



**INSTITUTO
FEDERAL**

Sudeste de
Minas Gerais

Campus
Manhuaçu

Web III - PHP Data Object (PDO)

Prof. Leonardo C. R. Soares - leonardo.soares@ifsudestemg.edu.br

Instituto Federal do Sudeste de Minas Gerais

13 de dezembro de 2024





Autoload

Carregamento automático de arquivos fonte

Conforme o sistema vai crescendo, o número de arquivos que precisam ser incluídos cresce proporcionalmente. Para facilitar o controle da inclusão destes arquivos, podemos implementar o método `spl_autoload_register()`. Este método (que deverá ser adicionado ao script) irá permitir que classes e interfaces sejam automaticamente carregadas caso não estejam definidas.





Autoload

Carregamento automático de arquivos fonte

Conforme o sistema vai crescendo, o número de arquivos que precisam ser incluídos cresce proporcionalmente. Para facilitar o controle da inclusão destes arquivos, podemos implementar o método `spl_autoload_register()`. Este método (que deverá ser adicionado ao script) irá permitir que classes e interfaces sejam automaticamente carregadas caso não estejam definidas.

 autoload.php

```
1  <?php
2  spl_autoload_register(function ($class_name){
3      |      include $class_name . '.class.php';
4  });
5  ?>
```





Recuperando informações

Quando executamos uma sentença SQL que retorna dados, este resultado é encapsulado em um *result set*. Para recuperarmos estes dados do *result set*, podemos utilizar os métodos:

- ▶ **PDOStatement::fetch**: Recupera a próxima linha de um *result set*;
- ▶ **PDOStatement::fetchAll**: Recupera todas as linhas restantes de um *result set*;
- ▶ **PDOStatement::fetchColumn**: Recupera **uma** coluna da próxima linha de um *result set*;
- ▶ **PDOStatement::fetchObject**: Retorna a próxima linha de um *result set* como um objeto.





Recuperando informações

Ao utilizarmos os métodos `PDOStatement::fetch` ou `PDOStatement::fetchAll` devemos informar o modo como os dados serão retornados. Para isso utilizamos uma das constantes do `PDO::FETCH_*`:





Recuperando informações

Ao utilizarmos os métodos `PDOStatement::fetch` ou `PDOStatement::fetchAll` devemos informar o modo como os dados serão retornados. Para isso utilizamos uma das constantes do `PDO::FETCH_*`:

- ▶ `PDO::FETCH_ASSOC`: Retorna um vetor associativo pelo nome das colunas do *result set*;





Recuperando informações

Ao utilizarmos os métodos `PDOStatement::fetch` ou `PDOStatement::fetchAll` devemos informar o modo como os dados serão retornados. Para isso utilizamos uma das constantes do `PDO::FETCH_*`:

- ▶ `PDO::FETCH_ASSOC`: Retorna um vetor associativo pelo nome das colunas do *result set*;
- ▶ `PDO::FETCH_NUM`: Retorna um vetor indexado numericamente pelas colunas do *result set*, sendo a primeira coluna de índice 0;





Recuperando informações

Ao utilizarmos os métodos `PDOStatement::fetch` ou `PDOStatement::fetchAll` devemos informar o modo como os dados serão retornados. Para isso utilizamos uma das constantes do `PDO::FETCH_*`:

- ▶ `PDO::FETCH_ASSOC`: Retorna um vetor associativo pelo nome das colunas do *result set*;
- ▶ `PDO::FETCH_NUM`: Retorna um vetor indexado numericamente pelas colunas do *result set*, sendo a primeira coluna de índice 0;
- ▶ `PDO::FETCH_BOTH` (default): Retorna um vetor indexado associativamente e numericamente pelas colunas do *result set*;





Recuperando informações

Ao utilizarmos os métodos `PDOStatement::fetch` ou `PDOStatement::fetchAll` devemos informar o modo como os dados serão retornados. Para isso utilizamos uma das constantes do `PDO::FETCH_*`:

- ▶ `PDO::FETCH_ASSOC`: Retorna um vetor associativo pelo nome das colunas do *result set*;
- ▶ `PDO::FETCH_NUM`: Retorna um vetor indexado numericamente pelas colunas do *result set*, sendo a primeira coluna de índice 0;
- ▶ `PDO::FETCH_BOTH` (default): Retorna um vetor indexado associativamente e numericamente pelas colunas do *result set*;
- ▶ `PDO::FETCH_OBJ`: Retorna um objeto anônimo com propriedades que correspondem aos nomes das colunas do *result set*.

Vejam todos em <https://www.php.net/manual/en/pdostatement.fetch.php>





Exemplo

Para exemplificar, retomaremos o projeto da aula anterior. Iremos criar o autoload, que estará em um arquivo chamado *autoload.php*. A classe *Disciplina* será alterado. Criaremos propriedades para representar os campos da tabela, métodos seletores e um método que irá retornar todas as disciplinas cadastradas.





Exemplo

```
require_once "Escola.class.php";  
class Disciplina{  
    private int $id;  
    private string $descricao;  
    public function setId(int $id):void{  
        $this->id = $id;  
    }  
    public function setDescricao(string $descricao):void{  
        $this->descricao = $descricao;  
    }  
    public function getId():int {  
        return $this->id;  
    }  
    public function getDescricao():string{  
        return $this->descricao;  
    }  
}
```





Exemplo

```
public function inserir(string $desc):void{  
    try{  
        $banco = new Escola();  
        $con = $banco->conectar();  
        $SQL = "insert into disciplinas (id,descricao) values (null,?)";  
        $pst = $con->prepare($SQL);  
        $pst->bindParam(1, $desc);  
        $pst->execute();  
    }catch(PDOException $p){  
        throw new PDOException($p->getMessage());  
    }  
}
```





Exemplo

```
public static function getDisciplinas():array {  
    try {  
        $banco = new Escola();  
        $con = $banco->conectar();  
        $SQL = "select * from disciplinas";  
        $pst = $con->prepare($SQL);  
        $pst->execute();  
        $dados = $pst->fetchAll(PDO::FETCH_OBJ);  
        $disciplinas = array();  
        foreach($dados as $dado){  
            $d = new Disciplina();  
            $d->setId($dado->id);  
            $d->setDescricao($dado->descricao);  
            array_push($disciplinas,$d);  
        }  
        return $disciplinas;  
    } catch(PDOException $p){  
        throw new Exception("Atenção: ".$p->getMessage());  
    }  
}
```





Agora faremos uma pequena aplicação *front-end* para exibir as disciplinas cadastradas.

```
<?php
require_once "autoload.php";
?>
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        tr:nth-child(odd){background:#ccc}
        tr:nth-child(even){background:#fff}
    </style>
    <title>Listagem de Disciplinas</title>
</head>
<body>
    <?php
    try{
        $disciplinas = Disciplina::getDisciplinas();
    }catch(PDOException $p){
        echo "<p>Ocorreu um erro ao recuperar as disciplinas.</p>";
    }
    ?>
```





Front-end

```
<table>
  <tr><th>ID</th><th>Descrição</th></tr>
  <?php
    foreach($disciplinas as $disciplina){
      ?>
      <tr>
        <td><?=$disciplina->getId();?></td>
        <td><?=$disciplina->getDescricao();?></td>
      </tr>
    <?php
  }
  ?>
</table>
</body>
</html>
```





Dúvidas?



