BE NodeJS

Tugas: Middleware Express JS

Leonardho R. Sitanggang

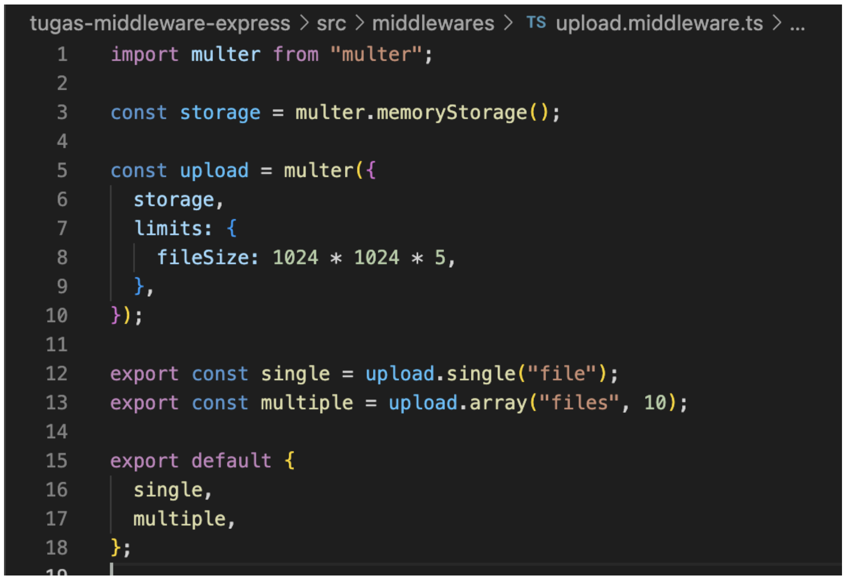
Soal :

1. Buatlah akun di **Cloudinary**
2. Pastikan bisa mendapatkan **CLOUD_NAME**, **API_SECRET** dan **API_KEY**, untuk panduan bisa akses **link berikut ini**
3. ubah source code **latihan-middleware-multer** agar bisa sesuai dengan kebutuhan berikut ini:

a. upload file satuan (single)
b. upload file lebih dari satu (multiple)
c. gunakan **Postman** untuk menguji API **/single** dan **/multiple,** ikut panduan mengupload file di postman

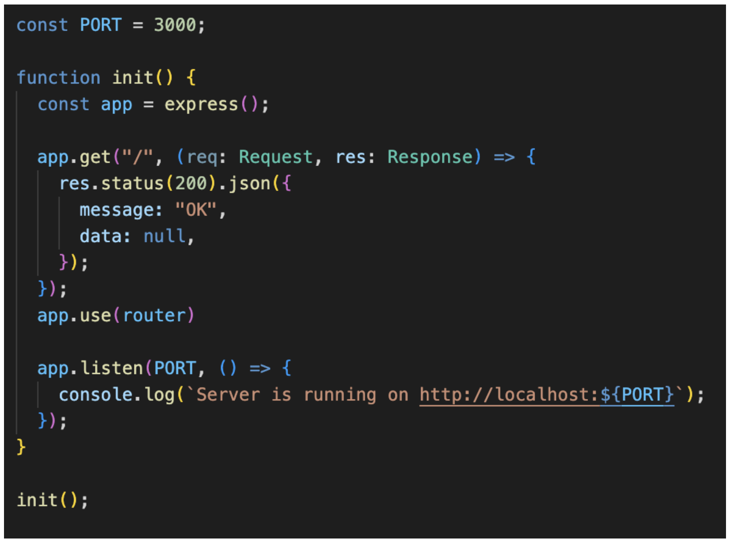
Jawab :

1. Source Code


tugas-middleware-express > src > middlewares > TS upload.middleware.ts > ...

