



Por: Carlos Ferreira

09/10/2017

Comentar

Obter Novidades

## Upload de arquivos no laravel com Request



### Compartilhe:

Em uma aplicação é muito comum surgir a necessidade de fazer upload de arquivos, e neste tutorial vou ensinar como fazer uploads de arquivos no laravel utilizando os recursos da classe Request, não iremos abordar os recursos File Storage.

Primeiro passo é entender as configurações de upload no arquivo  
`/config/filesystems.php`

Neste arquivo a primeira configuração é “default” que define qual “disks” de configuração de uploads que a aplicação vai manter por default. Por default é local:

```
1 | 'default' => env('FILESYSTEM_DRIVER', 'local'),
```

Ao deixar os dados assim com o padrão de upload como “local” no array “disks” vai pegar as configurações do array “local”:

```
1 | 'local' => [  
2 |     'driver' => 'local',  
3 |     'root' => storage_path('app'),  
4 | ],
```

**drive:** Como será feito o upload, se deixar “local” os arquivos ficaram armazenados na própria aplicação, também é possível configurar transferência automática via FTP ou até mesmo enviar para a Amazon com s3. Os valores suportados são: “local”, “ftp”, “s3” e “rackspace”.

**root:** É o diretório onde ficará armazenado os arquivos no momento que fizer upload. O helper `storage_path()` aponta para o diretório `storage/`, e como está assim `storage_path('app')` define que o diretório de upload será para `storage/app/`

Quando for fazer o upload de algum arquivo restrito, ou até mesmo que vai disponibilizar o download através de response ou exibir apenas o item, pode deixar essa configuração. Porém se for fazer upload de imagens/arquivos que precisarão ser exibidos posteriormente em uma view por exemplo, precisa utilizar as configurações de “public”, então no “default” configure como “public” veja:

```
1 | 'default' => env('FILESYSTEM_DRIVER', 'public'),
```

Ao fazer isso vai utilizar essas configurações de “disks”:

```
1 | 'public' => [  
2 |     'driver' => 'local',  
3 |     'root' => storage_path('app/public'),  
4 |     'url' => env('APP_URL').'/storage',  
5 |     'visibility' => 'public',  
6 | ],
```

Com essas configurações os uploads ficaram em `storage/app/public`

## Qual a vantagem?

Se for exibir as imagens em uma listagem na view vai ser mais simples de exibir. Não é possível exibir uma imagem que está em `storage/` em uma listagem na view, só é possível exibir itens na view que estão em `/public` (que são o conteúdo público de uma aplicação web (imagens, fontes, css, js)).

## Como resolver?

É possível através do próprio artisan criar um link simbólico de `storage/app/public` para dentro `public/storage` rode este comando:

```
1 | php artisan storage:link
```

Esse comando cria um link simbólico em `public/storage` que aponta para `storage/app/public`, ou seja, agora todos os arquivos disponíveis neste diretório estarão disponível publicamente na aplicação e agora será possível acessá-los facilmente.

**NOTA.** Antes de tentar fazer o upload é importante que o formulário que será utilizado para o envio dos arquivos (upload) contenha o atributo `enctype="multipart/form-data"`, exemplo:

```
1 | <form action="url-aqui" method="post" enctype="multipart/form-data">
2 |     <input type="file" name="image">
3 |     ...
4 | </form>
```

## Recuperar o arquivo de upload:

Neste exemplo nosso campo de imagem tem o nome "image", e para recuperar pode utilizar estas duas formas:

```
1 | $file = $request->file('image');
2 | // ou
3 | $file = $request->image;
```

### Verificações:

É importante antes de fazer o upload verificar se informou o arquivo e se é um arquivo válido:

```
1 | // Se informou o arquivo, retorna um boolean
2 | $file = $request->hasFile('image')
3 |
4 | // Se é válido, retorna um boolean
5 | $file = $request->file('image')->isValid()
```

### Recuperar algumas informações do arquivo:

É possível recuperar diversas informações a respeito do arquivo antes de fazer o upload, veja algumas das opções:

```
1 | // Retorna mime type do arquivo (Exemplo image/png)
2 | $request->imagem->getMimeType()
3 |
4 | // Retorna o nome original do arquivo
5 | $request->imagem->getClientOriginalName()
6 |
7 | // Extensão do arquivo
8 | $request->imagem->getClientOriginalExtension()
9 | $request->imagem->extension()
10 |
11 | // Tamanho do arquivo
12 | $request->imagem->getClientSize()
```

