Error: request entity too large

Asked 5 years, 11 months ago Active 17 days ago Viewed 266k times

405

117

I'm receiving the following error with express:

```
Error: request entity too large
    at module.exports
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/node_modules/connec
body/index.js:16:15)
    at json
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/node_modules/connec

    at Object.bodyParser [as handle]
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/node_modules/connec

    at next
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/node_modules/connec

    at Object.cookieParser [as handle]
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/node_modules/connec

    at next
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/node_modules/connec

    at Object.logger
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/node_modules/connec

    at next
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/node_modules/connec

    at Object.staticMiddleware [as handle]
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/node_modules/connec

    at next
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/node_modules/connec

TypeError:
/Users/michaeljames/Documents/Projects/Proj/mean/app/views/includes/foot.jade:31
    29| script(type="text/javascript", src="/js/socketio/connect.js")
    30|
  > 31| if (req.host='localhost')
    32|     //Livereload script rendered
    33|     script(type='text/javascript', src='http://localhost:35729/livereload.js')
    34|

Cannot set property 'host' of undefined
    at eval (eval at <anonymous>
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/jade/lib/jade.js:152:8),
<anonymous>:273:15)
    at
/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/jade/lib/jade.js:153:35
    at Object.exports.render
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/jade/lib/jade.js:197:10)
    at Object.exports.renderFile
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/jade/lib/jade.js:233:18)
    at View.exports.renderFile [as engine]
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/jade/lib/jade.js:218:21)
    at View.render
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/lib/view.js:76:8)
    at Function.app.render
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/lib/application.js:

    at ServerResponse.res.render
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/lib/response.js:801
```

```
    at Object.handle
(/Users/michaeljames/Documents/Projects/Proj/mean/config/express.js:82:29)
    at next
(/Users/michaeljames/Documents/Projects/Proj/mean/node_modules/express/node_modules/connec


POST /api/0.1/people 500 618ms
```

I am using meanstack. I have the following use statements in my express.js

```
//Set Request Size Limit
app.use(express.limit(100000000));
```

Within fiddler I can see the content-length header with a value of: 1078702

I believe this is in octets, this is 1.0787 megabytes.

I have no idea why express is not letting me post the json array I was posting previously in another express project that was not using the mean stack project structure.

javascript node.js http express

edited Dec 13 '16 at 18:40

GG.
13.7k 9 60 107

asked Nov 11 '13 at 22:40

mike james
2,700 3 13 19

3 quick note on this to anyone coming to this question - make sure your issue is actually the node server or body parser. For example I'm using body parser correctly but I got this error because I for to set the max body size in the NGINX conf file. – Stephen Tetreault Dec 11 '17 at 18:14

18 Answers

876

I had the same error recently, and all the solutions I've found did not work.

After some digging, I found that setting `app.use(express.bodyParser({limit: '50mb'}));` did set the limit correctly.

When adding a `console.log('Limit file size: '+limit);` in `node_modules/express/node_modules/connect/lib/middleware/json.js:46` and restarting node, I get this output in the console:

```
Limit file size: 1048576
connect.multipart() will be removed in connect 3.0
visit https://github.com/senchalabs/connect/wiki/Connect-3.0 for alternatives
connect.limit() will be removed in connect 3.0
Limit file size: 52428800
Express server listening on port 3002
```

We can see that at first, when loading the `connect` module, the limit is set to 1mb (1048576 bytes). Then when I set the limit, the `console.log` is called again and this time the limit is 52428800 (50mb). However, I still get a `413 Request entity too large .`

Then I added `console.log('Limit file size: '+limit);` in `node_modules/express/node_modules/connect/node_modules/raw-body/index.js:10` and saw another line in the console when calling the route with a big request (before the error output) :

```
Limit file size: 1048576
```

This means that somehow, somewhere, `connect` resets the limit parameter and ignores what we specified. I tried specifying the `bodyParser` parameters in the route definition individually, but no luck either.

While I did not find any proper way to set it permanently, you can "*patch*" it in the module directly. If you are using Express 3.4.4, add this at line 46 of `node_modules/express/node_modules/connect/lib/middleware/json.js` :

```
limit = 52428800; // for 50mb, this corresponds to the size in bytes
```

The line number might differ if you don't run the same version of Express. Please note that this is **bad practice** and it will be **overwritten** if you update your module.

So this temporary solution works for now, but as soon as a solution is found (or the module fixed, in case it's a module problem) you should update your code accordingly.

I have opened an issue on their GitHub about this problem.

