**MEAN Stack Training**

**Day 1**

**11-02-2022**

Mongo DB / MySQL Express JS Angular Node JS

Mongo DB / MySQL Express JS React JS Node JS

Mongo DB / MySQL Express JS Vue JS Node JS

https://[www.google.com](http://www.google.com) URL :

Uniform Resource Locator

http or s : protocol : hyper text transfer protocol secure

www : world wide web

google : domain

com : commercial

req(https/htt)------🡪

Client Server

🡨----res(http/https)------ html/html5

Html -🡪 display the content on browser

Css/css3

Css 🡪 it is use to display the content in proper format or presentation logic.

JS

JS -🡪 it is use to do programming on web page.

IDE :

VS Code :

HTML : Hyper text Mark up language

Html is use to create the web page.

Web page mainly use to display the content on browser in different format.

Html provide lot of pre-defined tags or elements.

Syntax

<tagName> opening tag

</tagName> closing tag

Html is not a case sensitive.

Html tags

1. Html
2. Head
3. Body

<html>

<head>

<title>This is my web page</title>

</head>

<body>

<p>Welcome to My Web Page</p>

</body>

</html>

.html or .html

1. Title tag
2. Paragraph tag p
3. Break tag br
4. Heading tag 6 heading
   1. H1 to h6

H1 is largest and h6 smallest.

attribute : attribute is known a properties of a tags.

Every tag contains one or more attribute in the form of key-value or name-value pairs. Attribute we have to use in opening tag. Value can be single or double or without quote.

Syntax

<tagName name1=”value1”> </tagName>

<tagName name1=”value1”> </tagName>

**Font tag** : This tag is use to change the color, size and style (face) of contents.

Hyper link : This tag is use to connect one web page to another web page or it is use to create book mark.

1. External hyper link
2. Internal hyper link or book mark

External hyper link

<a href=”pageName.html”>Text</a>

a anchor tag

href : hyper reference.

Add the images

<img src=”imageName.jpeg/gif/” />

List tags

It is use to display the item in proper format.

Unorder list

Order list

Table

**Id Name Age**

100 Ravi 21

101 Ramesh 22

table

tr 🡪 table row

th 🡪 table heading

tr🡪 table row

td 🡪 table data

Forms tag

Login Page

<form>

</form>

Html textfield syntax

<input type=”text/password/radio/checkbox/button/submit/reset/file” />

Before submit form

<file:///C:/Users/91990/Desktop/Real%20Variable%20MEAN%20Stack/2022%20-%20MEAN%20Stack%20Training%20Real%20Variable%20Client/Programs/HTML%20Programs/login.html>

After submit form

<file:///C:/Users/91990/Desktop/Real%20Variable%20MEAN%20Stack/2022%20-%20MEAN%20Stack%20Training%20Real%20Variable%20Client/Programs/HTML%20Programs/home.html?user=Ravi&pass=12345>

<file:///C:/Users/91990/Desktop/Real%20Variable%20MEAN%20Stack/2022%20-%20MEAN%20Stack%20Training%20Real%20Variable%20Client/Programs/HTML%20Programs/home.html?user=Raj&user=Deep&pass=1234>

By default every html form send data through

url using re-writing technique if method is get.

URL?key1=value1&key2=value2&key3=value3

If method is get it is not secure.

If you want to send the data through body part we can use post method.

Post method slower than get method.

Using get we can send maximum 255 character data.

Html 4

<!doctype html url =”url………dtd”/>

Document type definition

Root tag

Number of child tags

Optional tags.

Html5

<!doctype HTML/>

Online Shopping

index.html

6 pages

Banner and images

login page

Username TextField

Password PasswordField

Submit Reset

Home Page

Link1 Link2 Link3 Link4

**Day 2**

**14-02-2022**

CSS : Cascading Style sheet : CSS provide set of properties and values which help to apply good look and feel for web page. With html it is not possible or may be code become more complex.

With help of CSS we can achieve separation of concern.

Types of CSS

1. Inline css
2. Internal css or embedded css
3. External css

Inline CSS syntax

<tagName style=”property:value;property:value;”>

</tagName>

difference between tag and attribute.

Attribute is properties of tags.

<tag key1=”value1”> : root tag as well as complex tag

<child1>

<child2>Hi</child2> : simple tag

<child3>Hello</child3>

<child4>How r you</child4>

</child1>

</tag>

Div tag : it is known as container tag. Which is use to add more than one tag ie p, h1 to h6 as well as another div. Div tag is use to specify the section of web page.

Internal css or embedded css

Syntax

<style type=”text/css”>

</style>

We have to write this style tag in between head tag.

selector {property:value;property:value;}

types of selector

1. Universal selector : \* {property:value}
2. Specific selector : tagName {property:value}

p{color:red}

h1{color:green}

1. Multi specific selector : tagName,tagName,tagName{property : value}
2. Local Class selector : tagName.className {property:value}
3. Global class selector : .className {property:value}
4. Id selector : #idName{property:value}
5. Child selector : outerTag innerTag {property : value}

**Class selector Vs Id Selector**

Class : it is to refer group of tag of same type or different types.

Id: id is use to name tag unique. May be tag have same name or different name.

<p class=”abc” id=”p1”>First Para</p>

<p class=”xyz” id=”p2”>Second Para</p>

<p class=”abc” id=”p3”>Third Para</p>

<p class=”xyz” id=”p4”>Fourth Para</p>

External CSS

CSS Box Model

Border property

Every tag in html internally follow box model



**Day 3**

**15-02-2022**

**JavaScript : JavaScript** was object based interpreter scripting language.

Object based or prototype base Vs Object oriented

OOPs

Object, class, Inheritance, Polymorphism, Encapsulation and Abstraction.

Old Version of JavaScript provide lot of pre-defined or user-defined object.

Interpreter Vs Compiler : both translator which convert from one format to another format.

Interpreter it will convert the code line by line. Compiler convert whole code at time.

Programming Vs scripting : running on browser and we are not converting in any other format.

JavaScript using ES5 feature.

ECMA Script : EMCA is a concept. (European Computer Manufacture Association).

JavaScript is a one of the implementation of ECMA.

Using JavaScript we can do programming on web page.

Syntax for script tag

<script type=”text/JavaScript”> opening tag

document.write(“Welcome to JS);

</script> closing tag

We can write more than one script tag between head tag as well as body tag.

JavaScript is case sensitive. In JavaScript not mandatory every statement end with semicolon.

Variable : variable is a name which can hold value and value can change during the execution of a program. We can declare the variable using var keyword.

var a;

var name;

Data types : Data type is a type of data which tells what type of data it can hold.

var a; undefined

var b=10; number type consider

var c=20.20; number type consider

var name = “Ravi” string type consider

var res = true; boolean type consider

var obj = new Date(); object type consider

Operator :

Arithmetic operator : +, -, \*, /, %(remainder)

Conditional operator : >, >=, <, <=, ==, != ===

Logical operator : &&, ||, !

