Day 20 OAuth & KeyCloak

Day 19 JWT Revisit

JWT – JSON Web Token (Authentication as well as Authorization)

It’s helps to securely (handling claims) communicate between two parties.

JWT is consist of 3 sections Header, Payload and Signature. All are encrypted using different algorithms. HAS,RSA, PSA [256,384,512 bits]

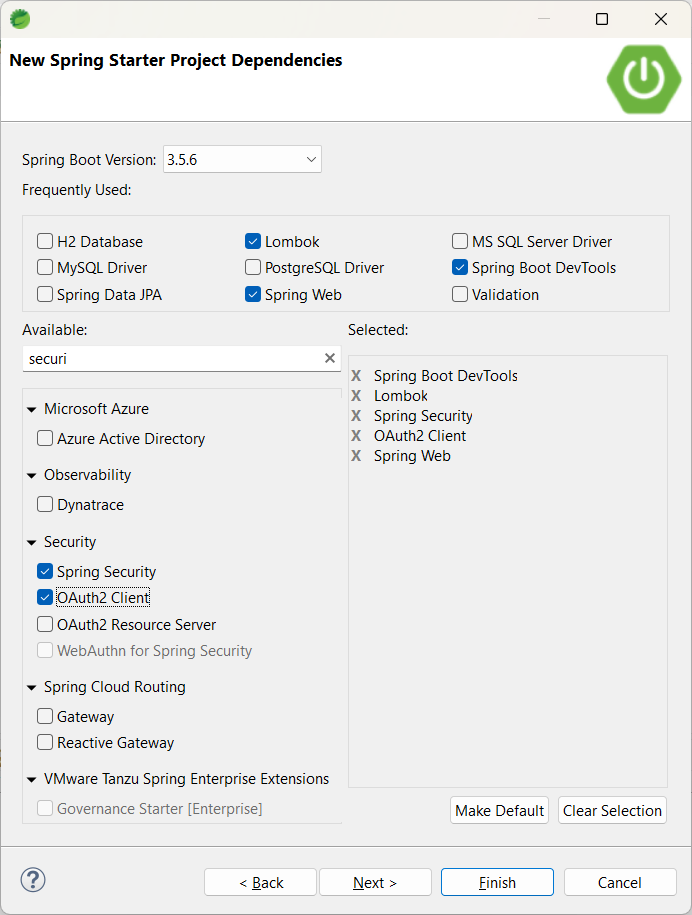
JWT.io

spring.main.allow-circular-references=true

DTO – Data Transfer Object (Used to send data from client to server or server to client) – Temp data. Not Saved anywhere.

JWT – Adv. Fine Grained control. (Each end point access we can control based on the role)

OAuth – Open Authorisation. [ Enable 3rd party login – Google, Github, LinkedIn, Microsoft, facebook] –



<https://www.geeksforgeeks.org/advance-java/implementing-oauth2-with-spring-security-a-step-by-step-guide/>

<https://console.cloud.google.com/auth/clients>

Spring Boot.

Week4 Revisit.

Day 16 – Spring Profile (How to set a spring boot project with multiple environments support)

Spring.active.profile = dev/test/stage/prod

Application.properties

Application-dev.properties

Application-test.properties

Application-stage.properties

Application-prod.properties

DEV, TEST, STAGE (PRE-PROD), PROD

DEV – Community version tools (in-memory db, sample/dummy/mock data)

TEST – Persistence DB (MySQL/Oracle XE etc., - Huge No of records to do Stress/Load Testing – Simulating 10000 users concurrent login. Performance Testing)

PRE-PROD/STAGE (UAT) – A cloned version of PROD environment with Dummy data.

ARS – Airline Reservation System

ARS-v1 – CLI version (Maven based java project using JDBC, MySQL/MS-SQL DB, CRUD operations on users, reservations, flights, payments, airports)

ARS-V2 – Spring Boot Monolith Application (Spring Starter Project with required dependencies)

Spring web, devtools, data-jpa, Lombok, validation, MySQL, Postgres, MS-SQL, h2, Swagger, Security, JWT)

Spring Security Module

Creates 2 end points (“/login”,”/logout”), 2 forms (login & logout) also configures sample user with dynamic password.

To avoid configuring the default password by spring boot, we can add following properties to spring boot application.properties file.

Spring.security.user.name=

Spring.security.user.password=

Spring.security.user.roles =

SecurityConfig.class

1. Form Based Authentication (Basic Authentication)
2. Token Based Authentication (JWT – Authentication & Authorization) – secret-key
3. OAuth – Enable Social login integration [Google, Facebook, Github etc.,]

KeyCloak – Open Source IAM service

Keycloak provides

user federation,

strong authentication,

user management,

fine-grained authorization

<https://www.keycloak.org/>

kc.bat start-dev