**Sample exercises on Centralized Authentication and SSO with Spring Boot 3 and Spring Cloud**

Exercise 1: Implementing Centralized Authentication with OAuth 2.1/OIDC

Securityconfig

package com.example.oauth\_login\_demo.config;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.web.SecurityFilterChain;

@Configuration

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http

.authorizeHttpRequests(auth -> auth

.anyRequest().authenticated())

.oauth2Login();

return http.build();

}

}

UserController

package com.example.oauth\_login\_demo.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

import java.security.Principal;

@RestController

public class UserController {

@GetMapping("/user")

public Principal user(Principal principal) {

return principal;

}

}

Application.yml

server:

port: 8080

spring:

security:

oauth2:

client:

registration:

google:

client-id: 639666112397-jfe5fchtoejhsimo9l0hnpc7masjntjt.apps.googleusercontent.com

client-secret: GOCSPX-Pa9-pQJbwnwAtQk6JUvXaQKITU\_O

scope:

- openid

- profile

- email

authorization-grant-type: authorization\_code

redirect-uri: "{baseUrl}/login/oauth2/code/{registrationId}"

provider:

google:

authorization-uri: https://accounts.google.com/o/oauth2/v2/auth

token-uri: https://oauth2.googleapis.com/token

user-info-uri: https://openidconnect.googleapis.com/v1/userinfo

user-name-attribute: sub





