**REST**

การโอนย้ายสถานะโดยใช้ตัวแทน เช่น GET POST PUT DELETE

**URI(Uniform Resource Identifier)/URL (Uniform Resource Locator)**

URI เป็นเหมือน "ชื่อ" ทั่วไป ในขณะที่ URL เป็นเหมือน "ที่อยู่" ที่บอกวิธีการไปถึงทรัพยากรนั้น ในที่นี้หมายถึง Resource (ทรัพยากร)

Uri = เลขที่บ้าน Url = ที่อยู่บ้าน

Request = client sent Response = server sent

Method Body ส่งไปได้ทุก syntax ของ rest GET POST PUT DELETE แต่อยู่ที่ server

จะรับหรือไม่

Params คือ เข้าถึงค่าของพารามิเตอร์ ที่มากับการ router และ Query String

อาจจะมาผ่านทางเดียวหรือสองทางพร้อมกันก็ได้

Router 1 ตัวไม่จำเป็นต้องมี พารามิเตอร์ตัวเดียว

พารามิเตอร์คือคู่ ชื่อ : ค่า (เช่น **'currency': 'USD'**) name:value

Query คือการ การเข้าถึงพารามิเตอร์ ที่มาจาก queryString มาพร้อมกับ method GET

**Response**

**Method**

**Res.send(text)**

**Res.sendFile(file)**

**Res.sendStatus(status)**

**100-info 200-succese 300-redirect 400-cliant 500-server**

**Res.json(jsonFormat)**

**Middleware = ตัวกลางที่คอยจัดการการเชื่อมต่อ ทุกๆด้าน**

Incoming request – client ชุดคำสั่ง software

Logging การลงบันทึกประวัติ

**Workshop start**

**Cmd cdเข้าโปรเจ็ค**

**{**

**"name": "restaurant\_api",**

**"version": "1.0.0",**

**"description": "Reatful\_api for restaurant Projest",**

**"main": "index.js",**

**"scripts": {**

**"dev":"nodemon index.js",**

**"start": "node index.js",**

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

**},**

**"author": "Phubate Nimnuan",**

**"license": "ISC",**

**"dependencies": {**

**"express": "^4.19.2"**

**}**

**}**

**ใส่ใน package.json ที่สร้างไว้**

**สร้างไฟล์ .env ไว้ที่โฟลเดอร์หลัก**

**ใส่เข้าไป**

**USER="default"**

**HOST="ep-square-night-a1yhcsnz-pooler.ap-southeast-1.aws.neon.tech"**

**PASSWORD="etzSGy0smAE9"**

**DATABASE="verceldb"**

**สร้างไฟล์ index.js**

**const express = require('express');**

**const app = express();**

**const PORT = 5000;**

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

**res.send("<h1>Hello restaurant api</h>")**

**});**

**app.listen(PORT, ()=>{**

**console.log("Listen to http://localhost:"+PORT);**

**})**

**สร้างโฟลเดอร์ config controllers middlewares models**

**ใน config มีไฟล์ db.config.js**

require("dotenv").config();

module.exports = {

    HOST: "ep-square-night-a1yhcsnz-pooler.ap-southeast-1.aws.neon.tech",

    USER: "default",

    PASSWORD:"etzSGy0smAE9",

    DB:"verceldb",

    dialect: "postgres",

    pool:{

        max:5,

        min:0,

        acquire:30000,

        idle:10000,

    },

};

**ใน models มีไฟล์ db.js**

const {Sequelize} = require("sequelize");

const dbConfig = require("../config/db.config");

const sequelize = new Sequelize(dbConfig.DB, dbConfig.USER,dbConfig.PASSWORD, {

    host: dbConfig.HOST,

    dialect: dbConfig.dialect,

    dialectOptions:{

        ssl:{require: true,

        rejectUnauthorized:false,

        },

    },

  });

testConnection = async () =>{

    try {

        await sequelize.authenticate();

        console.log("Connection has been established successfully.");

      } catch (error) {

        console.error("Unable to connect to the database:", error);

      }

}

testConnection();

module.exports = sequelize;

**\Restaurant\_API\models>npm install --save sequelize**

**\** **Restaurant\_API\models>npm install --save pg pg-hstore**

**เช็คว่าเชื่อมต่อ API สำเร็จ**

**Restaurant\_API\models>node db.js**