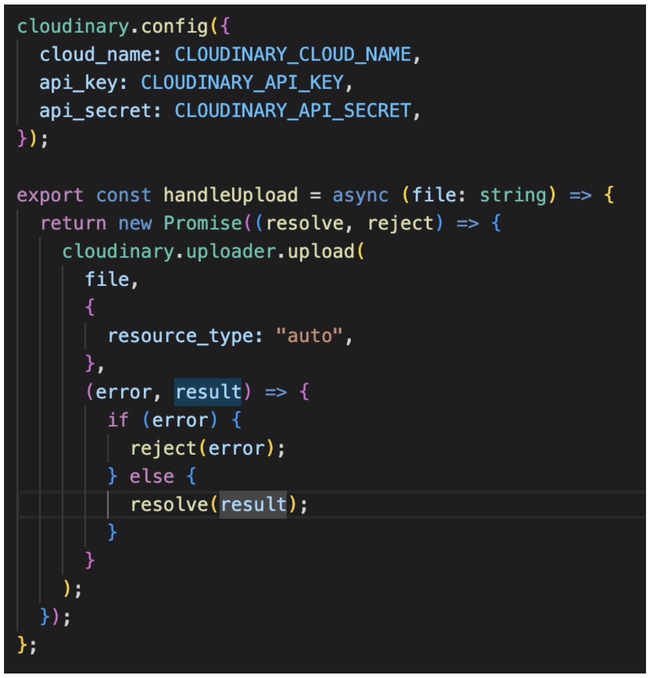
```
import multer from "multer";

const storage = multer.memoryStorage();

const upload = multer({
  storage,
  limits: {
    fileSize: 1024 * 1024 * 5,
  },
});

export const single = upload.single("file");
export const multiple = upload.array("files", 10);

export default {
  single,
  multiple,
};
```

Gambar 1.1 Middleware Multer untuk upload



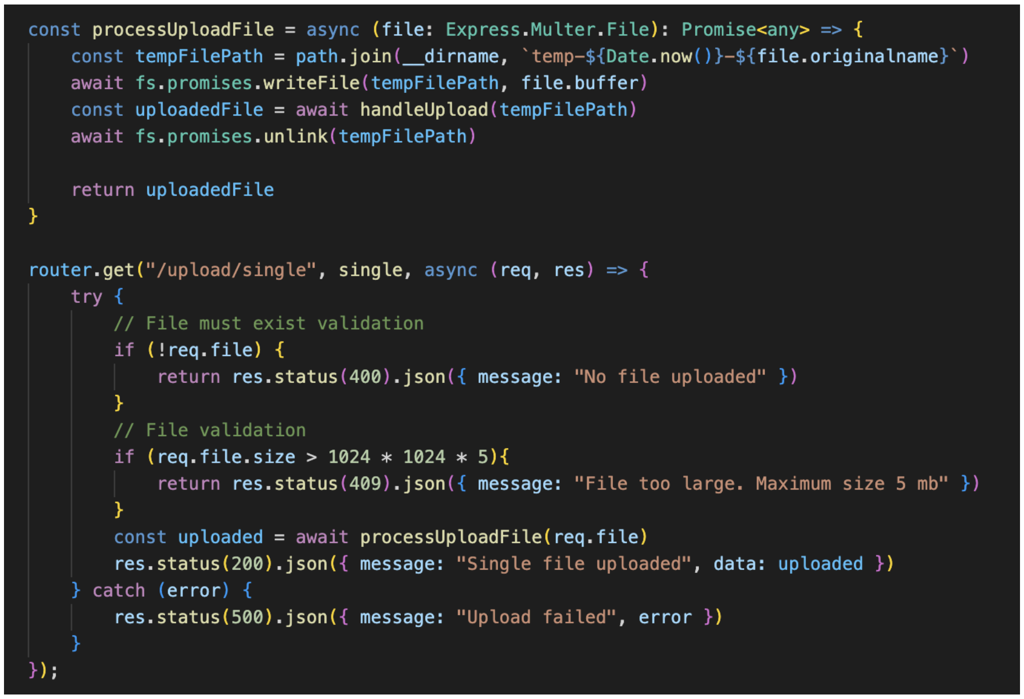
```
const PORT = 3000;

function init() {
  const app = express();

  app.get("/", (req: Request, res: Response) => {
    res.status(200).json({
      message: "OK",
      data: null,
    });
  });
  app.use(router)

  app.listen(PORT, () => {
    console.log(`Server is running on http://localhost:${PORT}`);
  });
}

init();
```

