August 09, 2018 · 3 mins, 441 words

Validate Max File Size in Laravel, PHP and Web-Server

Tutorial last revisioned on August 11, 2022 with Laravel 9

I got this question at least a dozen times: "I'm raising my file validation rule in Laravel to 20 MB but still get errors, it doesn't work". So decided to expand on it and explain the reason, and what to do. First, let's see how to validate file size in Laravel. In your Form Request files or **validate()** method you can pass array with this parameter:

```
[
  'image' => 'required|mimes:jpeg,bmp,png|size:20000',
]
```

The last part means that size should be not more than 20 MB (20000 kB). But that may be not enough, cause file restrictions exist not only on Laravel application level.

Update

Since Laravel 9.22 there's new fluent file validation rule. Above rule can now be written like this

```
'image' => ['required', File::image()->smallerThan(20000)
```

You can check pull request for this change here and read updated official documantation here.

PHP settings in php.ini

There are two settings related to max size in **php.ini** file. Here they are with their default values:

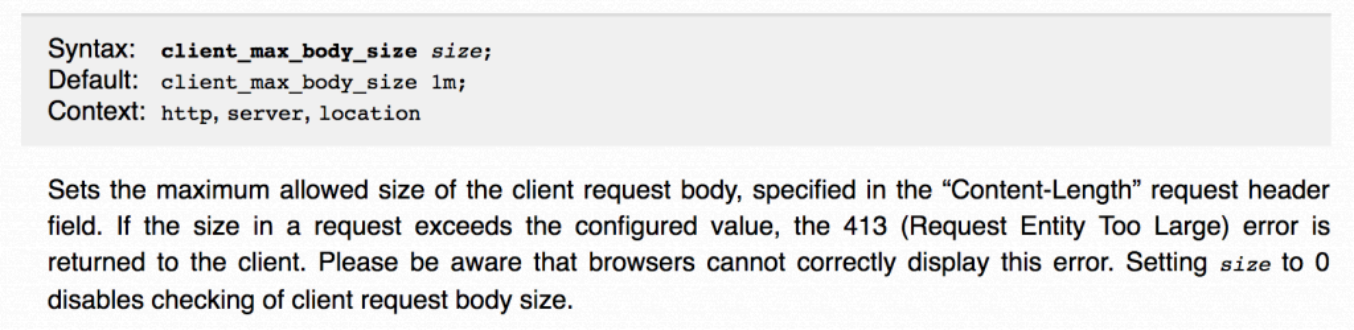
```
upload_max_filesize = 2M
post_max_size = 8M
```

As you can see, by default you can upload files only up to 2 MB, so you should change that to 20M. But even that won't allow you to upload 20 MB files, because of overall POST request restriction by **post_max_size**, it should be set to 20M, or rather even bigger than that, cause POST will likely to have more data than just the file, so I would set it to 21M at least,

in that case. Also, please make sure you're editing the correct **php.ini** file, cause there are cases with multiple files on the same server, for different PHP versions, also for FPM and CLI settings. But even this may be not enough. There's also web-server configuration.

Nginx and Apache Settings

By default Laravel Forge creates servers with LEMP stack, which uses Nginx as a web-server. Even without knowing how to configure it, you need to care about one setting in **nginx.conf** file: **client_max_body_size**. Here's a screenshot from official documentation:

