**Day 4 : 14 Sep 2024**

Verify the PostGres SQL Database in VM

C:\Program Files\PostgreSQL\16\bin

psql -U postgres

password : provide your password password in VM 123

\list to check all database present in your account.

create database testdb; it is use to create the database.

\c testdb; it is use to switch inside testdb database.

create table product(pid int primary key,pname varchar(50), price float);

insert into product values(1,'TV',67000);

insert into product values(2,'Computer',45000);

select \* from product;

mkdir express-product-restapi-with-postgres

cd express-product-restapi-with-postgres

npm init or npm init -y

npm install express pg

pg -🡪 for post gres database

mysql 🡪 for mysql database

mongodb-🡪 mongo db database.

pg is one the external third party module which provide set of api which hep to connect postgres sql database using node js

Creating express js application using MVC style

config --🡪Folder 🡪dbConfig.js which contains database connection details.

repository -🡪Folder (contains more than one file which contains database related code). 🡪ProductRepository, OrderRepository, AccountRepository, LogingRepository etc.

these file import or load dbConfig file and get the reference of Pool and do the operation on table etc.

controller -🡪 folder (it is responsible to receive the value from routing file). Base on controller method it call respective repository method and return result to router file. Controller contains request and response object. ProductController, OrderController, AccountController, LoginController

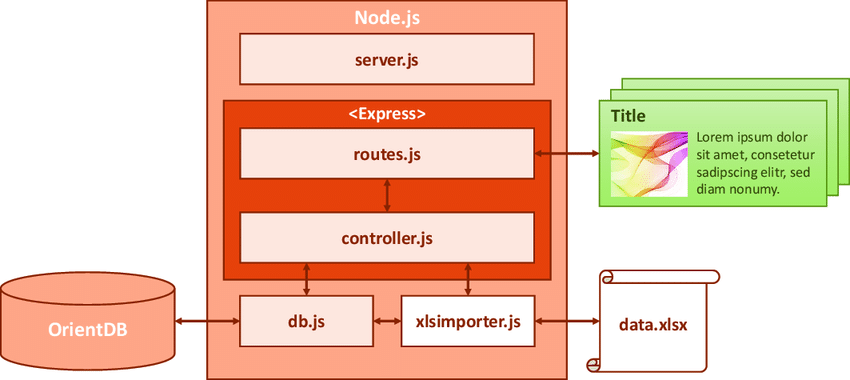
router -🡪 folder , this file routing the request base upon http method and sub path and it pass the request to controller method. router.js (generic file), productRouter.js, orderRouter.js etc.

app.js or main.js

this file responsible to load required module like express or middleware module like cors etc

then provide the main path for the application and redirect the request to router file base upon the main path.

It provide listener details ie port number and host name.



mkdir express-mvc-postgres-pg

cd express-mvc-postgres-pg

npm init -y this command is use to create package.json file

npm install express pg (pg like jdbc )

nodemon this module is use to refresh the application whenever we do any changes in the application. This module is third party modules we need to installed globally.

npm install nodemon -g

nodemon app.js

middleware modules

these modules add in between client and server modules. Whenever client send the request to specific end point before each request to server we want to perform some task that time we can use different type of middleware modules.

These middleware modules are can pre defined or user defined or custom modules.

Syntax to user middleware module

let express = require(“express”);

let app = express();

app.use(middlewaremodulename);

bodyParser is one of the type of middleware module to extract the data from request body part

generally we need to use while using post or put or patch method.

creating custom modules.

mkdir custom-middleware-module

cd custom-middleware-module

npm init -y

npm install express

creating custom validator middleware

Client Side validation 🡪 Using JavaScript, Using React or Angular Framework

React JS Validation using third party module is Formik and Yup Client side validation

To do backend side validation we can take the help of custom middleware validation.

Creating another middleware module which verify token present or not.

Rate limiting concept this is a technique we generally us to restrict the user to send request to application within a time period.

Custom rate limit middle are for express js.

Few pre defined external third party modules

morgan : it is a type of middleware module which help to do the keep the track about the application execution using logger features. That logger details we can display on console with different format or we can store in log file.

mkdir express-with-morgan

cd express-with-morgan

npm init -y

npm install express morgan

session tracking :

1st Req---------------------------------🡪

2nd req + session id

Client Server

🡨------- res + cookies (session Id)

session is a collection of request and response within a particular period of a time. by default http is stateless protocol. Using some technique we need handle the session.

1. Cookies : cookies is small text file create by server when client send first request to server. This file stored on client machine. Which contains unique session id, path, life of the cookies.
2. When cookies disable on client machine generally server side technique they send session id through URL.

URL?sessionID=%%%%###@@@@@2

This technique support only get method not other method.

1. Session api : expres-session which help to keep track account the session.
2. JWT token

REST API

Frontend backend

You need to hold

That id sessionStorage or localStorage

Client Whole application as backend with view as html, css and js

To work with cookies and session

Third party modules

cookie-parser equal to Cookies class

express-session equal to HttpSession

mkdir session-tracking-demo

cd session-tracking-demo

npm init -y

npm install express cookies-parser express-session

to connect post gres sql db we use pg module with pool api

to connect mysql sql db we use mysql module with pool api

we need to depends upon query.

when we want to connect database like RDBMS database

like mysql or oracle or db2 or post gres sql

ORM : Object Relation Mapping

In node JS side we need model class we defined table name or entity name with all attribute and data type as well as constraints.

Sequelize modules which provide ORM features using Node J or Express JS to store as object in RDBMS rather than depending sql query.