防XSS组件

Pom 文件引用

<dependency>

<groupId>com.jd.market</groupId>

<artifactId>jzt-common-utils</artifactId>

<version>1.0.0-SNAPSHOT</version>

</dependency>

## 1、拦截器方式，web.xml里配置如下,

<filter>

<filter-name>XssEscape</filter-name>

<filter-class>com.jd.common.xss.XssFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>XssEscape</filter-name>

<url-pattern>/\*</url-pattern>

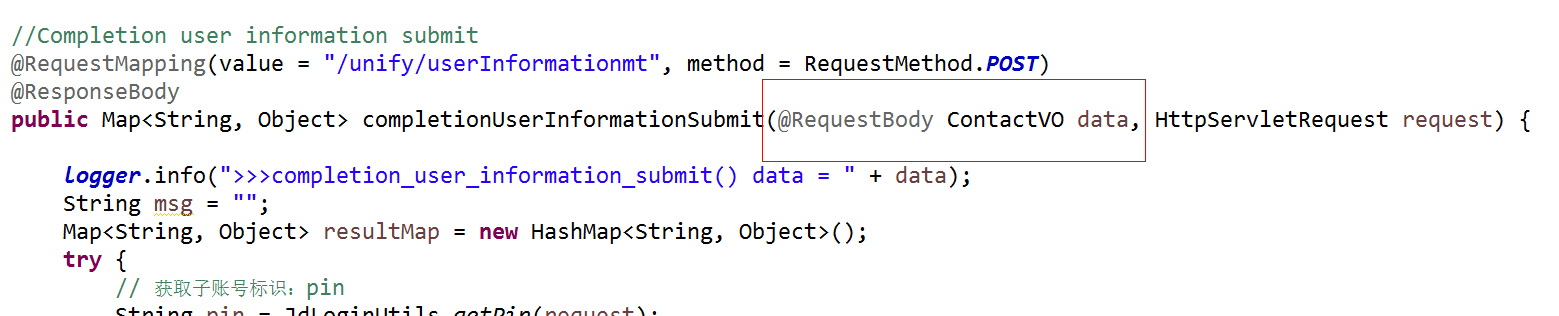
<dispatcher>REQUEST</dispatcher>

</filter-mapping>

这种拦截方式无法处理前台post JSON数据，需要使用第二种方式



后台采用@RequestBody接收参数



## 2、Spring mvc HttpMessageConverter 转换json格式

<mvc:annotation-driven>

<mvc:message-converters>

<!-- 将StringHttpMessageCOnverter的默认编码设为UTF-8 -->

<bean class="org.springframework.http.converter.StringHttpMessageConverter">

<constructor-arg value="UTF-8"/>

</bean>

<bean

class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter">

<property name="supportedMediaTypes">

<list>

<value>application/json;charset=UTF-8</value>

<value>text/html;charset=UTF-8</value>

</list>

</property>

<property name="objectMapper">

<bean class="com.jd.common.xss.CustomObjectMapper" />

</property>

</bean>

</mvc:message-converters>

</mvc:annotation-driven>

## 3、手工转换

XssEncodeUtils.xssEncode

如：

XssEncodeUtils.xssEncode(data.getUserName())