Gambar 1.2 Inisiasi Express



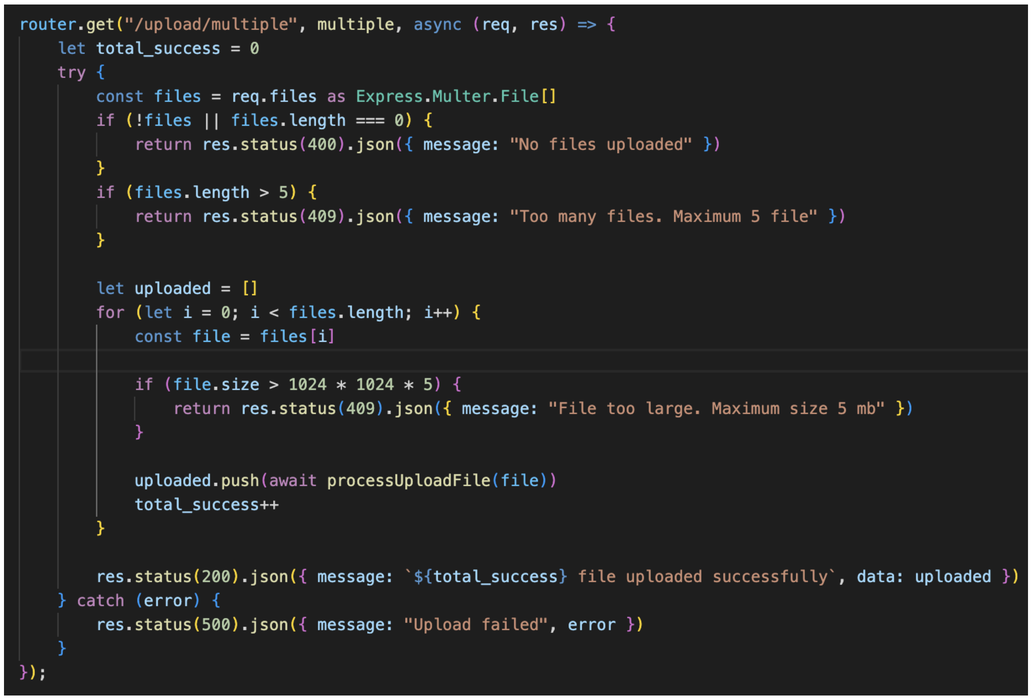
```
cloudinary.config({
  cloud_name: CLOUDINARY_CLOUD_NAME,
  api_key: CLOUDINARY_API_KEY,
  api_secret: CLOUDINARY_API_SECRET,
});

export const handleUpload = async (file: string) => {
  return new Promise((resolve, reject) => {
    cloudinary.uploader.upload(
      file,
      {
        resource_type: "auto",
      },
      (error, result) => {
        if (error) {
          reject(error);
        } else {
          resolve(result);
        }
      }
    );
  });
};
```

Gambar 1.3 Fungsi upload ke cloudinary



```
const processUploadFile = async (file: Express.Multer.File): Promise<any> => {
    const tempFilePath = path.join(__dirname, `temp-${Date.now()}-${file.originalname}`)
    await fs.promises.writeFile(tempFilePath, file.buffer)
    const uploadedFile = await handleUpload(tempFilePath)
    await fs.promises.unlink(tempFilePath)

    return uploadedFile
}

router.get("/upload/single", single, async (req, res) => {
    try {
        // File must exist validation
        if (!req.file) {
            return res.status(400).json({ message: "No file uploaded" })
        }
        // File validation
        if (req.file.size > 1024 * 1024 * 5){
            return res.status(409).json({ message: "File too large. Maximum size 5 mb" })
        }
        const uploaded = await processUploadFile(req.file)
        res.status(200).json({ message: "Single file uploaded", data: uploaded })
    } catch (error) {
        res.status(500).json({ message: "Upload failed", error })
    }
});
```

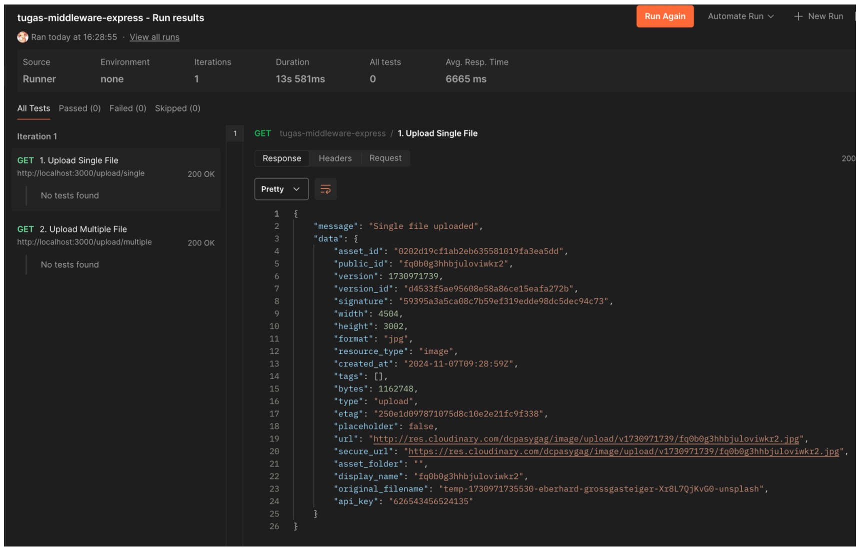
Gambar 1.4 Fungsi skema temporary file sebelum upload ke cloud dan API Get Single Upload