Assignment operator =

Increment and decrement operator ++, --

++ it increment the value by one – it decrement the value by 1

typeof operator

ternary operator : condition ? true:false;

ternary operator is shortcut of if statement.

Conditional statements

If statement

If(condition) {

Set of logic execute

}

If else

If(condition) {

True block

}else {

False block

}

If else if or if ladder

If(condition) {

}else if(condition) {

}else if(condition) {

}else {

}

Switch statement : it is use to execute the set of statement depends upon the user choice.

Looping : looping is use to execute the set of statement again and again till the condition become false.

While loop

Do while loop

For loop (classical for loop)

1st 2nd 4th

for(initialization; condition ; increment / decrement ) {

body of the loop; 3rd

}

function : function is use to write the set of instruction to perform a specific task.

Divided into two types

1. Pre-defined function
2. alert(“Msg”) : it is use to display the pop up message.
3. prompt() : it is use to take the value through keyboards.
4. parseInt(): it is use to convert string to integer
5. parseFloat(): it is use to convert string to float
6. eval() it is use to convert string to number.
7. Confirm() : it contains two button ok and cancel. If user click on ok button it return true else it return false.

Do {

Alert 1: add, 2:sub etc

Prompt() take the value

Conversation convert the value

switch(){

case 1 Addition

case 2 Subtraction

case 3: Multiplication

case 4: Division

}

Confirm do you want to continue

}while()

1. User-defined function

Normal function syntax

function functionName(parameterList) {

body of the function;

}

1. function no pass parameter as well as no return type.
2. Function passing parameter but no return type.
3. Function passing parameter and return value
4. No passing parameter but return value.

**Day 4**

**16-02-2022**

Event : Event is a interaction between user and component (html tag or dom).

Or Event is a delegation model or event provide bridge between html and JavaScript.

JavaScript provided lot of pre-defined event and all events start with pre-fix on followed by event name.

Types of events

onClick

onDblclick

onMouseOver

onMouseOut

onKeyUp

onKeyDown

onChange

onFocus

onBlur

onChange

onLoad

onUnload

etc

DOM : Document Object Model :

DOM Hierarchy

Index.html

**<html>**

**<head>**

**<title>Title Message</title>**

**</head>**

**<body>**

**<p>Welcome to My Web page</p>**

**</body>**

**</html>**

**Html -🡪 root tag**

**head body**

**title p**

**TextNode: Title Message**

**NextNode : Welcome to My Web Page**

**DOM API (Document Object Model Application Programming interface).**

**Lot of programming provided DOM API like Java, Python, C# and JavaScript which help read, write and update DOM dynamically.**

**document.getElementsByName(“name”)**

**Form Validation**

**JavaScript is use to do validation ie all fields required, min length, max length, valid email id, valid phone number or credit-card.**

**Using JavaScript we can do validation on client side.**

1. **We can do validation using JavaScript**
2. **We can do validation using HTML5 features**
3. **We can do validation using Angular Framework**
4. **We can do validation using React JS**

**object :**

**object : any real world entity is known as a object.**

**two types of object.**

1. **Pre-defined object.**
2. **User-defined object.**

**Every object hold two things.**

1. **Properties or state or fields or variables.**

**Have --🡪**

1. **Behaviour or function or methods.**

**Do/does-🡪**

**Pre-defined objects.**

**Basic pre-defined objects.**

**Syntax to create the reference of pre-defined or user-defined.**

**var refereceName = new ObjectName();**

**refereneName.fieldName;**

**referenceName.functionName();**

**String object.**

**Literal style**

**Creating memory using new keyword.**

**Date**

**array : ES5 Methods**

**literal style**

**object style using new keyword.**

**splice(index,numberOfElementDelete,n1,n2,n2)**

**1st parameter : index position**

**2nd parameter number of elements to delete**

**3rd parameter number, 4th parameter and 5th parameter till infinite parameter to add.**

**Day 5**

**17-02-2022**

**JavaScript using ES6 features**

**Till ES5 to declare the variable we were using var keyword.**

**From ES6 onwards we can use let and const keywords to declare the variable.**

**var : var is a keyword to declare the variable. Using var we can re-declare same variable once again with same value or different value.**

**using var keyword we can do global declaration.**

**using let keyword we can’t re-declaration.**

**Using let keyword we can do local variable declaration or block scope declaration.**

**Types of function :**

1. **Normal function declaration**
2. **Expression style function**

**let/var functionName = function() {**

**}**

1. **Arrow style function**

**let/var functionName = ()=> {**

**}**

**Using normal style or expression style we have to return the value using return keyword.**

**Arrow function return value without return keyword.**

**Callback : passing the function or function body or function itself to another function as a parameter is known as callback function.**

**Types of loop**

**While loop**

**Do while loop**

**For loop**

**forEach() function**

**for in loop**

**for of loop**

**for in loop it give index position**

**syntax**

**for(let/var variableName in arrayName) {**

**}**

**For of loop it give value directly**

**syntax**

**for(let/var variableName of arrayName) {**

**}**

**let/var obj = new ObjectName();**

**obj.propertyName; to get the value**

**obj.propetyName=value; set the value**

**obj.funtionName();**

**User-defined object**

**Object : any real word entity.**

**Example**

**Properties or state – have**

**Person**

**Behaviour -- do,does**

**Place**

**Animal**

**Bank**

**Customer**

**Order**

**In JavaScript we can create user defined object using**

1. **Literal style**
2. **Function style : using ES5**
3. **Class style : using ES6**

**let objetReferenceName = {key:value,functionName:function() {**

**}}**

Day 6

18-02-2022

Creating user-defined object in function style

Create user-defined object

Write one or more property

And one or more behaviour

1st behaviour to change property

2nd behaviour to display property

Using function style

Label Text Field

Add Name onClick = addName()

Get the value of text field using getElementById() or any other ways.

Those values store in array variables.

Display those names one by one using loop

In same page.



21-02-2022

JSON : Java Script Object Notation :

Json is use to share the data between two technologies.

JSON created using object literal style.

JSON is a pre-defined object which provide set of methods which help to covert string to json or json to object to string.

JSON.parse() : this method is use to covert string to json.

JSON.stringify() : This method is use to convert object/json to string.

Promise : Promise is a pre-defined object provided by JavaScript which help to handle asynchronous event of data.

**Synchronous**

**Asynchronous**

**Synchronous Statement: Every statement depends one by one**

**document.write(“Hi”);**

**document.write(“Hello”);**

**document.write(“How r you”);**

**Asynchronous Statement : every statement execute independently**

**document.write(“Hi”);**

**document.write(“Hello”);**

**document.write(“How r you”);**

**Synchronous function call**

fun1();

fun2()

fun3();

**Asynchronous Statement call**

fun1();

fun2()

fun3();

Synchronous call to server

1st req

2nd req

3r req

Client Server

Asynchronous call to server

1st req

2nd req

3r req

Client Server

JavaScript provided set of pre-defined function to do asynchronous task.

setInterval

setTimeout

clearInterval

scope object :

JavaScript provide two pre-defined scope object ie

sessionStorage and localStorage.

