

# Criando Helper no CodeIgniter

POR · 17 DE JUNHO DE 2015

Muitas das vezes estamos fazendo um projeto e no decorrer dele, temos que criar funções para nos auxiliar em alguma tarefa. Quem nunca precisou tirar os caracteres especiais de um texto para montar uma URL? Ou até mesmo fazer um tratamento em strings ?

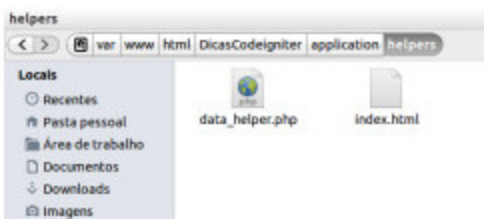
Pois é, toda vez que precisavamos formatar uma string temos que sempre fazer um **require()** em todas as páginas que usarmos tal função. Com o CodeIgniter podemos simplificar essa maneira, precisando somente chamar o nome da função direto na string. Isso é possível com a criação de um helper, que não é nada mais que um “ajudante”.

## Mas o que é um helper?

Helper não é nada mais que um ajudante, um grupo de funções aonde você poderá chamar no decorrer de seu projeto, assim que precisar.

Vamos fazer um exemplo aonde temos a data no formato **YYYY-mm-dd** e converter ela para **dd-mm-YYYY** usando um helper.

Para criar um helper você terá que criar um arquivo no formato **nomedoarquivo\_helper.php** . Então toda vez que você criar um helper utilize o sufixo **\_helper**. Esse arquivo deve ser colocado na pasta “helpers” localizado na sua pasta “application”. Nosso exemplo se chamará **data\_helper.php**



O conteúdo de um helper não é nada mais que funções. Simples né ?  
Para isso vamos criar uma função que irá fazer a conversão de uma data yyyy-mm-dd para dd-mm-yyyy.

### data\_helper.php

```
1 <?php
2 function FormatarData($entrada) {
3
4     $data = date('d/m/Y', strtotime($entrada));
5
6     return $data;
7 }
8 ?>
```

Pronto, nossa função já está pronta, agora só devemos chama-lá em nosso controller, caso ainda não saiba, você pode ler nosso artigo e aprender a [Como criar seu próprio controller no CodeIgniter](#).

### Exibir.php

```
1 <?php
2 class Exibir extends CI_Controller{
3
4     public function __construct() {
5         parent::__construct();
6     }
7
8     public function data(){
9
10        $this->load->helper('data');
11
12        $data = '2015-06-17';
13
14        echo FormatarData($data);
15    }
16 }
17 ?>
```

Na linha 10 chamamos o helper que foi criado, como ele foi renomeado data\_helper então devemos chamado como data.

Na linha 14 chamamos a função que criamos, no caso, FormatarData().

Pronto, seu helper foi criado e chamado com sucesso. Lembrando que sempre que você quiser utilizar o helper, você terá que adicionar *\$this->load->helper('data');* em seu código. Caso sua aplicação for utilizar frequentemente o helper criado, aconselho a chamar o helper no *\$autoload['helper']*.

Para chamar o helper no autoload abra o arquivo *application/autoload.php* e procure pela linha que contenha a array *\$autoload['helper']* e adicione o nome do helper criado. No meu ficaria assim:

```
1 $autoload['helper'] = array('url', 'html', 'data');
```

Agora você pode chamar a função criada no helper sem precisar carrega-ló toda vez que utilizar, ficando o seu código anterior assim:

```
1 <?php
2 class Exibir extends CI_Controller{
3
4     public function __construct() {
5         parent::__construct();
6     }
7
8     public function data(){
9
10        $data = '2015-06-17';
11
12        echo FormatarData($data);
13    }
14 }
15 ?>
```

Agora você já pode simplificar seu controller e sempre criar seus helpers com as funções que seu projeto irá precisar. Isso manterá a organização e agilidade para uma futura manutenção em seus projetos!

Abraços

Alisson Acioli