

# Learn Git and GitHub without any code!


Using the Hello World guide, you'll start a branch, write comments, and open a pull request.


Read the guide

 [expressjs](#) / [multer](#)

## Why does multer not save files with file extension by default? #439

New issue

 Open    mezod opened this issue on Jan 7, 2017 · 11 comments



mezod commented on Jan 7, 2017

What's the idea behind this decision? I'm asking because when serving the static files without file extension, for example, browsers cannot display images but they just download them. Seeing this is default I wonder how you serve the files then?

 1

Assignees

No one assigned

Labels


None yet

Projects

None yet

Milestone

No milestone




LinusU commented on Jan 7, 2017 • edited


Member

Good question, this should absolutely be added to the readme. I guess that my current best answer is this comment that I made before: [#170 \(comment\)](#).

In short, the client can send any file extension that they want, and blindly trusting that might be a bad idea. E.g. someone could upload a html file with some javascript, and give the .jpg extension. Since the server accepts images, it will accept this html file. When this file later is server, a "magic byte" or similar method is used to derive that this is an html file, and the file is served as such. The "attacker" now can run any arbitrary javascript under your domain name...

8 participants






mezod commented on Jan 7, 2017

Author

Hey Linus, thanks for your answer. I'm not sure if by linking to that answer you suggest to override the default behaviour by adding the file extension (even if its at our own risk of the problem you defined) or you completely disregard it.

I understand the problem and would like to keep it that way, but then, how am I supposed to serve such files? As they have no extension the library I'm using (koa-static) seems to not know the type of the file and send the wrong headers, which ultimately results in the browsers autodownloading the files instead of displaying them. I understand this is something koa-static should solve and not a general issue right?



LinusU commented on Jan 8, 2017 • edited

Member

Okay, it turns out that serve-static uses [mime](#) which only looks at the file extension... That's unfortunate...

I would probably recommend you to use the [file-type library by sindresorhus](#), it seems great and will actually look at the bytes instead of the client provided file extension.

You can use store the files to a temporary location on upload, then use the file-type library to get a file extension, and then move the file into place.

Since this is a very broadly requested feature, I think that we maybe should add it to v2.x ([#399](#)) in the form of a `detectedMimeType` and `detectedFileExtension`, what do you think?



mezod commented on Jan 8, 2017

Author

I think it would be a great feature, even though I must admit that I barely know how all of these works nor the security risks involved so my opinion shouldn't really matter here :d I am just surprised of the default behaviour when putting multer together with koa-static (serve-static). I'd have expected everyone to have my needs! Thanks for your help, I'll try to do it the way you suggest!



DiegoRBAquero commented on Jan 30, 2017

You can keep original file name and extension uploaded:

```
const multer = require('multer')
const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, 'uploads/')
  },
  filename: function (req, file, cb) {
    cb(null, file.originalname)
  }
})
const upload = multer({storage: storage})
```



18



8



4



1



abhishekdggeek referenced this issue on Mar 15, 2017

No way to enable/disable saving files with extensions. #469

Open



fasterthan commented on Apr 23, 2018

It would be convenient if this original file name snippet was included in the readme



LinusU commented on Apr 24, 2018

Member

It would be convenient if this original file name snippet was included in the readme

I would prefer not to do that since I don't think it's a good practice at all. It would be trivial for anyone to overwrite any file on the server by sending that name up. Also, it would cause problems if a user happens to upload a file with the same name as another file.

@fasterthan would you mind explaining your use case? I could give you a recommendation on how to solve it without the problems I just mentioned...

It would be trivial for anyone to overwrite any file on the server by sending that name up. Also, it would cause problems if a user happens to upload a file with the same name as another file.

```
const crypto = require('crypto')
let upload = multer({
  storage: multer.diskStorage({
    destination: (req, file, cb) => {
      cb(null, path.join(__dirname, '../uploads'))
    },
    filename: (req, file, cb) => {
      let customFileName = crypto.randomBytes(18).toString('hex'),
          fileExtension = file.originalname.split('.')[1] // get file extension from o
      cb(null, customFileName + '.' + fileExtension)
    }
  })
})
```



1



**mizozobu** commented on Oct 6, 2018

You can get the extension of the uploaded file with `path.extname(file.originalname)` as well.



**bikevit2008** commented on Apr 23

Hmm, I actually don't know exactly how koa-static works, but I think that most static servers (apache, nginx) serves the Content-Type based on the magic bytes 🤔

Okay, it turns out that serve-static uses `mime` which only looks at the file extension... That's unfortunate...

I would probably recommend you to use the [file-type library by sindresorhus](#), it seems great and will actually look at the bytes instead of the client provided file extension.

You can use store the files to a temporary location on upload, then use the file-type library to get a file extension, and then move the file into place.

Since this is a very broadly requested feature, I think that we maybe should add it to v2.x ([#399](#)) in the form of a `detectedMimeType` and `detectedFileExtension`, what do you think?

Hi, i think it's not correctly solution, because if i want filtering files on upload and reject files with disallowed extensions, but i can't, when i use `DiskStorage`. Thank you the library `file-type`, it's very useful for me.

I see now only one possible choose for me, but i think it's bad variant: i need use `MemoryStorage` then get Buffer, then check real file-type and save to fs, and after delete buffer from Memory. But i think it's too bad, because i need do identity functional like multer. What do you think about it?



**mathewk2017** commented on Jul 10

if I use `diskStorage`, how can give dynamic destinations?