轻松入门SpringSecurity

讲师：李新杰

问题(Question)

url和角色（权限）的对应关系应该从数据库加载

这里的核心问题是ConfigAttribute，表示访问一个url需要具备的某种资格，可以是角色，权限，或任何其它形式。

所以要搞清楚系统围绕它的逻辑和代码。

如何配置，如何存储，如何获取，供谁使用？

系统提供权限(How to provide)

配置：

代码里写死的

存储：

FilterInvocationSecurityMetadataSource

获取：

FilterInvocationSecurityMetadataSource

使用：

FilterSecurityInterceptor

如何替换(How to replace)

配置：

从数据库中读取

存储：

自己实现这个接口

FilterInvocationSecurityMetadataSource

获取：

自己实现这个接口

FilterInvocationSecurityMetadataSource

使用：

想办法设置到FilterSecurityInterceptor类里

加载时机(When to load)

安全元数据在系统启动时进行加载计算，且后面不会再更新。无论是写死在代理里，还是从数据库中查询

安全元数据不保存，任何时候需要时，都从数据库/缓存查。

修改了安全元数据后，人工触发更新。更新缓存/容器中的Bean。

修改了安全元数据后，由定时任务检测到后并负责更新。更新缓存/容器中的Bean。

后处理机制(Post process)

允许在某个点介入到系统提供的执行流程中，执行一些自己的逻辑。

ObjectPostProcessor<T>，对象后处理器，允许对一个对象进行一些额外的初始化或配置。

基于Web访问的实现原理(URL)

因为处理的都是对url的访问，所以全部处理流程和机制都是采用过滤器实现的。就是Java Web中的Servlet Filter。

系统定义了很多Filter，且这些Filter之间是有先后顺序的。一个请求到来，会依次经过各个Filter进行处理。

流程控制机制(Flow control)

我们见到更多的是按数值来控制流程的，如一个变量的值，一个方法的返回值等。

这个系统是按异常来控制流程的，绝大多数核心方法都没有返回值。
如果没有抛出异常，则表明执行正常。
如果遇到问题，就抛出对应类型的异常。

对Bean方法访问的实现原理(Method)

也可以实现非web的，对容器中的bean的方法的访问控制。

活学活用(Do what you want)

这是一个通用的框架，肯定有不适合你的地方。

搞明白原理和代码，按需定制。

Thank you! Good bye!

朋友：李新杰

公众号：编程新说