GDRP

Configuration:

Added Password Encryption:BCryptPasswordEncoder bean to securely encode passwords before storing them.

Access Control: Configured Spring Security to require authentication for accessing the /orders endpoint.

@Bean

public PasswordEncoder passwordEncoder() {

return new BCryptPasswordEncoder();

}

@Override

protected void configure(HttpSecurity http) throws Exception {

http

.authorizeRequests()

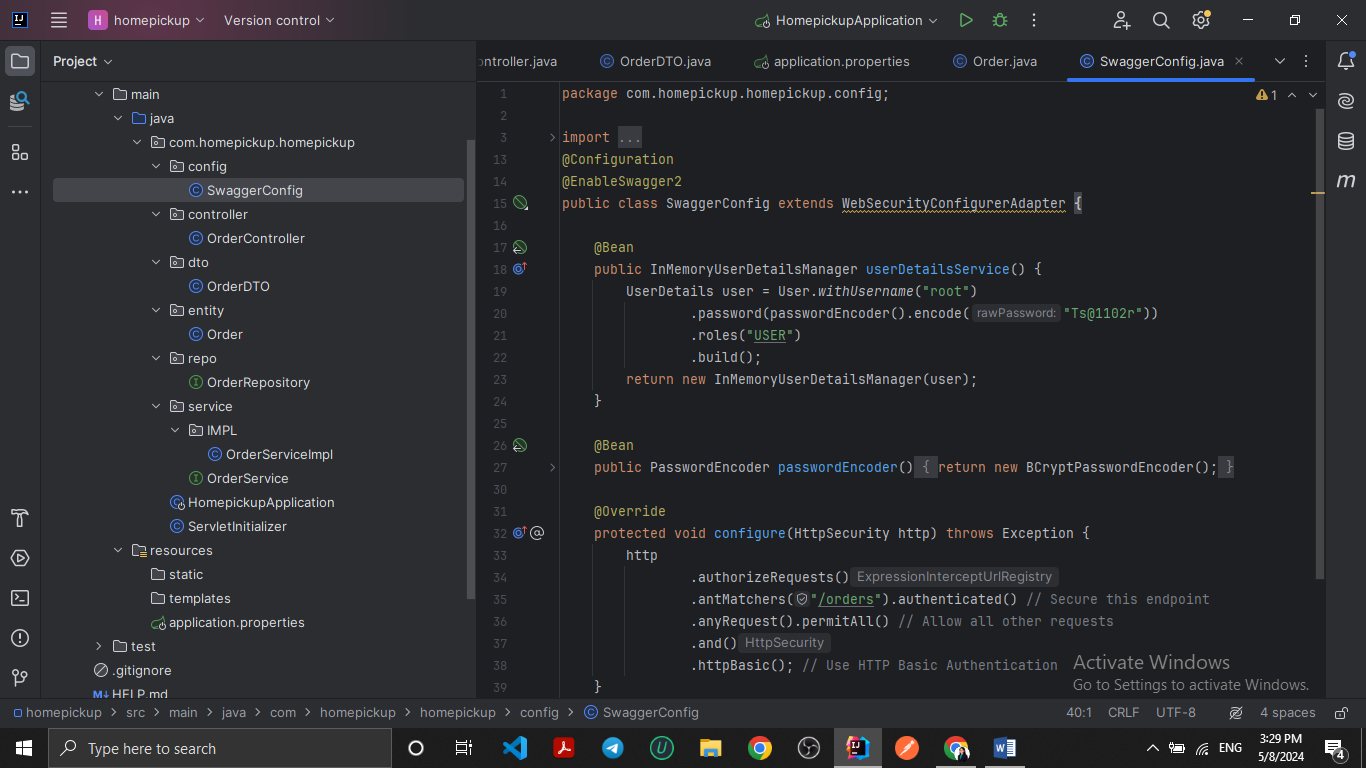
.antMatchers("/orders").authenticated() // Secure this endpoint

.anyRequest().permitAll() // Allow all other requests

.and()

.httpBasic(); // Use HTTP Basic Authentication

}



Controller:

Added GET Endpoint: I added a GET endpoint to retrieve order details by ID.

