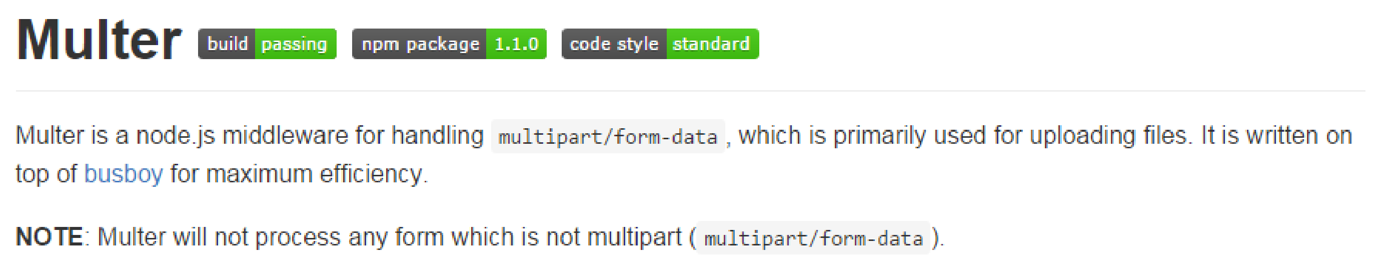
NodeJS File Upload Using Multer

Shams Nahid Follow
Oct 6, 2017 · 3 min read

Fork or clone the source code here



Multer is a node.js middleware for handling `multipart/form-data`, which is primarily used for uploading files. It is written on top of busboy for maximum efficiency.

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

Multer is a Node.js middleware, used to upload or handle multipart/form-data, on the top of Busboy. Here I'll explain the basic procedure of uploading a file in Node.js server.

Work-Flow

- First Get the seed code.
- Install dependencies.
- Create database named myproject.
- Install multer dependency.
- Configure multer.
- Voila!!! test through Postman

REQUIREMENTS

1. ***Node.js(6.x)***
2. ***NPM(3.x)***
3. ***Express.js(4.x)***

APP SETUP

Get the seed code from here.

First install all the node modules:

```
npm install
```

Create database named, myProject.

```
mongo
use myproject
```

Now run the project using

```
node server.js
or
nodemon
```

You have to see the line in console

```
The magic happens on port 8080
```

Multer Configaration

Install multer

```
npm install --save multer
```

Add the multer object to your routing file, ***routes/user/index.js***

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

We will upload the file in ***public/images/uploads*** directory. So our multer will contain the following configaration:

```
var storage = multer.diskStorage({
    destination: (req, file, cb) => {
      cb(null, 'public/images/uploads')
    },
    filename: (req, file, cb) => {
      cb(null, file.fieldname + '-' + Date.now())
    }
});
var upload = multer({storage: storage});
```

Now the multer is ready to upload the file in ***public/images/uploads*** folder

App Routing

Now we add the multer configuration in the routing file and add a dummy routing.

Now the ***routes/user/index.js*** will be looked like:

```
var express = require('express');
var router = express.Router();
var MongoClient = require('mongodb').MongoClient;
var assert = require('assert');
var url = 'mongodb://localhost:27017/myproject';

var multer = require('multer');
var storage = multer.diskStorage({
    destination: (req, file, cb) => {
      cb(null, 'public/images/uploads')
    },
    filename: (req, file, cb) => {
      cb(null, file.fieldname + '-' + Date.now())
    }
});
var upload = multer({storage: storage});


router.post('/fileUpload', upload.single('image'), (req, res, next)
=> {
    MongoClient.connect(url, (err, db) => {
        assert.equal(null, err);
        insertDocuments(db, 'public/images/uploads/' +
req.file.filename, () => {
            db.close();
            res.json({'message': 'File uploaded successfully'});
        });
    });
});


module.exports = router;


var insertDocuments = function(db, filePath, callback) {
    var collection = db.collection('user');
    collection.insertOne({'imagePath' : filePath }, (err, result) =>
{
        assert.equal(err, null);
        callback(result);
    });
}
```