If you want to share the between two pages we take the help of scope object.

sessionStorage hold the value till session open ie once application close the data store in session storage get lost.

If you want to store the data permanently. localStorage store the data permanently.

sessionStorage.setItem(“key”,value);

localStorage.setItem(“key”,value);

sessionStorage.getItem(“key”)

localStorage.getItem(“key”)

sessionStorage.removeItem(“key”);

localStorage.removeItem(“key”);

Promise : : Promise is a pre-defined object provided by JavaScript which help to handle asynchronous event of data.

Promise can be resolve and rejected.

We will create user-defined promise

If function return promise to handle this data we have to take the help then and catch().

then() will execute if promise resolve

catch() will execute if promise rejected.

JavaScript provide pre-defined function

Ie fetch(): This function help to call backend technologies REST API develop in any language.

fetch method return type is promise object.

22-02-2022

let emp = {id:1,name:”Ravi”,age:21};

let employees = [

{id:1,name:”Ravi”,age:21},

{id:2,name:”Ramesh”,age:24},

{id:3,name:”Raj”,age:25}

]

let empString = JSON.stringify(employees);

let empJson = JSON.parse(empString);

fetch vs axios

Node JS :Node JS is run time environment for JavaScript library, Framework or Programs.

Before Node JS JavaScript only known as client side scripting language But after Node JS JavaScript is known as Client side as well as server side scripting language.

­

Before Node JS if we want to run any JavaScript program we have to take the help of HTML page.

Inside Html page we are writing JavaScript code(JavaScript code can be internally script or external js).

After node js we can run the JavaScript program through commands.

In Node JS we can’t use DOM (Document Object Model) we can’t use.

document.write(“Welcome to JS”)

console.log(“Welcome to Node JS”);

Typescript : Typescript is known super set of JavaScript which support all features of ES6.

Still browser can’t understand typescript file we have to convert typescript to JavaScript with help of transpiler.

To install the transpiler we require node js.

npm (node package manager) which help download external dependencies or modules depends upon the application requirements.

npm install –g moduleName

To enable typescript transpiler we have to take the help of npm command

**npm --version**

**npm install –g typescript**

tsc --version

first create sample.ts

console.log("Welcome to Typescript program");

then convert ts to js using command as

tsc sample.ts

After converted we can see sample.js

node sample.js

Typescript provide or support data types

In ES5 or ES6 JavaScript

var a =10;

a=”Ravi”;

Typescript

let variableName:datatype;

let variableName:datatype=value;

**tsc datatype.ts : it will convert ts to js (ES5)**

**tsc dataType.ts --target es6 : it will convert ts to js (ES6)**

**array in typescript**

**23-02-2022**

**In Typescript we can create array to store same type value as well as different types of values.**

**tsc filename.ts –-target es6**

**user-defined object in JavaScript as well as typescript with data types using 3 ways**

1. **Literal style**
2. **Class style**
3. **Function style**

**Different types of function in Typescript**

**In JavaScript only function name must be match not mandatory number of parameter as well as type of parameter must be match**

**rest parameter : if we write array variable as parameter then we have to pass array variable and array variable can contains zero or 1 or many values. But we have to pass mandatory.**

**Rest parameter is a type of array only but it can receive zero or 1 or many parameter.**

**To declare the rest parameter we have to declare the variable using …variableName**

**In one function we can use only one rest parameter and it must be last parameter.**

**Spread parameter or operator : spread parameter is use to pass the value for rest parameter using array concept.**

**Optional : to declare optional parameter we have use variableName? optional parameter declaration must from right to left. We can’t leave mandatory parameter between two optional parameter.**

**Optional parameter default values are undefined.**

**If we want meaningful value then we can use default initialization.**

**Oops concept using typescript**

**Typescript support access specifiers concept.**

**Access specifiers help to provide the visibility of variable, function and class.**

**private**

**public**

**Create Product class in ES6 style**

**Which contains pid,pname,and price as private property**

**Using constructor set the value for pid,name,and price.**

**Then create more than product object and display product details.**

**Inside a constructor we have to write constructor price must be > 100 if else than 100 set value as 100.**

**24-02-2022**

**In Typescript we can use access specifiers in parameter variable insider constructor to make variable as instance variable.**

**Encapsulation : Binding or wrapping data and code in a single unit is known as Encapsulation.**

**class className {**

**property**

**behaviour**

**}**

**Typescript class with setter and getter methods.**

**Polymorphism : One name many forms.**

**Two types**

**Compile time**

**Function overloading : function have same name but different parameter list(type of parameter list or number of parameter list must be different).**

**Run time**

**Function overriding**

**Function have same name and same signature(number of parameter list and type of parameter list must be same).**

**To achieve function override we need inheritance.**

**Using abstract we can achieve partial abstraction.**

**Abstraction : hiding the internal implementation without knowing background details.**

**Interface : interface is use to achieve 100% abstract which contains only abstract functions.**

**In typescript we use interface to provide specification with all incomplete functions.**

**The class which implements that interface must be provide the implementation or body for that functions.**

**As well as with help to interface we can create literal style type of objects with variables.**

**String template with back ticks**

**Typescript Program Assignment you have to do**

****

**25-02-2022**

**Modules : module is like a package. Module is use to combine more than one variable, function, classes, interface which have same name but different purpose.**

**Using the module we can connect more than one file code with help of import and export.**

**Using the module we can divide the code base the functionality.**

**Create separate folder as typescript module example**

**Open this folder in vs code.**

**Typescript provide one file ie tsconfig.json file which help to typescript project configuration details.**

**To create the tsconfig.json file we have to run the command as**

**tsc –-init**

**abc.ts**

export function dis1(){

    console.log("This is dis1 function part of abc module")

}

**main.ts**

import {dis1} from './abc';

dis1();

**convert all ts to js you have to use the command as**

**tsc**

**to run the program you have to use the command as**

**node main.js**

**bootstrap : bootstrap is open source external css web framework which provide lot of pre-defined css classes which help to css responsive web application.**

**If we plan to apply css for web application we have to write all css rules from scratch.**

**git**

**28-02-2022**

**git :**

**Version control system : Version control system which help to records or keep track what are the changes we done on application or software or program.**

**Local Version Control system : RCS : Revision control system**

**Centralized version control system : SVN or CVS**

**Distributed Version control system : Git is type of Distributed version control system which provide local as well as remote repository.**

**Git is a open source distributed version control system or tool use to record the application changes.**

**Create the folder**

**git –version**

**git init : This command is use to make the folder as a local repository. This command only one time.**

**git status : This command give current status of repository**

**git add filename : This command is use to add the file from file system or os to staging area.**

**git commit –m “msg” : This command is use to move file or folder from staging area to local repository**

**git branch : git branch is like a pointer which keep more than one commit details.**

**By default git create default branch it may be master or main.**

**To check all branches we have to run the command as**

**git branch**

**to create the branch the we have to use command as**

**git branch branchname**

**to switch from one branch to another branch**

**git checkout branchname**

**git checkout –b branchName : this command it will create new branch as well as switch to that branch.**

**git branch –D branchName : This command is use to delete the branch**

**git merge branchname: This command is use to add branch task to current branch**

**Remote repository**

**GitHub**

**Git Lab**

**AWS**

**Azure**

**Google cloud**

**Etc**

git branch -M main : if local machine contains local branch name is master then using this command we can rename from master to main.

