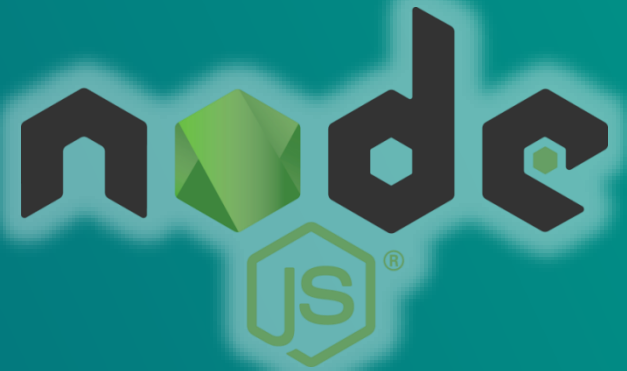


Back-End Development

Exploring

Multer Module



Multer

Multer is a node.js middleware for handling multipart/form-data, which is primarily used for uploading files.

NOTE: Multer will not process any form which is not multipart (multipart/form-data).

Documentation Link : <https://www.npmjs.com/package/multer>

Installation

- Run (npm install multer express cors) in your backend project, after you run (npm init), make sure the package.json already exist first.

```
{
  "name": "api-upload-image",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Baron",
  "license": "ISC",
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^4.17.1",
    "multer": "^1.4.2"
  }
}
```

← Libraries Installed

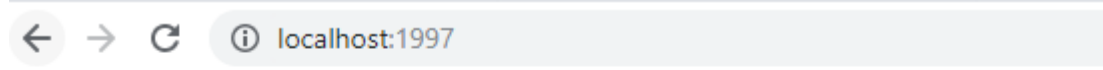
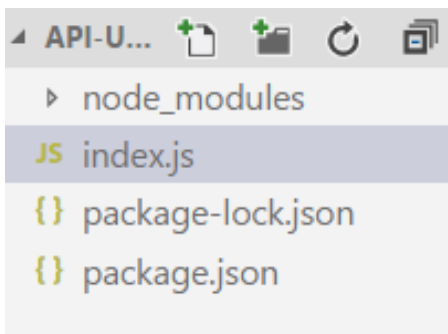
Create index.js

```
const express = require('express')
const cors = require('cors')

const port = 1997
const app = express()
app.use(cors())

app.get('/', (req, res) => {
  res.status(200).send('<h1>Selamat datang di Upload Image API!</h1>')
})

app.listen(port, () => console.log(`API aktif di port ${port}`))
```



Selamat datang di Upload Image API!

Create uploader.js

```
const multer = require('multer');
var fs = require('fs');

// Return multer object

module.exports = {
  uploader(destination, fileNamePrefix){
    let defaultPath = './public';
    const storage = multer.diskStorage({
      destination: (req, file, cb) => {
        const dir = defaultPath + destination;
        if (fs.existsSync(dir)) {
          console.log(dir, "exists")
          cb(null, dir);
        } else {
          fs.mkdir(dir, { recursive: true }, err => cb(err, dir));
          console.log(dir, "make")
        }
      },
      filename: (req, file, cb) => {
        let originalname = file.originalname;
        let ext = originalname.split('.');
        let filename = fileNamePrefix + Date.now() + '.' +
          ext[ext.length - 1];
        cb(null, filename);
      }
    });
  }
};
```

Continued in Next Slide

Create uploader.js

```
const imageFilter = (req, file, cb) => {  
  const ext = /\.(jpg|jpeg|png|gif|pdf|doc|docx|xlsx)$/;  
  if (!file.originalname.match(ext)) {  
    return cb(new Error('Only selected file type are allowed'), false);  
  }  
  cb(null, true);  
};  
  
return multer({  
  storage,  
  fileFilter: imageFilter  
});  
}
```

Edit index.js, add code below

```
const { uploader } = require('./uploader')
:
:
app.post('/addimage', (req,res) => {
  const path = '/test/images';
  const upload = uploader(path, 'TES').fields([{ name: 'image' }]);

  upload(req, res, (err) => {
    if(err){
      return res.status(500).json({ message: 'Upload files failed !',
        error: err.message });
    }



    const { image } = req.files;
    console.log(image)

    console.log(req.body.data)
    const data = JSON.parse(req.body.data);
    console.log(data)

    return res.status(200).send({ message: 'Upload File Success!' })
  })
})
```

Create React UI Project

Now run `create-react-app ui-upload-image` in the terminal, with the path same as where `api-upload-image` is located, so now you have both folder (api & ui)

Name	Date modified	Type	Size
 api-upload-image	13/09/2019 15:35	File folder	
 ui-upload-image	13/09/2019 16:02	File folder	

Install axios in ui project

npm install axios

```
"dependencies": {  
  "axios": "^0.19.0",  
  "react": "^16.9.0",  
  "react-dom": "^16.9.0",  
  "react-scripts": "3.1.1"  
},
```