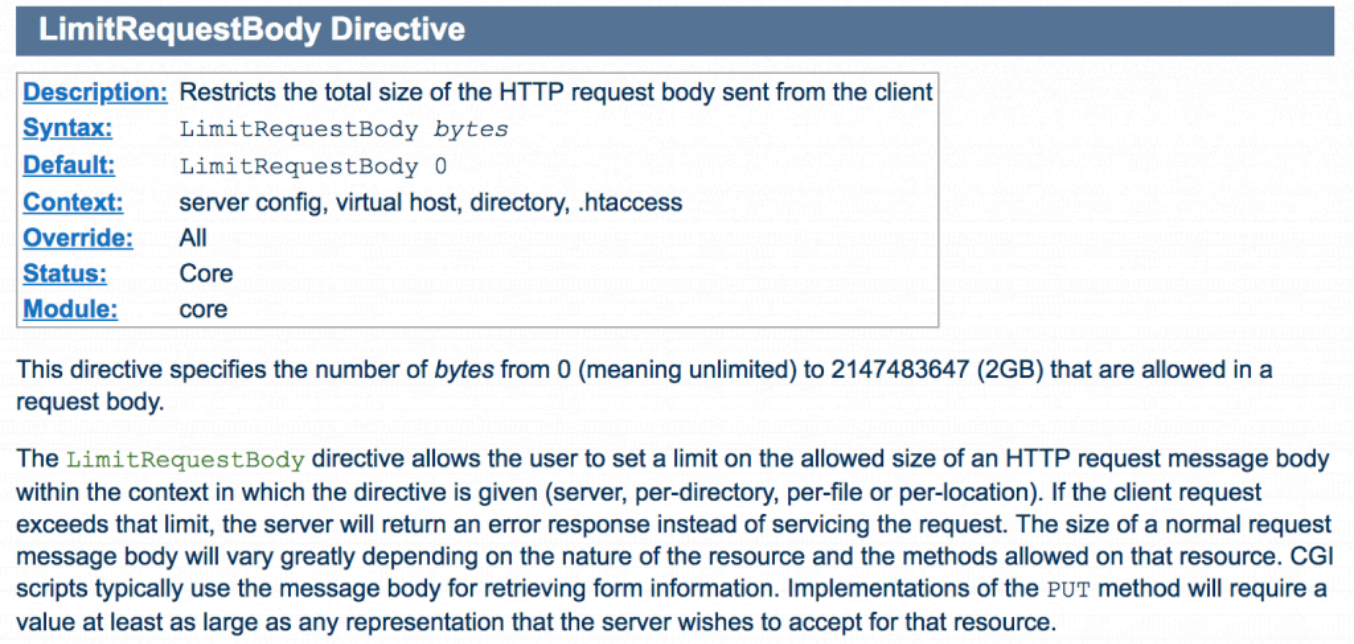


Syntax: `client_max_body_size size;`
Default: `client_max_body_size 1m;`
Context: `http, server, location`

Sets the maximum allowed size of the client request body, specified in the "Content-Length" request header field. If the size in a request exceeds the configured value, the 413 (Request Entity Too Large) error is returned to the client. Please be aware that browsers cannot correctly display this error. Setting *size* to 0 disables checking of client request body size.

As you can see, default value is only **1m**, which means that your whole POST request may be maximum 1MB. So you need to change that setting to 20m or higher.

If you're using Apache web-server, there's also a setting for that called **LimitRequestBody**. Here's a screenshot from the docs:



LimitRequestBody Directive

Description:	Restricts the total size of the HTTP request body sent from the client
Syntax:	`LimitRequestBody bytes`
Default:	`LimitRequestBody 0`
Context:	server config, virtual host, directory, .htaccess
Override:	All
Status:	Core
Module:	core

This directive specifies the number of *bytes* from 0 (meaning unlimited) to 2147483647 (2GB) that are allowed in a request body.

The `LimitRequestBody` directive allows the user to set a limit on the allowed size of an HTTP request message body within the context in which the directive is given (server, per-directory, per-file or per-location). If the client request exceeds that limit, the server will return an error response instead of servicing the request. The size of a normal request message body will vary greatly depending on the nature of the resource and the methods allowed on that resource. CGI scripts typically use the message body for retrieving form information. Implementations of the `PUT` method will require a value at least as large as any representation that the server wishes to accept for that resource.

The difference here is that Apache doesn't give any restrictions by default. So there's a chance that you will never need to edit this value, but just in case, it's good to know it exists.

So these are my tips on how to upload bigger files in Laravel. Anything I've missed?

Recent Premium Tutorials

November 17, 2022 · 12 mins, 2275 words · ⭐ PREMIUM

CRUDdy By Design by Adam Wathan: Summary, Examples, Opinions

January 05, 2023 · 11 mins, 2004 words · ⭐ PREMIUM

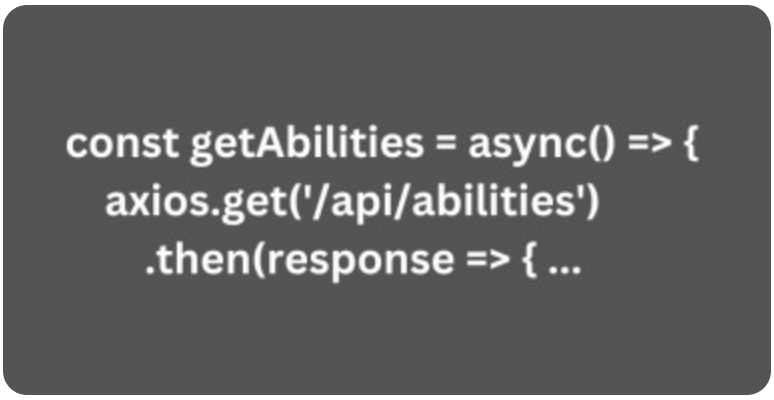
Laravel Testing: Mocking/Faking External 3rd Party APIs

March 21, 2023 · 18 mins, 3510 words · ⭐ PREMIUM

Laravel Api Auth with Vue and Sanctum: All You Need To Know

May 11, 2023 · 22 mins, 4342 words · ⭐ PREMIUM

Laravel Spatie Media Library: 8 Less-Known Features with Demos



May 16, 2023 · 9 mins, 1658 words · ⭐ PREMIUM

Laravel Vue SPA: Roles and Permissions Example with CASL

July 25, 2023 · 17 mins, 3239 words · ⭐ PREMIUM

Laravel Best E-commerce Shop Packages: Comparison Review