CRUD com PHP PDO

Veja nesse artigo como a realizar as operações de CRUD usando a API PDO do PHP.

Olá pessoal, em um artigo anterior falamos um pouco sobre a nova API PDO. Vimos como fazer para se conectar à um banco de dados MySQL e como retornar valores de um banco de dados.

Nesse artigo iremos continuar falando sobre esse assunto, só que dessa vez iremos ver como fazemos para realizar um CRUD.

Uma observação interessante é que o PDO fornece uma camada de abstração com o banco de dados, isso ocorre porque o PDO faz conexão com diversos bancos de dados diferentes, como MySQL, PLSQL, da mesma forma, tendo a única coisa diferente a string de conexão.

Mas afinal Ricardo, o que é um CRUD?

O CRUD vem do inglês, das palavras (Create, Read, Update, Delete) que quer dizer basicamente as 4 (quatro) principais operações com um banco de dados(inserir, ler, atualizar, excluir).

Quando pensamos em qualquer website dinâmico ou sistemas em gerais, a primeira coisa que vem na nossa cabeça é o CRUD, sem ele nada funciona, pois é com ele que realizamos as interações com o banco de dados.

Como a parte do Read (ler) nós já vimos no artigo anterior junto com a conexão (ver link no início do artigo), nesse artigo iremos tratar apenas o Insert, update e delete.

Vamos começar com o Insert usando o PDO.

Insert

Antes de vermos como fazer com o PDO, iremos ver como inserir dados no banco de dados com a api mysql, a qual não é mais recomendado seu uso.

Listagem 1: Inserindo dados na tabela com API mysql

```
        <?php
    $nome = $_POST["nome"];
    $email = $_POST["email"];
    $tel = $_POST["tel"];

    include_once 'conexao.php';

    $sql = "insert into cliente values(null,
            '".$nome."','".$email."','".$tel."')";
    //echo $sql;

    if(mysql_query($sql,$con)){
        $msg = "Gravado com sucesso!";
    }else{
        $msg = "Erro ao gravar!";
    }
    mysql_close($con);
?>
```

Como podemos ver é muito simples, mas foi considerado deprecated, ou seja, foi descontinuado pela galera do PHP, vamos ver como podemos fazer o insert usando PHP PDO.

Listagem 2: Insert usando PHP PDO

```
        <?php
try {
  $pdo = new PDO('mysql:host=localhost;dbname=meuBancoDeDados', $username, $password);
  $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

  $stmt = $pdo->prepare('INSERT INTO minhaTabela VALUES(:nome)');
  $stmt->execute(array(
```

```
    ':nome' => 'Ricardo Arrigoni'
  ));

  echo $stmt->rowCount();
} catch(PDOException $e) {
  echo 'Error: ' . $e->getMessage();

   ?>
```

Podemos ver que não muda muita coisa no código, apenas a forma como é gerado, usando try..catch e as funções de comunicação também, mas de resto continua bem parecida.

***Nota**: Essa maneira é a considerada ideal pela maioria dos desenvolvedores PHP hoje em dia, mas também existe a mysqli, que funciona bem parecido com a mysql, com a diferença de um "i" no final.*

Update

Para atualizar um registro na tabela do banco de dados utilizamos o comando UPDATE em SQL, só que essa parte é um pouco mais complexa, pois na verdade você vai precisar listar os registros, para ai sim escolher o que quer editar e atualizar.

No nosso exemplo não irei me atentar na forma de receber esses dados para ai sim atualizar em si, vamos dar ênfase apenas em como fazer o nosso código de atualizar.

