1. **Struct2是类级别的拦截，一个类对应一个request上下文。**

**SpringMVC是方法级别的拦截，一个方法对应一个request上下文。**

而**方法同时对应一个url**，所以从架构上说，SpringMVC更容易实现restful url。但是Structs2实现则费劲一些，因为**Structs2中Action的一个方法可以对应一个url**，但是**类属性却被其它方法所共享**，这就**无法用注解或者其它的方式标识其所属的方法了**。

1. **SpringMVC方法之间相互独立，独享request和response数据，请求数据通过参数获取，处理结果通过ModelMap交回给框架，方法之间变量不共享**；而Structs2搞得比较乱，虽然**方法独立，但是其Action变量是共享的。**不影响程序运行，但是编码和读程序时却带来麻烦，每次来一个请求则创建一个Action，一个Action对象对应一个request上下文。
2. Structs2对每个request进行封装，把一个一个的request，session等servlet生命周期的变量封装成一个一个的Map，供给每个Action使用，并保证线程安全，比较耗内存。
3. 拦截器的实现机制上**，Struct2有自己的interceptor机制，而SpringMvc采用独立的AOP方式，**这样导致了Structs2的配置文件量比SpringMVC大。
4. **SpringMVC的入口是Servlet,而Structs2的是filter。**导致了二者的机制不同。
5. **SpringMVC集成了Ajax，[注解@RequestBody则可以实现，直接返回响应文本。而Structs2](mailto:注解@RequestBody则可以实现，直接返回响应文本。而Structs2)拦截器集成了Ajax，在Action处理要安装插件或者自己写代码集成进去，使用起来相对不方便。**
6. SpringMVC验证支持JSR303，处理起来相对更加灵活方便，而Struts2验证比较繁琐，感觉太烦乱。
7. **SpringMVC与Spring无缝连接，从这点看来在项目管理和安全性上比Structs2高很多。**
8. **在设计思想上，Structs2符合面向对象的编程思想，SpringMVC则在servlet上扩展。**
9. **SpringMVC开发性能和效率高于Structs2。**
10. **SpringMVC可以认为100%的零配置了。**