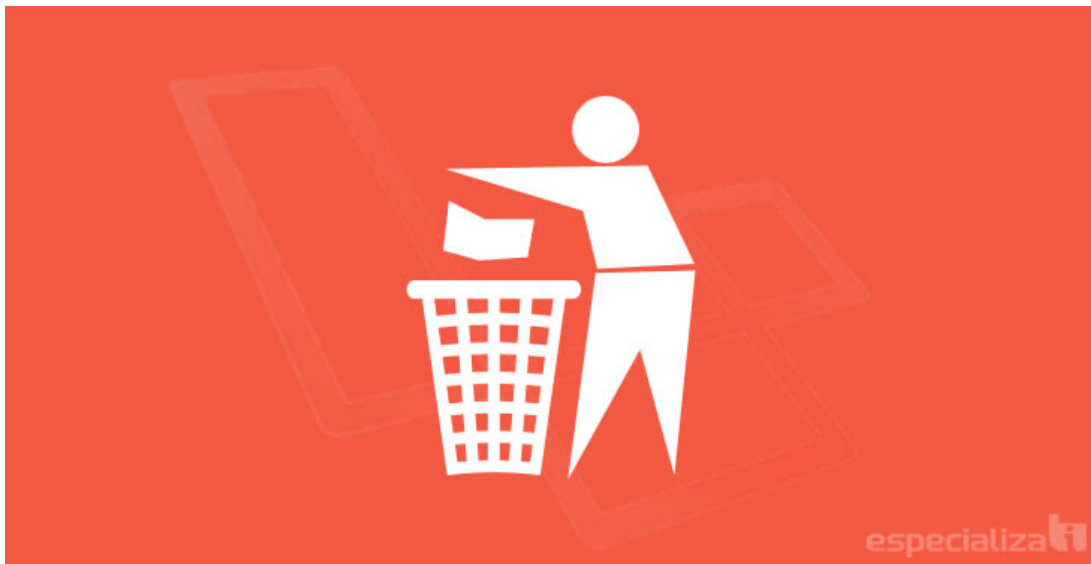




Por: Carlos Ferreira 📅 05/01/2018 💬 Comentar

Aprenda como Utilizar o Recurso de Soft Deleting do Laravel



Compartilhe:  (<https://www.facebook.com/sharer/sharer.php?u=https://blog.especializati.com.br/aprenda-como-utilizar-o-recurso-de-soft-deleting-do-laravel/>)  (<https://twitter.com/intent/tweet?text=https://blog.especializati.com.br/aprenda-como-utilizar-o-recurso-de-soft-deleting-do-laravel/>)  (<https://plus.google.com/share?url=https://blog.especializati.com.br/aprenda-como-utilizar-o-recurso-de-soft-deleting-do-laravel/>)

Outro recurso interessante do Laravel e que se mostra muito útil em diversos cenários é o Soft Deleting, este recurso permite “deletar” o registro sem excluir.

Como assim?

É possível criar um tabela com uma coluna “deleted_at”, essa coluna é do tipo date.

O Eloquent do Laravel por default (padrão) seleciona apenas os registros que não tem o campo “deleted_at” preenchidos.

Acontece o seguinte, quando está utilizando o recurso de Soft Deleting ao deletar o registro o mesmo não é apagado da tabela, apenas o campo deleted_at é preenchido com as informações de data. E quando consulta o registro, por

default só retorna os registros com a coluna `deleted_at` está igual a null, ou seja, registros que não foram “deletados”.

Isso é interessante quando se tem informações valiosas no banco de dados, porque não é interessante neste caso que estes registros sejam apagados, como por exemplo, registros de usuários. Não é interessante excluir definitivamente um registro de usuário no sistema.

Para criar a coluna `deleted_at` na tabela, no arquivo de migration basta adicionar isso:

```
1 | Schema::table('posts', function ($table) {  
2 |     $table->softDeletes();  
3 | });
```

O **`$table->softDeletes();`** vai criar a coluna **`deleted_at`**

Utilizando o Recurso de Soft Deleting:

Para utilizar é muito simples, basta utilizar a Trait **`SoftDeletes`** no Model correspondente, apenas isso. Veja um exemplo prático:

```
1 | namespace App\Models;  
2 |  
3 | use Illuminate\Database\Eloquent\Model;  
4 | use Illuminate\Database\Eloquent\SoftDeletes;  
5 |  
6 | class Post extends Model  
7 | {  
8 |     use SoftDeletes;  
9 |  
10 |    /**  
11 |     * Opcional, informar a coluna deleted_at como um Mutator de data  
12 |     *  
13 |     * @var array  
14 |     */  
15 |    protected $dates = ['deleted_at'];  
16 | }
```