[edit - found the solution]

After some research and testing, I found that when debugging, I added `app.use(express.bodyParser({limit: '50mb'}));` , but **after** `app.use(express.json());` . Express would then set the global limit to 1mb because the first parser he encountered when running the script was `express.json()` . Moving `bodyParser` above it did the trick.

That said, the `bodyParser()` method will be deprecated in Connect 3.0 and should not be used. Instead, you should declare your parsers explicitly, like so :

```
app.use(express.json({limit: '50mb'}));
app.use(express.urlencoded({limit: '50mb'}));
```

In case you need multipart (for file uploads) see this post.

Second edit

Note that in Express 4, instead of `express.json()` and `express.urlencoded()` , you must require the body-parser module and use its `json()` and `urlencoded()` methods, like so :

```
var bodyParser = require('body-parser');
app.use(bodyParser.json({limit: '50mb'}));
app.use(bodyParser.urlencoded({limit: '50mb', extended: true}));
```

If the `extended` option is not explicitly defined for `bodyParser.urlencoded()` , it will throw a warning (`body-parser deprecated undefined extended: provide extended option`). This is because this option will be required in the next version and will **not** be optional anymore. For more info on the `extended` option, please refer to the readme of `body-parser` .

edited May 9 at 13:25 — L. F.

answered Nov 13 '13 at 21:48 — Samuel Bolduc

 **9,592** 5 23 49  **10.9k** 4 26 53

7 My problem was that I had express.json() parser above my bodyParser... After understanding how it all works I removed bodyParser and set de limit in the other, like `app.use(express.json({limit:'50mb'}));` . I edited my answer to reflect this! – Samuel Bolduc Nov 14 '13 at 2:22

2 for express 4, the indicated code throws "body-parser deprecated undefined extended: provide extended option" on the `urlencoded` call – Mike 'Pomax' Kamermans Oct 4 '14 at 19:17

3 Thanks for updating this for express 4. You are a life saver! – Swills Nov 4 '15 at 6:00

30 This is as comprehensive as this answer gets, so there is one more point for future reference that I ran into if you are using nginx as a reverse proxy in front of your node.js/express instance, which is recommended practice as well. Nginx will throw the same `413::Request Entity is too large` exception. It wont forward the request to your express app. So we need to set `client_max_body_size 50M;` in the nginx config OR a specific server config OR even a specific location tag will work. – Aukhan Feb 16 '16 at 17:49

3 Thank you samuel for this ! Saved me from a world of headache, Cheers and +1 for the comprehensive answer! – BastianBuhrkall May 20 '16 at 9:29

92

I use Express 4.

In my case it was not enough to add these lines :

```
var bodyParser = require('body-parser');
app.use(bodyParser.json({limit: '50mb'}));
app.use(bodyParser.urlencoded({limit: '50mb', extended: true}));
```

I tried adding the parameterLimit option on urlencoded function as the documentation says and error no longer appears.

> The parameterLimit option controls the maximum number of parameters that are allowed in the URL-encoded data. If a request contains more parameters than this value, a 413 will be returned to the client. Defaults to 1000.

Try with this code:

```
var bodyParser = require('body-parser');
app.use(bodyParser.json({limit: "50mb"}));
app.use(bodyParser.urlencoded({limit: "50mb", extended: true, parameterLimit:50000}));
```

answered Apr 9 '16 at 8:05

 slorenzo
2,181 23 41

The parameterLimit was another gotcha I was missing, thanks for pointing this out. – crowebird Apr 19 '16 at 15:49

github.com/expressjs/body-parser#parameterlimit – jwerre Mar 2 '17 at 21:34

1 Adding the "parameterLimit:50000" solves the problem. – Andrien Pecson Aug 8 '17 at 7:32

1 I was struggling so hard on this until I found this answer. THANK YOU! #Relieve – Stephan Celis Aug 26 '18 at 7:30

This is exactly what I was looking for! – ApplePie Sep 14 '18 at 19:52

|

40

If someone tried all the answers, but hadn't had any success yet and uses NGINX to host the site add this line to /etc/nginx/sites-available

```
client_max_body_size 100M; #100mb
```

edited Feb 1 '18 at 22:04
bugwheels94
22.8k 3 30 54

answered Nov 22 '16 at 15:20

Alexander
856 1 11 21

3 This was my problem, thanks! One clue is that nginx's default is 1MB, so if you seem to be limited to that amount then it's probably nginx. Also, another clue: `body-parser` will return a `limit` property and a `length` property in the error object (along with `status:413`). Nginx doesn't do this. So if you can't see those properties in the error object, it's probably a nginx limit. Here's how to change this nginx setting (config files most likely in `/etc/nginx/sites-available/`) – user993683 Jun 2 '17 at 16:38

