**AuthController – API Authorization**

**Create a new util file -> enum -> Authority.java**

package org.studyeasy.SpringRestdemo.util.constants;

public enum Authority {

    READ,

    WRITE,

    UPDATE,

    USER,   //CAN UPDATE DELETE SELF OBJECT, READ ANYTHING

    ADMIN   //CAN READ UPDATE DELETE ANY OBJECT

}

**Modify Account.java**

private String role;

**to**

private String Authorities;

**Fix Error in AccountService.java and do in others where you get error**

Instead of getRole() and setRole(), replace with getAuthorities() and setAuthorities()

**In save() method – change the below line**

public Account save(Account account) {

        account.setPassword(passwordEncoder.encode(account.getPassword()));

        if(account.getAuthorities() == null){

            account.setAuthorities("ROLE\_USER);

        }

        return accountRepository.save(account);

    }

account.setAuthorities("ROLE\_USER);

Update to

 account.setAuthorities(Authority.USER.toString());

**Updated SeedData.java**

package org.studyeasy.SpringRestdemo.config;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.CommandLineRunner;

import org.springframework.stereotype.Component;

import org.studyeasy.SpringRestdemo.model.Account;

import org.studyeasy.SpringRestdemo.service.AccountService;

import org.studyeasy.SpringRestdemo.util.constants.Authority;

@Component

public class SeedData implements CommandLineRunner{

    @Autowired

    private AccountService accountService;

    @Override

    public void run(String... args) throws Exception {

        Account account01 = new Account();

        Account account02 = new Account();

        account01.setEmail("user@user.com");

        account01.setPassword("pass987");

        account01.setAuthorities(Authority.USER.toString());

        accountService.save(account01);

        account02.setEmail("admin@admin.com");

        account02.setPassword("pass987");

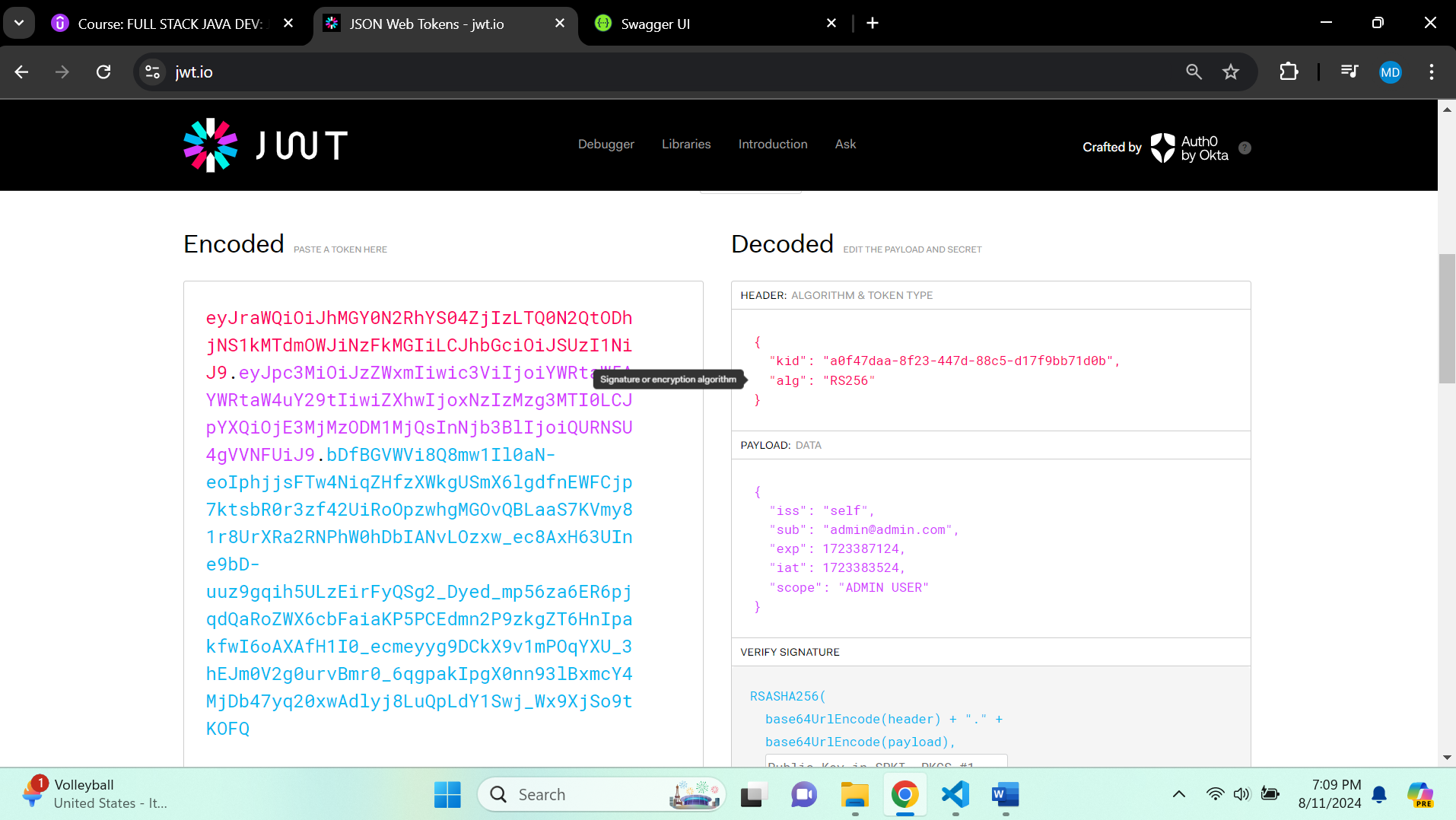
        account02.setAuthorities(Authority.ADMIN.toString() + " " + Authority.USER.toString());

        accountService.save(account02);

    }

}

**Output**

****

**In SecurityConfig.java**

**Update below line**

.requestMatchers("/auth/users").hasAuthority("SCOPE\_ROLE\_USER")

**To**

.requestMatchers("/auth/users").hasAnyAuthority("SCOPE\_ADMIN")

**Output**

In this, when you want to display list of users -> you get error.

So, you authorize using token by generating it and authorizing it. Try to display list of users (ADMIN), you get list of users.

So, you authorize using token by generating it and authorizing it. Try to display list of users (USER), you get error.