Mais detalhes na documentação oficial

(<http://api.symfony.com/3.0/Symfony/Component/HttpFoundation/File/UploadedFile.html>)

## Fazendo upload:

É bem simples o upload de arquivos no laravel, basta fazer assim:

```
1 | $upload = $request->image->store('products');
```

O arquivo será salvo em storage/app/public/products/ com um nome aleatório

É possível informar também o “driver” de upload no momento de salvar:

```
1 | $upload = $request->image->store('products', 's3');
```

Neste exemplo o upload vai pegar as configurações de “s3” e fazer o upload para a Amazon.

Se quiser salvar o arquivo com um nome especial, é possível utilizar o método storeAs(), veja como:

```
1 | $upload = $request->image->storeAs('products', 'novonome.jpg');  
2 | // Neste caso será salvo em storage/app/public/products/novonome.jpg
```

Também é possível informar dinamicamente qual é o driver de armazenamento, veja o exemplo:

```
1 | $upload = $request->image->storeAs('products', 'novonome.jpg', 's3');
```

## Um exemplo real:

Para finalizar vamos fazer um exemplo real, onde vamos fazer o upload do arquivo para storage/app/public/categories/ com nomes personalizados criados pela nossa própria aplicação, veja:

```

1  /**
2   * Store a newly created resource in storage.
3   *
4   * @param \Illuminate\Http\Request $request
5   * @return \Illuminate\Http\Response
6   */
7  public function store(Request $request)
8  {
9      // Define o valor default para a variável que contém o nome da imagem
10     $nameFile = null;
11
12     // Verifica se informou o arquivo e se é válido
13     if ($request->hasFile('image') && $request->file('image')->isValid()) {
14
15         // Define um aleatório para o arquivo baseado no timestamps atual
16         $name = uniqid(date('HisYmd'));
17
18         // Recupera a extensão do arquivo
19         $extension = $request->image->extension();
20
21         // Define finalmente o nome
22         $nameFile = "{$name}.{$extension}";
23
24         // Faz o upload:
25         $upload = $request->image->storeAs('categories', $nameFile);
26         // Se tiver funcionado o arquivo foi armazenado em storage/app/public/catego
27
28         // Verifica se NÃO deu certo o upload (Redireciona de volta)
29         if ( !$upload )
30             return redirect()
31                 ->back()
32                 ->with('error', 'Falha ao fazer upload')
33                 ->withInput();
34
35     }
36 }

```

## Como exibir?

Como criamos um link simbólico de public/storage que aponta para storage/app/public é possível exibir facilmente na view o arquivo de imagem, veja um exemplo de uma listagem dinâmica de categorias:

```

1  @foreach( $categories as $category )
2      {{ $category->name }}
3
4      name }}"
5  @endforeach

```

## Compartilhe:



(<https://www.especializati.com.br/curso-laravel-55>)

## Gostou deste conteúdo?

Junte-se a milhares de profissionais de ti inteligentes e seja o primeiro a receber as nossas novidades e dicas!



Nome:

E-mail:

Receber

Obter Novidades



Sobre o Autor:

### Carlos Ferreira

Carlos Ferreira é Analista de Sistemas Experiente, Empreendedor, Fundador da empresa EspecializaTi.

Certificações: Comptia Linux +, LPI, Novell Certification.

16 COMENTÁRIOS

especializati

**1** Iniciar sessão ▾

♥ Recomendar 2

🐦 Tweet

f Partilhar

Mostrar primeiro os mais votados ▾



Escreva o seu comentário...

INICIE SESSÃO COM O

OU REGISTE-SE NO DISQUS ?

Nome



**Rodrigo Carvalho** • há um ano

Carlos, não estou conseguindo exibir as imagens o que pode ser? Minha versão é 5.6

33 ^ | ▾ • Responder • Partilhar ▸

**Mauro Lucio** • há 10 meses

Como mostrar uma imagem com caminho e nome salvos em banco?

2 ^ | v • Responder • Partilhar ›

**Paul Sidewalk** • há 2 anos

Carlos, voce, como sempre, com uma didática diferenciada. Obrigado. Paulo Calçada.

1 ^ | v • Responder • Partilhar ›

**anonimo** • há um mês

Artigo legal,  
mas tem uns erros de português  
<http://prntscr.com/on7yn5>

^ | v • Responder • Partilhar ›

**Gilto Barcella** • há 4 meses

Não estou conseguindo exibir a imagem na view, alguém pode me ajudar?

