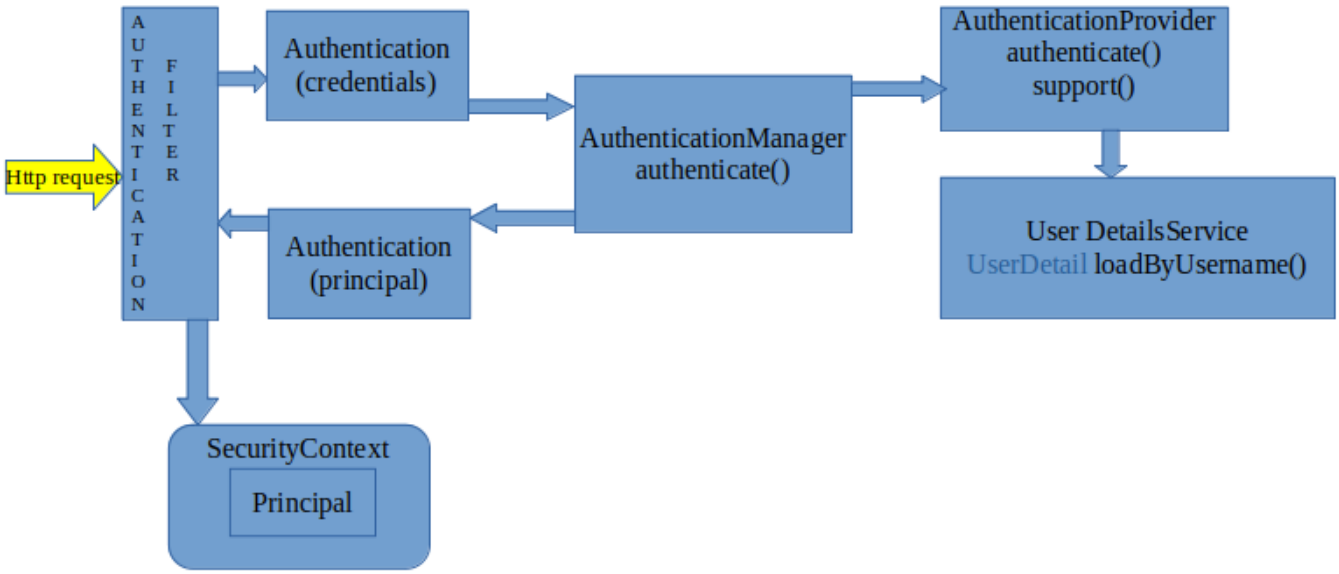
Spring security architecture

January 11, 2021 Lilly021 Team (https://lilly021.com/author/devlilly021-com/)
Lilly021 (https://lilly021.com/category/lilly021/)

(https://lilly021.com/spring-security-architecture/)

Spring security is framework used for securing Spring applications. It stands between client and application and gives possibility of configuring what data and functionalities are exposed to which users. Also, it handles common vulnerabilities. It is a huge topic, and in this post we are going to explain process of basic authentication in Spring security.
There are 5 core concepts in spring security:
Authentication (who are you)
Authorization (can you access or change some data)

Principal (currently logged in user)
Granted authority (permission)
Roles (group of permissions)
When you add spring security to your project, it will add filters to it, in order to intercepts requests
before they go to servlet. On the graph below is presented process of authentication in spring
security.



Authentication filter creates authentication object that contains credentials from request. Than it sends that object to authentication manger, to choose appropriate authentication provider. There are different authentication strategies, and for each strategy there is appropriate AuthenticationProvider. Each provider needs to get user from the system and to check its credentials (is it expired, is
password correct...). You can use different type of authentication providers in Spring security, or you can combine them.
Only part that is different between providers is how it finds user. That part is extracted in class UserDetailsService in loadByUsername() method which returns UserDetails object. UserDetails object contains data about user and account (is account expired, locked or enabled, are credentials non expired...) which authentication provider needs for authentication. If authentication is successful the Authentication object with the principal is returned to the Authentication Filter, which will set principal to SecurityContext . SecurityContext is associated
with current thread, and it provide principal for further authorization of request.
In the case that authentication is failed, spring throws an AuthenticationException, which goes back to the filter, so it can be catch or showed to user.