git remote add origin URL

git remote add origin <https://github.com/Kaleakash/test_app1.git>

Connecting local repository to remote repository

git push –u origin main : This command is use to push the code from local repository to remote repository

**whenever if you do any changes in local repository if you want to update these in remote repository.**

**git add .**

**git commit –m “message”**

**git push –u origin main**

**another way to create local repository**

**git clone URL : only one time ­­**

**git pull : it pull new changed done in remote repository to existing repository.**

**02-03-2022**

**Node JS : Node JS is a run time environment which help create server side programming.**

**Front end backend**

**Html/html5 Java**

**Asp.net**

**Css/css3 Php**

**Bootstrap CGI**

**Python**

**JavaScript**

**jQuery Node JS**

**Before Node JS JavaScript is known as Client side scripting language But after node js JavaScript is known as client side as well as server side scripting language.**

**Using JavaScript we can do file handling, creating server side technologies, creating rest full web service, connection databases, do security program.**

**With help of node js we can run JavaScript program through commands.**

**REPL Terminal : Read Eval Print loop.**

**Node JS Modules**

**Modules contains collection of function, variable and classes to do specific task.**

**Node JS using ES5 style**

**Node JS using ES6 style**

**Node JS with Typescript using ES6 style**

**Node JS using JavaScript**

**3 types**

1. **Core module : by default available with node js**
2. **User-defined modules**
3. **External module**

**Core module**

**Fs module**

**FS : File system : fs is a type of core module which provided lot pre-defined function which help to asynchronous as well as synchronous file handling program using JavaScript on server side.**

**In ES5 style to import the module we have use the function as require(“modulename”);**

**Syntax to load the module using ES5 styule**

**variableName refereceName = require(“moduleName”);**

**Taking the value through keyboard in node js**

**readline is core module which help to take the value through keyboards asynchronously.**

**readline-sync : This is external module which help to take the value synchronously.**

**External module installation**

**npm install readline-sync –g**

**or**

**npm install readline-sync**

1. Store employee record but id must unique

2. Delete the record using id if id present delete or else record not present.

3. update age using id if you pass wrong id display record not present.

4. using id display employee name and age if you pass wrong id display record not present.

5. display all employee details id, name, age

find()

findIndex

splice

03-03-2022

http module

node js provide pre-defined module ie http module.

Using http module we can create server side application using JavaScript as well as with the help of this module we can create the server also.

Servlet, jsp or Spring framework

Asp.net and C#

Php

Etc

We require server it may be web server or application server.

Tomcat server

Web logic server

Glashfish server

IIS Server

Apache server

Different between Non Node JS Server Vs Node JS.

Node JS server are event driven loop or architecture.

URL module : uniform resource locator : this module provide set of function which help to extract the data from url.

**04-02-2022**

Node JS Web Framework.

Express JS is one of the open source module base upon express module which help to develop the application very easily.

First create the express js application

Inside that folder create another sub folder

SimpleProject

In node js we have to create the package.json. This file contains all project configuration details.

Command to create the package.json file

npm init

it ask package name . give any name and then hit enter key one by one.

after created package.json file we have to install express js module

npm install express

node js provide pre-defined global property

\_\_dirname : This property provide current path of directory.

In Express JS if you want to enable post data we have to use middleware

app.use(middlewarename);

body-paser : it is external module which help to enable the post data receive from form.

Old version express js we were installing this module separately using command as npm install body-parser

In New version express js body-parser module automatically install with express module.

07-03-2022

Express JS introduce View Engine or dynamic HTML

Create folder as

Express js view engine

Open the command prompt and install express-generator module.

npm install express-generator –g

or

npm install express-generator

After install you can create project template using command as

express demo-app (demo-app) is project name.

now move inside a project folder using cd command

cd demo-app

npm install (to download the modules or dependencies.)

if we use express js view engine on view engine we can do dynamic task.

This view technologies tightly coupled with backend technologies ie express js.

Ex: JADE

Web Service :

Giving the service for web application when both application running using different technologies.

Browser Express JS

Client Amazon Payment gate way

Google pay

Paytm

Phone pay

Debit card

Credit card

1. SOAP Base web service
2. RESTFill Web Service

SOAP Base Web Service : Simple Object Access Protocol. SOAP base up SOA (Service Oriented Architecture).

Consumer Producer

Java (req)

HSBC XML/JSON HDFC

Java Express

Express (res)

Limitation of SOAP web service is we can consume and product data only in the form xml.

XML is heavy.

RESTFull : Representation State Transfer.

Using Restfull web service we can expose our resource (Employee, Customer, Order, Product) as a Web service. Using RestFull web service we can consume and product data in any format like XML or JSON or html or plain text etc.

If we make express js as a restfull web service any technologies can consume our data.

Like Java, Asp.net, php, Angular or React JS.

REST API (Application Programming interface).

Create REST API Folder

Then create sub folder

Simple REST API

Then VS Code in in Simple REST API folder using code .

Create the package.json file using npm init

npm install express

Get method we use to get the resources it can be product, employee, customer etc.

1. get method display welcome message.
2. Base upon the path retrieve string message.
3. Base upon the path retrieve josn message.
4. Retrieve product entity details in json format.
5. Retrieve more than one product details using json format.

Using get method REST Client can pass the value to Rest Consumer

1. Query param
2. Single value : URL?key=value
3. Multiple value : URL?key=value&key=value
4. Path param
   1. Singe value : URL/value
   2. Multiple value : URL/value1/value2

Get is equal to Select Query

post : Post method is use to store the resource or create the new resource

Post is equal to insert Query

Delete method : This method is use to delete the resource.

Express JS CRUD Operation using Typescript

Create the folder

Express JS with TypeScript

Install the typescript using command as

npm install typescript –g

we have to create the tsconfig.json

tsc –init

now we have create the package.json file

npm init

now create the folder as src

When we want to do Express JS with typescript we have to install two dependencies

npm install @types/node

npm install @types/express

npm install express

convert ts to js

tsc

now open package.json file

 "main": "dist/app.js",

  "scripts": {

    "start": "node .",

    "test": "echo \"Error: no test specified\" && exit 1"

  },

09-03-2022

File system

Data redundancy : duplicate records

Data consistency : format of the file or data format.

Security issue : we can give file in read mode or write mode.

Database

Data

Information

Database : storing the data in table format.

DBMS : Database management system it is a software which help to store the data in table format.

RDBMS : Relational Database Management System.