^ | v • Responder • Partilhar ›

**Gilto Barcella** • há 4 meses

Muito bom, me salvou!

^ | v • Responder • Partilhar ›

**Alexandre Possebon** • há 4 meses

Assim... ok, subiu corretmente e tb fiz update(Alem de salvar no diretorio estou colocando em  
MySQL)... mas, as imagens vao se acumulando no diretorio.  
Quero dizer, como apagar uma imagem após subir esta imagem?

^ | v • Responder • Partilhar ›

**Aline Chaves** • há 7 meses

Muito bom, me ajudou bastante, obrigada por compartilhar conhecimento

^ | v • Responder • Partilhar ›

**Marcelo Barth** • há 9 meses

Fantástico Carlos. Sou seu fã! Obrigado por mais essa ajuda!

^ | v • Responder • Partilhar ›

**Allan Proença Garcia** • há 10 meses

Não estou conseguindo exibir os arquivos, no meu caso estou querendo exibir videos..

^ | v • Responder • Partilhar ›

**Evecimar Silva** • há um ano

Exelente artigo. Parabéns

^ | v • Responder • Partilhar ›

**Elinardo Silva** • há 2 anos

Não to conseguindo fazer vizualizar a view as imagens. Dessa forma ai explicada eu quebro  
meu HTML, ou seja as classes da msm DIV n funcionam...

^ | v • Responder • Partilhar ›

**Rodrigo Costa** • há 2 anos

muito bom, era o que eu estava procurando.

^ | v • Responder • Partilhar ›

**Vitor Micillo Junior** • há 2 anos

Muito bom seu blog Professor, parabéns pelo ótimo trabalho que tem feito.  
E se eu quiser postar varias imagens ao mesmo tempo, como proceder ?

^ | v • Responder • Partilhar ›

**Mateus Cardoso** • há 2 anos

Bom tutorial , mas ainda estou com dificuldades, entre os seus cursos existe algum onde vc  
ensine a fazer o upload porém sendo enviado de um app e este como api?

^ | v • Responder • Partilhar ›



Responder • Partilhar ›



**Luiz Humberto B. Patrocínio** • há 2 anos

Bacana hein, agora mostre como excluir e como fazer upload de várias imagens.No mais tudo de bom no post.

Responder • Partilhar ›

TAMBÉM NO ESPECIALIZATI

Login Único no Laravel | EspecializaTi

3 COMENTÁRIOS • há 2 anos



**Paulo Roberto** — Como eu faço para implementar um log de acesso ?

Envio de E-mails no Laravel

6 COMENTÁRIOS • há um ano



**Ruan Nawê** — Gostei, excelente tutorial

Queues no Laravel (Filas)

3 COMENTÁRIOS • há um ano



**Elinardo Silva** — massa demais carlos! Suas aulas e tutoriais são disparadas melhor do que o documentação hehe. aproveitar pra tirar uma

Dois sentimentos que TODO desenvolver deve sentir

2 COMENTÁRIOS • há um ano



**Alan Victor Calsolari Bariani** — Eu me sinto diferente... eu sinto orgulho... porra.. a 5 anos eu sabia tão pouco... e consegui fazer isso

Subscrever • Acerca do DisqusAdicionar o DisqusAdicionar

Disqus Privacy Policy Política de privacidade

Obter Novidades

CONTEÚDOS RELACIONADO:

LARAVEL: REDIRECIONAR APÓS A AUTENTICAÇÃO

POR: CARLOS FERREIRA

([HTTPS://BLOG.ESPECIALIZATI.COM.BR/LARAVEL-REDIRECIONAR-APOS-AUTENTICACAO/](https://blog.especializati.com.br/laravel-redirecionar-apos-autenticacao/))



([HTTPS://BLOG.ESPECIALIZATI.COM.BR/LARAVEL-BLADE/](https://blog.especializati.com.br/laravel-blade/))



([HTTPS://BLOG.ESPECIALIZATI.COM.BR/ENVIO-DE-E-MAILS-NO-LARAVEL/](https://blog.especializati.com.br/envio-de-e-mails-no-laravel/))

# especializaTi | BLOG

Todos os direitos reservados © 2019 - EspecializaTi. É proibida a reprodução total ou parcial deste conteúdo.