|  |  |
| --- | --- |
| Shortcuts | What it does |
| Ctrl+b | Full screen or close VS side pane |

All components must start with capital letter because they’re really JSX (HTML written in JS code), small case reserved for HTML.

Must export default in each component.

Each class component must return a div or a react fragment <> </>that contains other JSX elements/html elements cannot return >1 .

For JS object member variable keys must be constant not variables

 {

                Username (cannot be event.target.name)

                :event.target.value

            }

Whenever we create REST service need to define method (GET/POST/PUT/DELETE) and URI

Tell Spring MVC that a class will handle REST requests, need to map /GET method to URI. use @RestController.

We create a Spring Boot project, through it we are using frameworks like spring, spring mvc and other things.

!!!!HelloWorldBean must have getter to remove error

Instead of returning a string, return a Bean which displays as JSON format.

Added spring boot starter web which haas dependency on Spring MVC framework thus get DispatcherServlet. Then AutoConfiguration said found DispatcherServlet on classpath. Configuration of Error pages by ErrorMVCAutoConfiguration; and also work by DIspatcherServlet; conversion to JSON format from Beans (done by HttpMessageConvertersAutoConfiguration, which initializes JacksonBeans) done by Spring Boot AutoConfiguration. all configuration done by Spring Boot AutoConfiguration which configures things on classpath.

Request first always sent to DispatcherSErvlet handles all incoming request, it is “/” the root of the web application. DSE knows all mappings of methods and URIs, then looks at method and URI for correct one to bean, converts it to JSON, then return the response back to user.

/users/{id} {id} is path variable, very important in mapping between method and URI, resource mappings, in restful applications, creating REST resources.

How to use Axios to make http call? Use return axios.get (url) to return a response

Each method in HelloWOrldSErvice will return a promise back. Handle error in Welcome Component which is calling the service.

When something not working go developer tools -> Network then log response

If react says Object cannot be child of React, then check and see if promise returning JSON object instead of String in Eclipse. Step 37.

NEED TO USE TICK WHEN USING PATH VARIABLES IN REACT AXIOS

Made HTTP service calls is easy using Axios. We made a simple API call which returned a JSON, and made API call with path variable.

Need APIs to get(GET), delete(DELETE), update(PUT), create(POST) comments.

Using Spring Data JPA syntax in COmmentResource

@Service annotation to say a class is a Spring component to be managed by it. @Autowired to add component.

When state of component changes render() will first be called. But when component first shown on screen (mount) it is componentDidMount() that is shown.

Use

let username = AuthenticationService.getLoggedInUserName()

        CommentDataService.retrieveAllComments(username)

        .then(

            response=>{

                // console.log(response)

                this.setState({comments:response.data})

            }

        )

to retrieve comments using API and update state in component to be displayed when it is rendered.

Component lifecycle: constructor()->render()->componendDidMount(). componentWIllUnmount() just before you change to another page, clear resource allocations here. When does a state change and when view is updated is determined by framework. shouldUpdateComponent() is the one that controls whether a view is rendered again after a state update, makes performance faster.

ResponseEntity<Void> allows us to return empty content as opposed to default success HTTP status when delete is successful. One important part of RESTful services is it allows use to build specific requests with specific state assigned; ResponseEntity allows us to use .noContent() if successfully delete or notFound() if problem deleting. [all this is in resource method of delete]

\*\*remember to use correct HTTP method in axios.

Function/method execution no need semicolon;

<Formik

                    initialValues={{

                        description:description,

                        inResponseTo:inResponseTo

                    }}

                >

\*\*The key-value pairs must be sticked together

Remember to have default constructor in Comment if you want something as a (PUT/POST) request body when creating or updating an object.

ResponseEntity.*created*(uri).build() create ResponseEntity and return status “created” and pass in URI

POST METHOD don’t put ID as creating new ID.

CommentResource is following REST standards for things it is returning.

onSubmit(values){

        let username = AuthenticationService.getLoggedInUserName()

        CommentDataService.updateComment(username, this.state.id, {

            //Use state values for those which are carried over from ListComments

            //Use values. if obtained from Formik.

            //

            id:this.state.id,

            description:values.description,

            inResponseTo:values.inResponseTo,

            targetDate: this.state.targetDate,

            username:this.state.username

        })

        console.log(values);

    }

When using put method for axios all fields must be sent as object to be added .

Graphical user interface, text, application, Word

Description automatically generated

[Whether to package front and backend together](https://stackoverflow.com/questions/49817462/react-spring-boot-deployment)?

I think it's a really bad idea to bundle your frontend app in with your backend for many reasons. Big repo, poor separation of concerns, complicated release cycles/deployment, tight coupling, you may have multiple UIs - e.g. web and native, more setup for UI devs who shouldn't need to concern themselves with java or maven etc. Why not have two projects, since they have no dependency on each other, the UI just consumes the API like any other client.

separating your front-end and back-end code (keep them in separate repositories).

Allowing all cross-origin requests

<https://stackoverflow.com/questions/39623211/add-multiple-cross-origin-urls-in-spring-boot>

Useful GIT guide: https://happygitwithr.com/pull-tricky.html

|  |  |
| --- | --- |
| Error | solution |
| Undefined value to defined value when typing initially to Formik text field | initialize text field it with an empty string first. |
| Formik not component “description” undefined | Remember to set initial values for Formik first. |

Utilize docker?

https://medium.com/geekculture/a-reactjs-web-application-with-a-spring-boot-backend-and-containerizing-it-using-docker-3eeaed8cb45a

https://elvisciotti.medium.com/dockerizing-areact-and-spring-boot-application-359fed5adb7d

<https://www.callicoder.com/spring-boot-mysql-react-docker-compose-example/>

# Difference between in28 minutes /02-todo-rest-api-h2 project (<https://www.udemy.com/course/deploy-java-spring-boot-to-aws-amazon-web-service/learn/lecture/15443732#content>) and changing from HardcodedCommentService to in memory H2-database myself

## application.properties

logging.level.org.springframework = INFO

#

#spring.mvc.view.prefix=/WEB-INF/jsp/

#spring.mvc.view.suffix=.jsp

#logging.level.org.springframework.web=INFO

spring.jpa.hibernate.ddl-auto=update

spring.jpa.defer-datasource-initialization: true

spring.sql.init.mode= embedded

spring.jpa.show-sql=true

spring.h2.console.enabled=true

spring.h2.console.settings.web-allow-others=true

## data.sql

**insert** **into** comment(id, description,urgency\_level,in\_response\_to,

target\_date,username)