Dr. EF Codd’s rules 12 rules start from 0 to 11

MySQL

Oracle

DB2

SQL Server

All RDBMS database are schema base

Employee

Id Name Age City Phno

100 Raj 21 null null

101 Ramesh 23 null null

102 Raju 34 Bangalore null

103 Mahesh 32 null 9900

TrainerStudent

TId TName Tech Sid SName Age

1 Raj Python 100 Reeta 21

1 Raj Python 101 Meeta 22

1 Raj Python 102 Leeta 23

Primary key and Foreign Key

Trainer

PK

TId TName Tech

1 Raj python

2 Ravi Angular

Student

PK FK

SId SName Age TSID

100 Reeta 21 1

101 Meeta 22 1

102 Leeta 23 1

103 Keeta 24 2

No SQL :

Mongo DB : Document base Database

HBase

Cassandra

Neo4j

Mongo DB

Mongo DB is a open source document base database which help to store the data in json format.

Open the command prompt in bin folder location

C:\Program Files\MongoDB\Server\5.0\bin

Then run mongod To start the server

It need some directory to create few files which to start the server

Inside a C Drive create data folder and inside data folder create db folder.

Another command open inside a bin folder

Then run mongo : It open mongo terminal to do mongo commands. This command is known as mongo client.

show databases

or

show dbs

use databaseName : this command if database not present it will create and switch to new database else it switch to existing database.

Database contains more than one collection(tables).

In mongo DB table is known as collection.

Syntax to create the collection.

db.createCollection(“Sample”);

in mongo db record is known as document.

db.collectionName.insert({jsonData});

to view the document from a collection

db.collectionName.find();

if we insert any document in collection every document by default contains \_id as a pre-defined fields with unique value.

\_id is consider a primary key in mongo db. If you want to change the value you can do it but you can’t change the field name.

Insert many record at a time

db.Emp.insertMany([{\_id:8,name:"Vikash",age:31,salary:45000,city:"Bangalore"},{\_id:9,name:"Vishal",age:38,salary:5000,city:"Pune"}]);

Emp

\_id,name,age,salary,city

5 to 8

Retrieve the document base upon the index value.

db.Emp.find()[0];

retrieve the specific value from a document using index position.

db.Emp.find()[6].name;

syntax

db.Emp.find({condition},{fieldNames});

filter fields form a document

db.Emp.find({},{name:1}) : it display name and \_id

db.Emp.find({},{name:1,\_id:0}) : it display one name

db.Emp.find({},{name:1,\_id:0,salary:1}) : it display name and salary

+ve or –ve : true

O is false.

db.Emp.find({},{name:1,\_id:0})[5].name;

db.Emp.find({},{name:1,\_id:0})[5]

filter the document using conditions.

db.Emp.find({\_id:1})

db.Emp.find({name:"Ajay"})

db.Emp.find({city:"Bangalore"})

db.Emp.find({city:"Bangalore"},{name:1})

db.Emp.find({city:"Bangalore"},{name:1,\_id:0})

db.Emp.find({age:{$gt:25}});

db.Emp.find({age:{$lt:25}});

db.Emp.find({age:{$gte:25}});

db.Emp.find({age:{$lte:25}});

db.Emp.find({age:{$eq:25}});

db.Emp.find({age:{$ne:25}});

> db.Emp.find({age:{$ne:25}},{name:1,city:1,\_id:0});

Two condition must be satisfies or any one condition.

$and $or

$and both the condition satisfies then we get the documents.

db.Emp.find({$and:[{name:"Ravi"},{age:21}]});

$or any one of the condition must be satisfies then we will get the document.

db.Emp.find({$or:[{name:"Ravi"},{age:25}]});

update the Document

This query update only one document

db.Emp.update({\_id:1},{$set:{age:23}});

This query can update more than one document.

db.Emp.updateMany({city:"Bangalore"},{$set:{salary:35000}});

both condition must be satisfies then only update

db.Emp.update({$and:[{\_id:1},{city:"Bangalore"}]},{$set:{salary:45000}});

db.Emp.update({\_id:1},{$set:{state:"Kar"}});

first check \_id document then if that document state field present then change the existing field value as Kar else it all the state field to that document.

All the fields for all documents with default value.

db.Emp.updateMany({},{$set:{state:""}});

remove the fields from specific document.

db.Emp.update({\_id:1},{$unset:{state:1}});

remove the document from a collection

db.Emp.remove({\_id:1})

db.Emp.remove({city:"Bangalore"});

storing array value in a fields.

db.Emp.insert({\_id:2,name:"Raju",age:23,marks:[72,80,77]});

db.Student.find({marks:80})

db.Student.find({marks:76})

db.Student.find({"marks.0":80});

db.Student.find({"marks.1":80});

Add the elements to array

db.Student.update({\_id:3},{$push:{marks:90}});

remove the elements from a array

db.Student.update({\_id:4},{$pop:{marks:1}});

Mongo DB relationship

We can achieve the relationship between two collection using

1. Embedded style
2. Linking style

Embedded style : we are storing all documents only one collection.

One to one

Employee document

{id:100,name:”Raj”,salary:12000}

Address document

{city:”Bangalore”,state:”Kar”}

[{pid:111,tech:”Java”},{pid:222,tech:”Python”}]

{\_id:100,name:”Raj”,salary:12000,”add”: {city:”Bangalore”,state:”Kar”},projects: [{pid:111,tech:”Java”},{pid:222,tech:”Python”}]

}

Linking style : we are storing all document in more than one collection.

TrainerInfo

PK

\_id TName Tech

100 Raj Java

101 Ravi Python

db.TrainerInfo.insert({\_id:101,tname:"Ravi",tech:"Python"});

db.TrainerInfo.insert({\_id:100,tname:"Raj",tech:"Java"});

StudentInfo

PK

\_id SName Age TSId

1 Seeta 21 100

2 Reeta 22 100

3 Meeta 23 101

4 Leeta 24 101

db.StudentInfo.insert({\_id:1,sname:"Seeta",age:21,tsid:TrainerInfo.find()[0].\_id});

db.StudentInfo.insert({\_id:2,sname:"Veeta",age:22,tsid:db.TrainerInfo.find()[0].\_id});

StudentInfo1

\_id SName Age TSId

1 Seeta 23 tid:{\_id:100,tname:”Raj”,tech:”Java”}

2 Reeta 22 tid:{\_id:100,tname:”Raj”,tech:”Java”}

3 Meeta 23 tid:{\_id:101,tname:”Ravi”,tech:”Python”}

4 Leeta 24 tid:[

{\_id:100,tname:”Ravi”,tech:”Java”},

{\_id:101,tname:”Raj”,tech:”Python”}

]

db.StudentInfo1.insert({\_id:3,sname:"Veeta",age:24,tsid:db.TrainerInfo.find()[1]});

db.StudentInfo1.insert({\_id:4,sname:"Meeta",age:25,tsid:[db.TrainerInfo.find()[1],db.TrainerInfo.find()[0]]});

**Aggregate function**

Aggregate function is use to group the multiple documents and then perform aggregate or mathematical operation and it return result.

