Revisit

* Micro- Service Security
* Inter-communication in Micro-Service (RestTemplate – Synchronous method)
* JWT – Token based Authentication and Authorization examples
* Adding Security Dependency in Spring Boot will automatically create 2 end points
* /login & /logout
* Login Screen with fields enter username and password will be created
* Filter will be added to check the UserDetails object in the Request.
* Authentication – Is the process of checking the credentials validity to ensure the user is already registered with the system/application
* Authorization – is the process of checking the permissions of currently logged in user to access resources
* JWT – JSON Web Token – It carries the data in the form of JSON. This will be alphanumeric string.
* JWT is divided into 3 parts. Header, Payload & Signature
* All the sections of JWT is encrypted using hashing algorithm (PSA,RSA,ESA,HSA – 256/384/512)

Medicines (Rarely available medicines – only with proper prescription)

Role based Authorization – User & Role (Authority) – UserDetails

OAuth

OIDC – Open ID Connect

LDAP – Lightweight Directory Access Protocol

CSRF – Cross Site Request Forgery

CORS – Cross Origin Resource Sharing

Spring Web MVC --- Dispatcher Servlet --- This servlet act as a Front Controller.

Normal Web Application – Web Servers like Tomcat, Weblogic, Glassfish (web.xml/@ServletMapping)

API documentation, Actuators (Health Checks / API Metrics)

Open API Specification

Spring Boot Actuator – Is a Operations related Dependency which adds few end-points to check & monitor the Applications Health & Other metrics.

SpringBoot – Is called as Opinionated Framework.

If you want to do some additional task

1. Add the Necessary Dependency
2. Add Annotations or Entry in Application.properties

Mini Project

Each project Team contains 5 members, Total 4 Project

1. Retail Banking Mgmt System
2. Pension Management System
3. Mail Order Pharmacy
4. Policy Administration System

Each project consist of 3 to 4 micro-service

4 Github private repo. – syskantechnosoft

<https://docs.google.com/spreadsheets/d/1dAD3wOc6vyO9XOIPErlDhi4oPmv9zS-7S27dT1OOIcc/edit?usp=sharing>

API Documentation – Swagger

OAS – Open API Specification

Open API Document – A document that describes the API.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Member Name | Micro-Service Name | Port | Github URL |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

<https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml>

Eureka Server, JWT

Adding Swagger 3 Steps

1. Create a Spring Starter Project with Spring Web & Spring Boot Dev Tools Dependency
2. Add the following Swagger Dependency

<dependency>

<groupId>org.springdoc</groupId>

<artifactId>springdoc-openapi-ui</artifactId>

<version>1.6.12</version>

</dependency>

1. Add the following Annotation in Starter Class @OpenAPIDefinition(info = @Info(title = "Sample API", version = "1.0", description = "Sample API"))

Adding Swagger will automatically add two end points

/swagger-ui/index.html (<http://localhost:8085/swagger-ui/index.html> )

/v3/api-docs

Github.com/username/mop/member1

Add Customer (id,Name, Email, Phone)

Account (id, type(savings/current/loan/deposit), minBalance, int\_rate)

AccountController

CustomeController

1. Eureka Server (Spring Boot Project – 8761) – Discovery-Server [ Spring Web, Spring Boot DevTools, Eureka Discovery Server, OpenAPIDoc, Actuator] @EnableEurekaServer, @OpenAPIDefinition
2. 4 Micro-Service [EurekaClient, Spring Web, Spring Data JPA, SpringBoot Dev Tools, Lombok, OpenAPI-ui, Actuator, h2, MySQL] @EnableEurekaClient, @OpenAPIDefinition
3. JWT micro-service [Auth-service]
4. Spring Web MVC Application – Web Client (Which consumes all the API)
5. Make sure to Test all the End points using Postman and Swagger. Take the screenshots and add it in Test (Word) Document.
6. Dbscript.sql
7. ReadMe.MD