



①: DispatcherServlet是springmvc中的前端控制器(front controller),负责接收request并将request转发给对应的处理组件.

②: HanlerMapping是springmvc中完成url到controller映射的组件.DispatcherServlet接收request,然后从HandlerMapping查找处理request的controller.

③: Cntroller处理request,并返回ModelAndView对象,Controller是springmvc中负责处理request的组件(类似于struts2中的Action),ModelAndView是封装结果视图的组件.

④ ⑤ ⑥: 视图解析器解析ModelAndView对象并返回对应的视图给客户端.

二、springmvc的工作机制

在容器初始化时会建立所有url和controller的对应关系,保存到Map<url,controller>中.tomcat启动时会通知spring初始化容器(加载bean的定义信息和初始化所有单例bean),然后springmvc会遍历容器中的bean,获取每一个controller中的所有方法访问的url,然后将url和controller保存到一个Map中;

这样就可以根据request快速定位到controller,因为最终处理request的是controller中的方法,Map中只保留了url和controller中的对应关系,所以要根据request的url进一步确认controller中的method,这一步工作的原理就是拼接controller的url(controller上@RequestMapping的值)和方法的url(method上@RequestMapping的值),与request的url进行匹配,找到匹配的那个方法;

确定处理请求的method后,接下来的任务就是参数绑定,把request中参数绑定到方法的形式参数上,这一步是整个请求处理过程中最复杂的一个步骤。springmvc提供了两种request参数与方法形参的绑定方法:

① 通过注解进行绑定,@RequestParam

② 通过参数名称进行绑定.

使用注解进行绑定,我们只要在方法参数前面声明@RequestParam("a"),就可以将request中参数a的值绑定到方法的该参数上.使用参数名称进行绑定的前提是必须要获取方法中参数的名称,Java反射只提供了获取方法的参数的类型,并没有提供获取参数名称的方法.springmvc解决这个问题的是用asm框架

读取字节码文件,来获取方法的参数名称.asm框架是一个字节码操作框架,关于asm更多介绍可以参考它的官网.个人建议,使用注解来完成参数绑定,这样就可以省去asm框架的读取字节码的操作.

三、源代码的分析

我们根据工作机制中三部分来分析springmvc的源代码.

其一,ApplicationContext初始化时建立所有url和controller类的对应关系(用Map保存):

其二,根据请求url找到对应的controller,并从controller中找到处理请求的方法:

其三,request参数绑定到方法的形参,执行方法处理请求,并返回结果视图.