

# SpringSecurity核心原理分析

MCA-Java架构师VIP体验课【Mashibing Certificat Architect】

主讲老师：**邓澎波老师**

时间: 2021-10-19 [20:00-23:30]

## 1.本节课的收获

1. SpringSecurity基本应用
2. FilterChainProxy的核心作用
3. SpringSecurity中相关过滤器的作用分析
4. 掌握SpringSecurity的核心流程
5. 掌握正确的查看源码的方式
6. 掌握SpringSecurity在分布式架构中的作用
7. 系统学习的必要性和紧迫性

## SpringSecurity

Shiro SpringSecurity

1.访问之后没有认证的状态会跳转到默认的登录页面

如何实现自定义的登录页面

2.提供的有默认的账号密码

认证和自定义的数据源实现

分析清楚SpringSecurity的执行流程

1.系统启动的时候SpringSecurity做了什么事情？

在Spring的初始化中完成 SpringSecurity的配置文件的加载解析操作

DelegatingFilterProxy 在系统启动的时候会完成初始化的操作

DelegatingFilterProxy--》 获取 代理的真实对象

周杰伦的经纪人 --》 周杰伦

DelegatingFilterProxy 在初始化的时候 从容器中根据名称和类型从容器中获取到了 FilterChainProxy 对象

名称是自定义的

对象什么时候放入到容器中的呢？--》SpringSecurity的初始化操作

```
protected void initFilterBean() throws ServletException {
    synchronized(this.delegateMonitor) {
        if (this.delegate == null) {
            if (this.targetBeanName == null) {
                // 获取 Filter的名称: springSecurityFilterChain
                this.targetBeanName = this.getFilterName();
            }
            // 获取spring 容器
            WebApplicationContext wac = this.findWebApplicationContext();
            if (wac != null) {
                this.delegate = this.initDelegate(wac);
            }
        }
    }
}
```

```
protected Filter initDelegate(WebApplicationContext wac) throws
ServletException {
    // 获取的是 springSecurityFilterChain
    String targetBeanName = this.getTargetBeanName();
    Assert.state(targetBeanName != null, "No target bean name set");
    // 根据 名称springSecurityFilterChain 和类型 Filter 从容器中获取对象
    Filter delegate = (Filter)wac.getBean(targetBeanName, Filter.class);
    if (this.isTargetFilterLifecycle()) {
        delegate.init(this.getFilterConfig());
    }

    return delegate;
}
```

2.第一次请求到来的时候，SpringSecurity的处理流程？

DelegatingFilterProxy 拦截的表达式 /\*

FilterChainProxy --> FilterChain 的代理对象 --》找到 真实 FilterChain 【过滤器链】 一组过滤器链路

JWT OAuth2.0

单点登录 分布式授权

Shiro SpringSecurity

基本功扎实

丰富的互联网实战经验

