



PHP

PDO (PHP Data Objects)

Maromo e Rodrigo

PDO em PHP

- Existem três API's de conexão com o banco de dados em PHP, são elas:
 - `mysql`: Pacote de funções para acesso ao MySQL, foi descontinuado no PHP7.
 - `mysqli`: Extensão da API `mysql` com suporte a funcionalidades adicionadas a versões posteriores ao MySQL 4.1 - <http://www.mysql.com/>
 - **PDO** - PHP Data Objects: Interface para acesso a dados do PHP.

API Robusta

- O PDO é uma API robusta que pode ser usada, independentemente do driver que você estiver usando.
- Muitas pessoas ficam um pouco intimidadas em usar o PDO de início, não por se tratar de uma API difícil de usar, muito pelo contrário, pois é muito fácil de usar, mas sim porque a API mysql é muito fácil de usar e as pessoas acabam se acostumando com ela.
- Mas vamos ver como realmente é fácil de usá-la.

Conexão com o Banco de Dados

- Primeiro vamos ver como se conectar usando o PDO.
- Nós vamos criar uma nova instância de classe e especificar o **driver** que vamos usar, no caso o **mysql**, o **nome do banco de dados**, **nome de usuário** e **senha**.

```
<?php
$conn = new
PDO('mysql:host=localhost;dbname=meuBancoDeDados',
    $username,
    $password);
```

Sobre a conexão

- Como vimos no código anterior, também é muito simples se conectar usando o PDO, mas como em toda conexão, é preciso tratar os erros, para se caso aconteça algum erro na conexão, mostrar isso ao usuário.

```
<?php
try {
    $conn = new
    PDO('mysql:host=localhost;dbname=meuBancoDeDados',
        $username,
        $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE,
    PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    echo 'ERROR: ' . $e->getMessage();
}
```

Sobre Erros

- O erro padrão do PDO é o `PDO::ERRMODE_SILENT`, mas no nosso código usamos o `PDO::ERRMODE_EXCEPTION` e abaixo vou listar as opções que temos:
 - `PDO::ERRMODE_SILENT`
 - `PDO::ERRMODE_WARNING`
 - `PDO::ERRMODE_EXCEPTION`

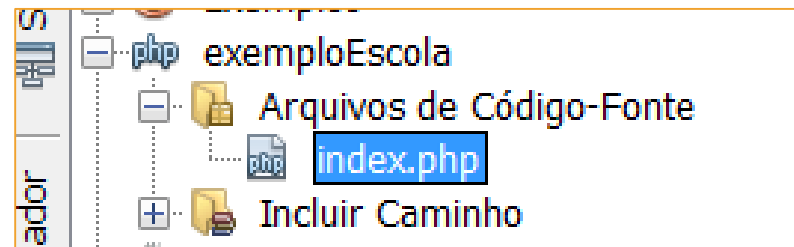
Banco de Dados na Prática

Banco: Escola



Aplicativo NoVo

Crie um novo aplicativo PHP nomeio como **exemploEscola**.



MySQL

Criar um Banco chamado escola

Em seguida criar uma tabela alunos, como abaixo:

```
CREATE TABLE `tb_alunos` (  
  `rm` int(11) NOT NULL,  
  `nome` varchar(60) DEFAULT NULL,  
  `email` varchar(60) DEFAULT NULL,  
  PRIMARY KEY (`rm`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Popule a tabela com alguns dados. Exemplo:

	rm	nome	email
	121	Marmo	maromo@gmail.com
	212	Rodrigo	rodrigo@gmail.com
»	NULL	NULL	

Arquivo conexao.php

```
<?php
2  $username='root';
3  $password='';
4  try {
5      $conn = new PDO('mysql:host=localhost;dbname=escola',
6                      $username,
7                      $password);
8      $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
9  } catch (PDOException $e) {
10     echo 'Erro ao conectar com o MySQL: ' . $e->getMessage();
11 }
```

Arquivo index.php

```
1  <?php
2  include_once './conexao.php';
3  ?>
4  <!DOCTYPE html>
5  <!--
6  Aulas professor Maromo.
7  -->
8  <html>
9      <head>
10         <meta charset="UTF-8">
11         <title>Apresentação de Conexão com PDO</title>
12     </head>
13     <body>
14         <?php
15             //Definir a codificação para UTF-8
16             $conn->exec("set names utf8");
17         ?>
18     </body>
19 </html>
```

Executando SELECT com PDO

Tendo a instância da classe PDO na variável \$conn, podemos realizar um simples SELECT desta forma:

```
<body>
    <?php
        //Definir a codificação para UTF-8
        $conn->exec("set names utf8");
        //Preparando o select
        $sql = "SELECT * FROM tb_alunos";
        $result = $conn->query($sql);
        $rows = $result->fetchAll( PDO::FETCH_ASSOC );
        //Exibindo o array associativo
        print_r($rows);
    ?>
```

Exemplo

```
Array
(
  [0] => Array
    (
      [rm] => 1
      [nome] => Maromo
      [email] => maromo@gmail.com
    )
  [1] => Array
    (
      [rm] => 2
      [nome] => Rodrigo
      [email] => rodrigo@gmail.com
    )
)
```

Fim

Obrigado
Maromo e Rodrigo