

## SpringMVC 运行流程分析

在整个 Spring MVC 框架中，**DispatcherServlet** 处于核心位置，负责协调和组织不同组件以完成请求处理并返回响应的工作

SpringMVC 处理请求过程：

- 1). 若一个请求匹配 **DispatcherServlet** 的请求映射路径(在 **web.xml** 中指定), **WEB** 容器将该请求转交给 **DispatcherServlet** 处理
- 2). **DispatcherServlet** 接收到请求后, 将根据请求信息(包括 **URL**、**HTTP** 方法、请求头、请求参数、**Cookie** 等)及 **HandlerMapping** 的配置找到处理请求的处理器(**Handler**). 可将 **HandlerMapping** 看成路由控制器, 将 **Handler** 看成目标主机。
- 3). 当 **DispatcherServlet** 根据 **HandlerMapping** 得到对应当前请求的 **Handler** 后, 通过 **HandlerAdapter** 对 **Handler** 进行封装, 再以统一的适配器接口调用 **Handler**。
- 4). 处理器完成业务逻辑的处理后将返回一个 **ModelAndView** 给 **DispatcherServlet**, **ModelAndView** 包含了视图逻辑名和模型数据信息
- 5). **DispatcherServlet** 借助 **ViewResolver** 完成逻辑视图名到真实视图对象的解析
- 6). 得到真实视图对象 **View** 后, **DispatcherServlet** 使用这个 **View** 对 **ModelAndView** 中的模型数据进行视图渲染

UML 时序图：

