

fs.writeFile fails without error when writing to a specific (not full) partition

I'm implementing an image cache for favicons found on the web. At one point I have the favicon's URL, and I'm sending it to my page, everything is fine.

Now alongside this, I write the icon to disk, using a pseudo-unique filename that I construct using the URL components.

And this works too, but **only if I write to /tmp/** !?

Here is my code:

```
router.get('/favicon', function(req, res) {
  favicon(req.query.url, function(err, iconUrl) {
    if (iconUrl) {
      var u = Url.parse(iconUrl);
      request.get({url: iconUrl}, function (err, response, body) {
        if (!err && typeof body !== 'undefined') {
          var h = u.host.replace(/\\/g, '\\\\'),
              p = u.path.replace(/\\/g, '\\\\'),
              fileName;

          fileName = path.join('/tmp', h + '.' + p);
          // fileName = path.join(__dirname, '..', 'cache', h + '.' + p);

          fs.writeFile(fileName, body, function(err) {
            if(err) {
              console.log(err);
            } else {
              console.log("SAVED! (%s)", fileName);
            }
          });
        } else {
          console.log('### ERROR iconUrl: (%s) err: (%s)', iconUrl, err);
        }
      });
      res.send(iconUrl);
    } else {
      res.status(500).send('No icon found');
    }
  });
});
```

This is called on the client (web browser) side like this:

```
$.get("/favicon", {
  url: decodeURI(feedHost),
  dataType: "json",
  timeout: 2000
}, function(iconURL) {
  console.info('OK (icon: %s)', iconURL);
}).done(function(iconURL) {
  console.info('OK (icon: %s)', iconURL);
}).fail(function() {
  console.error('ERROR');
});
```

I thought it was a "disk full" problem but no, here is the partition table on this machine, the project directory (where my image cache directory resides) is inside the /home partition, and /tmp/ is, well, on /:

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G   0  1.9G   0% /dev
tmpfs           388M  6.3M  382M   2% /run
/dev/sda6       19G   14G   3.8G  79% /
tmpfs           1.9G  23M   1.9G   2% /dev/shm
tmpfs           5.0M  4.0K   5.0M   1% /run/lock
tmpfs           1.9G   0   1.9G   0% /sys/fs/cgroup
/dev/loop0      82M   82M   0 100% /snap/core/4206
/dev/loop1     128M  128M   0 100% /snap/shotcut/20
/dev/loop2      82M   82M   0 100% /snap/core/4110
/dev/loop3      82M   82M   0 100% /snap/core/4017
/dev/loop4     298M  298M   0 100% /snap/shotcut/17
/dev/loop5     128M  128M   0 100% /snap/shotcut/21
/dev/sda5      121G  102G   14G   89% /home
tmpfs          388M  12K  388M   1% /run/user/1000
```

So when I use the code above, everything works fine, and /tmp/ fills with images, asynchronously, perfect.

As soon as I remove the leading / of the file path - pointing to a valid path, of course - then one, two, maximum 3 favicons are copied and then the process fails somehow, triggering the .fail() method on the client side, but *without any error on the server*.

Help!

Questions:

- What could be causing this weird "write 2 or 3 files then stop" problem? NB: The server is launched with [PM2](#).
- Is my ajax call (jQuery \$.get()) correct? Why does it make the other ajax calls in the page fail, instead of gently erroring and then move on?

node.js express fs

asked 57 mins ago

 **yPhil**
4,056 ● 1 ● 29 ● 53

It looks like that fileName contains some special chars. Can you add console.log(fileName) and see with which file it fails? – [Alex](#) 38 mins ago

Hi Alex: When I do this, I get the names of the 2 or 3 files that I also can see in the cache, and they are all named `subdomain.domain.tld.filename.extension`. But again, no error is triggered anywhere :(Bear in mind that when I use /tmp/ the same files are copied fine. – [yPhil](#) 32 mins ago

Hm, okay, I don't see if you dump error if any... can you add if (err) console.error(err) after function(err, response, body) { ? – [Alex](#) 28 mins ago

I also ran various tests where fileName was just Math.random() but same result: /tmp fine, local repo path fail :(– [yPhil](#) 28 mins ago

Yes, Alex, I am testing body for errors. And no need to test request.get() too, as it is not failing. I mean, if I remove the whole if (!err && typeof body !== 'undefined') everything runs fine. – [yPhil](#) 22 mins ago

