

SpringMVC

注解式的SpringMVC项目

核心对象：DispatcherServlet

- 使用步骤
 - 1.加入依赖 ： spring-webmvc, servlet依赖
 - 2.在web.xml 注册DispatcherServlet
- 作用
 - 1.创建springmvc的容器对象， WebApplicationContext. 在创建容器对象的同时，读取配置文件把controller控制器对象都创建好。 web项目启动后，所有对象都创建了，准备好了
 - 2.它是一个servlet，能够接收用户的请求，并显示请求的处理结果

- 1.@Controller: 创建控制器对象
- 2.@RequestMapping: 请求映射，把请求指定给某个方法处理
 - 1.value：请求的uri地址，唯一值
 - 2.method：表示请求的方式，get，post。 RequestMethod枚举值
- 3.接收请求的参数
 - 1.逐个接收： 请求中参数名需要和控制器方法的形参名一样 —— 逐个接收中，如果名称不一样使用@RequestParam
 - 2.java对象接收参数：要求属性名和请求中参数名一样
 - 3.提交请求参数有中文乱码，使用字符集的过滤器， CharacterEncodingFilter
- 4.返回值，表示请求的处理结果
 - 1.ModelAndView：表示数据和视图
 - 2.String：表示视图
 - 3.void：可以响应ajax请求
 - 4.Object：对象，表示数据，响应ajax请求
 - 1.返回一个Student，转为json对象
 - 2.返回列表<Student>,转为的json array
 - 3.返回String，表示数据
 - 5.springmvc中控制器方法返回对象，转为json响应ajax步骤
 - 1.加入处理json的工具库。加入依赖，默认是jackson处理json
 - 2.springmvc配置文件，加入<mvc:annotation-driven>
 - 3.在控制器方法的上面，加入@ResponseBody
 - 6.控制器方法返回对象，转为json：使用的核心对象HttpMessageConveter和他的实现类
- 5.静态资源
 - 为什么要处理静态资源：中央调度器的url-pattern设置为 “/” 导致静态资源的请求交给了中央调度器而默认框架中是没有处理静态资源的控制器对象的，所以访问静态资源是404
 - 解决方式：增加控制器对象，让加入的控制器对象处理静态资源的请求
 - 1.在配置文件加入<mvc:default-servlet-hanlder>，<mvc:annotation-driven> —— 创建DefaultServletHttpRequestHandler这个控制器对象
 - 2.在配置文件加入<mvc:resources mapping="请求静态资源的uri地址" location="静态资源的目录" /> <mvc:annotation-driven> —— 创建ResourceHttpRequestHandler这个控制器对象
- 6.SSM整合
 - 1使用容器创建，管理对象
 - 1.springmvc容器：管理控制器对象，也就是管理视图层的对象
 - 2.spring容器：管理的service，dao对象，管理业务逻辑层，持久层的对象
 - 2.创建service，dao，controller对象时间
 - 1.在创建容器对象的同时，把配置文件中的所有对象都创建好
 - 2.在web.xml使用DispatcherServlet创建springmvc容器对象，同时创建Controller对象
 - 3.在web.xml文件使用ContextLoaderListener创建spring容器对象，同时创建service，dao对象
 - 3.配置文件
 - 1.springmvc配置文件 —— 定义controller，视图解析器，注解驱动等内容
 - 2.spring配置文件 —— 声明service，mybatis的数据源， SqlSessionFactory, dao代理对象，事务配置
 - 3.mybatis主配置文件
 - 1.别名定义
 - 2.mapper文件的位置
 - 4.数据库的属性配置文件 —— jdbc.properties —— 数据库驱动，url，username，password
- 7.转发和重定向
 - forward：转发 —— 语法 forward:视图完整路径
 - redirect: 重定向 —— 语法redirect:视图完整路径
- 8.异常处理
 - 框架采用的是全局异常处理的方式，把异常集中到一个位置处理
 - 1.@ControllerAdvice: 放在类上面，表示此类是给控制器类增加异常处理的
 - 2.@ExcetionHandler:放在方法的上面，表示此方法那能够处理异常
- 9.拦截器
 - 拦截用户的请求，可以截断请求
 - 定义步骤：
 - 1.创建类实现HandlerInterceptor,里面有三个方法
 - 2.在springmvc的配置文件，声明拦截器对象
 - 重要方法 —— preHandle()
 - true:请求是正确，可以通过拦截器验证，能够执行控制器方法
 - false：请求不能通过验证，请求被截断，不能处理
 - 常用来做什么
 - 1.登录验证，检查用户是否登录
 - 2.权限管理，用户访问一个地址，通过拦截器验证用户能够执行这个操作
 - 3.做日志记录
 - 拦截器和过滤器区别
 - 1.拦截器是springmvc框架中的对象， 过滤器是servlet中对象
 - 2.拦截器有三个方法，三个执行时间；过滤器就一个执行时间，请求之前
 - 3.拦截器对象是springmvc容器创建的，过滤器是tomcat服务器创建的
 - 4.拦截器是用来拦截请求的，对请求做验证的； 过滤器是侧重对request，response的参数，属性做设置值的
 - 5.拦截器对Controller的请求做拦截的；对jsp，html，css，js用过滤器