$lookup

This operation is use to retrieve the document from more than one collection base up condition apply on field part two collections.

db.StudentInfo.aggregate([{$lookup:{from:"TrainerInfo",localField:"tsid",foreignField:"\_id",as:"TrainerDetails"}}])

db.TrainerInfo.aggregate([{$lookup:{from:"StudentInfo",localField:"\_id",foreignField:"tsid",as:"StudentDetails"}}])

using $look with $project to filter the fields.

b.StudentInfo.aggregate([{$lookup:{from:"TrainerInfo",localField:"tsid",foreignField:"\_id",as:"TrainerDetails"}},{$project:{sname:1,"TrainerDetails.tname":1}}])

using $look, $match, $project to filter the fields.

db.StudentInfo.aggregate([{$lookup:{from:"TrainerInfo",localField:"tsid",foreignField:"\_id",as:"TrainerDetails"}},{$match:{"TrainerDetails.tech":"Java"}},{$project:{sname:1,"TrainerDetails.tname":1}}])

$group

db.EmployeeDetails.aggregate([{$group:{\_id:"$city"}}]);

db.EmployeeDetails.aggregate([{$group:{\_id:"$deptId"}}]);

sum of salary group by city

db.EmployeeDetails.aggregate([{$group:{\_id:"$city",totalSalary:{$sum:"$salary"}}}]);

db.EmployeeDetails.aggregate([{$match:{city:"Bangalore"}},{$group:{\_id:"$city",minsalary:{$min:"$salary"}}}]);

Connection Mongo DB database through node js

1. mongodb module
2. mongoose module

using mongodb module

create mongodb module

then create the package.json file using **npm init** command

mongodb module we use for small application. It is known as native module.

Mongoose : mongoose module is base up on mongodb module which provide ODM (Object Data modelling ).

ORM : object relation mapping :

Mongoose module provide schema concept which help provide structure for collection.

Create folder

mongoose db module

**create package.json file using npm init**

**npm init mongoose**

Express JS with Mongoose Module using MVC style

MVC : Model View Controller



Express MVC :

Create MVC folder

Inside mvc folder

Folder : Backend folder

Inside backed two folder

Folder : Using JS

create package.json file using npm init

npm install express

npm install mongoose

Folder : Using TS

Folder : Frontend folder

Angular

Model : load the mongoose module. Then create the schema and model and export that model to another layer.

**Repository** : This file import or require model file and do the operation collection base upon the requirements. This file contains pure database logic using mongoose module or mongodb module.

Controller : This file import repository file and it take request and response object and do operation on collection. Base upon the request it will call repository method and base upon the repository result it will give the response to another layer.

router :This file map the sub path of request and http method (get, post, put and delete). Then base upon the method and sub path it will call specific controller functions.

**Main file**

App.js

Load all the module

Express, mongoose and more

Enable json body part

Connect the database

Then add middleware as a router file

Then run the application on specific port number.

14-03-2022

npm install nodemon –g

then run node js application using

nodemon app.js

if we do any change in application automatically refresh that application.

MVC project using TS

tsc –-init help to create the tsconfig.json

create the package.json file using command as

npm init

typescript dependencies

npm install @types/node

npm install @types/express

npm install @types/mongoose

js dependencies

npm install express

npm install mongoose

create the folder as

src

dist

inside src folder

router

controller

repository

model

config : this file contains database connection

app.js

15-03-2022

Front end technologies

Html, css, JS, Typescript

Html5, css3, ES6 and Typescript

Front end library and framework.

jQuery

Coffee JS

Backbone js

Angular JS

Angular Framework

React JS

Vue JS

D3 JS

Ext JS

etc

to read, write and update DOM properly.

jQuery is external JS library which provide lot of pre-defined function which help read, write and update DOM properly.

Library provide lot of pre-defined function which internally connected to each other to perform specific task.

Library doesn’t follow any standard.

Angular JS : Angular JS is framework which provide lot API to do specific task. All framework internally follow standard. Angular internally follow MVC on front end side.

Angular JS 1.0 1.x etc

HTML,CSS,JavaScript using ES5 etc.

AJAX : Asynchronous JavaScript and XML.

Angular Framework 2.x to 13.x

HTML,CSS,JavaScript and Typescript.

If we develop any application using any framework 60 to 70% task taken care by framework. But framework is not a final product it a protocol or template.

Angular framework is web base MVC framework provided by google company which help to develop SPA (Single Page application).

Single page application Vs multi page application.

All framework provide the implementation of design pattern.

Angular cli (command line interface). Using angular cli we can create the angular project

Ng (next generation for DOM or element).

npm install –g @angular/cli

to create new angular project we have to run the command as

ng new project-name

ng new welcome-app

routing 🡪 No

styling 🡪CSS

cd welcome-app

Then open project in vs code using code .

To run the project you have run the command as

ng serve

if it ask any policies option you can give yes or no.

After 100% compiled all files

Then open the browser write the url as <http://localhost:4200>

Default port number for angular in development mode is 4200

src

app

app.component.html (template)

app.component.ts

app.component.css

Angular is component base architecture.

Component is use to control the view or part of view.

Angular provide lot of pre-defined decorator.

All decorator start with prefix @ followed by decorator name.

Decorator also known as annotation. It is type of specific function provide meta data information.

@component

@NgModule

@Injectable

@Input

@Output

@Pipe

Etc

Using angular we are creating user defined tags.

With help of component we are creating user-defined tag with help selector attribute.

@component

selector : “app-root” : This attribute is use to create the user-defined tags.

templateUrl:”./app.component.html”,

styleUrls:[“./app.component.css”]

<app-root></app-root>

app.module.ts

@NgModule

Module is a combination of more than one component.

declaration : [all component declaration]

imports : [we can import pre-defined or user-defined modules]

providers : [we have to provide angular service details.]

bootstrap : [to display the parent component as first component]

main.ts

This file provide the main module details.

managermodule loginmodule customermodule

more than one compo more than one com more than component

creating the component using command prompt

ng generate component header

or

ng g c header

ng new data-binding

routing -🡪 no

styling 🡪 css

to run the project automatically in default browser we have to run the command as

ng serve –o

Data binding

Using data binding we can share the data between component to view (template) or template to component.

One way data binding : uni-directional

1. string interpolation :

Component ---🡪 View

syntax

{{}}

{{variableName}}

{{exression}} {{5+10}} {{10/5}}

{{functionCall}} {{display()}}

1. property binding :

component --🡪View

syntax

[]

<input type=”text” [value]=”lastname”/>

<p [innerText]=”lastname”></p>

1. event binding

View----🡪Component

()

DOM event Angular Event

onClick (click)

onDblClick (dbclick)

onMouseOver (mouseover)

onSubmit (ngSubmit)

two way data binding

using

event binding and string interpolation or property binding.

Passing the value from view to component

Template reference

<input type=”text” #nameRef/>

Two way data binding : bi-directional

[(ngModel)]

We can do two way data binding using ngModel attribute.

If we do any changes in component side automatically update on view and vice-versa.

To achieve two data binding angular use pre-defined attribute ngModel . ngModel pre-defined attribute is part of FormsModule so we have to import this module in app.module.ts file in import section.

Types of directives

**ng new types-of-directives**

using angular directive we are adding extra behaviour or functionality to DOM.

3 types of directives

1. component directive : using the component directive we are creating user-defined tags with help to selector.

