한다.  (2)HandlerMapping은 Handler 역할을 수행할 수 있는 객체를 찾기 위한 검색 객체이다.  (3)Handler를 찾기 위한 검색 방법에 따라 총 5가지 유형의 검색 객체가 지원된다.  (4)디폴트로 이미 등록된 HandlerMapping 은 BeanNameUrlHandlerMapping, SimpleUrlHandlerMapping  이다  (5)디폴트로 등록된 객체들은 스프링 빈 설정 파일에 명시할 필요가 없다. |
| 4 | 다음 설명 중 틀린 것은?  (1)스프링MVC에서 컨트롤러가 될 수 있는 객체는 총 4가지 유형이 있다.  (2)Servlet 도 컨트롤러 역할을 수행할 수 있도록 지원한다.  (3)HttpRequestHandler 객체도 컨트롤러의 역할을 수행할 수 있다.  (4)Controller 인터페이스를 구현한 객체도 컨트롤러 역할을 수행할 수 있다.  (5)일반 클래스는 컨트롤러 역할을 수행할 수 없다. |
| 5 | 다음 설명 중 틀린 것은?  (1)일반 클래스 즉 POJO 로도 컨트롤러의 역할을 수행시킬 수 있다.  (2)일반 클래스가 컨트롤러 역할을 수행하려면 @Controller 어노테이션을 명시하면 된다.  (3)일반 클래스로 컨트롤러를 정의하면, 메서드명은 개발자가 임으로 정의하므로 DispatcherServlet이 메서  드명을 알 수 없으며 따라서 호출도 불가능하다.  (4)스프링은 컨트롤러의 유형에 따라 1:1 대응되는 어댑터를 지원하는데, 즉 총 4가지 유형의 컨트롤러마다  지원되는 어댑터가 다르다.  (5)HandlerAdapter 로 인해, 컨트롤러로 정의될 수 있는 객체의 유형은 다양해 질 수 있다. |
| 6 | 빈 칸을 채우세요   | **컨트롤러 유형** | **지원 HandlerAdapter** | | --- | --- | | Servlet | SimpleServletHandlerAdapter | | httpRequestHandler | HttpRequestHandlerAdapter | | Controller 인터페이스 | SimpleControllerHandlerAdapter | | @Controller | AnnotationMethodHandlerAdapter | |
| 7 | 빈 칸을 채우세요   | **내용** | **Mybatis** | **Mybatis Spring** | | --- | --- | --- | | 라이브러리명 ( Jar 명 ) |  |  | | 쿼리 수행 객체 |  |  | | 쿼리 수행 객체 얻기 위한 컨테이너 |  |  | | 트랜잭션 처리 방법 |  |  | |
| 8 | 다음 설명 중 틀린 것은?  (1)컨트롤러는 요청을 처리 한 후 DispatcherServlet에게 뷰의 이름을 반환하게 된다.  (2)DispatcherServlet은 반환 받은 뷰의 이름을 직접 JSP로 가공하여 클라이언트에게 전송한다  (3)DispatcherServlet은 반환 받은 뷰의 이름만으로는 실제 뷰 페이지를 구성할 수 없으므로, ViewResolver  를 이용하여 View 객체를 생성 및 반환 받는다.  (4)View 객체는 JSP의 코드의 구성 및 실행을 담당하므로 View객체를 전달받은 DispatcherServlet 은 View  객체에 의해 생성된 최종 컨텐츠를 톰켓과 같은 웹컨테이너에 전달하고, 웹컨테이너는 바로 응답을 처리  한다.  (5)결국 ViewResolver 에 의해 최종적으로 반환되는 것은 View 객체이다 |
| 9 | 다음 설명 중 틀린 것은?  (1)컨트롤러는 언제나 ModelAndView를 반환해야 한다.  (2)컨트롤러가 ModelAndView를 반환하였을 경우엔 ViewResolver가 필요하다  (3)컨트롤러가 직접 View를 반환활 경우엔 ViewResolver가 필요없다.  (4)결국 ViewResolver가 필요한 경우는 뷰의 이름만으로는 실제 뷰페이지 구성이 불가능하기 때문에 이를  분석 및 완성하기 위해 해석자가 필요한 것이다. |
|  |  |