**values**(1, 'description 1', 'High', 3, LOCALTIMESTAMP - INTERVAL '30' MINUTE, 'username 1');

**insert** **into** comment(id, description,urgency\_level,in\_response\_to, target\_date,username)

**values**(2, 'description 2', 'Medium', 2, LOCALTIMESTAMP - INTERVAL '30' MINUTE, 'username 2');

**insert** **into** comment(id, description,urgency\_level,in\_response\_to, target\_date,username)

**values**(3, 'description 3', 'Low', 1, LOCALTIMESTAMP - INTERVAL '30' MINUTE, 'username 3');

## pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.7.5</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<groupId>com.mooc-app.rest.webservices</groupId>

<artifactId>restful-web-services</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>restful-web-services</name>

<description>RESTful APIs for mooc-app</description>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>javax.xml.bind</groupId>

<artifactId>jaxb-api</artifactId>

<version>2.3.0</version>

</dependency>

<dependency>

<groupId>com.sun.xml.bind</groupId>

<artifactId>jaxb-impl</artifactId>

<version>2.3.0</version>

</dependency>

<dependency>

<groupId>org.glassfish.jaxb</groupId>

<artifactId>jaxb-runtime</artifactId>

<version>2.3.0</version>

</dependency>

<dependency>

<groupId>javax.activation</groupId>

<artifactId>activation</artifactId>

<version>1.1.1</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.github.eirslett</groupId>

<artifactId>frontend-maven-plugin</artifactId>

<version>1.6</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

</dependency>

<dependency>

<groupId>org.webjars</groupId>

<artifactId>bootstrap</artifactId>

<version>3.3.6</version>

</dependency>

<dependency>

<groupId>org.webjars</groupId>

<artifactId>bootstrap-datepicker</artifactId>

<version>1.0.1</version>

</dependency>

<dependency>

<groupId>org.webjars</groupId>

<artifactId>jquery</artifactId>

<version>1.9.1</version>

</dependency>

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-tomcat</artifactId>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

<repositories>

<repository>

<id>spring-milestones</id>

<name>Spring Milestones</name>

<url>https://repo.spring.io/milestone</url>

<snapshots>

<enabled>false</enabled>

</snapshots>

</repository>

<repository>

<id>spring-snapshots</id>

<name>Spring Snapshots</name>

<url>https://repo.spring.io/snapshot</url>

<releases>

<enabled>true</enabled>

</releases>

</repository>

</repositories>

<pluginRepositories>

<pluginRepository>

<id>spring-milestones</id>

<name>Spring Milestones</name>

<url>https://repo.spring.io/milestone</url>

<snapshots>

<enabled>false</enabled>

</snapshots>

</pluginRepository>

<pluginRepository>

<id>spring-snapshots</id>

<name>Spring Snapshots</name>

<url>https://repo.spring.io/snapshot</url>

<releases>

<enabled>false</enabled>

</releases>

</pluginRepository>

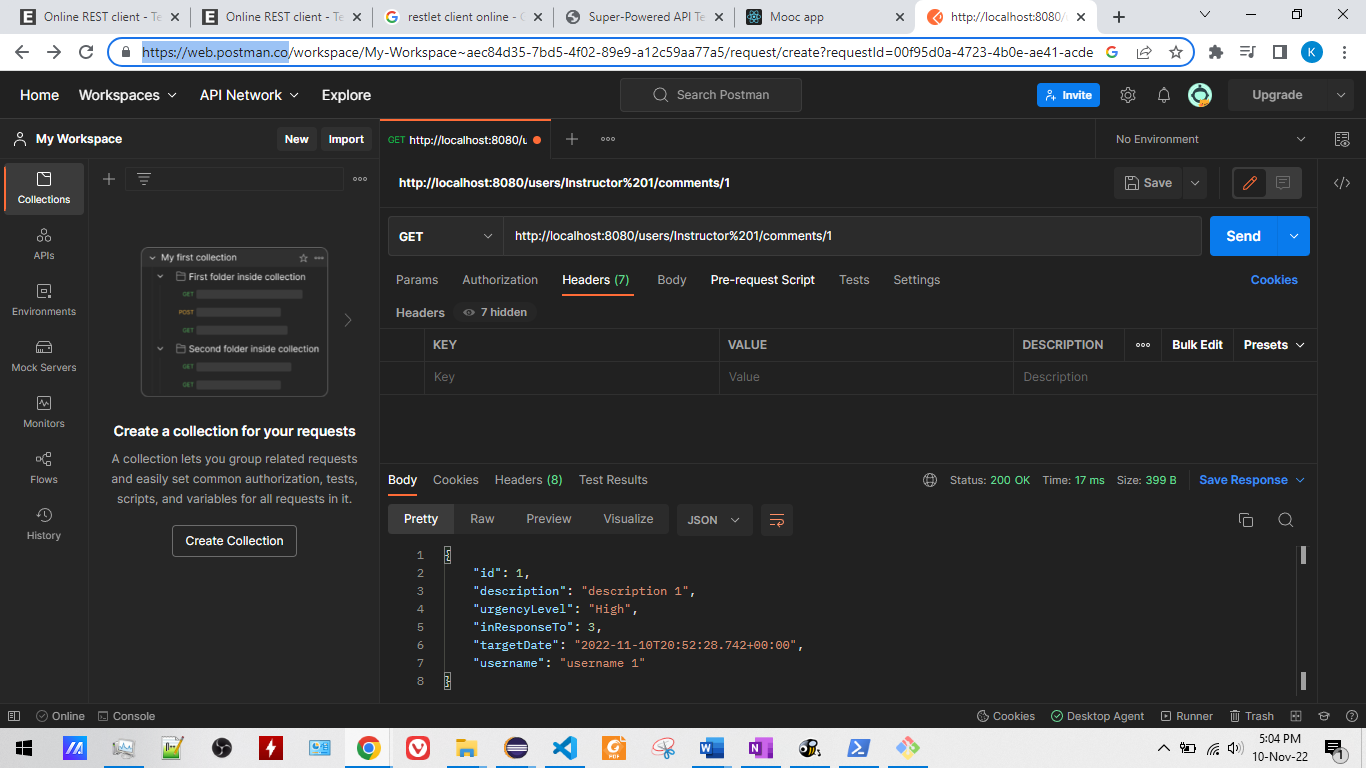
</pluginRepositories>

</project>

# How to use http client

Go to <https://web.postman.co/>

View <https://www.udemy.com/course/full-stack-application-with-spring-boot-and-react/learn/lecture/14018722#overview> for how to create a POST request and test it out.



