# 组件

## 前端控制器（DispatcherServlet）

1. 主要用作职责调度工作，控制流程。
2. 如果请求是multipart，将通过MultipartResolver进行文件上传解析。
3. 通过HandlerMapping，将请求映射到处理器(返回一个HandlerExecutionChain，他包括一个处理器，多个HandlerInterceptor拦截器)
4. HandlerAdapter支持多种类型的处理器(HandlerExecutionChain中的处理器)，
5. 通过ViewResolver解析逻辑视图名到具体视图实现；

## 处理器映射器HandlerMapping

根据url去查找处理器





@Controller是注解信息，表示该类是一个控制器类,可以被注解的处理器适配器找到。

@RequestMapping作用是指定一个url与该方法绑定。

为了能让注解的处理器映射器找到Handler控制器，就是为了找到@Controller，需要在springmvc.xml做下配置。



## 处理器

就是@Controller注解的类，是负责代码编写处理业务逻辑的一个类。

## 处理器适配器

执行处理器

处理器映射器是根据映射关系查找处理器，也就是@Controller，处理器适配器就是执行@Controller，调用controller里面的方法，返回ModelAndView。

## 视图解析器

1. @RequestHeader的使用获取请求头信息
2. 请求参数为pojo，例如一个表单提交，自动分装到对应的pojo类中，作为请求参数传递到后台。

ModelAndView，就是将返回结果放到这个对象中就行返回，modelAndView.addObject(“userName”,”张三”);结果处理就会根据主键userName获取到他的值。

# SpringMvc和Struts2区别

1. SpringMvc的入口是servlet，而struts2是filter。
2. Spring会比struts快一些，springmvc是基于方法的设计，而struts是基于类，每次请求都会实例一个action，每个action都会被注入属性。而spring基于方法，粒度更细。Spring3mvc是方法级别的拦截，拦截到方法后根据参数上的注解，把request数据注入进去，在spring3mvc中，一个方法对应一个request上下文。而struts2架构是类级别的拦截，每次来了请求就创建一个Action，然后调用setter，getter方法把request中的数据注入。Struts2实际上是通过setter,getter方法与request打交道，一个Action对象对应一个request上下文。
3. Struts2在接收参数的时候，可以用属性接收参数，这就说明参数注解是让多个方法共享的。
4. Struts2更加符合oop的编程思想，springmvc比较谨慎，在servlet上扩展。



Springmvc是spring的一个模块，所以spring对于springmvc的控制器管理更加简单方便，而且提供了全注解的方式进行管理，各种功能的注解都比较全面，使用简单。而struts2需要采用xml很多的配置参数来管理。

Springmvc是方法级别的拦截，一个方法对应一个request上下文。而方法同时又跟一个url对应，所以说从架构本身上spring3mvc更容易实现restful url。

Springmvc使用注解@ResponseBody。会自动将对象转换为Json数据。

# 拦截器过滤器

Springmvc和核心控制器是servlet，而struts2是Filter。

## 过滤器

依赖于servlet容器，在实现上基于函数回调，他可以对几乎所有的请求进行过滤。但是过滤器实例只能在容器初始化时调用一次。

目的：用来做一些过滤操作，例如在Javaweb中，对传入的request，response提前过滤掉一些信息，或者提前设置一些参数，然后再传入servlet或者controller进行业务逻辑操作。

## 拦截器

依赖于web框架，在实现上基于java的反射，属于面向切面的一种运用，就是在service或者一个方法前调用一个方法，或者在一个方法后调用一个方法，比如动态代理就是拦截器的一种实现。一个拦截器实例可以在controller声明周期之内多次调用，但是缺点只能对controller请求进行拦截，对其它的一些比如静态资源的请求则没办法进行拦截处理。