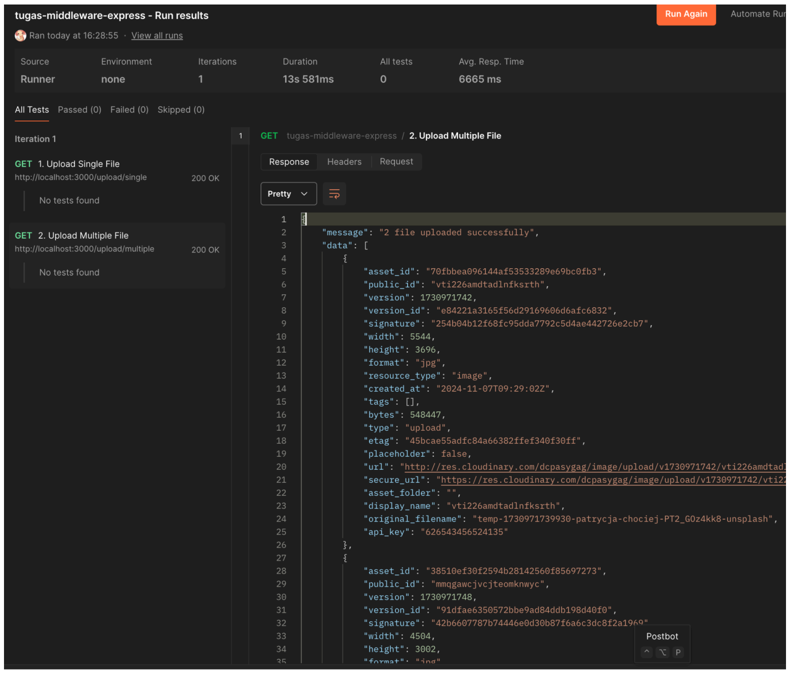
```
router.get("/upload/multiple", multiple, async (req, res) => {
    let total_success = 0
    try {
        const files = req.files as Express.Multer.File[]
        if (!files || files.length === 0) {
            return res.status(400).json({ message: "No files uploaded" })
        }
        if (files.length > 5) {
            return res.status(409).json({ message: "Too many files. Maximum 5 file" })
        }

        let uploaded = []
        for (let i = 0; i < files.length; i++) {
            const file = files[i]

            if (file.size > 1024 * 1024 * 5) {
                return res.status(409).json({ message: "File too large. Maximum size 5 mb" })
            }

            uploaded.push(await processUploadFile(file))
            total_success++
        }

        res.status(200).json({ message: `${total_success} file uploaded successfully`, data: uploaded })
    } catch (error) {
        res.status(500).json({ message: "Upload failed", error })
    }
});
```

Gambar 1.5 API Get Multiple Upload

Output / Testing


tugas-middleware-express - Run results
Run Again
Automate Run
New Run
Ran today at 16:28:55 · View all runs
Source Runner
Environment none
Iterations 1
Duration 13s 581ms
All tests 0
Avg. Resp. Time 6665 ms
All Tests
Passed (0)
Failed (0)
Skipped (0)
Iteration 1
GET 1. Upload Single File
http://localhost:3000/upload/single 200 OK
No tests found
GET 2. Upload Multiple File
http://localhost:3000/upload/multiple 200 OK
No tests found
GET tugas-middleware-express / 1. Upload Single File
Response
Headers
Request
Pretty

```
{
    "message": "Single file uploaded",
    "data": {
        "asset_id": "0202d19cf1ab2eb635581019fa3ea5dd",
        "public_id": "fq0b0g3hhbjuloviwkr2",
        "version": 1730971739,
        "version_id": "d4533f5ae95608e58a86ce15eafa272b",
        "signature": "59395a3a5ca08c7b59ef319edde98dc5dec94c73",
        "width": 4504,
        "height": 3002,
        "format": "jpg",
        "resource_type": "image",
        "created_at": "2024-11-07T09:28:59Z",
        "tags": [],
        "bytes": 1162748,
        "type": "upload",
        "etag": "250e1d097871075d8c10e2e21fc9f338",
        "placeholder": false,
        "url": "http://res.cloudinary.com/dcpasygag/image/upload/v1730971739/fq0b0g3hhbjuloviwkr2.jpg",
        "secure_url": "https://res.cloudinary.com/dcpasygag/image/upload/v1730971739/fq0b0g3hhbjuloviwkr2.jpg",
        "asset_folder": "",
        "display_name": "fq0b0g3hhbjuloviwkr2",
        "original_filename": "temp-1730971735530-eberhard-grossgasteiger-Xr8L7QjKvG0-unsplash",
        "api_key": "626543456524135"
    }
}
```