# When Hibernate keeps inserting records with pk starting from 1 whenever it restarts even though there’s already records inside.

<https://stackoverflow.com/questions/2011528/hibernate-auto-increment-id>

Use

@Id

@GenericGenerator(name="startfromlargestid" , strategy="increment")

@GeneratedValue(generator="startfromlargestid")

Annotations at PK column of entity

# Filename too long git bash

<https://stackoverflow.com/questions/52699177/how-to-fix-filename-too-long-error-during-git-clone>

# Run docker in cmd

Text

Description automatically generated

Docker container list

Docker container stop dc7e5ef4fc7d

Docker rm dc7e5ef4fc7d

docker run --detach --env MYSQL\_ROOT\_PASSWORD=dummypassword --env MYSQL\_USER=todos-user --env MYSQL\_PASSWORD=dummytodos --env MYSQL\_DATABASE=todos --name mysql --publish 3306:3306 mysql:5.7

shift + ctrl +c to copy code from terminal, right click to paste

# Check mysql running using mysql shell (search for program)

\connect todos-user@localhost:3306

dummytodos

\use todos

\sql

Select \* from comment

# Creating AWS EBS environment

\*\*USE JAR FILE

Just enter environment name and select mooc-app-jar app. Select tomcat. “Additional configuration” use tomcat lowest version, and below username and password.

todosuser

dummypass

# AWS deploying static front end

https://www.udemy.com/course/deploy-java-spring-boot-to-aws-amazon-web-service/learn/lecture/15443858#overview

# If “Docker engine stopped” error or runtime error when launching Docker desktop program

Go to Services manager and start DokcerDesktopService manually

# How to create jar/war file

Ensure below block is below <package> block in pom.xml

<groupId>com.mooc-app.rest.webservices</groupId>

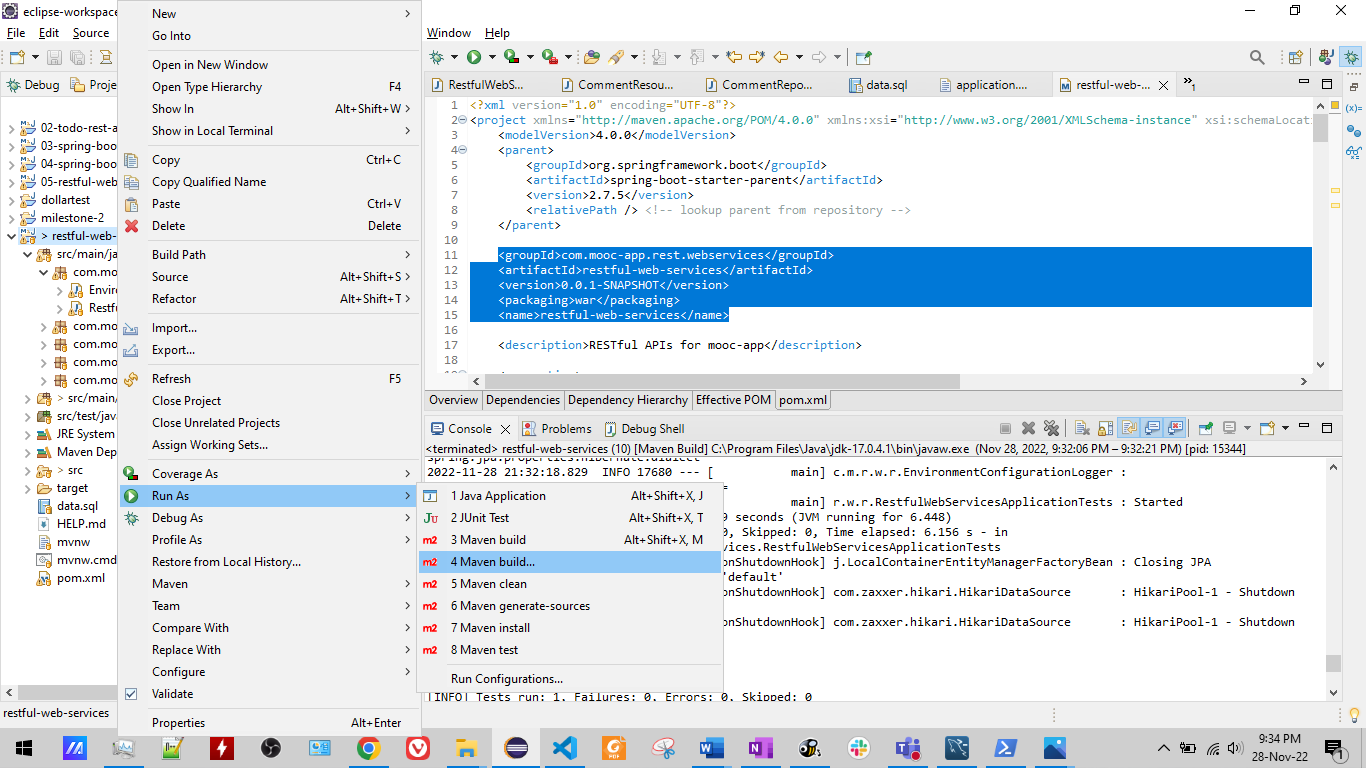
<artifactId>restful-web-services</artifactId>

