1. Spring MVC提供了基于注解的控制器开发模式，请说明一下注解的作用
2. @RestController：是一个组合注解，组合了@Controller和@ ResponseBody，当后端只返回数据时（如json），就可以使用此注解。
3. @PathVariable：用来接收路径参数，可以将URL中的值映射到方法参数中。
4. @RequestBody：可以配置在类和方法上，用来指定类或者方法是针对哪个URI的处理器。还可以根据请求参数、请求首部、media type等进一步缩小匹配范围
5. @ResponseBody：表示将返回值放在response体内，而不是返回一个页面。
6. @SessionAttribute：用在处理器的类上，用于在多个请求之间传递参数，类似于Session的Attribute，但不完全一样，一般来说@SessionAttribute设置的参数只用于暂时的传递，而不是长期的保存
7. @ModelAttribute：
8. 用于方法上时：通常用来在处理@RequestMapping之前，为请求绑定需要从后台查询的model；
9. 用于参数上时： 用来通过名称对应，把相应名称的值绑定到注解的参数bean上；要绑定的值来源于：

A） @SessionAttributes 启用的attribute 对象上；

B） @ModelAttribute 用于方法上时指定的model对象；

C） 上述两种情况都没有时，new一个需要绑定的bean对象，然后把request中按名称对应的方式把值绑定到bean中。

1. DispatcherServlet处理一个请求的流程

一个http请求到达服务器，被DispatcherServlet接收。DispatcherServlet将请求委派给合适的处理器Controller，此时处理控制权到达Controller对象。Controller内部完成请求的数据模型的创建和业务逻辑的处理，然后再将填充了数据后的模型即model和控制权一并交还给DispatcherServlet，委派DispatcherServlet来渲染响应。DispatcherServlet再将这些数据和适当的数据模版视图结合，向Response输出响应。

3. 面向Web开发的MVC框架种类繁多，请查阅资料对Spring MVC和其它一种MVC框架（如struct2）进行对比

a. 框架机制

spring mvc 和 struts2的加载机制不同：Spring MVC的入口是servlet，而struts2是filter

b. 拦截机制

Struts2框架是类级别的拦截，每次请求就会创建一个Action，和Spring整合时Struts2的ActionBean注入作用域是原型模式prototype（否则会出现线程并发问题），然后通过setter，getter吧request数据注入到属性。

Struts2中，一个Action对应一个request，response上下文，在接收参数时，可以通过属性接收，这说明属性参数是让多个方法共享的。Struts2中Action的一个方法可以对应一个url，而其类属性却被所有方法共享，这也就无法用注解或其他方式标识其所属方法了

SpringMVC是方法级别的拦截，一个方法对应一个Request上下文，所以方法基本上是独立的，独享request，response数据。而每个方法同时又和一个url对应，参数的传递是直接注入到方法中的，是方法所独有的。处理结果通过ModeMap返回给框架。在Spring整合时，SpringMVC的Controller Bean默认单例模式Singleton，所以默认对所有的请求，只会创建一个Controller，有应为没有共享的属性，所以是线程安全的。

c. 性能方面  
SpringMVC实现了零配置，由于SpringMVC基于方法的拦截，有加载一次单例模式bean注入。而Struts2是类级别的拦截，每次请求对应实例一个新的Action，需要加载所有的属性值注入，所以，SpringMVC开发效率和性能高于Struts2。

d. 配置方面  
spring MVC和Spring是无缝的。从这个项目的管理和安全上也比Struts2高，SpringMVC可以认为已经100%零配置。

e. 设计思想  
Struts2更加符合OOP的编程思想， SpringMVC就比较谨慎，在servlet上扩展。