SPRING SECURITY

Monday, 8 July 2024 7:03 PM

Spring

Spring Boot

Spring DATA

Spring Security

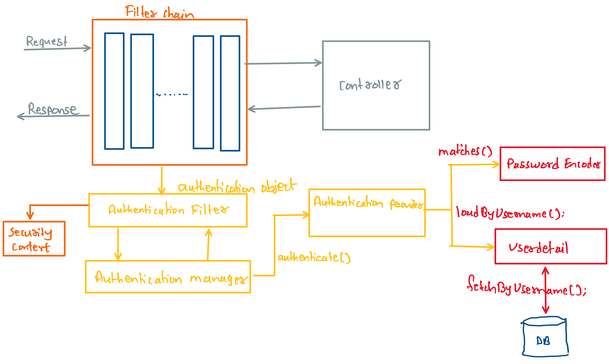
Authentication proving who you are

Authorization what you allowed to do after who you are.

Key Security principles:-

* Least privilege
* Secure by Design
* Fail-safe by default
* Secure information
* Input Validation
* Auditing and logging.
* updates and Patch management.

How Spring Security works?



* All the API's are protected
* Form Based Authentication
* Auto generated password with user as defut name
* Spring.Security.user.name
  spring.Security.user.password
  to change default user name and password.

Security filters:-

@EnableWebSecurity:- tells to enable webSecurity features in Application and gives liberaty to customize the SecurityConfig.

Basic Auth it is not a form based Authentication and there is /logout API

JSESSIONID Cookies (can be disabling the session to stateless).

Basic Auth uses base64 Encoding to Encode username and password

Ex:- admin:Hello ⟶ Encoder ⟶ YWRtaW46SGVsbG8=
(admin = username, Hello = password)

In Memory Authentication:-

```
no usages
@Bean
public UserDetailsService userDetailsService() {
    UserDetails user1 = User.withUsername("admin1")
            .password("{noop}Hello1")
            .roles("USER")
            .build();

    UserDetails user2 = User.withUsername("admin2")
            .password("{noop}Hello2")
            .roles("ADMIN")
            .build();
    return new InMemoryUserDetailsManager(user1,user2);
}
```

username
static method returns UserBuilder object
password
role

↳ implements UserDetailsManager
↳ extends UserDetailsService

Role Based Authentication:-

@PreAuthorize:- used to check the authorization before executing the method and specific condition.

@EnableMethodSecurity:- By adding method Security preAuthorize will start Working.

* 403 (Forbidden) Able to login but server refusing respond to it.

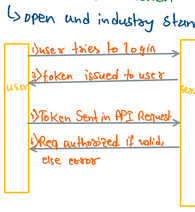
Hashing:- Converting data into unreadable format

BCryptPasswordEncoder = data + Xy324 (salt) ⟶ Encoder ⟶ $2a$...
(data = Original data, Xy324 = Autogenerated String, $2a$... = Encoded Password)

JWT Authentication:-

* Without JWT
  → No advance feature like Expiration time
  → Can be decoded easily
  → Custom token System:- once user login returns token and this token will can be pass in the API Request instead of username & Password.

JWT :- JSON Web Token
↳ open and industry standards



How token sent
Header
Authorization: Bearer <token>

JWT contains 3 parts divided with .
1) Header (algorithm & Token Type)
2) Payload Data
3) Signature

Implementing JWT:-

JwtUtils:- * contains utility method for generating, parsing and validating JWTs
* generation of token, validation of token and Extracting username from the token.

AuthTokenFilter:- * Filters incoming requests to check for a valid JWT in the header, setting the authentication context if token is valid.
* Extract JWT from the request header, validates it and configure the Spring Security context with user detail

AuthEntryPointJwt:- * Custom handling for unauthorized requests, typically when authentication is required but not supplied or valid.
* Unauthorized request is detected, it logs the error and returns a JSON response with an Error message, statuscode and the path attempted.

SecurityConfig:- Security Filter chain, Permitting or denying access based on paths and roles. Session management to stateless, crucial for JWT usage.