

Técnico em Desenvolvimento de Sistemas

Programação WEB 3

Linguagem PHP – Orientação a Objetos

Prof. Laércio Silva

laercio.silva31@etec.sp.gov.br

Lndsilva@hotmail.com

PDO :: PHP Data Objects

- O PHP conecta-se a uma enorme variedade de gerenciadores de bancos de dados disponíveis no mercado.
- Para cada tipo de banco de dados existe uma gama de funções associadas para realizar operações como conexão, consulta, retorno, desconexão, dentre outras.

PDO :: PHP Data Objects

- Em razão da crescente adoção do PHP, surgiu a necessidade de unificar o acesso às diferentes extensões de bancos de dados presentes no PHP.
- Assim surgiu a PDO (PHP Data Objects), cujo objetivo é prover uma API limpa e consistente, unificando a maioria das características presentes nas extensões de acesso a banco de dados.

PDO :: PHP Data Objects

- A PDO não é uma biblioteca completa para abstração do acesso à base de dados, uma vez que ela não faz a leitura e tradução das instruções SQL, adaptando-as aos mais diversos drivers de bancos de dados existentes.
- Ela simplesmente unifica a chamada de métodos, delegando-os para as suas extensões correspondentes e faz uso do que há de mais recente no que diz respeito à orientação a objetos presente no PHP5.

PDO :: PHP Data Objects

- Para conectar em bancos de dados diferentes, a única mudança é na string de conexão.

Banco	String de conexão
SQLite	<code>new PDO('sqlite:testes.db');</code>
FireBird	<code>new PDO("firebird:dbname=C:\\base.GDB", "SYSDBA", "masterkey");</code>
MySQL	<code>new PDO('mysql :unix_socket=/tmp/mysql.sock;host=localhost;port=3307;dbname=livro', 'user', 'senha');</code>
Postgres	<code>new PDO('pgsql :dbname=example;user=user;password=senha;host=localhost');</code>

Programação a Objetos em PHP:

Acesso a banco de dados

```
conectando_pdo.php x
1 <?php
2 $servername = "localhost";
3 $username = "root";
4 $password = "";
5 try {
6     $conn = new PDO("mysql:host=$servername;dbname=livro", $username, $password);
7
8     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
9
10    echo "Banco de dados conectado!!!";
11 }
12 catch(PDOException $e) {
13     echo "Erro ao conectar o banco de dados: " . $e->getMessage();
14 }
15
16 ?>
```

Inserção, alteração e exclusão

- Neste primeiro exemplo, assim como no exemplo de inserção de dados visto anteriormente, o programa irá se conectar ao banco de dados em localhost, chamado livro, com o usuário admin.
- Em seguida, ele irá inserir dados relativos a clientes na base de dados.
- Por fim, ele fechará a conexão ao banco de dados.
- Caso ocorra algum erro durante a execução das instruções SQL, será gerada uma exceção, tratada adequadamente pelo bloco try/catch.

Programação a Objetos em PHP:

Inserção

```
1  <?php
2
3      $servername = "localhost";
4      $username = "root";
5      $password = "";
6
7      try
8      {
9          //instancia objeto PDO, conectando no mysql
10
11          $conn = new PDO("mysql:host=$servername;dbname=livro", $username, $password);
12
13          //executa uma série de instruções SQL
14
15          $conn->exec("INSERT INTO famosos(codigo,nome) VALUES (1,'Érico Veríssimo')");
16          $conn->exec("INSERT INTO famosos(codigo,nome) VALUES (2,'John Lennon')");
17          $conn->exec("INSERT INTO famosos(codigo,nome) VALUES (3,'Mahatma Gandhí')");
18          $conn->exec("INSERT INTO famosos(codigo,nome) VALUES (4,'Ayrton Senna')");
19          $conn->exec("INSERT INTO famosos(codigo,nome) VALUES (5,'Charlie Chaplin')");
20          $conn->exec("INSERT INTO famosos(codigo,nome) VALUES (6,'Anita Garibaldi')");
21          $conn->exec("INSERT INTO famosos(codigo,nome) VALUES (7,'Mário Quintana')");
22
23          // fecha a conexão
24
25          $conn = null;
26      }
27      catch (PDOException $e)
28      {
29          //caso ocorra uma exceção, exibe na tela
30          print "Erro!: " . $e->getMessage() . "\n";
31          die();
32      }
33  ?>
```


Programação a Objetos em PHP:

Busca

```
pdo_lista.php x
1  <?php
2
3      $servername = "localhost";
4      $username = "root";
5      $password = "";
6
7      try
8      {
9          //instancia objeto PDO, conectando no mysql
10         $conn = new PDO("mysql:host=$servername;dbname=livro", $username, $password);
11
12         //executa uma instrução SQL de consulta
13
14         $result = $conn->query("SELECT codigo, nome FROM famosos");
15
16         if ($result){
17
18             //percorre os resultados via iteração
19             foreach($result as $row)
20             {
21                 //exibe os resultados
22                 echo $row['codigo'] . ' - ' . $row['nome'] . "<br>\n";
23             }
24         }
25         //fecha a conexão
26         $conn = null;
27     }
28     catch(PDOException $e){
29         print "Erro!:". $e->getMessage(). "<br/>";
30         die();
31     }
32     ?>
```

Buscar

- Podemos também retornar os dados de uma consulta por meio da função `fetch()`, a qual aceita como parâmetro o "estilo de fetch":
De acordo com o estilo, o retorno poderá ser um dos relacionados a seguir:

Parâmetros	Descrição
PDO: <code>:FETCH_ASSOC</code>	Retorna um array indexado pelo nome da coluna.
PDO: <code>:FETCH_NUM</code>	Retorna um array indexado pela posição numérica da coluna.
PDO: <code>:FETCH_BOTH</code>	Retorna um array indexado pelo nome da coluna e pela posição numérica da mesma.
PDO: <code>:FETCH_OBJ</code>	Retorna um objeto anônimo (stdClass), de modo que cada coluna é acessada como uma propriedade.

Buscar com função `fetch()`

- Repetimos o mesmo programa que lista os códigos e os nomes dos famosos do banco de dados.
- A diferença é que agora utilizaremos a função **`fetch()`** para listar os resultados.
- O estilo de fetch desejado será o `PDO::FETCH_OBJ`, que retorna os dados da consulta em forma de objeto.
- Veja que agora a variável `$row` é um objeto, e cada coluna retornada (codigo, nome) é acessada como uma propriedade deste objeto.

Programação a Objetos em PHP:

Busca com função fetch()

```
pdo_lista_obj.php x
1  <?php
2      $servername = "localhost";
3      $username = "root";
4      $password = "";
5
6      try{
7
8          //instancia objeto PDO, conectando no mysql
9          $conn = new PDO("mysql:host=$servername;dbname=livro", $username, $password);
10
11         //executa uma instrução SQL de consulta
12
13         $result = $conn->query("SELECT codigo, nome FROM famosos");
14
15         if ($result){
16             //percorre os resultados via fetch()
17             while ($row = $result->fetch(PDO::FETCH_OBJ))
18             {
19                 //exibe os dados na tela, acessando o objeto retornado
20                 echo $row->codigo . ' - ' . $row->nome . "<br/>";
21             }
22         }
23         //fecha a conexão
24         $conn = null;
25     }
26     catch (PDOException $e){
27         print "Erro!: " . $e->getMessage() . "<br/>";
28         die();
29     }
30  ?>
```

Programação a Objetos em PHP:

Excluindo registro:

```
1  <?php
2      $servername = "localhost";
3      $username = "root";
4      $password = "";
5
6      try {
7
8          //instancia objeto PDO, conectando no mysql
9          $conn = new PDO("mysql:host=$servername;dbname=livro", $username, $password);
10
11         //definindo o modo de exceção para o erro
12         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
13
14         //comando para excluir registro
15         $sql = "DELETE FROM famosos WHERE codigo=3";
16
17         //utilizando a função exec() porque nenhum resultado é retornado
18         $conn->exec($sql);
19         echo "Registro excluído com sucesso!!!";
20     } catch(PDOException $e){
21         echo $sql . "<br>" . $e->getMessage();
22     }
23     $conn = null;
24  ?>
```

Programação a Objetos em PHP:

Alterando registro:

```
pdo_update_my.php x
1  <?php
2      $servername = "localhost";
3      $username = "root";
4      $password = "";
5
6      try {
7          //instancia objeto PDO, conectando no mysql
8          $conn = new PDO("mysql:host=$servername;dbname=livro", $username, $password);
9
10         //definindo o modo de exceção para o erro
11         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
12
13         $sql = "UPDATE famosos SET nome='Ayrton Senna da Silva' WHERE codigo=4";
14
15         //Preparando a instancia
16         $stmt = $conn->prepare($sql);
17
18         //executando a query
19         $stmt->execute();
20
21         // mensagem caso a alteração ocorra
22         echo $stmt->rowCount() . " Alteração realizada com sucesso!!!";
23     } catch(PDOException $e) {
24         echo $sql . "<br>" . $e->getMessage();
25     }
26     $conn = null;
27     ?>
```

Muito Obrigado

Até a Próxima Aula

Prof. Laércio Silva
Email: Indsilva@hotmail.com