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

7 de dezembro de 2024

PHP Data Object (PDO)

PHP e bancos de dados

Muitas aplicações PHP fazem persistência de informações em uma ampla variedade de banco de dados como MySQL, PostgreSQL, SQ-Lite, MSSQL, Oracle etc. Cada banco de dados ofere sua própria extensão PHP para o estabelecimento de comunicação entre o PHP e o banco de dados. O MySQL, por exemplo, utiliza a extensão mysqli, que adiciona diversas funções mysqli_*(). O MSSQL utiliza a extensão MSSQL que adiciona diversas funções mssql_*() etc.

PHP Data Object (PDO)

PHP e bancos de dados

Muitas aplicações PHP fazem persistência de informações em uma ampla variedade de banco de dados como MySQL, PostgreSQL, SQ-Lite, MSSQL, Oracle etc. Cada banco de dados ofere sua própria extensão PHP para o estabelecimento de comunicação entre o PHP e o banco de dados. O MySQL, por exemplo, utiliza a extensão mysqli, que adiciona diversas funções mysqli_*(). O MSSQL utiliza a extensão MSSQL que adiciona diversas funções mssql_*() etc.

E qual o problema com isso?

PHP Data Object (PDO)

PHP e bancos de dados

Muitas aplicações PHP fazem persistência de informações em uma ampla variedade de banco de dados como MySQL, PostgreSQL, SQ-Lite, MSSQL, Oracle etc. Cada banco de dados ofere sua própria extensão PHP para o estabelecimento de comunicação entre o PHP e o banco de dados. O MySQL, por exemplo, utiliza a extensão mysqli, que adiciona diversas funções mysqli_*(). O MSSQL utiliza a extensão MSSQL que adiciona diversas funções mssql_*() etc.

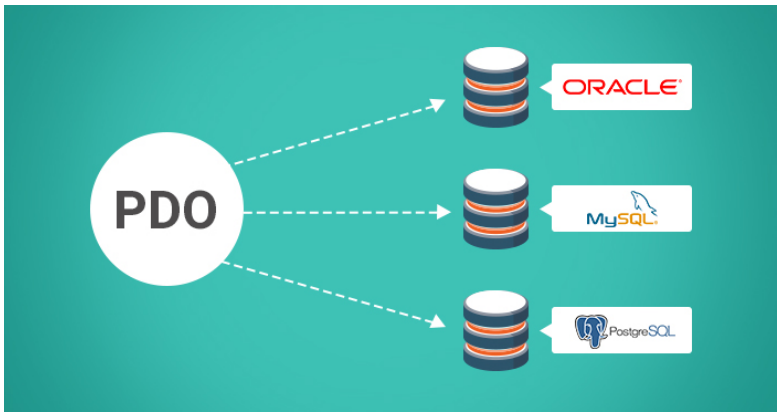
E qual o problema com isso?

Como corrigir?

PHP Data Object (PDO)

O PDO (PHP Data Objects) surgiu visando unificar o acesso às diferentes extensões de bancos de dados em uma API limpa e consistente.

O PDO não é uma biblioteca completa para abstração do acesso à base de dados visto que não faz a tradução das instruções SQL, adaptando-as aos mais diversos sistemas de banco de dados. Ela simplesmente unifica a chamada de métodos, delegando-os para as suas respectivas extensões.



PHP Data Object (PDO)

Para que um banco de dados seja utilizado é preciso que o PDO possua seu driver. Os seguintes drivers estão atualmente implementados:

Driver name	Supported databases
PDO_CUBRID	Cubrid
PDO_DBLIB	FreeTDS / Microsoft SQL Server / Sybase
PDO_FIREBIRD	Firebird
PDO_IBM	IBM DB2
PDO_INFORMIX	IBM Informix Dynamic Server
PDO_MYSQL	MySQL 3.x/4.x/5.x
PDO_OCI	Oracle Call Interface
PDO_ODBC	ODBC v3 (IBM DB2, unixODBC and win32 ODBC)
PDO_PGSQL	PostgreSQL
PDO_SQLITE	SQLite 3 and SQLite 2
PDO_SQLSRV	Microsoft SQL Server / SQL Azure
PDO_4D	4D

PHP Data Object (PDO)

O método construtor da classe PDO (*PDO::__construct*) possui a seguinte assinatura:

```
1   public PDO::__construct ( string $dsn [,
        string $username [, string $passwd [, array
         $options ]]] )
```

Não ocorrendo erros, o método construtor irá instanciar um objeto PDO que representará a conexão com o banco de dados requisitado.

