

陈续渊 Lv2

2019年08月04日 阅读 338

Spring Security基于方法级别的自定义表达式（可以完成任何权限判断）

背景

需求是这样的：项目采用的前后端分离的架构，且使用的RESTFUL风格，请求是一样的url，但是http method不一样。

如果要允许一个人获取某个资源，但是不能创建它，显然基于url的不够。

当我查阅到了可以基于方法的权限控制之后，我认为这应该是个最佳问题，一个方法针对不同的入参可能会触发不同的权限。比如说，一限，但是没有查看B目录的权限。而这两个动作都是调用的同一个Controller来区分查看不同的目录。

关于作者

陈续渊 Lv2

Java工程师 @ 追一...



获得点赞 88



文章被阅读 7,632

掘金小册



详解 Laravel 源码中优秀的设计模式

新人价 ¥9.95 ~~¥19.9~~

SpringBoot 源码解读与原理分析

新人价 ¥14.95 ~~¥29.9~~

新人专享好礼



送你 **45元** 买小册

[立即领取](#)

相关文章

基于postman的api自动化测试实践



18



2

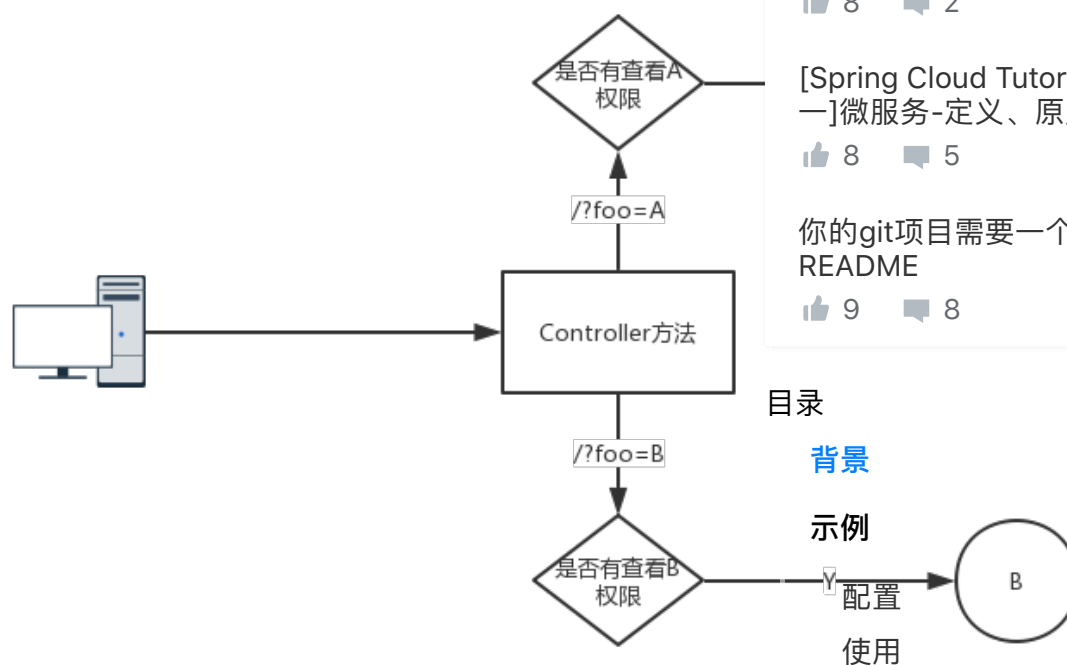
rap2使用姿势——前端、后端、测试必看（多gif图）



8



8



默认的 `hasAuthority` 和 `hasRole` 表达式都无法满足需求，因为它们只能判断一个硬编码的权限或者角色字符串。所以我们需要用到自定义表达式来高度自定义权限判断以满足需求。下面我们就来具体介绍如何使用它。

