

Tela de Login com Bootstrap e CodeIgniter

POR **FÁBIO S. RESZKO** · PUBLISHED 12 DE OUTUBRO DE 2014 · UPDATED 26 DE AGOSTO DE 2016

UPDATE!

Estão ocorrendo 2 erros que muitos têm me perguntado a causa.

O primeiro é este:

```
1 A PHP Error was encountered
2
3 Severity: Notice
4
5 Message: Only variable references should be returned by reference
6
7 Filename: core/Common.php
8
9 Line Number: 257
```

Este é um bug nesta versão do CodeIgniter, 2.2.0. É um erro no retorno do valor de uma variável por referência. A solução é atualizar o CI para a versão 3.+ ou então abrir o código core/Common.php na linha 257 e fazer a seguinte mudança:

```
1 De:
2 return $_config[0] =& $config;
3
4 Para:
5 $_config[0] =& $config;
6 return $_config[0];
```

A solução ideal é atualizar o CI.

O outro erro é em relação a uma mensagem na tela inicial dizendo que a variável \$erro não existe. Eu tenho o costume de trabalhar com a opção short_open_tag habilitada no PHP, o que significa que basta eu digitar isso: <? ?> em vez disso: <?php ?> pra fazer tudo funcionar. Porém, a opção short_open_tag deve ser habilitada no PHP. Veja isso no php.ini, ou no WAMP, clique no ícone na bandeja, vá em PHP -> PHP Settings e habilite ao opção short_open_tag, então após reiniciar os servidor, seu código irá funcionar.

Olá.

No post de hoje vamos aprender a como integrar aquela tela de login que existe no exemplo no site do [Bootstrap](#) no [CodeIgniter](#), fazendo um sistema simples de autenticação.

Como a intenção aqui é apenas mostrar como podemos fazer um sistema de login bem interessante no CodeIgniter e também mostrar como utilizar aquela bela tela de login do Bootstrap, não iremos utilizar Banco de Dados.

O usuário/senha serão *hardcode* e também após o usuário conseguir logar-se, será apresentado apenas uma página com um texto informando o sucesso do login.

Para começar, vamos utilizar de uma funcionalidade que a Orientação à Objetos nos fornece para que possamos criar este projeto.

Vamos lançar mão da herança na orientação à objetos, onde nós conseguimos herdar as propriedades, métodos, etc da classe pai.

Para ser mais específico, o que nós vamos fazer pedir ao CodeIgniter para que todos os controllers que iremos proteger, estendam o controller chamado MY_Controller, que é onde iremos colocar o código de verificação de login do usuário.

Dentro do controller MY_Controller, na função __construct(), que é chamada sempre que uma classe é construída, nós vamos consultar uma variável na sessão para saber se o usuário passou pela tela de login.

Caso o valor desta variável seja igual a 1, então nada é feito e o fluxo continua, porém, caso o valor seja diferente de 1, indica então que o usuário não está logado, e assim nós redirecionamos ele novamente para a tela de login.

O legal aqui é que nós não vamos precisar especificar a tela de login quando precisarmos redirecionar o usuário.

Nós vamos simplesmente redirecioná-lo para a url base do sistema, pois como o CodeIgniter está configurado, neste caso, para que ele chame o *controller home* automaticamente que é o controller principal, e como este controller estende o MY_Controller, então, por mais que nós chamemos a URL base, se o usuário não estiver logado, ele vai ser redirecionado para a tela de login.

Se você se perdeu no texto acima, não se preocupe, logo mais você entenderá.

Vamos então começar a construir nosso sistema de login.

O primeiro passo é configurarmos o CodeIgniter para carregar automaticamente a biblioteca *session* e o helper *url*

Para isto, abra o arquivo **application/config/autoload.php** e na linha 55, que onde você informa quais bibliotecas deseja carregar, digite dentro da array a string *session* ficando assim:

```
1 $autoload['libraries'] = array('session');
```

Na linha, 67, deixe assim:

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

O helper *url* nos traz função relacionadas à url do sistema.

Feito isso, agora você precisa configurar uma chave secreta para que a biblioteca *session* seja utilizada. Para isto abra o arquivo config.php que também está neste mesmo diretório.

Desça até a linha 227 e configure uma chave de criptografia. Veja meu exemplo abaixo:

```
1 $config['encryption_key'] = 'Aqui vai uma chave bem dificil de encriptação.';
```

Você pode colocar qualquer texto aí.