@GetMapping("/{id}")

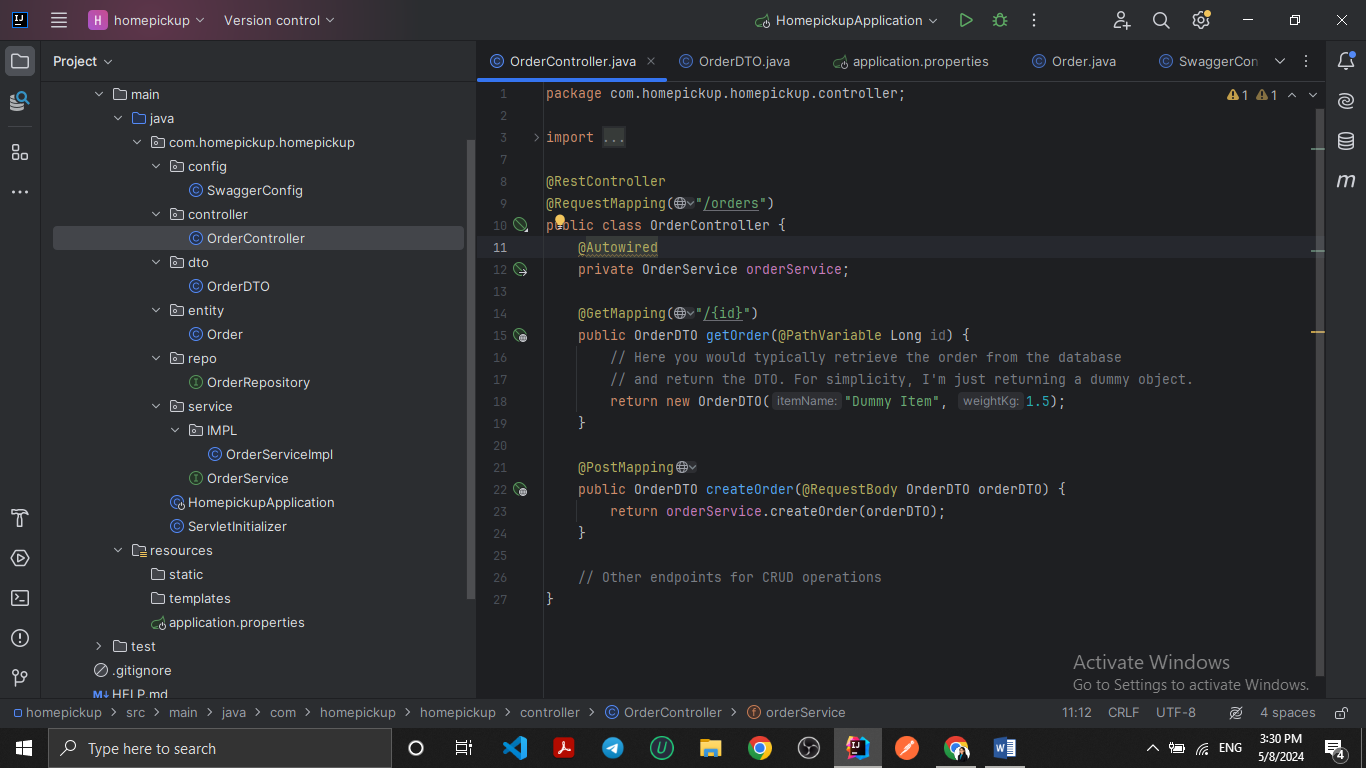
public OrderDTO getOrder(@PathVariable Long id) {

// Here you would typically retrieve the order from the database

// and return the DTO. For simplicity, I'm just returning a dummy object.