@Component({

Selector:”my-tag”,

templateUrl:”my-tag.html”

})

class MyComponent {

}

Component is a type of directive which help to create the user-defined or custom tag.

1. structure directive : using this type of directive we can add or remove dom elements.

\*ngIf

\*ngFor

Using structure directive we can use if and for loop in html page.

Creating model class or interface

ng g class employee

Or

ng g interface employee

Attribute directive attribute directive mainly use to do css.

ngStyle

ngClass

**Angular Forms**

ng new angular-forms

Template reference

In HTML and JS

<input type=”text” id=”n1” name=”user”/>

JS

var name = document.getElementById(“n1”).value;

In Angular

Template

<input type=”text” #nameRef>

Ts

Inside a function

let name = nameRef.value

Using angular forms we can pass the value from template to component in json format.

1. Template driven form

Flow of the application

View ----------🡪Component

More code on template side

Easy to develop

1. Reactive form or model driven form

Flow of the application

Component 🡨-------------view

More code on component side

It is good for complex form.

ng g c tdf-login-page

ng g c mdf-login-page

In Template driven form we have crate the reference of form using ngForm and ngModel attribute.

<form #loginRef =”ngForm” (ngSubmit)=”checkUser(loginRef)”>

<label>UserName</label>

<input type=”text” name=”user” ngModel />

</form>

ngForm is pre-defined attribute

#loginRef user-defined name start with pre-fix #name

ngForm is a pre-defined attribute part of FormsModule

so you have to import FormsModule in app.module.ts file in import section.

Model Driven Form

Angular provide pre-defined API

Ie FormGroup and FormControl

FormControl bind to textfield, passwordfield, radiobutton, checkbox, dropdowntox.

According to Reactive form we can’t create FormControl without FormGroup.

FromGroup is a collection of more than one FormControl.

Form formgroup

UserName TextField formcontrol

Password passwordfield formControl

Submit Reset

In model driven or reactive form Angular provided pre-defined attribute ie formGroup and formControlName which help to bind FormGroup and FormControl reference part of component.

formGroup and formControlName are pre-defined attribute part of ReactiveFormsModule. So we have to import ReactiveFormsModule in app.module.ts file in import section.

Angular Form Validation

Angular framework provided six classes to do validation

ng-valid

ng-invalid

ng-dirty

ng-pristine

ng-touched

ng-untouched

**state class true class false**

control ng-touched ng-untouched

has visited

control has ng-dirty ng-pristine

change the

value

control ng-valid ng-invalid

value is value

loginRef.controls['user'].invalid && (loginRef.controls['user'].dirty || loginRef.controls['user'].touched)

contr

21-03-2022

ng new angular-service

no routing

styling css

Angular Service

If we do any business logic inside a component that code become local to that component so same code we can’t access in another component.

Angular service mainly use to do business logic which can be access in more than one component.

Template component service

First first-component

Second second-component service

Third third-component

Angular service mainly divided into two types

1. User-defined service
   1. Creating the object explicitly
   2. Creating the object using DI (Dependency Injection)
2. Pre-defined service : Angular provided pre-defined API it HttpClient which help to call REST full web service develop in any language.

IOC : Inversion of control

: it is a concept rather than creating any resource or object explicitly allow to create by container. If container create it will create property as well as maintain the resource properly.

DI : Dependency Injection

DI is a implementation of IOC

Constructor base DI

Setter base DI

Angular support only constructor base DI.

Html -🡪Component ---🡪 UserDefined service (using new keyword or using DI) --🡪

Angular provided pre-defined API ie HttpClient which help to call backend technologies REST API. Using HttpClient we can call get, post, put and delete methods.

We have to do DI for HttpClient in user-defined service class

FakeService

Fetch(): fetch is a pre-defined api part of JavaScript which help to call rest api.

fetch function return type if promise object. if it return promise we have to then and catch. Then is use to load the data and catch is use to handle it error.

In Angular HttpClient api’s get, post, put and delete method return is Observable.

Observable is part of RxJS (Reactive JS).

It is use to handle asynchronous event of data.

Promise can’t cancel but Observable can be cancel using unsubscribe.

Promise is use to load the data all at time. But Observable load the data one by one.

If any api return type is Observable to load the data we have to subscribe(). Which will take 3 parameter as a callback. 1st parameter is like a next which load the record one by one. 2nd parameter it exception generate it will call. 3rd parameter call after loaded all data successfully.

HttpClient is a pre-defined API part of HttpClientModule. So we have to import HttpClientModule in app.module.ts file in import section.

22-03-2022

"start": "tsc-watch --onSuccess \"node ./dist/app.js\"",

Please install external module in using tsc folder

npm install tsc-watch -g

or

npm install tsc-watch

in angular project

ng g c product

ng g interface product

ng g s product

two domain or server going communicate each others.

Angular application 🡪 4200

Express JS 🡪 9090

CORS Policy

Cross Origin Resource Sharing

In node JS we have to install cors module and add the middleware to enable to access resource for front end technologies.

npm install cors

npm install @types/cors

Angular pipes

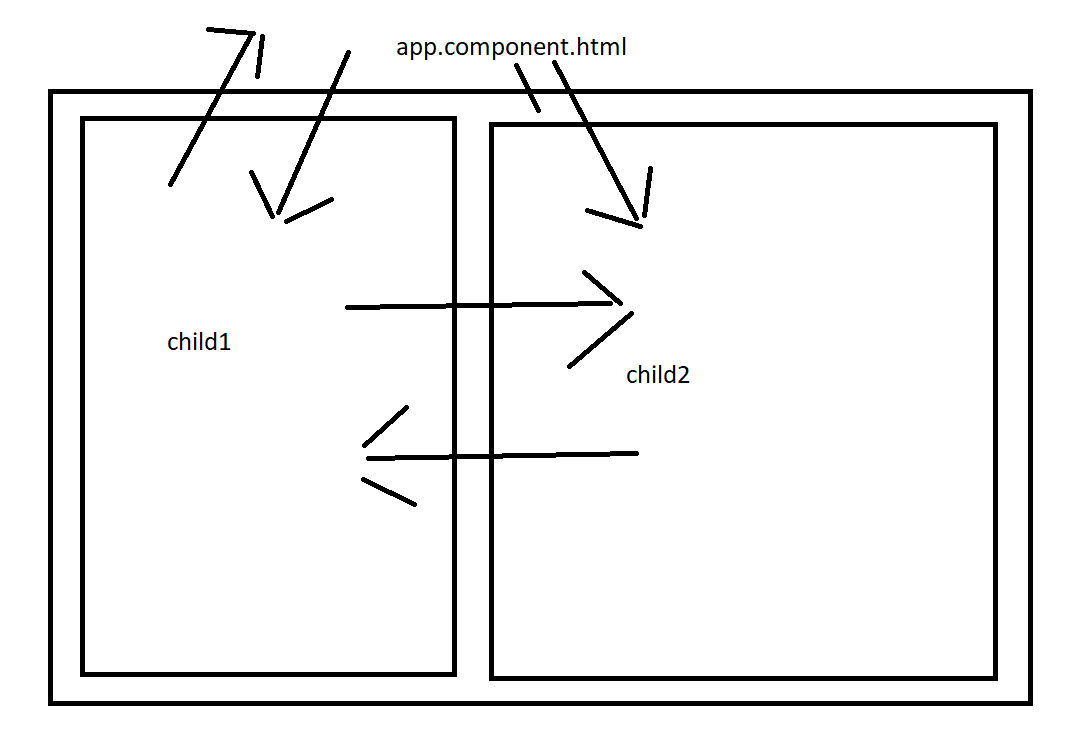
Angular pipes are use to transform the data on view side (template).

