目录

[Spring-security文档 1](#_Toc13263)

[Hello Spring Security 1](#_Toc12893)

[Updating Dependencies 1](#_Toc17996)

[Starting Hello Spring security boot 1](#_Toc9957)

[Spring boot Auto Configuration 1](#_Toc3676)

Spring-security文档

Hello Spring Security

这部分包括最小化配置spring boot使用spring security

* 为方便，可以https://github.com/spring-projects/spring-security/tree/5.4.2/samples/boot/helloworld找到完整的应用实例；可以下载一个最小的spring boot + spring security的应（https://start.spring.io/starter.zip?type=maven-project&language=java&packaging=jar&jvmVersion=1.8&groupId=example&artifactId=hello-security&name=hello-security&description=Hello%20Security&packageName=example.hello-security&dependencies=web,security）

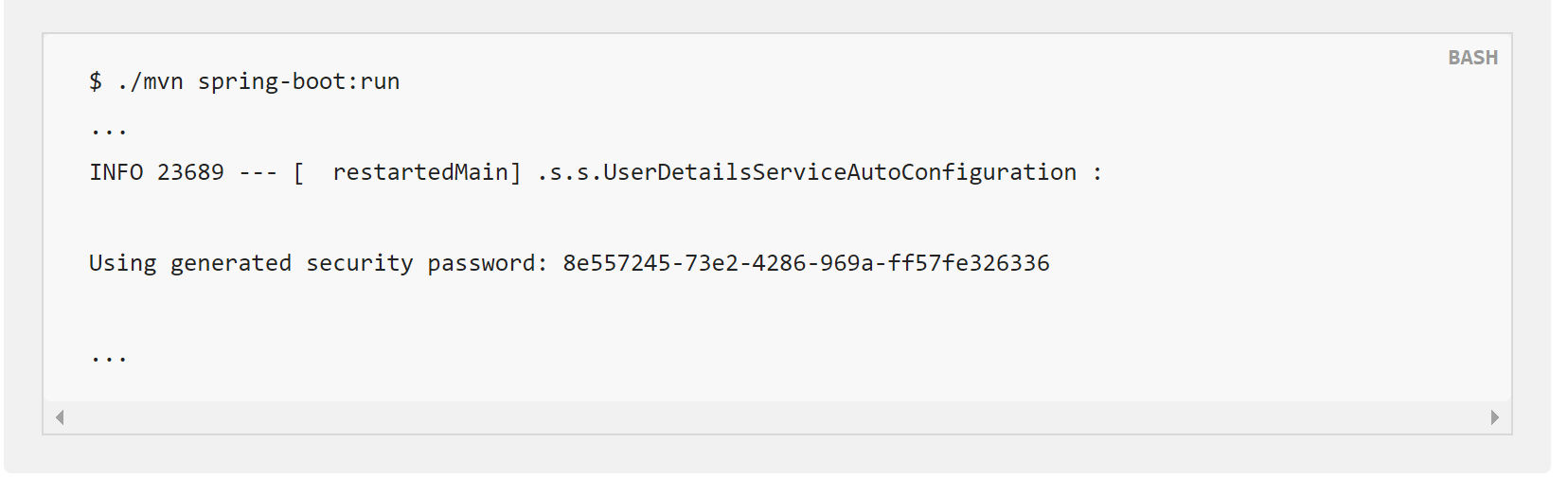
Updating Dependencies

仅需要用Maven或者Gradle设置你需要去更新的依赖

Starting Hello Spring security boot

你可以现在用Maven插件run命令运行Spring boot 应用，举例子展示怎么去做（以及开始输出）

**Example：**



Spring boot Auto Configuration

**Spring Boot automatically（spring boot 自动配置）:**

* 开启Spring Security 的默认配置，它将创建servlet过滤器为名为springSecurityFilterChain的bean；他将在你的应用中负债所有的安全（保护应用程序url、验证提交的用户名和密码、重定向到登录表单等等）
* 创建一个UserDetailsService Bean, 生成一个用户名为user，密码为随机生成打印在log控制台
* 为每个请求向servlet容器注册一个名为springSecurityFilterChain的Filter

**Spring boot 不需要做大量的配置就可以完成很多事，下面特性总结：**

* 为应用交互提供用户认证
* 为你生成默认的登入表单
* 让用户名为user的用户登入控制台，使用基础表单进行身份认证
* 使用BCrypt保护密码存储
* 让用户退出
* 防止CSRF攻击 [CSRF attack](https://en.wikipedia.org/wiki/Cross-site_request_forgery) prevention
* Session固定保护[Session Fixation](https://en.wikipedia.org/wiki/Session_fixation) protection
* 安全header集成Security Header integration

1. [HTTP Strict Transport Security](https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security) for secure requests
2. [X-Content-Type-Options](https://msdn.microsoft.com/en-us/library/ie/gg622941(v=vs.85).aspx) integration
3. Cache Control (can be overridden later by your application to allow caching of your static resources)
4. [X-XSS-Protection](https://msdn.microsoft.com/en-us/library/dd565647(v=vs.85).aspx) integration
5. X-Frame-Options integration to help prevent [Clickjacking](https://en.wikipedia.org/wiki/Clickjacking)

* 集成以下的 Servlet API方法

1. [HttpServletRequest#getRemoteUser()](https://docs.oracle.com/javaee/6/api/javax/servlet/http/HttpServletRequest.html" \l "getRemoteUser())
2. [HttpServletRequest.html#getUserPrincipal()](https://docs.oracle.com/javaee/6/api/javax/servlet/http/HttpServletRequest.html" \l "getUserPrincipal())
3. [HttpServletRequest.html#isUserInRole(java.lang.String)](https://docs.oracle.com/javaee/6/api/javax/servlet/http/HttpServletRequest.html" \l "isUserInRole(java.lang.String))
4. [HttpServletRequest.html#login(java.lang.String, java.lang.String)](https://docs.oracle.com/javaee/6/api/javax/servlet/http/HttpServletRequest.html" \l "login(java.lang.String, java.lang.String))
5. [HttpServletRequest.html#logout()](https://docs.oracle.com/javaee/6/api/javax/servlet/http/HttpServletRequest.html" \l "logout())

Servlet Security: The Big Picture

本节讨论springsecurity在基于Servlet的应用程序中的高级体系结构，我们为Authentication, Authorization, Protection Against Exploits 部分建立了极易理解的参考资料。