

Helpers

Ao acessarmos o `index.php` pela uri <http://localhost/mercado/index.php/produtos/index> (<http://localhost/mercado/index.php/produtos/index>), o nosso css não funciona. Se olharmos na aba de desenvolver do navegador, o css recebe um 404, pois o path para o arquivo está errado!

 `view-source:localhost/mercado/index.php/produtos/css/bootstrap.css`

O problema é que quando linkamos o css na página colocamos:

```
<link rel="stylesheet" href="css/bootstrap.css">
```

Este caminho é relativo a uri que está no navegador, e não ao arquivo.

Mas como iremos arrumar isso? Existe uma função php chamada `base_url` que irá resolver isto para nós, deixando o import do css assim:

```
<link rel="stylesheet" href="<?= base_url("css/bootstrap.css") ?>">
```

Acesse a página agora e veja que está tudo em branco! Caimos novamente no caso parecido com o do banco de dados, você se lembra? Para acessarmos a função, precisamos antes importá-la. Lembre-se sempre disto. E como fazemos para importar essa função que ajuda, este helper ?

Precisamos alterá-la no nosso Controller, na função `index` , antes de carregarmos a view

```
$this->load->helper("url");  
$this->load->view("produtos/index.php", $dados);
```

Confira agora no seu navegador que o import do css está correto, ele deve estar assim:

```
<link rel="stylesheet" href="http://localhost/mercado/css/bootstrap.css">
```

Os helpers servem para ajudar na view, você pode encontrar todos os helpers na documentação:

<http://ellislab.com/codeigniter/user-guide/> (<http://ellislab.com/codeigniter/user-guide/>)

Nós iremos ver alguns durante o curso. Vamos começar vendo o helper para formatar dinheiro, mas o helper padrão formata em dólar, uma vez que o formato padrão é moeda americana. Não é isto que queremos, certo? Então vamos aprender a criar o nosso para formatar em real.

Todos os helpers devem ser criados no diretório `/application/helpers` . Vamos criar um arquivo chamado `currency_helper.php` com uma função que transforma número em reais:

O código abaixo recebe um número, e usa o `number_format` do próprio PHP. O `number_format` recebe o número a ser formatado, o separador de casas decimais, neste caso a vírgula, e o separador de milhares, neste caso o ponto,

concatenando no final o R\$.

```
<?php
function numeroEmReais($numero) {
    return "R$ " . number_format($numero, 2, ",", ".");
}
```

Agora podemos alterar o index.php para mostrar os números em reais, alterando esta linha:

```
<?=$produto["preco"]?>
```

Para ficar assim:

```
<?= numeroEmReais($produto["preco"])?>
```

Lembre-se de importar o helper no seu controller, antes da função que chama a view

```
$this->load->helper("url");
$this->load->helper("currency");
$this->load->view("produtos/index.php", $dados);
```

O resultado deve ser este:

Produtos

Bola de Futebol	Bola de futebol...	R\$ 300,00
HD Externo	Marca HD-Mega	R\$ 400,00
Garrafa de 2 Litros	Um refrigerante	R\$ 10,50