Testing

Testing : Testing is use to find the defects or bugs or errors in application.

Layer architecture or model architecture

Node JS (Express JS using MVC Style).

2 types

Black box testing

Input Process Output

White box testing

Input ------------------------🡪Process -------------------🡪Output

Unit testing : it is a type of white box testing. Unit the smallest price of code to do some specific task written in function or method or procedure or module.

**Jasmine :** Jasmine is one of the type of open source library which provide set of API which help to do the testing for JavaScript application (JavaScript program can be client or server side).

Testing Client side JavaScript Programs using Jasmine

Jasmine Provide set of pre-defined function

1. Test suite : Test suite is use or combine more than one test case.

To make the test suite Jasmine framework provide pre-defined function ie

describe(“Message”,callbackfunction)

callbackfunction we can use in expression style or arrow style

1. Test case : Test case is a type of function which help do the testing for the specific function.

To make the test case Jasmine provide pre-defined function ie it()

It(“message”,callbackfunction)

1. Expect : Jasmine provide lot of pre-defined function in the form of expect which help to check actual and expectation output.

Describe : it can contains more than one it

Every it can contains more than one expect which help to check actual and expectation output.

Jasmine provide pre-defined function to make suite and spec and expectation.

To run application we were depends upon the Karma. Karma is known as test runner to run the application.

Karma is a tool which help to provide the result ie test may be pass or fail.

But new version Jasmine provide own runner to run the application.

21-11-2021

Jasmine testing hook functions

describe(“Message”,()=> {

beforeAll(()=> {

call only once before all it function

})

beforeEach(()=> {

any initialization for every it function it get call

})

it(“1st Testing”,()=> {})

It(“2nd Testing”,()=> {})

afterEach(()=> {

closing any resource after each testing function

})

afterAll(()=> {

it call only once after all it function.

})

})

Testing for the Angular Application

Angular Framework internally provide Jasmine and Karma for testing Angular component, angular template and angular service classes.

In Angular

Jasmine : It is a open source framework provide set of function ie describe, it and more than one expect functions.

Karma : Karma is test runner which is use display the result in browser.

Angular Utilities : Angular framework provided set of API which help to test the Angular component, template and service.

All API for Angular part of

@angular/core/test these modules.

TestBed Angular API

to test the application we have to use the command as

ng test

component testing :

property

function

template testing:

tags values

service testing

ng g s employee

setTime(()=> {

service.loadEmployeeInfo().subscribe(result=> {

write more than one expect

})

done();

},2000)

Testing for Backend technologies ie

Node JS

If we are planning to do testing for node JS

npm install jasmine

npm install jasmine-node

First create the package.json file

npm init command

using jasmine command we have to run one command ie

jasmine init

This command use to create spec directory and configuration json file for testing. This spec directory is use to store all our testing files.

Jasmine : JavaScript

Angular

Node JS

Mocha with Chai -🡪 Node JS

JEST : React JS

Selenium : End to End Testing : Java, JavaScript, Python etc

Testing for Express JS Application for Get, Post method

Supertest : it is a type of external module which help to do the Testing for Express JS Application.

Jasmine with SuperTest is use to do the Testing for Express JS or Http module testing.

Create the Folder Express JS Application folder

Please create the package.json file

Using npm init

npm install express

npm install jasmine –D

npm install jasmine-node –D

npm install supertest –D

jasmine init (This command is use to create spec folder)

28-11-2021

Docker: Docker is an advanced OS virtualization software platform that makes it easy to create , deploy and run the application in Docker Container.

The Docker container is a very light weighted (takes less resource or memory space of our machine) package that allows the developer to package up an application and deploy it as one with help of inbuilt libraries and other dependencies.

Docker is also known as engine.

Virtualization : Virtualization is the means of employing software (Such as hypervisor) to create a virtual machine or version of resources such as a server, data storage, data base or application.

Virtualization let us divided a system into series of separate section, each one acting as a distinct individual system. The virtual environment is known as virtual machine.

Window, Linux or Unix or Mac etc.

VM Ware software

VM ware software allow to run the guest OS in Base machine.

My Base Machine is Window 10

With RAM 16 RAM and memory 1tb

If I am planning to run Guest OS on VM ware

Mac or Unix or Linux etc

4 GB RAM for Guest OS and 50 GB Memory space.

12 GM

I want to run 10 Guest OS in base machine.

1GM

Virtual Machine : A VM is a computing environment or software that aids developer to access an operating system via physical machine.

VM Ware ; This type of software is used to do Virtualization. Virtualization is an abstract version of an physical machine.