示例

我们将创建一个 `canRead` 的表达式。当入参为"A"时，将判断当前用户是否有查看A的权限；当入参为"B"时，将判断当前用户是否有查看B的权限。

配置

为了创建自定义表达式，我们首先需要实现root表达式：

java 复制代码

```
public class CustomMethodSecurityExpressionRoot
```

```
    extends SecurityExpressionRoot implements MethodSecurityExpressionOperations {

    private Object filterObject;
    private Object returnObject;

    public CustomMethodSecurityExpressionRoot(Authentication authentication) {
        super(authentication);
    }

    // 我们的自定义表达式
    public boolean canRead(String foo) {
        if (foo.equals("A") && !this.hasAuthority("CAN_READ_A")) {
            return false;
        }

        if (foo.equals("B") && !this.hasAuthority("CAN_READ_B")) {
            return false;
        }

        return true;
    }

    @Override
    public Object getFilterObject() {
        return this.filterObject;
    }

    @Override
    public Object getReturnObject() {
        return this.returnObject;
    }

    @Override
    public Object getThis() {
        return this;
    }

    @Override
    public void setFilterObject(Object obj) {
        this.filterObject = obj;
    }
}
```

```
}

@Override
public void setReturnObject(Object obj) {
    this.returnObject = obj;
}
}
```

接下来，我们需要把 `CustomMethodSecurityExpressionRoot` 注入到表达式处理器内：

java 复制代码

```
public class CustomMethodSecurityExpressionHandler
    extends DefaultMethodSecurityExpressionHandler {
    private AuthenticationTrustResolver trustResolver =
        new AuthenticationTrustResolverImpl();

    @Override
    protected MethodSecurityExpressionOperations createSecurityExpressionRoot(
        Authentication authentication, MethodInvocation invocation) {
        CustomMethodSecurityExpressionRoot root =
            new CustomMethodSecurityExpressionRoot(authentication);
        root.setPermissionEvaluator(getPermissionEvaluator());
        root.setTrustResolver(this.trustResolver);
        root.setRoleHierarchy(getRoleHierarchy());
        return root;
    }
}
```

然后需要把 `CustomMethodSecurityExpressionHandler` 写到方法安全配置里面：

java 复制代码

```
@Configuration
@EnableGlobalMethodSecurity(prePostEnabled = true)
public class MethodSecurityConfig extends GlobalMethodSecurityConfiguration {
    @Override
    protected MethodSecurityExpressionHandler createExpressionHandler() {
        CustomMethodSecurityExpressionHandler expressionHandler =
            new CustomMethodSecurityExpressionHandler();
        expressionHandler.setPermissionEvaluator(new CustomPermissionEvaluator());
    }
}
```

```
        return expressionHandler;  
    }  
}
```

使用


[复制代码](#)

```
@PreAuthorize("canRead(#foo)")  
@GetMapping("/")  
public Foo getFoo(@RequestParam("foo") String foo) {  
    return fooService.findAll(foo);  
}
```

如果用户访问A，但是没有 `CAN_READ_A` 权限，接口将会返回403。

关注下面的标签，发现更多相似文章

[Spring](#)

陈续渊  Java工程师 @ 追一科技
发布了 18 篇专栏 · 获得点赞 88 · 获得阅读 7,632

[关注](#)

安装掘金浏览器插件

打开新标签页发现好内容，掘金、GitHub、Dribbble、ProductHunt 等站点内容轻松获取。快来安装掘金浏览器插件获取高质量内容吧！

评论

相关推荐

专栏 · 江南一点雨 · 1月前 · Spring

手码两万余字，SpringMVC 包教包会

👍 83 💬 6

专栏 · 张大佛爷_zhang · 14天前 · Spring / Java

[Spring基本功系列]Spring源码之IOC原理

👍 31 💬 1

专栏 · 小姐姐味道 · 29天前 · Java / Spring

高逼格面试：线程封闭，新名词✓

👍 36 💬 4

专栏 · 江南一点雨 · 1月前 · Spring

Spring 学习，看松哥这一篇万余字干货就够了！

👍 53 💬 11

专栏 · 小傅哥 · 8天前 · Spring

源码分析 | 手写mybait-spring核心功能(干货好文一次学会工厂bean、类...

👍 3 💬

专栏 · Van_Fan · 17天前 · Spring

BeanUtils 如何拷贝 List?

👍 13 💬 5

专栏 · 码农清风 · 27天前 · Spring

面试还不懂这10道Spring问题，回去等通知了

👍 21

💬 3

专栏 · 江南一点雨 · 1月前 · Spring

学Maven，这篇万余字的教程，真的够用了！

👍 51

💬 8

专栏 · 宜春 · 27天前 · Spring

Spring注解之@Autowired、@Qualifier、@Resource、@Value

👍 17

💬 8