Shiro学习笔记--web

|  |  |
| --- | --- |
|  | 🢂 内容概览 |
|  | Why：此文档用来做什么？它存在的意义是什么？为解决什么问题？   |  | | --- | |  |   What：当前包含了那些内容？   |  | | --- | |  |   How：此文档应如何参考？   |  | | --- | |  |   Who：此文档适用于那些人员阅读参考？   |  | | --- | |  | |

目录

[1 基本配置 3](#_Toc444841286)

[1.1 shiro配置文件默认位置 3](#_Toc444841287)

[1.2 修改配置文件位置 3](#_Toc444841288)

[1.2.1 通过ServletContext参数配置 3](#_Toc444841289)

[1.2.2 通过InitParam参数配置 3](#_Toc444841290)

[1.2.3 直接在Web.xml书写配置 4](#_Toc444841291)

[1.3 shiro配置说明 4](#_Toc444841292)

[1.4 定制WebEnvironment 4](#_Toc444841293)

[2 [urls]配置 4](#_Toc444841294)

[2.1 定义规则 5](#_Toc444841295)

[2.2 顺序相关 5](#_Toc444841296)

[2.3 自定义过滤器 6](#_Toc444841297)

[2.4 默认过滤器 6](#_Toc444841298)

[2.5 启用/禁用过滤器 7](#_Toc444841299)

[3 会话管理 7](#_Toc444841300)

[3.1 cookie 8](#_Toc444841301)

[3.1.1 禁用cookie 8](#_Toc444841302)

[4 RememberMe 8](#_Toc444841303)

[5 JSP标签 8](#_Toc444841304)

# 基本配置

向web中引入shiro的最基本配置方式是在web.xml添加如下配置：

|  |
| --- |
| *<!--01 配置shiro监听-->* <**listener**>  <**listener-class**>org.apache.shiro.web.env.EnvironmentLoaderListener</**listener-class**> </**listener**>  *<!--02 配置shiro过滤器-->* <**filter**>  <**filter-name**>ShiroFilter</**filter-name**>  <**filter-class**>org.apache.shiro.web.servlet.ShiroFilter</**filter-class**> </**filter**>  *<!--03 配置shiro过滤器规则-->* <**filter-mapping**>  <**filter-name**>ShiroFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  <**dispatcher**>REQUEST</**dispatcher**>  <**dispatcher**>FORWARD</**dispatcher**>  <**dispatcher**>INCLUDE</**dispatcher**>  <**dispatcher**>ERROR</**dispatcher**> </**filter-mapping**> |

## shiro配置文件默认位置

配置中未指定shiro配置文件的位置，则shiro默认会从如下两个位置查找，以找到的第一个shiro.ini作为配置文件；

* WEB-INF/shiro.ini
* classpath\*:shiro.ini

## 修改配置文件位置

### 通过ServletContext参数配置

如果需要修改配置文件位置，则可以在web.xml配置文件中添加ServletContext参数配置；

|  |
| --- |
|  |

配置文件路径格式样例：

|  |
| --- |
| /WEB-INF/some/path/shiro.ini  file:/home/foobar/myapp/shiro.ini  classpath:com/foo/bar/shiro.ini  url:http://confighost.mycompany.com/myapp/shiro.ini |

### 通过InitParam参数配置

|  |
| --- |
|  |

### 直接在Web.xml书写配置

|  |
| --- |
|  |

## shiro配置说明

1. EnvironmentLoaderListener：初始化WebEnvironment实例，存放shiro所需要的所有实例，包括SecurityManager；并将其放入到ServletContext中；如果需要获取WebEnvironment实例，则可以通过WebUtil.getRequiredWebEnvirotment(servletContext)方法获取；
2. ShiorFilter：ShiroFilter将依照filter-mapping配置的规则过滤请求，并使用WebEnvironment处理请求；

注：为了使shiro能够更好的过滤请求，建议将ShiroFilter过滤器定义到所有过滤器的最前面；

## 定制WebEnvironment

shiro默认使用基于INI配置文件的IniWebEnvironment；如果需要定制，则可以在web.xml配置ServletContext变量，修改默认的WebEnvironment；

|  |
| --- |
|  |

# [urls]配置

|  |
| --- |
|  |

## 定义规则

shiro可以在urls中，定义过滤器。相比web.xml中配置过滤器而言，shiro的配置更加简明，灵活；配置格式如下：

|  |
| --- |
| 过滤url = 过滤器名称1[过滤器规则参数][,过滤器名称2[过滤器规则参数],…] |

示例：

|  |
| --- |
|  |

**说明：**

1. 过滤器规则可以定义多条，每条可以匹配不同的url；
2. 过滤器名称可以配置shiro默认提供的过滤器，也可以在[main]中自定义过滤器规则；
3. 过滤器名称后面的规则参数，根据不同的过滤器类型而不同；当没有参数时，可以去掉[]；