Agora ao selecionar os registros utilizando o Model (Eloquent) só vai trazer os registros que não foram apagados, ou seja, os registros que a coluna `deleted_at` é null, faça o teste:

```
1 | #controller  
2 | public function index(Post $post)  
3 | {  
4 |     $posts = $post->get();  
5 |  
6 |     return view('path.name-view', compact('posts'));  
7 | }
```

Para ver a query final, faça isso e irá notar que tem um where adicional em `deleted_at`, verificando se é diferente a null:

```
1 | $posts = $post->toSql();  
2 | dd($posts);
```

Por default não vai trazer os registros deletados, mas, é possível forçar isso, de forma que não traga estes registros nem mesmo em relacionamento de tabelas, veja os exemplos:

```
1 // Exibindo os posts que não foram deletados do usuário logado
2 $posts = $post->where('user_id', auth()->user()->id)->withTrashed()->get();
3
4 // Registros que não foram deletados, mesmo com um relacionamento:
5 $posts = $post->comments()->withTrashed()->get();
```

Como exibir apenas os registros “deletados” com Soft Deleted?

É bem simples fazer isso, veja com o exemplo prático:

```
1 public function onlyTrashed(Post $post)
2 {
3     $postsTrashed = $post->onlyTrashed()->get();
4
5     return view('path.name-view', compact('postsTrashed'));
6 }
```

Restaurando um Registro deletado com Soft Deleted:

Note através do exemplo prático, como é simples fazer isso:

```
1 public function restore(Post $post, $id)
2 {
3     // Recupera o post pelo ID
4     $post = $post->find($id);
5
6     // Restaura:
7     $post->restore();
8
9     // Também é possível fazer assim, em uma única linha:
10    // $post = $post->find($id)->restore();
11 }
```

Outro método de restaurar um ou mais registros:

```
1 $post->withTrashed()
2     ->where('user_id', auth()->user()->id)
3     ->restore();
```


Como remover o registro permanentemente?

Em alguns casos pode que seja necessário remover definitivamente o registro, não só de maneira suave como vimos, mas sim, remover o registro do banco de dados definitivamente. Isso também é possível, veja o exemplo prático:

```
1 public function forceDelete(Post $post, $id)
2 {
3     // Recupera o post pelo ID
4     $post = $post->find($id);
5
6     // Deleta definitivamente:
7     $post->forceDelete();
8
9     // Também é possível fazer assim, em uma única linha:
10    // $post = $post->find($id)->forceDelete();
11 }
```

Este recurso de Soft Deleted é muito útil e simples, e em diferentes cenários certamente será necessário utilizar. Portanto espero que este tutorial tenha sido útil, e qualquer dúvida, deixa aquele comentário! 😊

Abraços 🤗

Compartilhe:  (<https://www.facebook.com/sharer/sharer.php?u=https://blog.especializati.com.br/aprenda-como-utilizar-o-recurso-de-soft-deleting-do-laravel/>)  (<https://twitter.com/intent/tweet?text=https://blog.especializati.com.br/aprenda-como-utilizar-o-recurso-de-soft-deleting-do-laravel/>)  (<https://plus.google.com/share?url=https://blog.especializati.com.br/aprenda-como-utilizar-o-recurso-de-soft-deleting-do-laravel/>)



(<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



Sobre o Autor:

Carlos Ferreira

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

Certificações: Comptia Linux +, LPI, Novell Certification.

TAMBÉM NO ESPECIALIZATI

Eventos no Laravel – Parte 2

há 3 anos • 2 COMENTÁRIOS

Esse tutorial é a continuação do tutorial sobre Eventos no ...

7 motivos para aprender PHP em 2018

há 3 anos • 1 COMENTÁRIO

Que o PHP é uma linguagem realmente muito agradável, isso é ...

Laravel Eloquent Global Scope

há 3 anos • 2 COMENTÁRIOS

No último tutorial detalhei sobre Local Scopes no Laravel, agora dando ...

Login i

há 3 anos

Em algu desenvo Laravel,

6 COMENTÁRIOS

especializati

Disqus' Privacy Policy

1 Iniciar sessão ▾

Recomendar

Tweet

f Partilhar

Mostrar primeiro os mais votados ▾



Escreva o seu comentário...

INICIE SESSÃO COM O

OU REGISTE-SE NO DISQUS ?

Nome

**Ricardo Costa** • há 3 anos

Excelente. Realmente existem situações onde vamos precisar destes recursos.

44 ^ | v 1 • Responder • Partilhar ›

**uhelliton** • há 2 anos

Excelente!!!

CONTEÚDOS RELACIONADO:



(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

YouTube f t in G
(<https://www.youtube.com/watch?v=...>)
sub_confirmation=1)

Todos os direitos reservados © 2021 - EspecializaTi. É proibida a reprodução total ou parcial deste conteúdo.