tugas-middleware-express - Run results
Run Again
Automate Run
Ran today at 16:28:55 · View all runs
Source Runner
Environment none
Iterations 1
Duration 13s 581ms
All tests 0
Avg. Resp. Time 6665 ms
All Tests
Passed (0)
Failed (0)
Skipped (0)
Iteration 1
GET 1. Upload Single File
http://localhost:3000/upload/single 200 OK
No tests found
GET 2. Upload Multiple File
http://localhost:3000/upload/multiple 200 OK
No tests found
GET tugas-middleware-express / 2. Upload Multiple File
Response
Headers
Request
Pretty
Postbot

```
{
    "message": "2 file uploaded successfully",
    "data": [
        {
            "asset_id": "70fbbea096144af53533289e69bc0fb3",
            "public_id": "vti226amdtadlnfksrth",
            "version": 1730971742,
            "version_id": "e84221a3165f56d29169606d6afc6832",
            "signature": "254b04b12f68fc95dda7792c5d4ae442726e2cb7",
            "width": 5544,
            "height": 3696,
            "format": "jpg",
            "resource_type": "image",
            "created_at": "2024-11-07T09:29:02Z",
            "tags": [],
            "bytes": 548447,
            "type": "upload",
            "etag": "45bcae55adfc84a66382ffef340f30ff",
            "placeholder": false,
            "url": "http://res.cloudinary.com/dcpasygag/image/upload/v1730971742/vti226amdtad
            "secure_url": "https://res.cloudinary.com/dcpasygag/image/upload/v1730971742/vti22
            "asset_folder": "",
            "display_name": "vti226amdtadlnfksrth",
            "original_filename": "temp-1730971739930-patrycja-chociej-PT2_GOz4kk8-unsplash",
            "api_key": "626543456524135"
        },
        {
            "asset_id": "38510ef30f2594b28142560f85697273",
            "public_id": "mmqgawcjvcjteomknwyc",
            "version": 1730971748,
            "version_id": "91dfae6350572bbe9ad84ddb198d40f0",
            "signature": "42b6607787b74446e0d30b87f6a6c3dc8f2a1969"
            "width": 4504,
            "height": 3002,
            "format": "jpg"
```

Gambar 1.6 Runner dari API Get Single & Get Multiple Upload

Repository :

- *Github* *https://github.com/FlazeFy/sanbercode-nodejs-xpress/tree/main/tugas-middleware-express*
- *Postman Documentation* *https://documenter.getpostman.com/view/18882129/2sAY51911r*

Refrensi :

- *Materi Pekan 2.1 Framework Express JS* *https://sanbercode.com/bootcamp/class/684/materi/2-1-framework-expressjs/dc0618b7-fd52-4eec-a578-5ed88c4d5b3c*
- *Materi Pekan 2.2 Routing Rest API* *https://sanbercode.com/bootcamp/class/684/materi/routing-rest-api/de70a7be-9056-40e0-bed0-c280c6bc1de0*
- *Materi Pekan 2.3 Middleware Express*

https://sanbercode.com/bootcamp/class/684/materi/2-3-middleware-expressjs/c33672de-a33d-4916-b07c-9221b9e836b5

- *Cloudinary*
  https://cloudinary.com/
- *Postman File Upload*
  https://apidog.com/blog/postman-upload-file-detailed-guide/
- *Cara Upload Photo via Cloudinary dengan Express*
  https://bilkisismail07.medium.com/cara-upload-photo-image-dengan-cloudinary-pada-node-js-dan-express-part-1-c967aa14074c