PHP Data Object (PDO)

Parâmetros

dsn

O *Data Source Name* contém a informação requerida para conectar ao banco de dados. Em geral, um DSN consiste do nome do *driver* PDO seguido de dois pontos e de informações específicas para a conexão com aquele *driver*. A documentação completa pode ser vista em PDO driver-specific documentation.

PHP Data Object (PDO)

Parâmetros

dsn

O *Data Source Name* contém a informação requerida para conectar ao banco de dados. Em geral, um DSN consiste do nome do *driver* PDO seguido de dois pontos e de informações específicas para a conexão com aquele *driver*. A documentação completa pode ser vista em PDO driver-specific documentation.

username

O nome do usuário para a conexão com o banco de dados.

PHP Data Object (PDO)

Parâmetros

dsn

O *Data Source Name* contém a informação requerida para conectar ao banco de dados. Em geral, um DSN consiste do nome do *driver* PDO seguido de dois pontos e de informações específicas para a conexão com aquele *driver*. A documentação completa pode ser vista em PDO driver-specific documentation.

username

O nome do usuário para a conexão com o banco de dados.

passwd

A senha do usuário.

PHP Data Object (PDO)

Parâmetros

dsn

O *Data Source Name* contém a informação requerida para conectar ao banco de dados. Em geral, um DSN consiste do nome do *driver* PDO seguido de dois pontos e de informações específicas para a conexão com aquele *driver*. A documentação completa pode ser vista em PDO driver-specific documentation.

username

O nome do usuário para a conexão com o banco de dados.

passwd

A senha do usuário.

options

Um vetor de opções específicas para o *driver* selecionado.

Exemplo

```php
<?php
  $dsn = "mysql:host=localhost;dbname=banco";
  $username = "root";
  $password = "senhadoroot";
  $options = array (PDO::ATTR_ERRMODE => PDO::
      ERRMODE_EXCEPTION);
  try {
  $dbh = new PDO ($dsn, $username, $password);
    echo "Conectado com sucesso.";
  } catch (PDOException $p) {
    echo " Nao foi possivel efetuar a conexao :
        ". $p-> getMessage();
  }
?>
```

Projeto Exemplo

Faremos uma pequena aplicação para exemplificar a conexão com banco de dados e a inserção de dados em tabelas. O banco de dados que utilizaremos chama-se escola. Inicialmente, crie dentro deste banco de dados uma tabela chamada disciplinas. Esta tabela pode ser criada com o comando SQL: *CREATE TABLE disciplinas (id int NOT NULL AUTO_INCREMENT, descricao varchar(200) DEFAULT NULL, PRIMARY KEY (id));*

Classe para conexão com banco de dados

```php
<?php
class Escola {
    public function conectar():PDO {
        try {
            return new PDO("mysql:host=localhost;dbname=escola;","root","senhadoroot");
        }catch(PDOException $p){
            throw new PDOException($p->getMessage());
        }
    }
}
?>
```

Classe Disciplina

Agora criaremos uma classe para gerenciar as ações sobre a tabela de disciplinas. Inicialmente, escreveremos apenas o método responsável por inserir uma nova disciplina na tabela.

Classe para conexão com banco de dados

```php
<?php
require_once "Escola.class.php";
class Disciplina{
    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());
        }
    }
}
?>
```

Testando a aplicação

Para testar as classes implementadas, implemente um pequeno formulário HTML que permita ao usuário cadastrar algumas disciplinas.


рахмат
danke
謝謝
ngiyabonga
teşekkür ederim
Баярлалаа
спасибо
mersi
barka
welalin
tack
spas
kia ora
taafetai lava
vinaka
blagodaram
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
kiitos
dankie
nanni
nandri
dhanyavad
спасибі
thank you
gracias
хвала
asante
manana
obrigada
tenki
bedankt
bayarlalaa
gracie
hvala
mauruuru
köszönöm
dankon
aciu
akun
djiere dieuf
tau
mochchakkeram
murakoze
enkosi
dziękuję
chnorakaloutioun
gratias ago
gracies
sulpay
дякую
mamnun
go raibh maith agat
chokrane
sobodi
dėkuji
obrigado
sagolun
sukriya
kop khun krap
taiku
arigatô
takk
dakujem
trugarez
mèsi
didi madloba
najis tuke
ありがとう
tanemirt
rahmet
grazie
kam sah hamnida
rahmat
terima kasih
dhanyavadagalu
shukriya
merce
мерси
diolch
감사합니다
xièxie
ευχαριστώ
merci