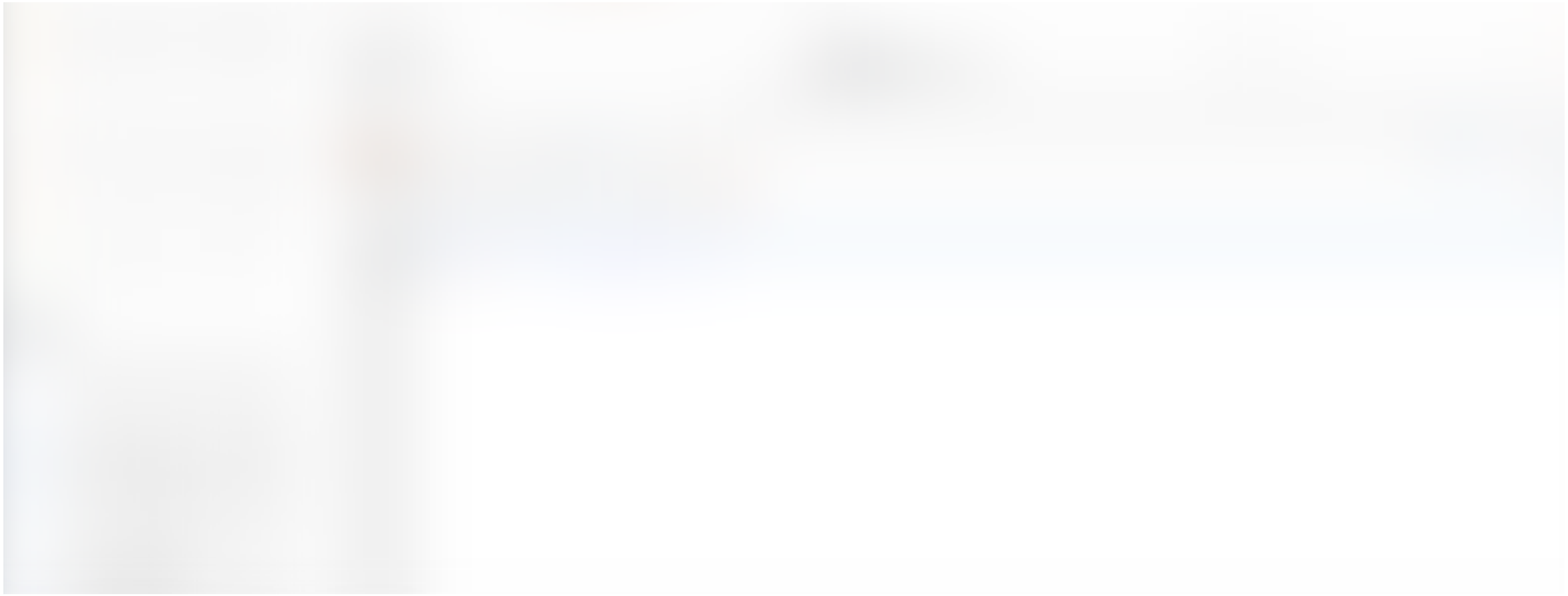
Now our code is ready for test.

Test

We will perform API test using Postman:

In postman

```
Request Type: POST
Address: http://localhost:8080/user/fileUpload
Body: form-data
Header: Null
Key: image
value: select a file from your machine
```



Now In your project structure, ***public/images/uploads***, you will find the uploaded file and as well as in the database of myproject the file path is saved already.

Conclusion

So Here we are, completed the basic file uploading using multer in node.js and express.js server. Stay tuned and if there is a confusing term or something, response below or create an issue in ***github***.

Nodejs Expressjs Multer Fileupload Mongodb

1K claps

WRITTEN BY

Shams Nahid

Follow

Love to travel, listen to music and hang out with friends. I believe in simplicity and peace.

See responses (9)

More From Medium

Related reads

Uploading Files and Serve Directory Listing Using NodeJS

Bilguun Batbold in Quick Code
Feb 18 · 4 min read

557

Related reads

Using Passport, Bcrypt, Express, & Handlebars in a Nodejs Full-Stack App for User Authentication

Roberto Baldizon in Silibrain
Jun 19, 2018 · 8 min read ★

645

Also tagged Expressjs

Building your GraphQL API with Node and MongoDB

Ishan Manandhar
Nov 24 · 5 min read ★

11