**REST API Upload File + MySQL**

Pada modul ini, kita akan belajar bagaimana cara mengunggah file melalui REST API Node JS yang diintegrasikan dengan database MySQL. Ikuti praktikum berikut ini dengan penuh ketelitian!

**Langkah praktikum**

* Buatlah database “olshop” dengan ketentuan tabel seperti berikut ini!



* Siapkan folder project “olshop” dan install library pendukungnya yaitu express, cors, body-parser, multer, mysql
* Buatlah file api.js dan inisiasi library yang telah diinstal.

|  |
| --- |
| const express = require("express")  const app = express()  const multer = require("multer") // untuk upload file  const path = require("path") // untuk memanggil path direktori  const fs = require("fs") // untuk manajemen file  const mysql = require("mysql")  const cors = require("cors")  app.use(express.static(\_\_dirname));  app.use(express.json())  app.use(express.urlencoded({extended: true}))  app.use(cors()) |

* Buatlah folder “image” pada root folder untuk menyimpan file image.
* Buat variabel untuk konfigurasi proses upload file

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

* Buat variable untuk konfigurasi koneksi ke database

|  |
| --- |
| const db = mysql.createConnection({      host: "localhost",      user: "root",      password: "",      database: "olshop"  }) |

* Buat endpoint untuk menambah data barang

|  |
| --- |
| // endpoint untuk menambah data barang baru  app.post("/barang", upload.single("image"), (req, res) => {      // prepare data      let data = {          nama\_barang: req.body.nama\_barang,          harga: req.body.harga,          stok: req.body.stok,          deskripsi: req.body.deskripsi,          image: req.file.filename      }      if (!req.file) {          // jika tidak ada file yang diupload          res.json({              message: "Tidak ada file yang dikirim"          })      } else {          // create sql insert          let sql = "insert into barang set ?"          // run query          db.query(sql, data, (error, result) => {              if(error) throw error              res.json({                  message: result.affectedRows + " data berhasil disimpan"              })          })      }  }) |

* Buatlah endpoint untuk mengubah data barang!

|  |
| --- |
| // endpoint untuk mengubah data barang  app.put("/barang", upload.single("image"), (req,res) => {      let data = null, sql = null      // paramter perubahan data      let param = { kode\_barang: req.body.kode\_barang }      if (!req.file) {          // jika tidak ada file yang dikirim = update data saja          data = {              nama\_barang: req.body.nama\_barang,              harga: req.body.harga,              stok: req.body.stok,              deskripsi: req.body.deskripsi          }      } else {          // jika mengirim file = update data + reupload          data = {              nama\_barang: req.body.nama\_barang,              harga: req.body.harga,              stok: req.body.stok,              deskripsi: req.body.deskripsi,              image: req.file.filename          }          // get data yg akan diupdate utk mendapatkan nama file yang lama          sql = "select \* from barang where ?"          // run query          db.query(sql, param, (err, result) => {              if (err) throw err              // tampung nama file yang lama              let fileName = result[0].image              // hapus file yg lama              let dir = path.join(\_\_dirname,"image",fileName)              fs.unlink(dir, (error) => {})          })      }      // create sql update      sql = "update barang set ? where ?"      // run sql update      db.query(sql, [data,param], (error, result) => {          if (error) {              res.json({                  message: error.message              })          } else {              res.json({                  message: result.affectedRows + " data berhasil diubah"              })          }      })  }) |

* Buatlah endpoint untuk menghapus data barang berdasarkan kode\_barang!

|  |
| --- |
| // endpoint untuk menghapus data barang  app.delete("/barang/:kode\_barang", (req,res) => {      let param = {kode\_barang: req.params.kode\_barang}      // ambil data yang akan dihapus      let sql = "select \* from barang where ?"      // run query      db.query(sql, param, (error, result) => {          if (error) throw error            // tampung nama file yang lama          let fileName = result[0].image          // hapus file yg lama          let dir = path.join(\_\_dirname,"image",fileName)          fs.unlink(dir, (error) => {})      })      // create sql delete      sql = "delete from barang where ?"      // run query      db.query(sql, param, (error, result) => {          if (error) {              res.json({                  message: error.message              })          } else {              res.json({                  message: result.affectedRows + " data berhasil dihapus"              })          }      })  }) |

* Buatlah endpoint untuk mendapatkan data barang!

|  |
| --- |
| // endpoint ambil data barang  app.get("/barang", (req, res) => {      // create sql query      let sql = "select \* from barang"      // run query      db.query(sql, (error, result) => {          if (error) throw error          res.json({              data: result,              count: result.length          })      })  }) |

* Jangan lupa untuk mengatur port dimana server berjalan

|  |
| --- |
| app.listen(8000, () =>{      console.log("Server run on port 8000");  }) |

* Jalankan file app.js
* Buka Postman dan kita akan menambahkan data baru dengan endpoint seperti berikut ini!



Pada bagian tab **Body**, pilih opsi **form-data** (karena data yang dikirimkan mengandung tipe file) dan isi data yang akan dikirimkan termasuk memilih file dari barang yang akan ditambah.



* Karena kita menggunakan tipe **form-data**, maka dibagian tab **Headers**, kita *disable* (*uncheck*) “**Content-Type**” yang berisi “**application/x-www-form-urlencoded**”



* Klik tombol **Send** dan kita akan mendapatkan *response* seperti berikut ini.



* Akses data barang melalui endpoint berikut.



Dan kita akan mendapatkan *response* berikut ini.



Pastikan proses upload file berhasil dengan mengecek file yang ada pada folder **image**.



* Silahkan uji endpoint untuk mengubah dan menghapus data dan pastikan ada perubahan file pada folder image! Selamat mencoba!