

Univiçosa

ADS 103 – Desenvolvimento web back-end

Prof. Hermes Nunes Pereira Júnior

CodeIgniter, adicionando novas funções de seleção no Controller

Na aula de hoje adicionaremos novas funções para selecionarmos outros registros gravados no banco de dados e exibidos na view “Conteúdo.php”.

1. Criando novas funções para buscar os dados no bd.

Como vimos anteriormente, para exibir os dados na View “Conteúdo.php” é necessário que o banco esteja povoado, que o “app/Models/MuseuModel.php” e o “app/Controllers/Home.php” estejam com suas configurações corretas.

Ao abrir o controller “Home.php” temos a função index() e a função exibir(), esta última função foi usada para fazer o select no banco e exibir na View “Conteúdo.php”.

Observe o quadro abaixo a função exibir.

```
public function exibir()
{
    // vai chamar uma view que exibe todas os registros

    // criar uma variável para ter acesso à MuseuModel
    $museuModel = new \App\Models\MuseuModel();

    // vamos armazenar o select na variavel $data
    // o método find() faz o select de todos os campos listados no model
    // MuseuModel
    $data['dados'] = $MuseuModel->where('Id', 1)->findAll();

    // o controller vai chamar a view “Conteúdo.php”
    echo view('Conteúdo', $data);
}
```

2. Criando uma nova function para exibir um novo registro na View

Olhando o código da function exibir, podemos criar uma nova function igual a ela, mas desta vez alterando o nome e o valor do “Id”.

Vamos tomar como **exemplo** que no banco de dados, o Id = 2 refere-se a sistemas operacionais. O Título é “Sistemas Operacionais” e o texto trata deste assunto.

Observe o quadro abaixo.

```

public function SistemasOperacionais()
{
    // vai chamar uma view que exibe todas os registros

    // criar uma variável para ter acesso à MuseuModel
    $museuModel = new \App\Models\MuseuModel();

    // vamos armazenar o select na variavel $data
    // o método find() faz o select de todos os campos listados no model
    // MuseuModel
    $data['dados'] = $MuseuModel->where('Id', 2)->findAll();

    // o controller vai chamar a view "Conteudo.php"
    echo view('Conteudo', $data);
}

```

Explicando as alterações.

- O nome da função é “SistemasOperacionais”. Como neste exemplo hipotético o Id=2 tem o “Titulo” e o “Texto” com este assunto, vamos usar um nome para a função que remete ao assunto. Não se deve usar espaços ou acentos no nome da função;
- Na linha “**\$data['dados'] = \$MuseuModel->where('Id', 2) -> findAll();**” o Id recebe o valor 2. O select usará o 2 como base para a pesquisa na tabela. Se fôssemos usar o sql puro ficaria assim: “select * from Conteudo where Id = 2”.

03. Criando a rota para acessar a página

Por questão de segurança o framework não permite que páginas ou views sejam acessadas diretamente. Para isso um arquivo é o responsável por organizar todas as rotas do sistema: “app/Config/Routes.php”.

Vamos inserir a seguinte linha no final do arquivo

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

ao escrever na url do sistema a palavra “sistemasoperacionais”, será chamado o controller “Home” e a função “SistemasOperacionais”.

Algumas observações:

- o nome “sistemasoperacionais” será usado como parâmetro na URL, como o sistema é case sensitive, ele diferencia letras minúsculas e MAIÚSCULAS. Então cuidado com os nomes;
- não há necessidade de colocar o mesmo nome da função na rota. A função se chama “SistemasOperacionais”, a rota se chama “sistemasoperacionais”. Pode ser uma abreviação dos nomes, mas cuidado para que se houver outras funções o padrão das abreviações sejam as mesmas.

A url ficará como no quadro abaixo.

<code>http://192.168.51.5/aXXXXX/adsXXX/museu/public/index.php/sistemasoperacionais</code>
--

Aí será exibida a página com os dados que foram gravados no banco de dados referentes a sistemas operacionais.

4) Considerações Finais sobre a aula

Vimos nesta aula que a exibição dos dados na view usando os recursos do framework é muito mais simples que criar manualmente em html. Como as informações estão no banco de dados, usar uma única página de forma dinâmica para exibir cada registro torna o sistema muito mais ágil e fácil de fazer manutenção.

Bons estudos.