**PROJECT TOKO KOMPUTER**

**(CRUD CUSTOMER)**

## Persiapan Library

1. Install library yang dibutuhkan yaitu multer dengan command

npm i --save multer

Multer adalah middleware node.js untuk menangani multipart/form-data yang biasanya digunakan untuk mengunggah file

1. Buat file customer.js pada routes
2. Import library dan model seperti berikut

| //import express  const express = require("express")  const app = express()  app.use(express.json())  // import md5  const md5 = require("md5")  //import multer  const multer = require("multer")  const path = require("path")  const fs = require("fs")  //import model  const models = require("../models/index")  const customer = models.customer |
| --- |

1. Tambahkan konfigurasi destination dan filename seperti berikut

| //config storage image  const storage = multer.diskStorage({  destination:(req,file,cb) => {  cb(null,"./image/customer")  },  filename: (req,file,cb) => {  cb(null, "img-" + Date.now() + path.extname(file.originalname))  }  })  let upload = multer({storage: storage}) |
| --- |

1. Tambahkan endpoint untuk menampilkan semua data customer

| app.get("/", (req, res) =>{  customer.findAll()  .then(result => {  res.json({  customer: result  })  })  .catch(error => {  res.json({  message: error.message  })  })  }) |
| --- |

1. Tambahkan endpoint untuk menampilkan data customer berdasarkan id

| app.get("/:customer\_id", (req, res) =>{  customer.findOne({ where: {customer\_id: req.params.customer\_id}})  .then(result => {  res.json({  customer: result  })  })  .catch(error => {  res.json({  message: error.message  })  })  }) |
| --- |

1. Tambahkan endpoint untuk menambahkan data customer baru

| app.post("/", upload.single("image"), (req, res) =>{  if (!req.file) {  res.json({  message: "No uploaded file"  })  } else {  let data = {  name: req.body.name,  phone: req.body.phone,  address: req.body.address,  image: req.file.filename,  username: req.body.username,  password: md5(req.body.password)  }  customer.create(data)  .then(result => {  res.json({  message: "data has been inserted"  })  })  .catch(error => {  res.json({  message: error.message  })  })  }  }) |
| --- |

1. Tambahkan endpoint untuk mengubah data customer berdasarkan id

| app.put("/:id", upload.single("image"), (req, res) =>{  let param = { customer\_id: req.params.id}  let data = {  name: req.body.name,  phone: req.body.phone,  address: req.body.address,  username: req.body.username  }  if (req.file) {  // get data by id  const row = customer.findOne({where: param})  .then(result => {  let oldFileName = result.image    // delete old file  let dir = path.join(\_\_dirname,"../image/customer",oldFileName)  fs.unlink(dir, err => console.log(err))  })  .catch(error => {  console.log(error.message);  })  // set new filename  data.image = req.file.filename  }  if(req.body.password){  data.password = md5(req.body.password)  }  customer.update(data, {where: param})  .then(result => {  res.json({  message: "data has been updated",  })  })  .catch(error => {  res.json({  message: error.message  })  })  }) |
| --- |

1. Tambahkan endpoint untuk menghapus data customer berdasarkan id

| app.delete("/:id", async (req, res) =>{  try {  let param = { customer\_id: req.params.id}  let result = await customer.findOne({where: param})  let oldFileName = result.image    // delete old file  let dir = path.join(\_\_dirname,"../image/customer",oldFileName)  fs.unlink(dir, err => console.log(err))  // delete data  customer.destroy({where: param})  .then(result => {    res.json({  message: "data has been deleted",  })  })  .catch(error => {  res.json({  message: error.message  })  })  } catch (error) {  res.json({  message: error.message  })  }  }) |
| --- |

