

项目进阶，构建安全高效的企业服务

牛客Java高级工程师 第七章

► 1. Spring Security

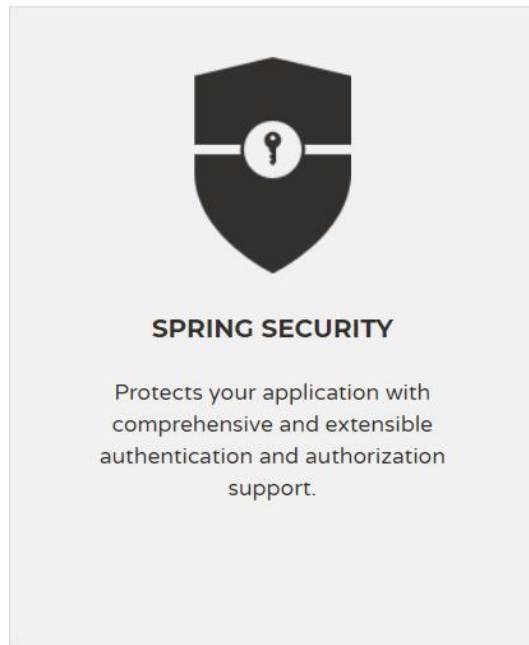
- 简介

Spring Security是一个专注于为Java应用程序提供身份认证和授权的框架，它的强大之处在于它可以轻松扩展以满足自定义的需求。

- 特征

- 对身份的 **认证** 和 **授权** 提供全面的、可扩展的支持。
- 防止各种攻击，如会话固定攻击、点击劫持、csrf攻击等。
- 支持与Servlet API、Spring MVC等Web技术集成。

<https://spring.io/projects/spring-security>

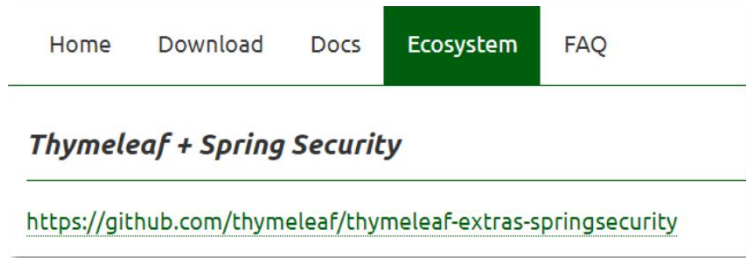


► 2. 权限控制

- 登录检查
 - 之前采用拦截器实现了登录检查，这是简单的权限管理方案，现在将其废弃。
- 授权配置
 - 对当前系统内包含的所有的请求，分配访问权限（普通用户、版主、管理员）。
- 认证方案
 - 绕过Security认证流程，采用系统原来的认证方案。
- CSRF配置
 - 防止 CSRF 攻击的基本原理，以及表单、AJAX相关的配置。

► 3. 置顶、加精、删除

- 功能实现
 - 点击 置顶，修改帖子的类型。
 - 点击 “加精”、“删除”，修改帖子的状态。
- 权限管理
 - 版主可以执行 “置顶”、“加精” 操作。
 - 管理员可以执行 “删除” 操作。
- 按钮显示
 - 版主可以看到 “置顶”、“加精” 按钮。
 - 管理员可以看到 “删除” 按钮。



► 4. Redis高级数据类型

- HyperLogLog
 - 采用一种基数算法，用于完成独立总数的统计。
 - 占据空间小，无论统计多少个数据，只占12K的内存空间。
 - 不精确的统计算法，标准误差为 0.81% 。
- Bitmap
 - 不是一种独立的数据结构，实际上就是字符串。
 - 支持按位存取数据，可以将其看成是byte数组。
 - 适合存储大量的连续的数据的布尔值。

► 5. 网站数据统计

- UV (Unique Visitor)
 - 独立访客，需通过用户 IP 排重统计数据。
 - 每次访问都要进行统计。
 - HyperLogLog，性能好，且存储空间小。
- DAU (Daily Active User)
 - 日活跃用户，需通过用户 ID 排重统计数据。
 - 访问过一次，则认为其活跃。
 - Bitmap，性能好、且可以统计精确的结果。

■ 网站 UV

年 / 月 / 日

年 / 月 / 日

开始统计

统计结果

0

■ 活跃用户

年 / 月 / 日

年 / 月 / 日

开始统计

统计结果

0

► 6. 任务执行和调度

- JDK 线程池
 - ExecutorService
 - ScheduledExecutorService
- Spring 线程池
 - ThreadPoolTaskExecutor
 - ThreadPoolTaskScheduler
- 分布式定时任务
 - Spring Quartz



<http://www.quartz-scheduler.org>

► 7. 热帖排行

- **Hacker News**

$$\text{Score} = (P-1) / (T+2) ^ G$$

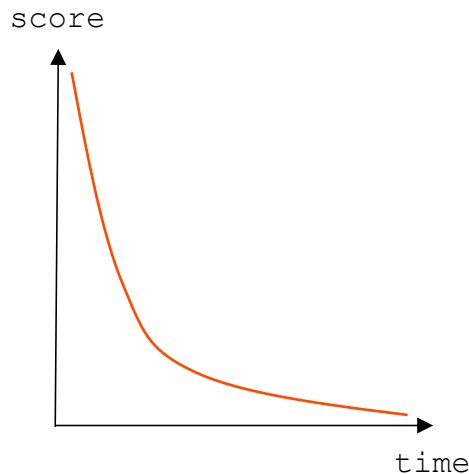
- **StackOverflow**

$$(\log(Q\text{views}) * 4) + ((Q\text{answers} * Q\text{score}) / 5) + \text{sum}(A\text{scores})$$

$$((Q\text{ageInHours} + 1) - ((Q\text{ageInHours} - Q\text{updated}) / 2)) ^ 1.5$$

- **Nowcoder**

$$\log(\text{精华分} + \text{评论数} * 10 + \text{点赞数} * 2 + \text{收藏数} * 2) + (\text{发布时间} - \text{牛客纪元})$$



► 8. 生长图

- wkhtmltopdf
 - wkhtmltopdf url file
 - wkhtmltoimage url file
- java
 - Runtime.getRuntime().exec()

<https://wkhtmltopdf.org>



► 9. 将文件上传至云服务器

- 客户端上传
 - 客户端将数据提交给云服务器，并等待其响应。
 - 用户上传头像时，将表单数据提交给云服务器。
- 服务器直传
 - 应用服务器将数据直接提交给云服务器，并等待其响应。
 - 分享时，服务端将自动生成的图片，直接提交给云服务器。



<https://www.qiniu.com>

► 10. 优化网站的性能

- 本地缓存
 - 将数据缓存在应用服务器上，性能最好。
 - 常用缓存工具：Ehcache、Guava、Caffeine等。
- 分布式缓存
 - 将数据缓存在NoSQL数据库上，跨服务器。
 - 常用缓存工具：MemCache、Redis等。
- 多级缓存
 - > 一级缓存（本地缓存）> 二级缓存（分布式缓存）> DB
 - 避免缓存雪崩（缓存失效，大量请求直达DB），提高系统的可用性。

Thanks

