springMVC前端控制器

<servlet>

<servlet-name>springmvc</servlet-name> <servlet-class>oro.springframework.web.servlet.DispatcherServlet</servlet-class>

</servlet>

<!—contextConfigLocation配置springmvc加载的配置文件（配置处理器。适配器等等）。自己建的spring-mvc文件（如果不配置这个文件，则默认加载的是”/WEB-INF/servlet名称(springmvc-servlet.xml)”） -->

<init-param>

<param-value>contextConfigLocation</param-value>

<param-value>classpath:spring-mvc.xml</param-value>

</init-param>

spring拦截器配置

工具接口类

定义拦截器，提供三个方法

public class handlerInterceptor1 implements HandlerInterceptor{

//在进入handler方法之前执行

//用于身份认证和身份授权

//比如身份认证，如果认证不通过，表示用户没有登陆，需要此方法拦截，不再向下执行

@Override

public boolean preHandle(HttpServletRequest request,HttpServletResponse response,Object handler)throws Exception{

return false;

}

//进入handler方法之后，在返回modleAndView之前执行

//应用场景从ModleAndView出发：例如将公用的模型数据，在这里传到视图，也可以在这里同意指定视图，例如菜单导航，

@Override

public void postHandle(HttpServletRequest request,HttpServletResponse response,Object handler,ModleAndView modleAndView)throws Exception{

return false;

}

//执行handler完成执行此方法

//统一的异常处理，统一的日志处理

@Override

public void afterComplecation(HttpServletRequest request,HttpServletResponse response,Object handler,Exception ex)throws Exception{

return false;

}

}