1. Uppercase
2. Lowercase
3. Date
4. Currency
5. Json

etc

23-03-2022

Component communication



Sharing the data

1. Parent to child : @Input
2. Child to parent : @Output with EventEmit api

@ViewChild

Sibling

1. localStorage or sessionStorage
2. shared common service
3. RxJS Observable concept

ng new component-communication

Angular routing : Angular routing is use to navigate from one page to another page base upon path provided in routing file.

ng g c aboutus

ng g c contactus

ng g c login

ng g c home

Angular provided pre-defined tag ie

<router-outlet></router-outlet>

This tag is like a placeholder which help to load the content of page base upon path provided in routing file.

Auth guard

Angular provided pre-defined auth guard ie interface which is like a middleware between login authentication and home page.

CanActive

CanDeactive

CanLoad

Testing is use to find the defect or error or bugs in application

Layer architecture

Unit testing : unit : it is use to test function or method or class or module. Using unit testing we are testing function functionality working or not.

Testing

1. Black box testing

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

1. White box testing

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

UI

Front end

Jasmine : Jasmine is open source testing framework. Which provided lot of pre-defined function which help to do the testing for client side as well as server side JavaScript.

Test suite : test suite is like a container which contains more than on test spec as well as another test suite.

To make test suite testing framework provide pre-defined function ie

describe(“”,callback)

Test spec : Test spec is use to do the test the function functionality

it(“”,callback)

Expect functions

describe(“Sample testing ”,()=> {

it(“first test”,()=> {

// logic call the business logic.

expect(expectOut).toXXX(actual)

})

it(“second test”,()=> {

})

})

Hook

Backend end

expect().toBeDefined()

expect().toBeFalse()

expect().toBeFalsy()

expect().toBeGreaterThan(expected)

expect().toBeGreaterThanOrEqual(expected)

expect().toBeInstanceOf(expected)

expect().toBeLessThan(expected)

expect().toBeLessThanOrEqual(expected)

expect().toBeNaN()

expect().toBeNegativeInfinity()

expect().toBeNull()

expect().toBePositiveInfinity()

expect().toBeTrue()

expect().toBeTruthy()

expect().toBeUndefined()

expect().toContain(expected)

expect().toEqual(expected)

expect().toHaveBeenCalled()

expect().toHaveBeenCalledBefore(expected)

expect().toHaveBeenCalledOnceWith()

expect().toHaveBeenCalledTimes(expected)

expect().toHaveBeenCalledWith()

expect().toHaveClass(expected)

expect().toHaveSize(expected)

expect().toMatch(expected)

expect().toThrow(expectedopt)

expect().toThrowError(expectedopt, messageopt)

expect().toThrowMatching(predicate)

expect().withContext(message)

Angular Testing

Testing Framework

Jasmine

Mocha

JEST

Plain JS : Jasmine, Mocha and JEST

Angular Framework : Jasmine and Karma

React Js : JEST

Node JS Testing

: Jasmine

: Mocha and Chai

Jasmine is a open source framework provided lot of API which help to do the testing.

Karma : Karma is test runner which help to display the output on browser.

Angular internally configure Jasmine and Karma.

TestBed pre-defined class provide by angular which help to do the testing angular special classes.

Component, Service, pipe, routing etc

Node JS Testing (Server Side script testing )

Node JS Testing we can do using jasmine and mocha.

Jasmine provide all function ie spec, suite and assert function.

Mocha provide only spec and suite and allowing to use other assert library

Should js

Chai

Create the folder using mocha and chai

npm init

npm install mocha -D

npm install chai –D

Node JS testing using Jasmine

npm init

npm install jasmine

npm install jasmine-node

Docker :Docker is a advanced OS Virtualization software platform that makes it easier to create, deploy and run the application in Docker container.

Base Machine : 16 RAM

Virtual Image 1 : 6 RAM

10 Virtual images OS

Virtualization : it is use to employing software (such as hypervisor) to create a virtual version of resources such as server, application, database or software.

Virtualization let you divide a system into a series of separate section, each one acting as a distinct individual system.

Docker container is a very light weighted container that allow developer to package up an application and deploy it with the help of in built libraries.

Virtualization is an abstract version of physical machine. While Docker (Containerization) is the abstract version of an application develop in any technologies.

Container This is a runtime process / instance of an image.

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

Docker file : A Docker file a blue print / set of instruction that defined how to create the image.

Docker commands

docker --version

docker images : This command is use to check running images in local machine

docker pull hello-world : This command is use to pull the pre-defined image available in Docker hub

docker run imageName/imageId

Docker hub account :

It is like a git hub which help to pull and push/publish images.

Creating user-defined images

**Dockerfile**

FROM busybox:latest

CMD ["date"]

docker build –t imageName:version . –f Dockerfile

Docker image to run the node js program

App.js

function sayHello(name){

    return "Welcome to Simple Node JS running through docker "+name;

}

console.log(sayHello("Ravi"));

Dockerfile

FROM node:latest

COPY app.js .

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

docker build -t my-node:1.0 . -f Dockerfile

now you can run the docker image

docker run my-node:1.0

28-03-2022

Creating the image for Express JS application

Create the package.json file using npm init command

npm install express

app.js

let express = require("express");

let app = express();

let employee=[

    {id:100,name:"Ravi",age:21},

    {id:101,name:"Ramesh",age:22},

    {id:1002,name:"Raju",age:23},

]

app.get("/",(req,res)=> {

    res.send("Welcome to Express with Docker")

})

app.get("/user/:name",(req,res)=> {

    let name = req.params.name;

    res.send("Welcome user "+name+" with docker concept");

})

app.get("/employee",(req,res)=> {

    res.json(employee);

})

app.listen(3000,()=>console.log(`Server running on port number 3000`))

**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:1.0 . f Dockerfile

if image contains to run the server we have to run this command as

docker run –d –p 3000:3000 imageName/imageId

creating the image to run the static html, css and js file

create the html, css or js file

**Dockerfile**

FROM nginx:latest

COPY index.html /usr/share/nginx/html

docker build -t my-web-page:1.0 . -f Dockerfile

nginx default port number is 80

Creating the image Angular project

First create the project

You have to build the project

ng build

once image created you have to push this image in docker hub

before push create the tag for the image

docker tag imageName:version dockerhubaccont/imageName:version

docker push dockerhubaccount/imageName:version

docker pull akashkale/my-angular