Agora vamos informar ao CodeIgniter para carregar como padrão nosso controller *home*. Para isto, abra o arquivo *routes.php* também dentro desta mesma pasta, e vá até o final do arquivo.

A linha 41 informa ao CodeIgniter que o controller padrão que será carregado, caso nenhum seja informado, é o *welcome*. Mude para *home* que será agora o controller principal.

Muito bem, feito isso, vamos agora criar o controller que será o responsável pela verificação do login do usuário.

Vamos criar um controller chamado MY_Controller e colocá-lo dentro da pasta *application/core/* com o seguinte conteúdo:

```
1 <?php
2
3 class MY_Controller extends CI_Controller {
4
5     public function __construct()
6     {
7         parent::__construct();
8
9         $logado = $this->session->userdata("logado");
10
11         if ($logado != 1)
12             redirect(base_url('index.php/login'));
13     }
14 }
```

O caminho para este controller ficará assim: *application/core/MY_Controller.php*

Agora, dentro da pasta *application/controllers* modifique o controller *welcome.php* para *home.php* lembrando de alterar também o nome da classe dentro deste arquivo, alterando de *Welcome* para *Home* e mude também a classe pai, que originalmente está como CI_Controller para MY_Controller.

Aqui que está o pulo do gato.

Todo controller que você quiser proteger, terá que estender o controller *MY_Controller*, pois é dentro deste que está o código de verificação do login do usuário.

Abaixo veja como ficou o código do controller home.php:

```
1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Home extends MY_Controller {
4
5     public function index()
6     {
7         $this->load->view('v_home');
8     }
9 }
```

Este controller basicamente só chama a view *v_home*, que traz a mensagem de sucesso quando o usuário estiver logado.

Vamos criar agora outro controller dentro desta pasta, chamado *login.php*. Este controller será responsável pelas funções relacionadas ao login do usuario.

Ele terá 3 funções dentro.

A primeira função *index()* simplesmente carrega a view login. Veja abaixo:

```
1     public function index()
2     {
3         $this->load->view('v_login');
4     }
```

A função *logar()* é para onde o action do formulário de login deverá apontar. Aqui nós fazemos a comparação do usuário/senha digitados com os gravados.

Neste exemplo o usuário/senha estão *hardcode*. Mas você fará a modificação para um banco de dados, caso necessite.

Veja abaixo o código:

```
1 public function logar(){
2
3     $usuario = $this->input->post("usuario");
4     $senha = sha1($this->input->post("senha"));
5
6     //Código sha1 da senha 123456 7c4a8d09ca3762af61e59520943dc26494f8941b
7     //O usuário no exemplo aqui será usuario@email.com.br
8     //Mas em um projeto real, você trará isto do banco de dados.
9
10    //Se o usuário e senha combinarem, então basta eu redirecionar para a url base, pois agora o usuário irá
11    //pela verificação que checa se ele está logado.
12    if ($usuario == "usuario@email.com.br" && $senha == '7c4a8d09ca3762af61e59520943dc26494f8941b' ) {
13        $this->session->set_userdata("logado", 1);
14        redirect(base_url());
15    } else {
16        //caso a senha/usuario estejam incorretos, então mando o usuário novamente para a tela de login com
17        $dados['erro'] = "Usuário/Senha incorretos";
18        $this->load->view("v_login", $dados);
19    }
20 }
```

E a última função é a *logout()* onde eu simplesmente destruo a variável *logado* que é responsável por indicar se o usuário está ou não logado no sistema. Veja:

```
1 /*
2  * Aqui eu destruo a variável logado na sessão e redireciono para a url base. Como esta variável não existe m
3  * será direcionado novamente para a tela de login.
4  */
5 public function logout(){
6     $this->session->unset_userdata("logado");
7     redirect(base_url());
8 }
9
```

Atenção!

Algo muito importante. O controller *login.php* **não pode estender a classe MY_Controller**, pois não há sentido em queremos proteger a tela de login.

Caso você estenda a classe *MY_Controller*, o navegador irá entrar em loop.

Este controller deve estender normalmente o controller *CI_Controller* do CodeIgniter.