Docker is use to achieve containerization. Containerization is the abstract version of an application. Application can be java, node , python or angular etc.

Docker Container : Running instance of Docker image, container turn the actual application or run the actual application. The Container includes an application and all of its dependencies.

Docker Image : It is a template that holds set of instruction needed to create a working container.

Or The file system and configuration of our application which are used to create the container.

Docker file : A Docker file is a blue print that defined how our image to build. It is a series of steps that you have to defined to create the images.

Docker Daemon : Docker Daemon is use to run background service of Docker engine. It manages building, running, and distributing Docker container.

Docker registries : It is use to store Docker images. There are two type of registry

Private registry

Public registry

Docker hub : it is like a git hub which help to publish the images in registry ie may be public or private.

In Docker hub you can create only one private registry and more than one public registry.

So if we publish our image in Docker any other person pull that image from Docker and run the image ie run the application without installing any software in their machine.

Open the command prompt

Docker commands

docker --version : This command is use to check the version of docker

docker images : This command is use to check all images in docker in local images

hello-world image

docker pull hello-world : This command pull the image from Docker hub.

docker run imageName/imageId : This command is use to run the image

busybox :busybox is a small tiny unix base OS.

docker run -it busybox (it means iterative mode)

if you want to create Docker image we have to take the help of docker file

file name must be

Dockerfile (without extension).

Create simple image to run simple data command

**Dockerfile**

FROM busybox:latest

CMD ["date"]

docker build -t my-date-image . -f Dockerfile.txt

Creating image to run the node Application

Dockerfile

FROM node:latest

COPY app.js .

CMD ["node","app.js"]

docker build -t my-node-js . -f Dockerfile

docker run my-node-js

Creating Docker image to run The express JS application.

Create the folder.

Create the package.json file using npm init

npm install express

Now create app.js file and create more than one REST API.

Dockerfile

FROM node:latest

RUN mkdir /usr/src/app

WORKDIR /usr/src/app

COPY package.json /usr/src/app

RUN npm install

COPY app.js /usr/src/app

CMD ["node","app.js"]

docker build -t my-express-123 . -f Dockerfile

If images contains to run the web application we have to run the command as

docker run –p 9090:9090 image-name

docker run –p 9191:9090 image-name

right side or red color port number is actual server port number

left side or green color port number is expose port number. Expose port number can be same or different.

docker run –d –p 9191:9090 image-name

-d detached mode

-p port number

If you want to find the container details then you can use the command as

docker ps (process status)

to stop the container we have to use the command as

docker stop containerId

delete the container

docker rm containerid

if you get the error then we have to stop and then remove

docker stop containerId

docker rm containerId

or

docker rm –f containerId

docker ps –a : This command display stop container list.

Command to delete the Docker image

docker rmi imageName/imageId

if image link with container then we will get the error.

First stop the container and delete the container and then delete the image or else

Delete the image force fully.

docker rmi –f imageName/imageId

Creating the image for web application ie html, css and js

First create the folder and create htm page, css and js file

**What is nginx?**

Nginx (pronounced "engine-x") is an open source reverse proxy server for HTTP, HTTPS, SMTP, POP3, and IMAP protocols, as well as a load balancer, HTTP cache, and a web server (origin server).

Nginx serve by default run on port number 80

Dockerfile

FROM nginx:latest

COPY . /usr/share/nginx/html

docker build -t my-html-123 . -f Dockerfile

Create the angular project

Angular application running on web server provided by google on port number 4200.

This application running on development environment.

After development we have to build the project and this build file we have to give production environment server ie tomcat, web logic, jboss, IIS Server etc.

Once development task finish we have to stop the project and we have build the project using commands as

ng build

After this command in project folder it will create dist folder.

Inside dist folder it will create another folder with project name and inside that folder all build file are present.

Dockerfile

FROM nginx:latest

COPY . /usr/share/nginx/html

docker build -t my-angular-123 . -f Dockerfile.txt

Now we will publish the image in docker hub

Before publish the image we have create the tag or provide the tag for that image.

Tag is just like a identity for the image.

docker tag imageName dockerhubaccountId/imageName:latest

docker tag my-angular-123 akashkale/my-angular-123:latest

after created tag now we can publish this image in Docker hub.

docker push dockerhubaccountid/imageName

docker push akashkale/my-angular-123

Jenkin

docker pull jenkins/Jenkins

After pull Jenkin image in local machine to run the Jenkin we have to use the command as

docker run -p 8080:8080 -p 50000:50000 jenkins/jenkins:lts-jdk11

After running successfully

<http://localhost:8080>

it will ask the password please copy and paste the password from command prompt

and first time install the suggested plugins.