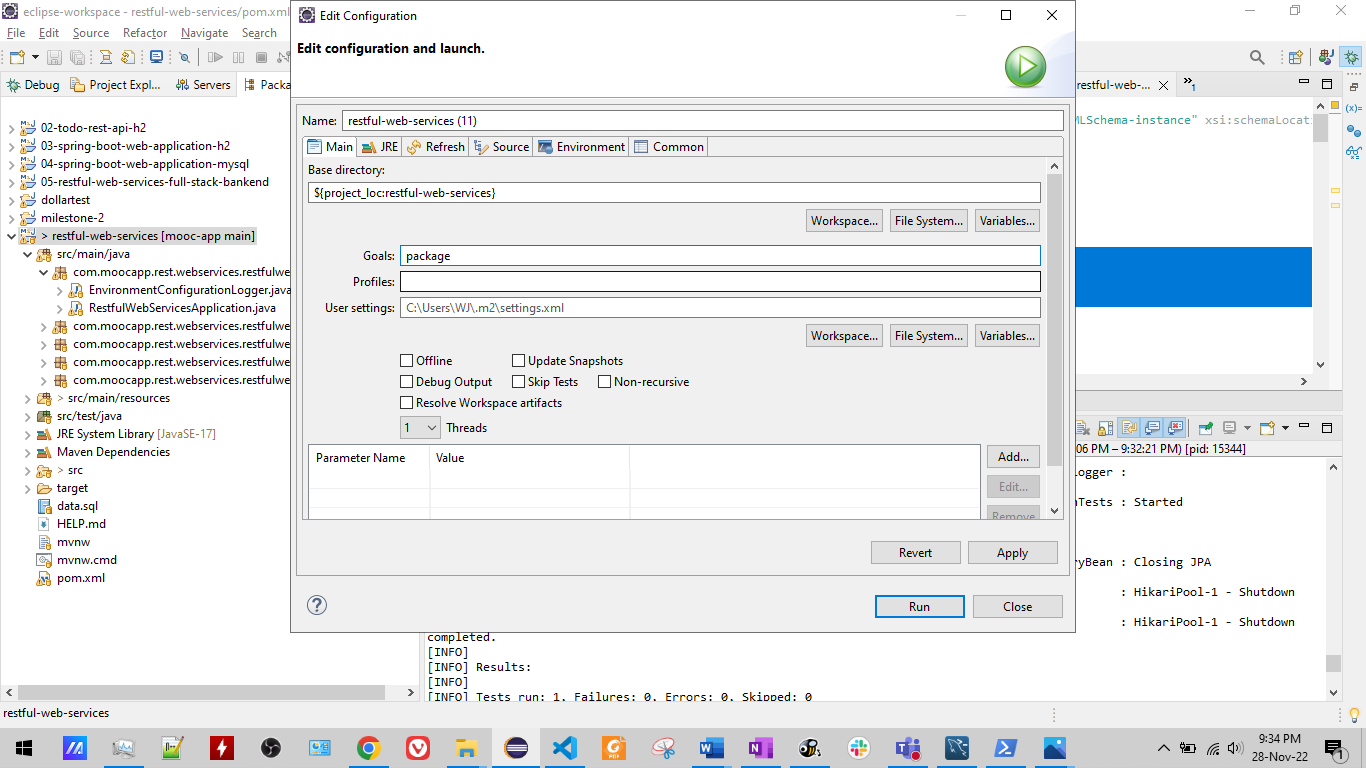
<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

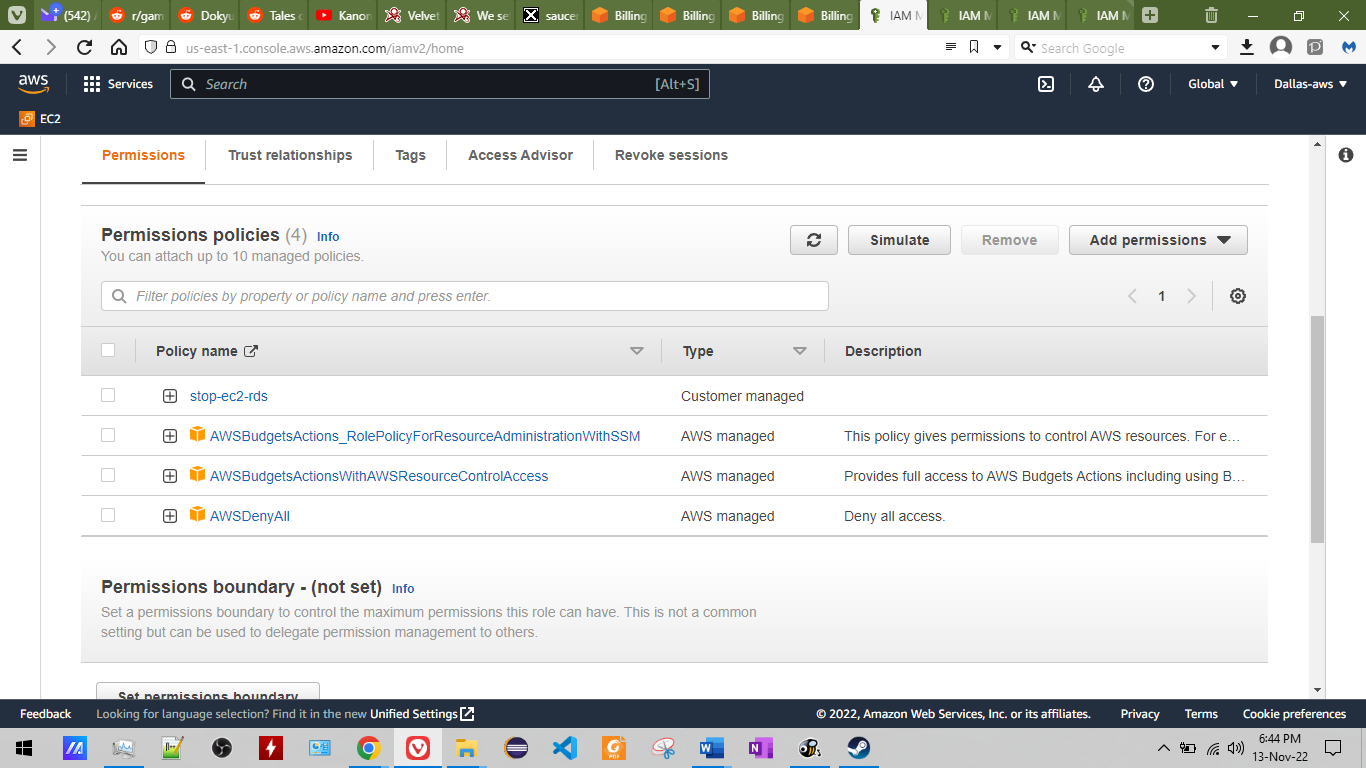
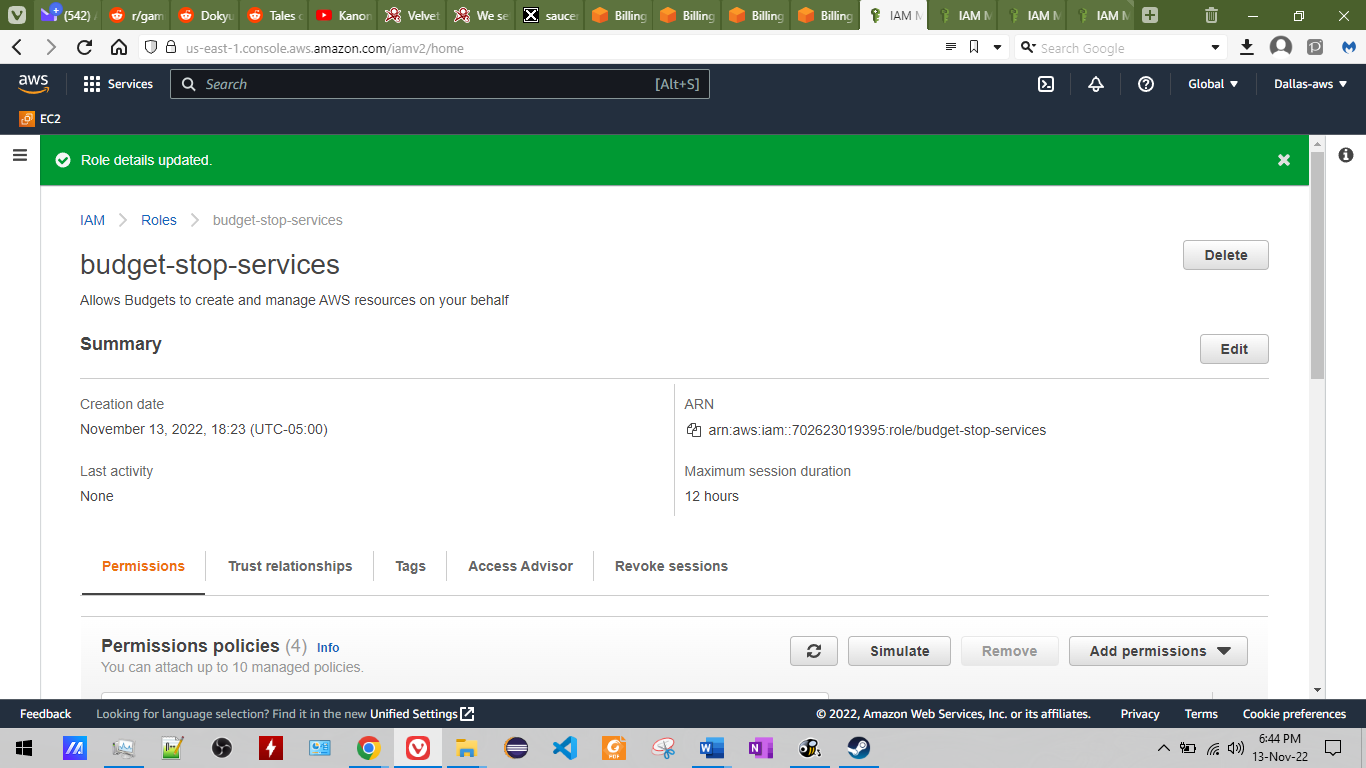
<name>restful-web-services</name>

