

Design web application using expressjs

1. Using npm init create paclkage.json file
2. Add following depencies in it

"dependencies": {

    "ejs": "^3.1.5",

    "nodemon": "^1.2.1",

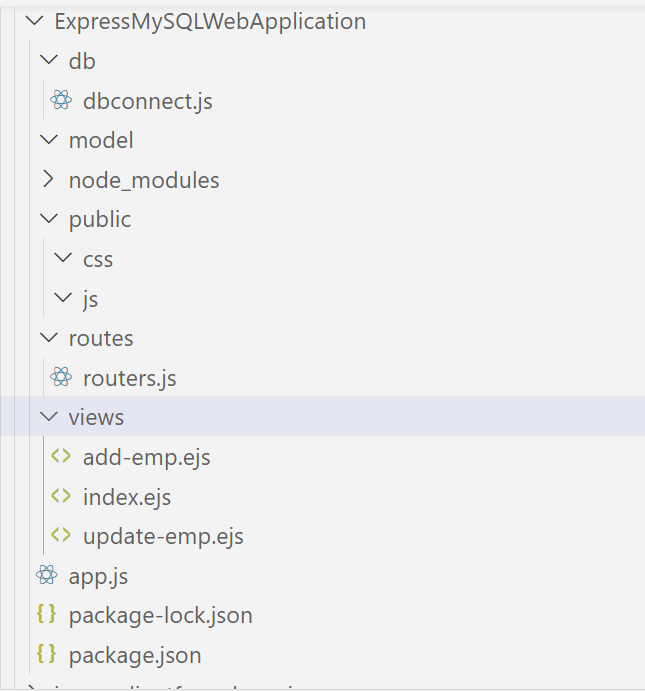
    "body-parser": "^1.19.0",

    "express": "^4.16.4",

    "mysql": "^2.17.1"

}

1. And then use npm install to install all depenedencies
2. Create folder structure as shown below



1. Add followigncode in app.js

//add all libraries

const express=require("express");

//add all libraries

const express=require("express");

const app=express();

const path=require("path")

const bodyparser=require('body-parser');

//all url handling(routing) details will be written in this file

const routes=require("./routes/router.js")

//configure the application

app.set("views",path.join(\_\_dirname,"views"))

app.set("view engine","ejs")

//add static setting

//app.use("/css",express.static(path.resolve(\_\_dirname,"public/css")))

//app.use("/js",exprss.static(path.resolve(\_\_dirname,"public/js")))

//app.use("/images",exprss.static(path.resolve(\_\_dirname,"public/images")))

//add middlewares

app.use(bodyparser.urlencoded({extended:false}));

//pass the control to router for handling url

app.use("/",routes);

//start the server

app.listen(9000,function(){

console.log("server started at prot 9000")

});

//it will make all the setting and configuration accessible in

//all the files

module.exports=app;

1. Add following code in dbconnect.js , for creating connection with mysql server

//the file for database connection

const mysql=require("mysql");

var mysqlconnection=mysql.createConnection({

host: '127.0.0.1',

user: 'root',

password: 'root123',

database: 'test',

port:3306

});

mysqlconnection.connect((err)=>{

if(!err){

console.log("connected successfully")

}else{

console.log("error in connection ",err);

}

});

module.exports=mysqlconnection;

1. Add following code in router.js

const express=require("express");

var myrouter=express.Router();

//connection is same as mysqlconnection in dbconnet.js

var connection=require("../db/dbconnect")

//to display all products in tabular form

myrouter.get("/products",function(req,resp){

connection.query("select \* from product",(err,data,fileds)=>{

if(err){

resp.status(500).send("no data found");

}

else{

//to create table view using data, write file

//index.ejs in view folder, to access the file write as follows

resp.render("index",{proddata:data})

}

})

});

//it will pass the reference of myrouter in router variable of app.js file

module.exports=myrouter;

1. Write index.ejs file views folder as follows

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<header>

<h1>Product Management system</h1>

</header>

<div>

<form action="/addproduct">

<button type="submit" name="add"id="add" value="add" >add new product</button>

</form>

<table border="2">

<tr>

<th>Product id</th>

<th>Name</th>

<th>qty</th>

<th>price</th>

<th>action</th>

</tr>

<% for(var i=0;i<proddata.length;i++){%>

<tr>

<td><%=proddata[i].pid%></td>

<td><%=proddata[i].pname%></td>

<td><%=proddata[i].qty%></td>

<td><%=proddata[i].price%></td>

<td><a href="/deleteprod/<%=proddata[i].pid%>">delete</a></td>

</tr>

<%}%>

</table>

</div>

</body>

</html>

1. Add following code in myroter.js to show empty form

//to display empty form to accept new product

myrouter.get("/addproduct",function(req,resp){

resp.render("add-prod")

})

1. Add add-prod.ejs file in views folder

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<form action="/insertprod" method="post">

Product id: <input type="text" name="pid" id="pid"

placehoder="Enter pid"><br>

Product name: <input type="text" name="pname" id="pname"

placehoder="Enter pname"><br>

Product qty: <input type="text" name="qty" id="qty"

placehoder="Enter qty"><br>

Product price: <input type="text" name="price" id="price"

placehoder="Enter price"><br>

<button type="submit" name="btn" id="btn" value="btn">Add new Product</button>

</form>

</body>

</html>

1. Add following code in router.js

//to add new product in the table

myrouter.post("/insertprod",function(req,resp){

connection.query("insert into product values(?,?,?,?)",[req.body.pid,req.body.pname,req.body.qty,req.body.price],(err,result)=>{

if(err){

resp.status(500).send("no data added");

}else{

resp.redirect("/products")

}

})

})

1. To delete data add following code in router.js

myrouter.get("/deleteprod/:prodid",function(req,resp){

connection.query("delete from product where pid=?",[req.params.prodid],(err,result)=>{

if(err){

resp.status(500).send("no data deleted");

}else{

resp.redirect("/products")

}

})

})