2 thank you so much for this answer i believe this answer should be upvoted because it is important , personally i forgot about nginx configuration ... again big thank you – Fadi Abo Msalam Mar 12 '18 at 20:21

How to set limit only in express? – alex dykyi Oct 31 '18 at 14:32

@alexdykyi Answer from Vivek22 : app.use(bodyParser({limit: '50mb'})); – Alexander Nov 1 '18 at 23:20

28

I don't think this is the express global size limit, but specifically the connect.json middleware limit. This is 100kb by default when you use `express.bodyParser()` and don't provide a `limit` option.

Try:

```
app.post('/api/0.1/people', express.bodyParser({limit: '5mb'}), yourHandler);
```

edited Aug 30 '16 at 17:04

answered Nov 11 '13 at 23:47

Peter Lyons
117k 22 236 249

1 Hi - I am using the following: app.get('/api/0.1/people/', express.bodyParser({limit:'100mb'}), tracks.all); I am still receiving the same error... – mike james Nov 12 '13 at 20:50

1 I have the following code: codepen.io/mike-jam-es/pen/hHbwJ – mike james Nov 13 '13 at 20:49

1 this worked for me. thank you. – chovy Jun 26 '15 at 19:21

the default limit is 100kbyte as it seems now github.com/expressjs/body-parser#limit – Qiong Wu Aug 30 '16 at 16:14

12

in my case .. setting `parameterLimit:50000` fixed the problem

```
app.use( bodyParser.json({limit: '50mb'}) );
app.use(bodyParser.urlencoded({
  limit: '50mb',
  extended: true,
  parameterLimit:50000
}));
```

edited May 16 '16 at 12:51

FredMaggiowski
1,810 2 16 36

answered May 16 '16 at 12:11

Mohanad Obaid
121 1 3

This worked for me. "Request entity too large" seems to refer to large strings (like JSON) too. I was posting ~20Kib JSON and had this problem (notice that default `body-parser` payload **size** limit is OK for me). Was solved with `parameterLimit` only (no need to set any `limit` s). – aesede Jul 20 '16 at 16:01

10

2016, none of the above worked for me until i explicity set the 'type' in addition to the 'limit' for bodyparser, example:

```
  var app = express();
  var jsonParser       = bodyParser.json({limit:1024*1024*20, type:'application/json'});
  var urlencodedParser = bodyParser.urlencoded({
extended:true,limit:1024*1024*20,type:'application/x-www-form-urlencoded' })

  app.use(jsonParser);
  app.use(urlencodedParser);
```

edited Oct 3 at 18:28

Morfinismo
3,471 3 9 26

answered May 6 '16 at 20:42

user1709076
1,027 3 21 36

Yes, this is what I needed to do also. And it has been set up so you can drive the config settings from your app script. – Robb Sadler Jun 30 '16 at 13:35

2 did you mean `application/x-www-form-urlencoded` (rather than `application/x-www-form-urlencoding`)? – grenade Jun 15 '17 at 8:09

8

The following worked for me... Just use

```
app.use(bodyParser({limit: '50mb'}));
```

that's it.

Tried all above and none worked. Found that even though we use like the following,

```
app.use(bodyParser());
app.use(bodyParser({limit: '50mb'}));
app.use(bodyParser.urlencoded({limit: '50mb'}));
```

only the 1st `app.use(bodyParser());` one gets defined and the latter two lines were ignored.

Refer: https://github.com/expressjs/body-parser/issues/176 >> see 'dougwilson commented on Jun 17, 2016'

edited Mar 3 '17 at 20:18
Tunaki
98.3k 24 228 312

answered Mar 3 '17 at 20:15
Vivek22
399 5 14

1 Only this worked for me as well. Thanks! – seemvision May 11 '18 at 16:51

1 THANK YOU! I also had an old app.use(bodyParser.json()); in my code before and indeed, only the first one is taken! – Nico Nov 24 '18 at 1:01