Change value of <package> tag to JAR if you want to create JAR file.

Then “run as” “maven build” enter *package* in the *goals* field.



# Stuck at update loop page when setting budgets

Crete IAM role with below permissions. For Trust relationship tab and the custom stop-ec2-rds permission policy look below.

**Edit Trust relationships tab of the IAM role to this:**

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Principal": {

"Service": "budgets.amazonaws.com"

},

"Action": "sts:AssumeRole",

"Condition": {

"ArnLike": {

"aws:SourceArn": "arn:aws:budgets::123456789012:budget/\*"

},

"StringEquals": {

"aws:SourceAccount": "123456789012"

}

}

}

]

}

**Create a custom Policy with this and attach to the IAM role:**

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"ec2:DescribeInstanceStatus",

"ec2:StartInstances",

"ec2:StopInstances",

"iam:AttachGroupPolicy",

"iam:AttachRolePolicy",

"iam:AttachUserPolicy",

"iam:DetachGroupPolicy",

"iam:DetachRolePolicy",

"iam:DetachUserPolicy",

"organizations:AttachPolicy",

"organizations:DetachPolicy",

"rds:DescribeDBInstances",

"rds:StartDBInstance",

"rds:StopDBInstance",

"ssm:StartAutomationExecution"

],

"Resource": "\*"

}

]

}

Relevant links:

<https://docs.aws.amazon.com/cost-management/latest/userguide/budgets-controls.html> (what to do to create IAM role to allow budget actions)

<https://docs.aws.amazon.com/cost-management/latest/userguide/billing-example-policies.html#example-budgets-applySCP> (correct section from previous link)

<https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_manage-attach-detach.html#embed-inline-policy-console> (creating a custom policy)

<https://docs.aws.amazon.com/IAM/latest/UserGuide/roles-managingrole-editing-console.html#roles-managingrole_edit-trust-policy> (editing the trust relationships tab)

# 404 not found

<https://stackoverflow.com/questions/39670615/spring-boot-tomcat-on-aws-elastic-beanstalk-only-showing-404-page>

# Postman

JSON data to post in body:

http://moocappjar.eba-if84mhyz.us-east-1.elasticbeanstalk.com/users/Instructor%201/comments

