Rest of the content is already explained in the Readme.md file. Here, we are doing the Github Action setup as we are providing the docker username and password that is required in yml ile:

Graphical user interface, text, application, email

Description automatically generated

line no, 42, we need docker username and password. That setup we will configure here.

Steps:

1. Go to your repo and go under security.

Graphical user interface, text, application, email

Description automatically generated

1. Under security tab, click on action under “secrets and variables.”

Graphical user interface, text, application, email

Description automatically generated

1. Click on “new repository secret.”

Graphical user interface, text, application, email

Description automatically generated

1. Provide the same variable name defined in yml file:

Graphical user interface, text, application, email

Description automatically generated

1. Add both username and password for docker hub:

Graphical user interface, text, application, email

Description automatically generated

Once this is done, we need to create **“.github/workflows/maven.yml**” file.

**Working maven.yml file:**

name: project cicd flow

on:

push:

branches: [ feature/kubernetes-exercise-1 ]

pull\_request:

branches: [ feature/kubernetes-exercise-1 ]

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v2

- name: Set up JDK 17

uses: actions/setup-java@v1

with:

java-version: '17'

distribution: 'adopt'

cache: maven

- name: Build with Maven

run: mvn clean install

- name: Test cases execution

run: mvn clean test

#if condition, if the branch is equal to main then create the docker image for it and push to docker hub

#bookmarker-api-jib:latest image name defined in docker push command below will pick from image

# name defined in pom. file in JIB plugin. If you want to user dockerFile then you should define

# file path but here we are using JIB so it's picking from pom.

- if: ${{ github.ref == 'refs/heads/feature/kubernetes-exercise-1' }}

name: Build and Publish Docker Image

run: |

echo "This job runs in custom branch"

./mvnw clean package jib:dockerBuild -DskipTests

docker login -u ${{ secrets.DOCKER\_USERNAME }} -p ${{ secrets.DOCKER\_PASSWORD }}

docker push ${{ secrets.DOCKER\_USERNAME }}/bookmarker-api-jib:latest

Make sure spring boot project folders like this:

Graphical user interface, website

Description automatically generated

Earlier, I had folder like “Documents/spring-boot-projects/branch name” and due to that my Github action was not working.

The image name we defined online

no 37, picked from pom.xml file jib plugin:

Text, letter

Description automatically generated

Timeline

Description automatically generated

Once that is done, you are good to go….

Graphical user interface, text, application, email

Description automatically generated

16. Now, we will add post method for bookmark. For bookmark method, we are not giving all details. We are passing only 2 arguments so that's why I have created the "BookmarkRequest" class for taking the input and add @Valid @NotEmpty annotation.

17. Now, if there is something error in your payload then it will through big error, that’s good

but we need an accurate error due to which our post method is failed. So, we need to add one annotation.

**to fix it:**

<problem-spring-web.version>0.27.0</problem-spring-web.version>

**<dependency>**

**<groupId>org.zalando</groupId>**

**<artifactId>problem-spring-web-starter</artifactId>**

**<version>${problem-spring-web.version}</version>**

**</dependency>**

Before applying this annotation, the error will be like this:

Graphical user interface, text, application

Description automatically generated

**After adding this dependency:**

**Graphical user interface, text, application, email

Description automatically generated**