## 顺序相关

过滤器定义规则与在INI文件中定义的顺序相关，先定义个过滤器将被先调用；对于映射相同请求的过滤器规则，遵循就近原则；

示例：将匹配第一条规则

|  |
| --- |
| /account/\*\* = ssl, authc  /account/signup = anon |

## 自定义过滤器

如果用户想实现自定义的过滤器，则需要实现PathMatchingFilter接口；配置示例：

|  |
| --- |
|  |

## 默认过滤器

shiro提供了一些默认过滤器供用户使用，shiro为这些过滤器提供了默认的名称，可以根据名称在配置文件中使用；默认过滤器名称定义在DefaultFilter枚举类中；

|  |
| --- |
|  |

如下为默认过滤器及对应的类：

|  |
| --- |
|  |

## 启用/禁用过滤器

shiro所有过滤器都继承自OncePerRequestFilter抽象类，该类增加了enabled的属性及相关setter/getter方法；新增的过滤器可以通过重写isEnabled方法来改变启用/禁用规则；

在INI配置文件中可以配置是否启用过滤器，默认为启用；

示例：

|  |
| --- |
|  |

# 会话管理

在web环境中，shiro的SessionManager默认实现为ServletContainerSessionManager。该类衔接了web和shiro的session；使用默认配置即可以正常运行；但是缺乏对特定功能，比如session集群的支持；

为了实现Session功能的扩展，shiro提供了本地化Session；针对web实现的本地化session类为：DefaultWebSessionManager；

|  |
| --- |
|  |

## cookie

DefaultWebSessionManager实现了如下两个方法：

* sessionIdCookieEnabled (a boolean)：使session生效；
* sessionIdCookie, a [Cookie](http://shiro.apache.org/static/current/apidocs/org/apache/shiro/web/servlet/Cookie.html) instance：Cookie实例；

Cookie默认实现类为SimpleCookie；

### 禁用cookie

|  |
| --- |
|  |

# RememberMe

当使用的AuthenticationToken实现了RememberMeAuthenticationToken接口后，就可以提供RmemeberMe功能；UsernamePasswordToken类已经实现了该接口；

|  |
| --- |
|  |

## 使用RememberMe功能

### 编程方式

|  |
| --- |
|  |

### 登录表单方式

对于web项目而言，认证过滤器的实现类默认为FormAuthenticationFilter，默认使用username，password，rememberMe作为是认证请求参数名；

|  |
| --- |
|  |

|  |
| --- |
|  |

可以在INI文件中配置认证请求参数名；

|  |
| --- |
|  |

### 定制rememberMe功能

|  |
| --- |
|  |

通过实现RememberMeManager接口，可以定制RememberMe功能；

|  |
| --- |
|  |

# JSP标签

## 添加标签库

|  |
| --- |
| <%@ taglib prefix="shiro" uri="http://shiro.apache.org/tags" %> |

## guest

当用户处于游客状态，即用户不存在，或者没有登录或者没有rememberMe功能时，显示标签中的内容；

|  |
| --- |
|  |

## user

当用户的在系统中存在时，显示标签中的内容；

|  |
| --- |
|  |

## authenticated

当用户登录成功后，显示标签中内容；

|  |
| --- |
|  |

## noAuthenticated

当用户未登录时，显示标签内容：

|  |
| --- |
|  |

## principal

显示用户信息；

|  |
| --- |
|  |

等价于：

|  |
| --- |
|  |

打印指定类型的Principal

|  |
| --- |
|  |

等价于：

|  |
| --- |
|  |

打印指定属性名称的信息

|  |
| --- |
|  |

等价于：

|  |
| --- |
|  |

## hasRole

当用户具有指定的角色时，显示标签中的内容；

|  |
| --- |
|  |

## lacksRole

当用户没有指定的角色时，显示标签中的内容；

|  |
| --- |
|  |

## hasAnyRole

当用户具有指定角色中的任意一个，则显示标签中的内容；

|  |
| --- |
|  |

## hasPermission

当用户具有指定权限时，显示标签中的内容；

|  |
| --- |
|  |

## lacksPermission

当用户没有指定权限时，显示标签内容；

|  |
| --- |
|  |

# 实例