{

        "description": "test-aws",

        "urgencyLevel": "1",

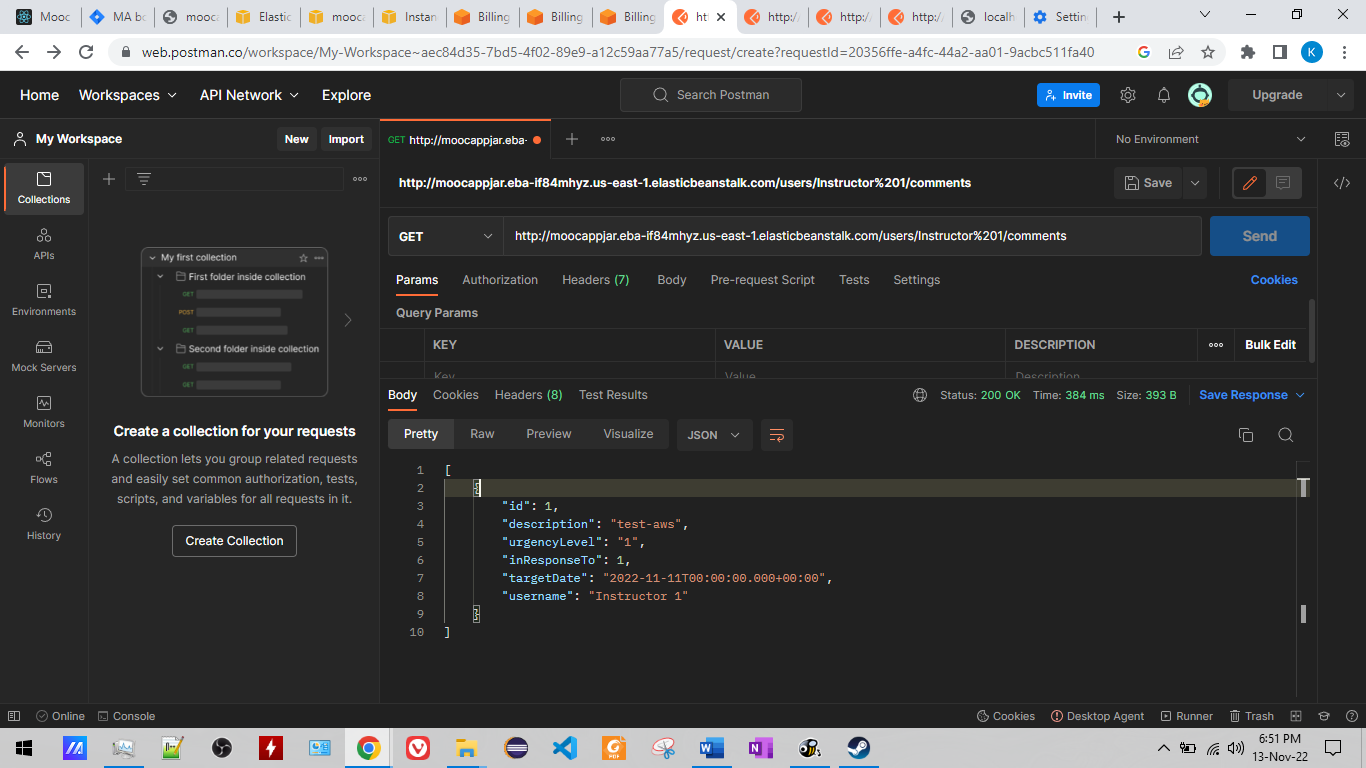
        "inResponseTo": 1,

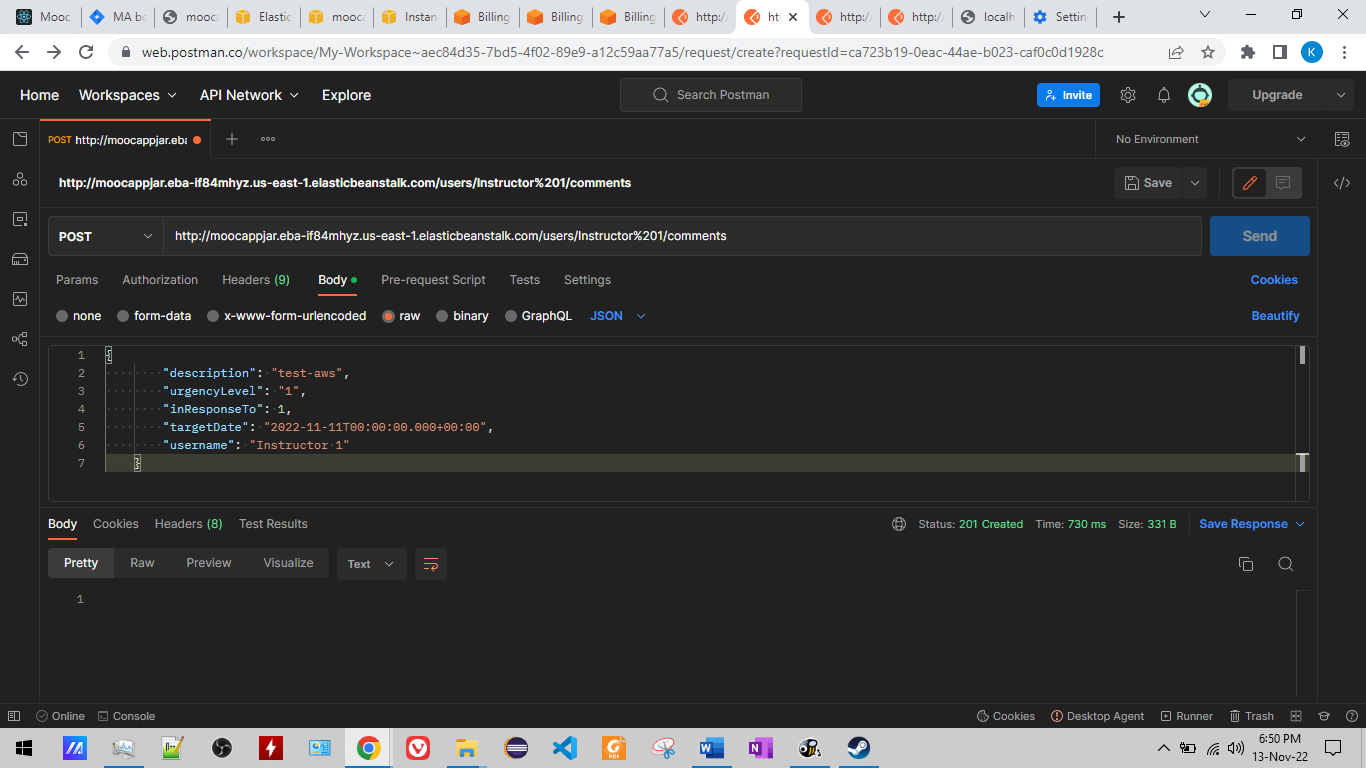
        "targetDate": "2022-11-11T00:00:00.000+00:00",

        "username": "Instructor 1"

    }

http://moocappjar.eba-if84mhyz.us-east-1.elasticbeanstalk.com/users/Instructor%201/comments





# Stashing changes

<https://www.atlassian.com/git/tutorials/saving-changes/git-stash#stashing-your-work>

# AWS CORS no “Access-Control-Allow-Origin” error

Include below code in main RestfulWebServicesApplication class:

// AWS

@Bean

**public** WebMvcConfigurer corsConfigurer() {

**return** **new** WebMvcConfigurer() {

@Override

**public** **void** addCorsMappings(CorsRegistry registry) {

registry.addMapping("/\*\*").allowedMethods("\*").allowedOrigins("\*");

}

};