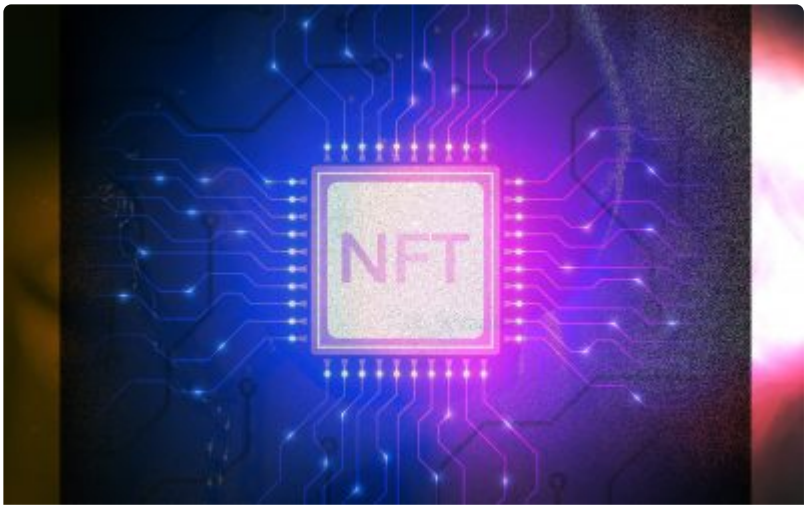
If you want to add spring security to your project you can follow next steps:

1. Add spring-boot-starter-security dependency
2. Extend the class WebSecurityConfigurerAdapter and add annotation @EnableWebSecurity on that class
3. Use mentioned class to configure authentication. In spring security, AuthenticationManagerBuilder is in charge for authentication, so you need to override method “configure” which receives AuthenticationMangerBuilder as param, and use it to set type of authentication. You can implement UserDetailsService to specify how users should be retreived from the system. If you have need, you can create your custom UserDetails by implementing UserDetails interface.
4. Configure authorization using HttpSecurity object, by overreading method configure of class WebSecurityConfigurerAdapter which receives HttpSecurity as a param. Here you can set restrictions to different paths by using method chain to set different paths and allowed roles. When you are chaining your paths you must start from the most restrictive one.

Previous (https://lilly021.com/wpf/)

Next (https://lilly021.com/what-are-ui-design-patterns/)

You may also like

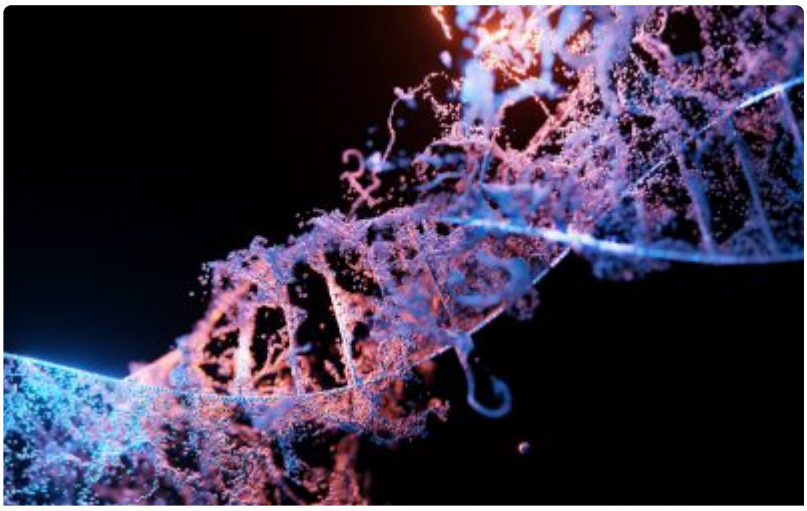


(https://lilly021.com/nft-and-its-applications/)

May 9, 2022 (https://lilly021.com/nft-and-its-applications/) Lilly021 (https://lilly021.com/category/lilly021/)

NFT and it's applications (https://lilly021.com/nft-and-its-applications/)

Read more (https://lilly021.com/nft-and-its-applications/)



(https://lilly021.com/genetic-programming/)

May 4, 2022 (https://lilly021.com/genetic-programming/) Lilly021 (https://lilly021.com/category/lilly021/)

Genetic programming (https://lilly021.com/genetic-programming/)

Read more (https://lilly021.com/genetic-programming/)

(https://lilly021.com/javascript-page-lifecycle/)

April 26, 2022 (https://lilly021.com/javascript-page-lifecycle/) Lilly021 (https://lilly021.com/category/lilly021/)

JavaScript page Lifecycle (https://lilly021.com/javascript-page-lifecycle/)

Read more (https://lilly021.com/javascript-page-lifecycle/)

About

About (https://lilly021.com/about/)

Careers (https://lilly021.com/careers/)

Become Partner (https://lilly021.com/become-partner/)

Portfolio (https://lilly021.com/portfolio/)

Blog (https://lilly021.com/blog/)

Contact (https://lilly021.com/contact/)

Latest news

NFT and it’s applications (https://lilly021.com/nft-and-its-applications/)

Genetic programming (https://lilly021.com/genetic-programming/)

JavaScript page Lifecycle (https://lilly021.com/javascript-page-lifecycle/)

What is WebdriverIO: (https://lilly021.com/what-is-webdriverio/)

Docker vs. VM (Virtual Machine) (https://lilly021.com/docker-vs-vm-virtual-machine/)

Lilly021

Software development | Web development | Mobile development