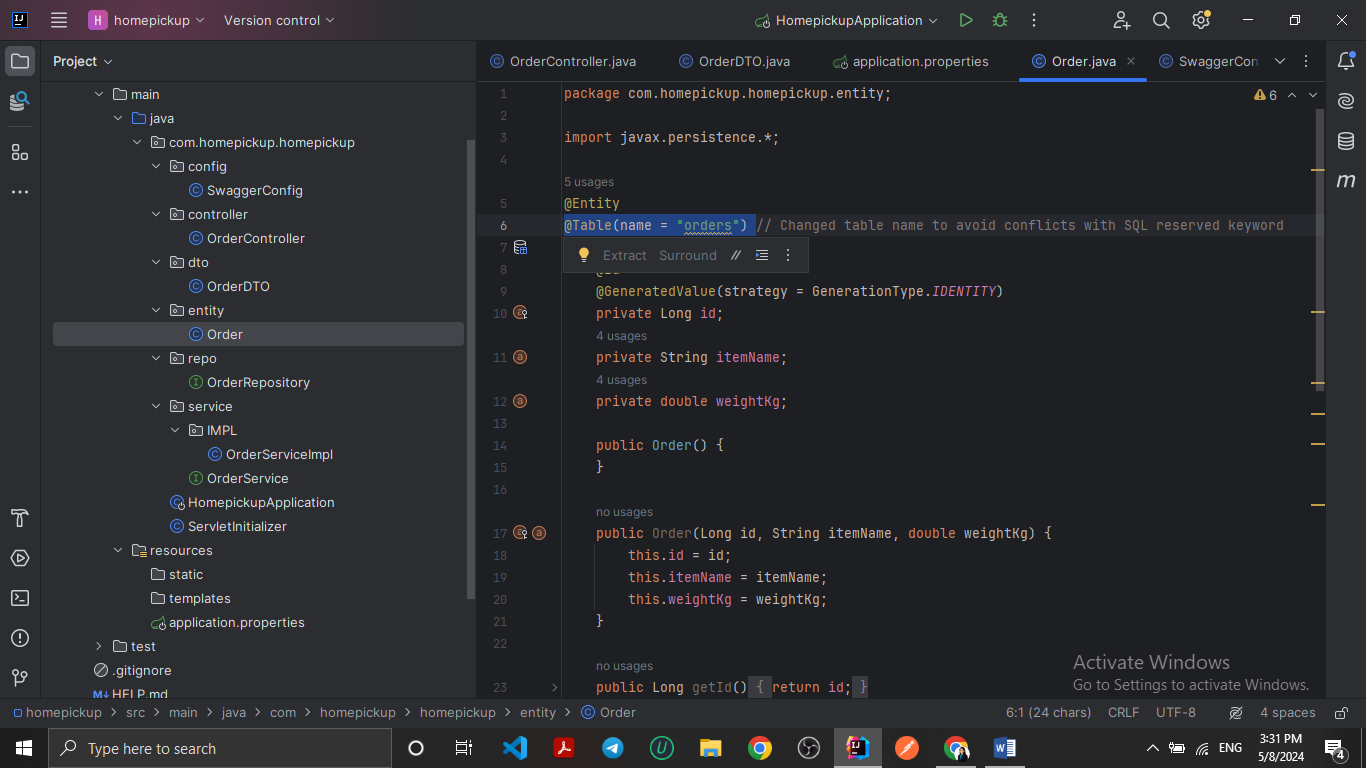
return new OrderDTO("Dummy Item", 1.5);

}



Entity:

Table Name: Changed the table name to orders to avoid conflicts with SQL reserved keywords.



@Table(name = "orders")

Repository:

Removed @EnableJpaRepositories: This annotation is typically placed at the configuration level, not at the repository level.

@Repository

public interface OrderRepository extends JpaRepository<Order, Long> {

}

Service Implementation:

Refactored Package Name: Renamed the package from com.homepickup.homepickup.service.IMPL to com.homepickup.homepickup.service.impl (following Java conventions).

package com.homepickup.homepickup.service.impl;