}

**Edit bucket policy in S3:**

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "AddPerm",

"Effect": "Allow",

"Principal": "\*",

"Action": "s3:\*",

"Resource": "arn:aws:s3:::mooc-app-jar/\*"

}

]

}

**Edit CORS of bucket under “Permissions” tab:**

[

{

"AllowedHeaders": [

"\*"

],

"AllowedMethods": [

"GET",

"PUT",

"POST",

"DELETE"

],

"AllowedOrigins": [

"\*"

],

"ExposeHeaders": []

}

]

Google: spring boot add response header access-control-allow-origin

<https://stackoverflow.com/questions/51802102/spring-boot-security-no-access-control-allow-origin-header-is-present-on-the-r>

https://stackoverflow.com/questions/59300723/java-spring-boot-access-control-allow-origin-not-working

https://stackoverflow.com/questions/46065156/access-control-allow-origin-with-spring-boot

https://stackoverflow.com/questions/35091524/spring-cors-no-access-control-allow-origin-header-is-present

<https://stackoverflow.com/questions/67838960/configuring-cors-with-nginx-and-aws-elastic-beanstalk>

<https://stackoverflow.com/questions/44617825/passing-headers-with-axios-post-request>

<https://docs.aws.amazon.com/AWSEC2/latest/APIReference/cors-support.html>

# Error 404 not found code no such key aws s3 ebs when refreshing

<https://stackoverflow.com/questions/49034229/amazon-aws-s3-404-error-on-page-refresh>

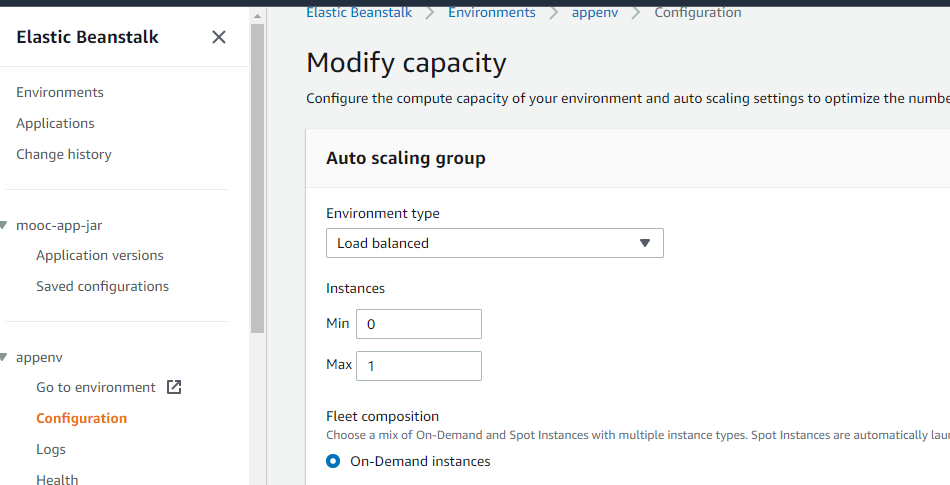
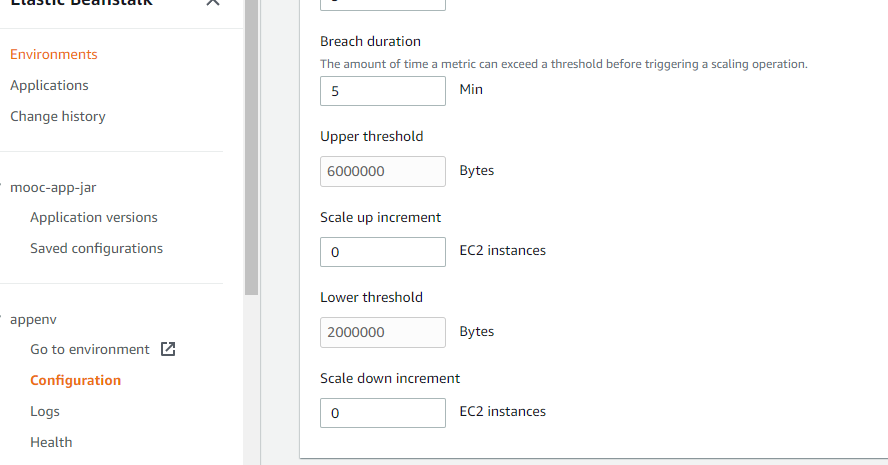
# (budgets, EBS environment) how to prevent EC2 instance from being terminated once automatically stopped by alerts from budget.

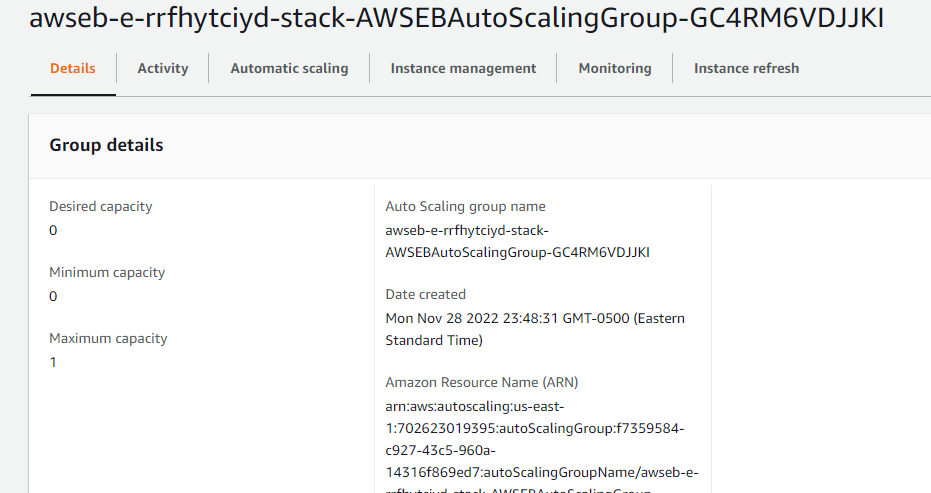
Detach instance from auto-scaling group that was automatically created when you created the EBS environment. Remember to “save” budget alerts again to have it trigger actions.

<https://stackoverflow.com/questions/19194893/why-does-my-aws-ec2-instance-terminates-when-stopped>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/detach-instance-asg.html>

In Capacity configuration of EBS environment, configure as below:

 If above fails, then try setting min instances to 1, , scale up/down increment to 1, -1 then let an instance be launched first, then change min instance to 0, scale up/down increment to 0 to prevent a new instance from being launched when current one will be stopped by AWS budget alert actions.

