

Criando uma interface/página para registros dados no BD

1. Introdução

Um sistema normalmente armazena os registros no banco de dados. Estes registros são manipulados pelo sistema, por exemplo exibindo em uma página quando forem solicitados.

A inserção destes registros no banco de dados normalmente acontecem através de páginas onde os usuários podem dar entrada a novos registros ou editar algum registro armazenado no banco.

Nesta aula criaremos uma interface/página para fazer a inserção no banco de dados do museu, criaremos uma função dentro do controller Home e adicionaremos uma rota no arquivo Routes.php.

2. Criação da função para abrir a página de cadastro

Vamos editar o arquivo “app/Controllers/Home.php”. No final do arquivo vamos **acrescentar** uma nova função para abrir a página de cadastro de novos registros no banco de dados.

```
public function inserirRegistro()
{
    echo View('InserirRegistro'); // a palavra reservada “View” é uma função que chamará a
                                // página “app/Views/InserirRegistro.php”
}
```

Explicando o Código:

- Esta função deve ser inserida no final do arquivo, antes da chave “}” final. Não se deve apagar nenhuma função;
- o nome da função é “**inserirRegistro**”. Este nome é sugerido para ser facilmente lembrado, nomes que remetem ao que a função faz é mais fácil de ser lembrada. Observe que neste caso específico a primeira letra da primeira palavra é minúscula e a primeira letra da segunda palavra é MAIÚSCULA. **Não há espaços ou acentos em nomes de funções**;
- A view “InserirRegistro” é o nome da página “InserirRegistro.php” que estará em “app/Views/”.

3. Criação da View para inserção dos novos registros

A view a ser criada é uma página html que contém um form e os campos que serão salvos no banco de dados. Neste exemplo faremos uma página simples, sem css. Posteriormente ela ser formatada, excluindo a <table></table>, as <tr></tr> e substituindo as <td></td> por <div>, inserindo as CDN do bootstrap para formatação da página.

Crie a página **“InserirRegistro.php”** dentro do diretório “app/Views”. Digite o código abaixo.

```
<html>

    <head>
        <title>Página de Inserção de Registros</title>
        <meta charset="utf-8">

    </head>

    <body>
        <h2 align='center'>Página de Inserção de Registros</h2>

        echo $msg;

    <form method="post">

        <table border='1'>
            <tr><td>Título:</td><td><input type="text" name="Titulo" /></td></tr>
            <tr><td>Texto:</td><td><textarea name="Texto" rows="10" cols="30"
/></textarea></td></tr>
            <tr><td><input type="submit" value="inserir"></td></tr>

        </table>
    </form>
    </body>

</html>
```

Explicando o Código.

- Nas tags “<title></title>” e “<h2></h2>” não há nenhuma interação dinâmica, não há passagem de variáveis. Posteriormente estas informações serão dinâmica;
- O “<form>” não passa o parâmetro “action”, o codeigniter automaticamente enviará este form para o controller;
- Como o texto a ser inserido pode ser longo, usamos uma “textarea” para o usuário inserir ou colar o texto.

4. Criando a rota para acessar a página

Para acessar a página recém criada através da URL vamos adicionar no arquivo “**app/Config/Routes;.php**” a linha para chamar a página. Digite após a última linha:

```
$routes->get('inserirRegistro', 'Home::inserirRegistro');
```

Explicando a linha

- o primeiro parâmetro “inserirRegistro” será usado na URL para chamar a página. Este parâmetro pode conter um outro nome: “inserir”. Neste exemplo usamos este primeiro nome para que fique bem explícito;
- o segundo parâmetro “ Home::inserirRegistro” é o Controller e a função criada.

Para acessar a página digite no navegador:

```
http://ip.do.servidor/usuario/disciplina/nome.do.museu/public/index.php/inserirRegistro
```

OBS.: A parte em vermelho da URL deve ser digitada exatamente como está aí. A parte inicial deve ser alterada conforme necessário.

Se tudo correu bem a página de cadastro será aberta e exibida no navegador. Observe que não há nenhuma formatação. Como foi explicado no item 3 a formatação será feita usando o bootstrap.

5. Fazendo o insert no banco de dados

Com toda a infraestrutura pronta (Controller, Route, View) abrindo a página para a inserção dos novos registros, vamos passar para a atualização do Controller e da View para que o insert seja feito de forma correta.