Veja abaixo o código completo do controller *login.php*:

```
1 <?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Login extends CI_Controller {
4
5     public function index()
6     {
7         $this->load->view('v_login');
8     }
9
10    public function logar(){
11
12        $usuario = $this->input->post("usuario");
13        $senha = sha1($this->input->post("senha"));
14
15        //Código sha1 da senha 123456 7c4a8d09ca3762af61e59520943dc26494f8941b
16        //O usuário no exemplo aqui será usuario@email.com.br
17        //Mas em um projeto real, você trará isto do banco de dados.
18
19        //Se o usuário e senha combinarem, então basta eu redirecionar para a url base, pois agora o usuário irá
20        //pela verificação que checa se ele está logado.
21        if ($usuario == "usuario@email.com.br" && $senha == '7c4a8d09ca3762af61e59520943dc26494f8941b' ) {
22            $this->session->set_userdata("logado", 1);
23            redirect(base_url());
24        } else {
25            //caso a senha/usuario estejam incorretos, então mando o usuário novamente para a tela de login com
26            $dados['erro'] = "Usuário/Senha incorretos";
27            $this->load->view("v_login", $dados);
28        }
29    }
30
31    /*
32     * Aqui eu destruo a variável logado na sessão e redireciono para a url base. Como esta variável não existe
33     * será direcionado novamente para a tela de login.
34     */
35    public function logout(){
36        $this->session->unset_userdata("logado");
37        redirect(base_url());
38    }
39
40 }
```

A parte de controllers termina aqui. Agora vamos criar as views *login* e *home*.

A view login é a tela de login que vamos utilizar lá do Bootstrap.

Originalmente o exemplo do bootstrap tem uma checkbox para relembrar o login. Como esta é uma etapa um pouco mais complexa, não vou abordar aqui, por isso removi este input do formulário.

Veja abaixo como ficou o código completo da view *v_login.php*:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <meta name="description" content="">
8     <meta name="author" content="">
9     <link rel="icon" href="../../favicon.ico">
10
11     <title>Template de Login do Bootstrap</title>
12
13     <!-- Latest compiled and minified CSS -->
14     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
15
16     <!-- Custom styles for this template -->
17     <link href="<?= base_url('includes/signin.css') ?>" rel="stylesheet">
18
19     <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->
20     <!--[if lt IE 9]>
21     <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
22     <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
23     <![endif]-->
24 </head>
25
26 <body>
27     <div class="container">
28
29         <form class="form-signin" role="form" method="post" action="<?= base_url('index.php/login/logar') ?>">
30             <h2 class="form-signin-heading">Por favor, logue-se</h2>
31             <input type="email" class="form-control" placeholder="Email address" required autofocus name="usuario">
32             <input type="password" class="form-control" placeholder="Password" required name="senha">
33             <button class="btn btn-lg btn-primary btn-block" type="submit">Fazer login</button>
34             <? if (isset($erro)): ?>
35                 <div class="alert alert-danger" role="alert" style="margin-top: 10px;"><?= $erro; ?></div>
36             <? endif; ?>
37         </form>
38     </div> <!-- /container -->
39 </body>
40 </html>
```

A outra view é a *v_home.php*. Esta seria a view principal do seu sistema que é protegido.

O usuário só vai conseguir ver esta view, se conseguir passar com sucesso pela tela de login.

Aqui eu só coloquei uma mensagem de sucesso e um link para o usuário deslogar. Veja:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <meta name="description" content="">
8     <meta name="author" content="">
9     <link rel="icon" href="../../favicon.ico">
10
11     <title>Template de Login do Bootstrap</title>
12
13     <!-- Latest compiled and minified CSS -->
14     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
15
16     <!-- Custom styles for this template -->
17     <link href="<?= base_url('includes/signin.css') ?>" rel="stylesheet">
18
19     <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->
20     <!--[if lt IE 9]>
21     <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
22     <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
23     <![endif]-->
24 </head>
25
26 <body>
27     <div class="container">
28
29         <h1 class="text-center">Parabéns, você está logado.</h1>
30         <p>Lembre-se, para proteger seus controllers, basta estender o controller MY_Controller, e assim, automa
31         seus controllers passarão pela verificação de senha.
32         </p>
33         <link aqui para deslogar: <a href="<?= base_url('index.php/login/logout') ?>">Deslogar</a></p>
34
35     </div>
36 </body>
37 </html>
```

Bom, é isto. O sistema de login termina aqui.

Abaixo segue este sistema funcionando para você testar. Os dados de acesso são:

Usuário: usuario@email.com.br

Senha: 123456