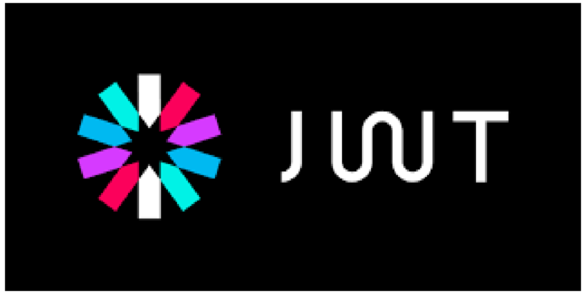
Secured API with Spring Security



Base concepts regards credential based logging

Authentication

Verifying identity of the user
This process has been done after provide credentials of particular user
Establish the user identity of the system
Status code regards : 401

Authorization

Determine what kind of actions to be allowed
Process done after successfully identified the user / authenticated the user
Status code regards: 403

Security mechanisms (some of)

Basic Auth

- Use the Authorization HTTP header
- Provided credentials are transmit via this header
- Header details are encoded as Base64 format
- Has vulnerability

OAuth 2.0

- Open standard authorization protocol

  Inspire to use tokens
- Try to not expose user's sensitive information

Explore about Spring Security

https://docs.spring.io/spring-security/reference/servlet/getting-started.html

Main points – Main beans

SecurityFilterChain

Responsible to handle incoming HTTP request with enforce the configured security stratergies

AuthenticationProvider

Normally it is implemented DaoAuthenticationProvider. It is responsible for fetching user info and encode/ decode passwords

AuthenticationManager

Responsible for authenticating the Authentication object, handling all necessary tasks

Key classes, interfaces and methods

OncePerRequestFilter

Invoke once per request

request.getHeader("Authorization");

Get the header that taken the authorization data and normally it is the credentials

SecurityContextHolder

The container that hold authenticated principal (user) of an application

UserDetailsService

responsible for retrieving user-related data during the authentication process

UserDetails

An interface that represents core user information retrieved by the UserDetailsService

Key Classes, interfaces and methods cont.

UsernamePasswordAuthenticationToken

Create auth token for given user details

UsernamePasswordAuthenticationToken(

userDetails – Instance of UserDetails. Included user's information like username and password.

null – Represent user's password. Here the token create after the user authentication. That's why its null.

userDetails.getAuthorities() - Usually, it is represent user's role

)