Para fazer a atualização usando a API mysql é só fazer assim:

Listagem 3: Update com mysql_

```
        <?php

    $nome = $_POST["nome"];
    $email = $_POST["email"];
    $tel = $_POST["tel"];
    $id = $_POST["id"];

    include_once 'conexao.php';

    $sql = "update cliente set
            nome = '".$nome."', email = '".$email."',telefone = '".$tel."'
            where idcliente = ".$id;
```

```
    if(mysql_query($sql,$con)){
        $msg = "Atualizado com sucesso!";
    }else{
        $msg = "Erro ao atualizar!";
    }
    mysql_close($con);

    ?>
```

Agora utilizando o PDO basta utilizar esse código:

Listagem 4: Update usando PDO

```
        <?php

$id = 5;
$nome = "Novo nome do Ricardo";

try {
  $pdo = new PDO('mysql:host=localhost;dbname=meuBancoDeDados', $username, $password);
  $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

  $stmt = $pdo->prepare('UPDATE minhaTabela SET nome = :nome WHERE id = :id');
  $stmt->execute(array(
    ':id'   => $id,
    ':nome' => $nome
  ));

  echo $stmt->rowCount();
} catch(PDOException $e) {
  echo 'Error: ' . $e->getMessage();
}
?>
```

Lembrando que estamos usando exemplos distintos em cada API, o que queremos é mostrar como funciona as funções do PDO, fique livre para adaptar ao seu projeto da maneira que achar melhor.

Delete

A função de delete é praticamente igual ao insert na prática, a diferença é que ao invés de inserir dados, vamos estar excluindo eles da tabela, para isso precisamos identificar o registro de alguma maneira(normalmente usamos o próprio ID do registro) para então excluir da tabela.

Excluindo dados usando o mysql_ é da forma que mostramos na listagem 5:

Listagem 5: Excluindo dados usando mysql_

```
        <?php
    $id = $_GET["id"];
    include_once 'conexao.php';

    $sql = "delete from cliente where idcliente = ".$id;

    if(mysql_query($sql,$con)){
        $msg = "Deletado com sucesso!";
    }else{
        $msg = "Erro ao deletar!";
    }
    mysql_close($con);

    ?>
```

Para realizar a exclusão com o PDO ficaria dessa forma na listagem 6:

Listagem 6: Excluindo com PDO

```
        <?php
$id = 5;

try {
  $pdo = new PDO('mysql:host=localhost;dbname=meuBancoDeDados', $username, $password);
  $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

  $stmt = $pdo->prepare('DELETE FROM minhaTabela WHERE id = :id');
  $stmt->bindParam(':id', $id);
  $stmt->execute();

  echo $stmt->rowCount();
} catch(PDOException $e) {
  echo 'Error: ' . $e->getMessage();
}
?>
```

Como podemos ver estamos utilizando o bindParam e caso utilize ele sempre deve ser passado uma variável no segundo parâmetro, caso contrario terá uma mensagem de erro.

SELECT

Abaixo é possível ver um código simples de exemplo de como executar um select no banco de dados usando PDO.

Listagem 7: Select em PDO

```
<?php
$consulta = $pdo->query("SELECT nome, usuario FROM login;");


while ($linha = $consulta->fetch(PDO::FETCH_ASSOC)) {
    echo "Nome: {$linha['nome']} - Usuário: {$linha['usuario']}<br />";
}
?>
```

Conclusão

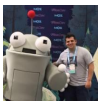
Neste artigo não abordei as conexões e nem a listagem de dados, focamos apenas em aprender como inserir, atualizar e excluir registros no banco de dados MySQL utilizando o PDO do PHP.

Se tiverem alguma dúvida, podem ficar a vontade em usar os comentários abaixo para perguntar que terei prazer em respondê-las.

Espero que tenham gostado e até o próximo artigo.

Veja também

- Introdução ao PHP PDO
- Usando PDO (PHP Data Objects) para aumentar a produtividade
- Biblioteca PDO - PHP



Ricardo Arrigoni

Cursando Marketing na UniCarioca, trabalhando com Marketing digital desde 2012 com especialização em SEO e links patrocinados. Profissional certificado Google Adwords Expert Search. Portfolio: http://www.ricardoarrigoni.com.br