1. Terakhir, export file customer.js

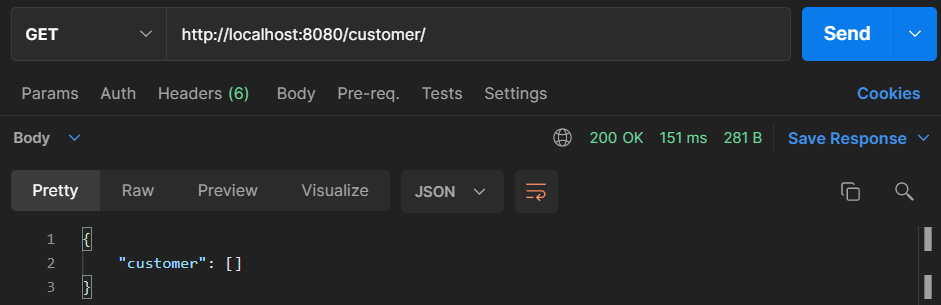
| module.exports = app |
| --- |

1. Agar bisa dijalankan, jangan lupa import pada file index.js

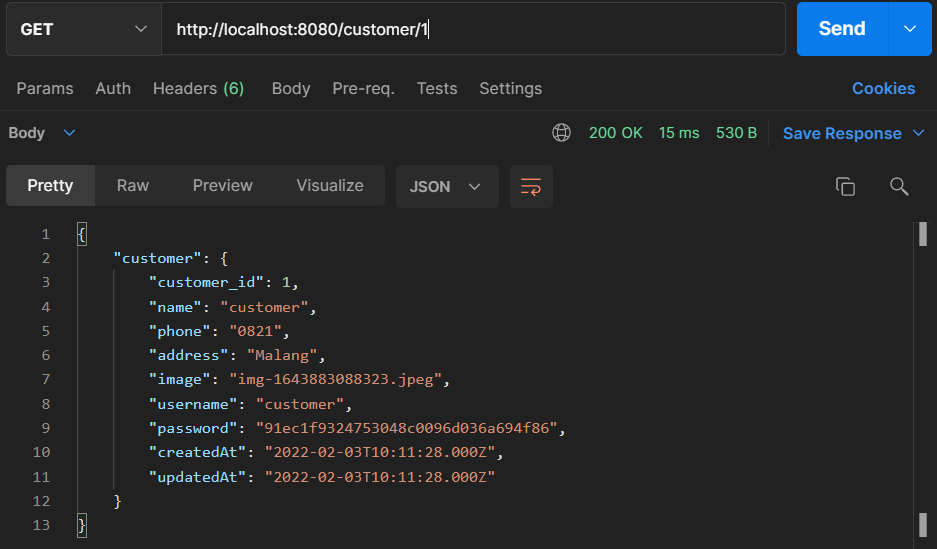
| //endpoint customer  const customer = require('./routes/customer');  app.use("/customer", customer) |
| --- |

1. Jalankan dan Test hasil API pada postman

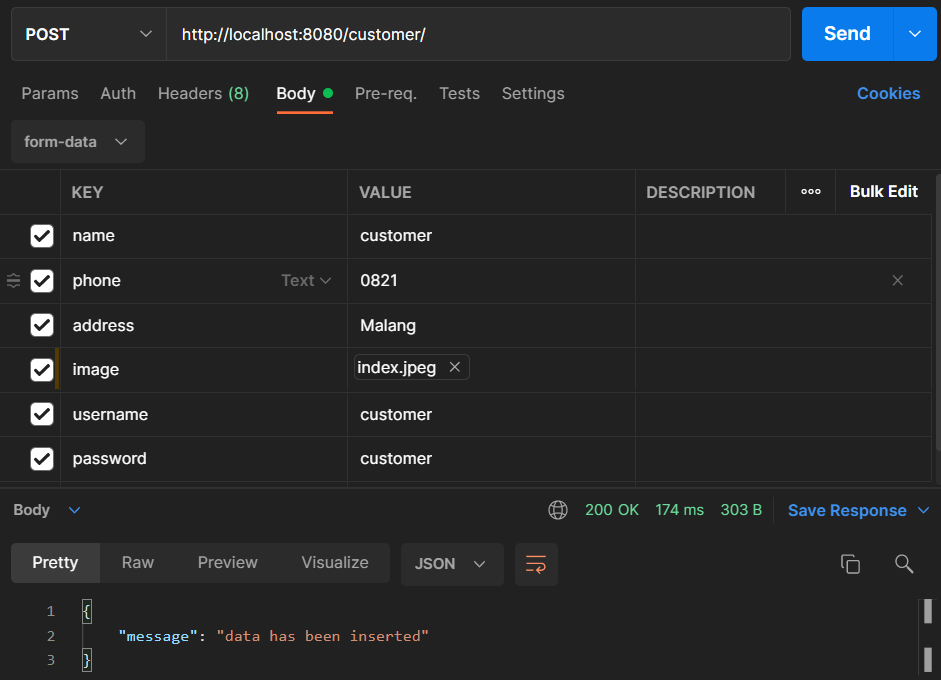
GET ALL Customer



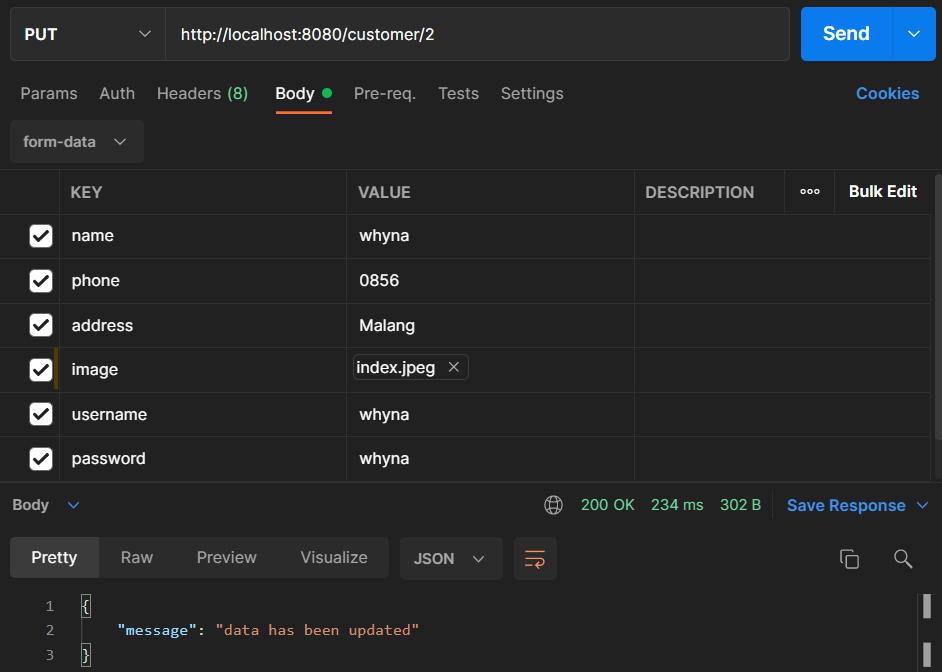
GET BY ID Customer



POST Customer



PUT Customer



DELETE Customer