5

For express ~4.16.0, express.json with limit works directly

```
app.use(express.json({limit: '50mb'}));
```

answered Jan 14 at 8:42
Stenal P Jolly
344 3 12

4

After тo many tries I got my solution

I have commented this line

```
app.use(bodyParser.json());
```

and I put

```
app.use(bodyParser.json({limit: '50mb'}))
```

Then it works

edited Feb 10 at 8:51
Boaz
15.8k 7 48 58

answered Sep 18 '18 at 4:17
Maulik Patel
324 1 3 19

4

Little old post but I had the same problem

Using express 4.+ my code looks like this and it works great after two days of extensive testing.

```
var url         = require('url'),
    homePath    = __dirname + '/../',
    apiV1       = require(homePath + 'api/v1/start'),
    bodyParser  = require('body-parser').json({limit:'100mb'});

module.exports = function(app){
    app.get('/', function (req, res) {
        res.render( homePath + 'public/template/index');
    });

    app.get('/api/v1/', function (req, res) {
        var query = url.parse(req.url).query;
        if ( !query ) {
            res.redirect('/');
        }
        apiV1( 'GET', query, function (response) {
            res.json(response);
        });
    });

    app.get('*', function (req,res) {
        res.redirect('/');
    });

    app.post('/api/v1/', bodyParser, function (req, res) {
        if ( !req.body ) {
            res.json({
                status: 'error',
                response: 'No data to parse'
            });
        }
        apiV1( 'POST', req.body, function (response) {
            res.json(response);
        });
    });
};
```

answered Jan 11 '17 at 7:48

Alexander Sasha Shcherbakov

103 8

4

In my case the problem was on **Nginx configuration**. To solve it I have to edit the file: `/etc/nginx/nginx.conf` and add this line inside server block:

```
client_max_body_size 5M;
```

Restart Nginx and the problems its gone

```
sudo systemctl restart nginx
```

answered Sep 18 '18 at 8:33

Jaime Fernandez

810 8 7

I've used another practice for this problem with multer dependancie.

3

Example:

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

var uploading = multer({
  limits: {fileSize: 1000000, files:1},
});

exports.uploadpictureone = function(req, res) {
  cloudinary.uploader.upload(req.body.url, function(result) {
    res.send(result);
  });
};

module.exports = function(app) {
    app.route('/api/upload', uploading).all(uploadPolicy.isAllowed)
        .post(upload.uploadpictureone);
};
```

edited Mar 17 '16 at 18:28

answered Mar 16 '16 at 10:44

Quentin Malguy

154 1 5

3

A slightly different approach - the payload is *too BIG*

All the helpful answers so far deal with increasing the payload limit. But it might also be the case that the payload is indeed too big but for no good reason. If there's no valid reason for it to be, consider looking into why it's so bloated in the first place.

Our own experience

For example, in our case, an Angular app was greedily sending an entire object in the payload. When one bloated and redundant property was removed, *the payload size was reduced by a factor of a 100*. This significantly improved performance and resolved the 413 error.

edited Jul 7 '18 at 13:21

answered Jun 10 '18 at 9:08

Boaz

15.8k 7 48 58

2

for me following snippet solved the problem.

```
var bodyParser = require('body-parser');
app.use(bodyParser.json({limit: '50mb'}));
```

edited Jul 25 at 5:36

WasiF

5,118 1 42 58

answered Jun 20 at 17:21

Sanjeeva Kumar Acham

21 3

The better use you can specify the limit of your file size as it is shown in the given lines:

1

```
app.use(bodyParser.json({limit: '10mb', extended: true}))
app.use(bodyParser.urlencoded({limit: '10mb', extended: true}))
```

You can also change the default setting in node-modules body-parser then in the lib folder, there are JSON and text file. Then change limit here. Actually, this condition pass if you don't pass the limit parameter in the given line app.use(bodyParser.json({limit: '10mb', extended: true})).

edited Jul 9 at 18:51
Marco
161 10

answered May 24 at 14:36

Ravi Beniwal
11 2

0

I too faced that issue, I was making a silly mistake by repeating the `app.use(bodyParser.json())` like below:

```
app.use(bodyParser.json())
app.use(bodyParser.json({ limit: '50mb' }))
```

by removing `app.use(bodyParser.json())` , solved the problem.

answered Aug 2 at 6:03

WasiF
5,118 1 42 58

0

For me the main trick is

```
app.use(bodyParser.json({
  limit: '20mb'
}));

app.use(bodyParser.urlencoded({
  limit: '20mb',
  parameterLimit: 100000,
  extended: true
}));
```

bodyParse.json first bodyParse.urlencoded second

answered Feb 18 at 17:58

Nicollas Matheus
124 7

0

In my case removing `Content-type` from the request headers worked.

answered Sep 17 at 4:56
carkod
484 5 19