Change App.js Content

```
import React, { Component } from 'react';

class App extends Component {
  state = { files: null }

  fileChange = (e) => {
    console.log(e.target.files)
    if(e.target.files[0]) {
      this.setState({ files: e.target.files })
    } else {
      this.setState({ files: null })
    }
  }

  render() {
    return (
      <div>
        <input type="file" onChange={this.fileChange} multiple={true} />
        <input type="button" value="Upload" />
      </div>
    )
  }
}

export default App;
```

Add this function in App.js

```
onBtnUploadClick = () => {
  if(this.state.files) {
    var formData = new FormData()
    var options = {
      headers: {
        'Content-Type': 'multipart/form-data'
      }
    }
    var data = {
      caption: 'testing bro'
    }

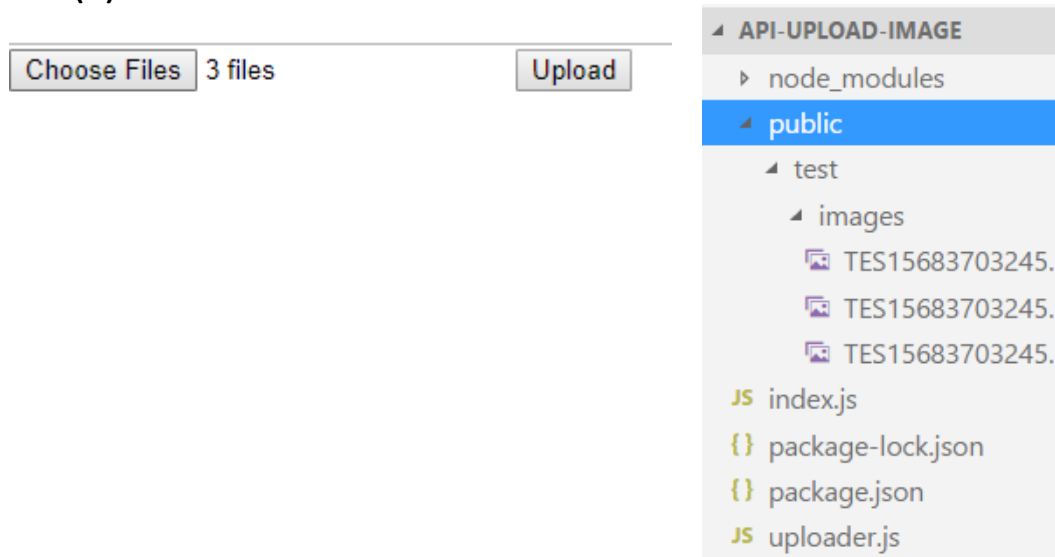
    for(var i = 0; i < this.state.files.length; i++) {
      formData.append('image', this.state.files[i])
    }
    formData.append('data', JSON.stringify(data))

    axios.post("http://localhost:1997/addimage", formData, options)
      .then((res) => {
        console.log(res.data)
      })
      .catch((err) =>{
        console.log(err.response)
      })
  }
  else {
    alert('Image harus diisi!')
  }
}
```

Don't forget to put the function on button onClick event :

```
render() {  
  return (  
    <div>  
      <input type="file" onChange={this.fileChange} multiple={true} />  
      <input type="button" value="Upload" onClick={this.onBtnUploadClick} />  
    </div>  
  )  
}
```

Now you can try it. You can upload 1 file or more. Check the console in the browser, and the terminal in the backEnd Project after you upload the file(s)



Now add this line of code in index.js of the backEnd Project, if you want to access the image from the API

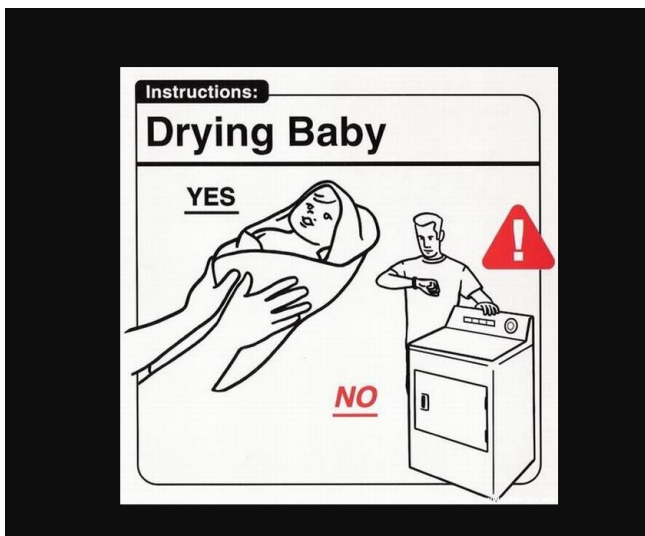
```
app.use(express.static('public'))
```

Open url like this in your browser to access the image

<http://localhost:1997/test/images/> (File Name)

Example :

<http://localhost:1997/test/images/TES1568370324559.jpg>



Back-End Development

Exploring

Multer Module

