SSH和SSM定义

SSH 通常指的是 Struts2 做控制器(controller)，spring 管理各层的组件，hibernate 负责持久化层。

SSM 则指的是 SpringMVC 做控制器(controller)，Spring 管理各层的组件，MyBatis 负责持久化层。

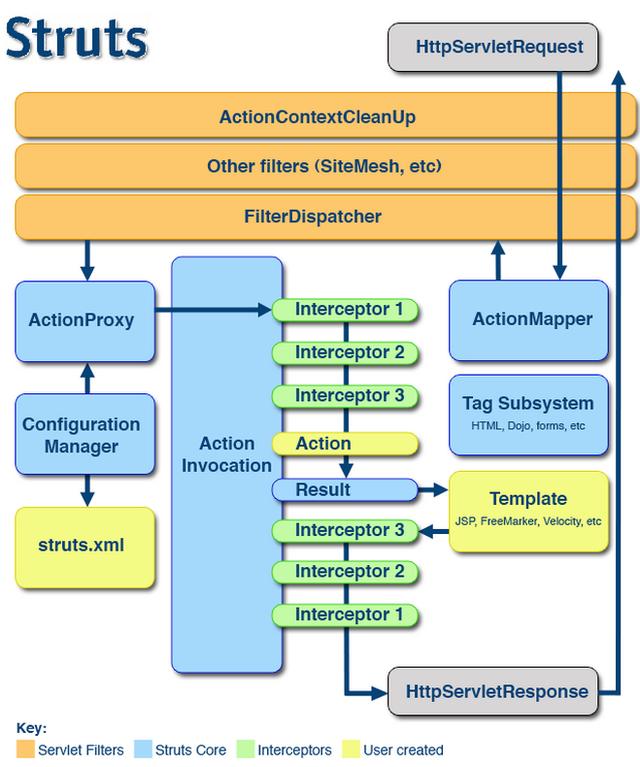
共同点：1.Spring依赖注入DI来管理各层的组件。2.使用面向切面编程AOP管理事物、日志、权限等。

不同点：1.Struts2 和 SpringMVC 控制器(controller)控制视图和模型的交互机制的不同，

Struts2是**Action类级别**，SpringMVC是**方法级别**，更容易实现RESTful风格。

SSH 和 SSM 的实现原理

1.Struts2 的实现原理



Struts2框架执行步骤（Struts2使用Filter嵌入）：

1、客户端初始化一个指向Servlet容器（例如Tomcat）的请求

2、这个请求经过一系列的过滤器（Filter）（这些过滤器中有一个叫做ActionContextCleanUp的可选过滤器，这个过滤器对于Struts2和其他框架的集成很有帮助

3、接着FilterDispatcher被调用，FilterDispatcher询问ActionMapper来决定这个请求是否需要调用某个Action

4、如果ActionMapper决定需要调用某个Action，FilterDispatcher把请求的处理交给ActionProxy

5、ActionProxy通过Configuration Manager询问框架的配置文件，找到需要调用的Action类

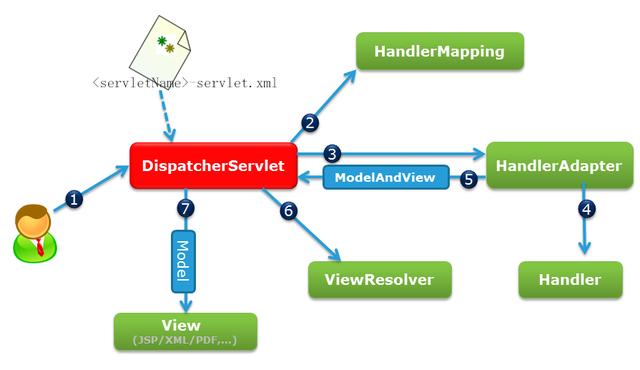
6、ActionProxy创建一个ActionInvocation的实例。

7、ActionInvocation**实例使用命名模式来调用**，在调用Action的过程前后，涉及到相关拦截器（Intercepter）的调用。

8、一旦Action执行完毕，ActionInvocation负责根据struts.xml中的配置找到对应的返回结果。**返回结果通常是（但不总是，也可 能是另外的一个Action链）一个需要被表示的JSP或者FreeMarker的模版。**

9、将处理结果返回给客户端

2.SpringMVC 的实现原理



SpringMVC框架执行步骤（SpringMVC使用Servlet嵌入）：

1、客户端发出一个http请求给web服务器，web服务器对http请求进行解析，如果匹配[DispatcherServlet](https://www.baidu.com/s?wd=DispatcherServlet&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/u013977285/article/details/_blank)的请求映射路径（在web.xml中指定），web容器将请求转交给DispatcherServlet.

2、DipatcherServlet接收到这个请求之后将根据请求的信息（包括URL、Http方法、请求报文头和请求参数Cookie等）以及HandlerMapping的配置找到处理请求的处理器（Handler）。

3-4、DispatcherServlet根据HandlerMapping找到对应的Handler,将处理权交给Handler（Handler将具体的处理进行封装），再由具体的HandlerAdapter对Handler进行具体的调用。

5、Handler对数据处理完成以后将返回一个ModelAndView()对象给DispatcherServlet。

6、Handler返回的ModelAndView()只是一个[逻辑视图](https://www.baidu.com/s?wd=%E9%80%BB%E8%BE%91%E8%A7%86%E5%9B%BE&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://blog.csdn.net/u013977285/article/details/_blank)并不是一个正式的视图，DispatcherSevlet通过ViewResolver将逻辑视图转化为真正的视图View。

7、Dispatcher通过model解析出ModelAndView()中的参数进行解析最终展现出完整的view并返回给客户端。

Hibernate 和 MyBatis 两种ORM框架对比

两者的相同点

Hibernate与MyBatis都可以是通过SessionFactoryBuider由XML配置文件生成SessionFactory，然后由SessionFactory 生成Session，最后由Session来开启执行事务和SQL语句。其中SessionFactoryBuider，SessionFactory，Session的生命周期都是差不多的。

Hibernate和MyBatis都支持JDBC和JTA事务处理。

两者各自优势

MyBatis可以进行更为细致的SQL优化，可以减少查询字段。

MyBatis容易掌握，而Hibernate门槛较高。

Hibernate的DAO层开发比MyBatis简单，Mybatis需要维护SQL和结果映射。

Hibernate对对象的维护和缓存要比MyBatis好，对增删改查的对象的维护要方便。

Hibernate数据库移植性很好，MyBatis的数据库移植性不好，不同的数据库需要写不同SQL。

Hibernate有更好的二级缓存机制，可以使用第三方缓存。MyBatis本身提供的缓存机制不佳，更新操作不能指定刷新指定记录，会清空整个表，但是也可以使用第三方缓存。

Hibernate 封装性好，屏蔽了数据库差异，自动生成SQL语句，应对数据库变化能力较弱，SQL语句优化困难。

MyBatis仅实现了SQL语句和对象的映射，需要针对具体的数据库写SQL语句，应对数据库变化能力较强，SQL语句优化较为方便。

总结

SSM和SSH不同主要在MVC实现方式，以及ORM持久化方面不同（Hiibernate与Mybatis）。SSM越来越轻量级配置，将注解开发发挥到极致，且ORM实现更加灵活，SQL优化更简便；而SSH较注重配置开发，其中的Hiibernate对JDBC的完整封装更面向对象，对增删改查的数据维护更自动化，但SQL优化方面较弱，且入门门槛稍高。

附Spring家族：

