**Jasmine: testing framework**

**Zest**

**App.component.spec.ts**

**Describe(){ //test suite**

**It() //test case**

**It()**

**It()**

**It()**

**}**

**Creating the Node.js File Upload Handler**

You need a file upload handler to handle the request sent from the Angular application. You'll be making use of the Express framework for creating the Node handler.

Create a folder called file-upload-server and initialize the Node project inside it.

mkdir file-upload-folder

cd file-upload-folder

npm init

Once you have the Node project initialized, install the express framework using npm.

npm install --save express

Create a file called app.js inside the project folder. You'll be making use of body-parser to parse the post parameter to the request handler. Install it now:

npm install --save body-parser

You'll also be using the connect-multiparty module to upload the file.

Once you have the required dependencies for writing the file upload handler, let's start by creating the Express app. Add the following code to the app.js file:

const express = require('express')

const app = express()

const port = 3000

const bodyParser = require("body-parser");

const multipart = require('connect-multiparty');

const multipartMiddleware = multipart({

uploadDir: './uploads'

});

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({

extended: true

}));

app.post('/api/upload', multipartMiddleware, (req, res) => {

res.json({

'message': 'File uploaded succesfully.'

});

});

app.listen(port, () => console.log(`Example app listening on port ${port}!`))

Angular Code

<input (change)="fileChange($event)" type="file" class="custom-file-input" id="inputGroupFile01" aria-describedby="inputGroupFileAddon01">

fileChange(element) {

this.uploadedFiles = element.target.files;

}

--

<span class="input-group-text upload" (click)="upload()" id="btnUpload">Upload</span>

--

upload() {

let formData = new FormData();

for (var i = 0; i < this.uploadedFiles.length; i++) {

formData.append("uploads[]", this.uploadedFiles[i], this.uploadedFiles[i].name);

}

this.http.post('/api/upload', formData)

.subscribe((response) => {

console.log('response received is ', response);

})

}

Complete Angular code

import { Component, OnInit } from '@angular/core';

import { HttpClient } from '@angular/common/http';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit {

uploadedFiles: Array < File > ;

constructor(private http: HttpClient) {

}

ngOnInit() {

}

fileChange(element) {

this.uploadedFiles = element.target.files;

}

upload() {

let formData = new FormData();

for (var i = 0; i < this.uploadedFiles.length; i++) {

formData.append("uploads[]", this.uploadedFiles[i], this.uploadedFiles[i].name);

}

this.http.post('/api/upload', formData)

.subscribe((response) => {

console.log('response received is ', response);

})

}

}

Complete NodeJS Code

#Create Node.js backend

First, install the following node modules.

npm install express multer body-parser dotenv --save

Install nodemon as a dev dependency.

npm install nodemon --save-dev

Create a new directory inside root of angular project called uploads.

Okay, now create one file in the angular project root folder called server.js.

Write the following piece code inside the server.js file.

// server.js

const path = require('path');

const express = require('express');

const multer = require('multer');

const bodyParser = require('body-parser')

const app = express();

const DIR = './uploads';

let storage = multer.diskStorage({

destination: (req, file, cb) => {

cb(null, DIR);

},

filename: (req, file, cb) => {

cb(null, file.fieldname + '-' + Date.now() + '.' + path.extname(file.originalname));

}

});

let upload = multer({storage: storage});

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({extended: true}));

app.use(function (req, res, next) {

res.setHeader('Access-Control-Allow-Origin', 'http://localhost:4200');

res.setHeader('Access-Control-Allow-Methods', 'POST');

res.setHeader('Access-Control-Allow-Headers', 'X-Requested-With,content-type');

res.setHeader('Access-Control-Allow-Credentials', true);

next();

});

app.get('/api', function (req, res) {

res.end('file catcher example');

});

app.post('/api/upload',upload.single('photo'), function (req, res) {

if (!req.file) {

console.log("No file received");

return res.send({

success: false

});

} else {

console.log('file received');

return res.send({

success: true

})

}

});

const PORT = process.env.PORT || 4000;

app.listen(PORT, function () {

console.log('Node.js server is running on port ' + PORT);

});