Remember to change auto scaling group settings to below to:

Maybe try using in28minutes project 5 backend and then paste your code and files inside?

# Setting up app on ec2 instead of through Elastic beanstalk

https://cloudkatha.com/how-to-deploy-spring-boot-application-on-aws-ec2/

https://asf.alaska.edu/how-to/data-recipes/moving-files-into-and-out-of-an-aws-ec2-instance-windows/

# UDEMY

Section 7 is about integrating MySQL DB with REST API, but its front end is a jsp. Need this to make sure DB runs correctly.

Section 8 ‘s front end is a react project , but it’s REST API uses H2 DB. Need this to make sure CORS runs correctly.

# React scripts start is not a command node js

Make sure package.json: "start": "set PORT=4200 && react-scripts start",

Delete package-lock.JSON

Delete node\_modules

Npm install again, then npm audit fix –force, then try npm run start even if got issues.

# [**'react-scripts' is not recognized as an internal or external command**](https://stackoverflow.com/questions/47928735/react-scripts-is-not-recognized-as-an-internal-or-external-command) error removing certain .bin directories under node\_modules

Delete “node\_modules” directory under root folder C:\Users\WJ\Desktop\Georgia Tech\Fall 2022\CS 8803 MAS (Special Topics) - Mobile Applications and Services\Assignment\mooc-app\mooc-app . Then “npm install” at same directory with administrator privileges

https://stackoverflow.com/questions/47928735/react-scripts-is-not-recognized-as-an-internal-or-external-command

# [**'react-scripts' is not recognized as an internal or external command**](https://stackoverflow.com/questions/47928735/react-scripts-is-not-recognized-as-an-internal-or-external-command) error removing certain .bin directories under node\_modules

Delete “node\_modules” directory under root folder C:\Users\WJ\Desktop\Georgia Tech\Fall 2022\CS 8803 MAS (Special Topics) - Mobile Applications and Services\Assignment\mooc-app\mooc-app . Then “npm install” at same directory with administrator privileges

https://stackoverflow.com/questions/47928735/react-scripts-is-not-recognized-as-an-internal-or-external-command

# Mysql connection failed when building war (how to skip tests when building war in eclipse)

<https://stackoverflow.com/questions/8923002/how-to-configure-maven-install-to-skip-tests-in-eclipse>

# Steps

Use 05-spring-boot-react-full-stack-h2 as base project.

Add mysql dependency in pom.xml

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

</dependency>

Remove H2 dependency in pom.xml

Add mysql settings in application.properties:

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

hibernate.id.new\_generator\_mappings =false

spring.jpa.defer-datasource-initialization: true

spring.sql.init.mode= always

spring.jpa.show-sql=true

#spring.h2.console.enabled=true

#spring.h2.console.settings.web-allow-others=true

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://${RDS\_HOSTNAME:localhost}:${RDS\_PORT:3306}/${RDS\_DB\_NAME:todos}

spring.datasource.username=${RDS\_USERNAME:todos-user}

spring.datasource.password=${RDS\_PASSWORD:dummytodos}

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL57Dialect

Remove h2 settings in pom.xml

Delete all JWT packages

Remove spring security dependency in pom.xml

Comment out pringSecurityConfigurationBasicAuth.java and BasicAuthenticationController

Add spring.datasource.initialization-mode=always to application.propreties

(by now, POSTMAN test in local and in Elastic Beanstalk with Java environment and lowest MySQL version succeeds, remember to create JAR by skipping tests)

Copy over files from the Feedback, Comment and their subpackages, organize imports and package name

Update data.sql to

/\*https://www.browserling.com/tools/bcrypt Use Rounds 10\*/

**delete** **from** todo;

**delete** **from** comment;

**delete** **from** feedback;

--/\*in28minutes/dummy\*/

--INSERT INTO USER (ID, USERNAME, PASSWORD, ROLE)

--VALUES (1, 'in28minutes', '$2a$10$3zHzb.Npv1hfZbLEU5qsdOju/tk2je6W6PnNnY.c1ujWPcZh4PL6e','ROLE\_USER');

--

--/\*in28minutes2/mypassword\*/

--INSERT INTO USER (ID, USERNAME, PASSWORD, ROLE)

--VALUES (2, 'in28minutes2', '$2a$10$i9AckmxMkb4yKtLCdxeQheCm2pXWB3qZ2G189/Ph/DUci1DvLO.Rq','ROLE\_USER');

--

**insert** **into** todo(id, username,description,target\_date,is\_done)

**values**(10001, 'in28minutes', 'Learn JPA', sysdate(), **false**);

**insert** **into** todo(id, username,description,target\_date,is\_done)

**values**(10002, 'in28minutes', 'Learn Data JPA', sysdate(), **false**);

**insert** **into** todo(id, username,description,target\_date,is\_done)

**values**(10003, 'in28minutes', 'Learn Microservices', sysdate(), **false**);

**insert** **into** comment(id, description,urgency\_level,in\_response\_to, target\_date,username, votes)

**values**(1, 'description 1', '3', 0, LOCALTIMESTAMP - INTERVAL '30' MINUTE, 'Student 1', 1);

**insert** **into** comment(id, description,urgency\_level,in\_response\_to, target\_date,username, votes)

**values**(2, 'description 2', '2', 1, LOCALTIMESTAMP - INTERVAL '30' MINUTE, 'Student 2', 1);

**insert** **into** comment(id, description,urgency\_level,in\_response\_to, target\_date,username, votes)

**values**(3, 'description 3', '1', 2, LOCALTIMESTAMP - INTERVAL '30' MINUTE, 'Student 3', 1);

**insert** **into** feedback(id, feedback\_comment,feedback\_rating, lesson\_id)

**values**(1, 'feedback comment 1', 1, 1);

**insert** **into** feedback(id, feedback\_comment,feedback\_rating, lesson\_id)

**values**(3, 'feedback comment 2', 2, 1);

**insert** **into** feedback(id, feedback\_comment,feedback\_rating, lesson\_id)

**values**(2, 'feedback comment 3', 3, 1);

**insert** **into** feedback(id, feedback\_comment,feedback\_rating, lesson\_id)

**values**(4, 'feedback comment 4', 3, 2);

**insert** **into** feedback(id, feedback\_comment,feedback\_rating, lesson\_id)

**values**(5, 'feedback comment 5', 3, 2);

Localhost:4200 to AWS EB should be working now. Remember, base project is 05.