Entendendo o funcionamento

Depois de abrir a página de inserção e digitando o título e texto, ao clicar no botão de “inserir” a página será submetida e o Controller será novamente chamado. Ele receberá os dados enviados via “POST”, fará a checagem, fará o tratamento dos dados e o posterior insert no banco de dados.

Nesta etapa abriremos o Controller Home.php para editar a função “inserirRegistro”. Observe o código abaixo.

```
public function inserir()  
{  
  
    $data['titulo'] = 'Cadastrar um novo Registro no Banco de dados;  
    $data['acao'] = 'inserir';
```

```

// Vamos criar uma variável para ser usada no insert
$data['msg'] = '';

// este bloco de códigos irá verificar em qual momento está a view

if($this->request->getMethod() === 'post')
{
    $museuModel = new \App\Models\MuseuModel();

    $museuModel->set('Titulo', $this->request->getPost('Titulo'));
    $museuModel->set('Texto', $this->request->getPost('Texto'));
    $museuModel->insert();

    if ($museuModel->insert())
    {
        // deu certo
        $data['msg'] = 'Registro inserido com sucesso';
    }
    else
    {
        // deu errado
        $data['msg'] = 'Erro ao inserir Registro';
    }
}

echo View('InserirRegistro', $data);
}

```

Explicando o código.

- Criamos um array chamado “data” e atribuímos a ele 3 valores: titulo, acao e msg. Eles serão usados na view para exibir de forma dinâmica os textos;
- Com o comando “**if(\$this->request->getMethod() === 'post')**” verificamos se os parâmetros chegaram via método post;
- Criamos um objeto \$museuModel à partir do model MuseuModel;
- Depois o sistema recebe os campos que vieram da view “InserirRegistro” e armazenam no objeto \$museuModel ;

- Por fim o objeto \$museuModel usa o comando insert() do framework para gravar no banco de dados;
- O comando if testa se o insert foi feito, se for positivo ele escreve na view InserirRegistro.php que “ Registro inserido com sucesso”, senão escreve “ Erro ao inserir Registro;
- O último comando “echo View('InserirRegistro', \$data)”, abre a View, passando os parâmetros através da variável “\$data”

6. Alterando a view com as novas configurações

Nesta etapa vamos fazer algumas pequenas alterações no código da view para que ela receba corretamente os comandos e variáveis vindos do controller.

Abra a view InserirRegistro.php. As alterações e inserções estão assinaladas de **VERMELHO**.

```
<html>

    <head>
        <title><?php echo $titulo ?></title>
        <meta charset="utf-8">

    </head>

    <body>
        <h2 align='center'> <?php echo $titulo ?></h2>

        <?php echo $msg; ?>

        <form method="post">
<table border='1'>
    <tr><td>Título:</td><td><input type="text" name="Titulo" /></td></tr>
    <tr><td>Texto:</td><td><textarea name="Texto" rows="10" cols="30"
/></textarea></td></tr>
    <tr><td><input type="submit" value="<?php echo $acao ?>"</td></tr>
</table>
        </form>
    </body>

</html>
```

Explicando o código.

- Inserimos na tag <title></title> o php que exibirá o título da página. Neste momento não haverá mudanças no título, mas abrimos a possibilidade de exibir título e cabeçalhos diferentes dependendo da situação;
- Na tag <h2></h2> inserimos o título tal qual fizemos no título da página;
- O comando seguinte, echo \$msg exibirá se o insert foi ou não feito com sucesso;
- Por fim, dentro do form o botão de submeter recebe o valor vindo da variável \$acao.

7) Inserindo uma nova rota no arquivo Routes.php

Para que o insert seja feito é importante inserir uma nova linha no arquivo Routes.php, como o conteúdo a ser inserido vem através do método post, precisamos adicionar uma nova linha. Abra o arquivo “app/Config/Routes.php”, adicione logo após a última linha o seguinte comando:

```
$routes->post('inserirRegistro', 'Home::inserirRegistro');
```

Esta linha é responsável por fazer a rota para chamar o controler.

8. Conclusão

Uma página para fazer o insert de registros em um sistema é fundamental. Neste caso específico a página não é complexa, posteriormente ela poderá ser novamente chamada para a edição de um registro específico, só que desta vez trabalhando de forma diferente.

Bons estudos