

Definição

O (PDO - PHP Data Objects)

Define uma interface **leve e consistente** para acessar bancos de dados em PHP. Cada driver de banco de dados que implementa a interface PDO pode expor recursos específicos do banco de dados como funções de extensão regulares. Observe que você não pode executar nenhuma função do banco de dados usando a extensão PDO por si só; você deve usar um driver PDO específico do banco de dados para acessar um servidor de banco de dados.

O PDO fornece uma camada de abstração de acesso a dados, o que significa que, **independentemente do banco de dados** que você está usando, você usa as mesmas funções para emitir consultas e buscar dados. O PDO não fornece uma abstração de banco de dados; não reescreve o SQL ou emula os recursos ausentes. Você deve usar uma camada de abstração completa, se precisar desse recurso.

O PDO é fornecido com o **PHP 5.1** e está disponível como uma extensão PECL para o PHP 5.0; O PDO requer os novos recursos de OO no núcleo do PHP 5 e, portanto, não será executado com versões anteriores do PHP.

Não possui dependência, surgiu a necessidade de juntar o acesso de diversas extensões de banco de dados presentes na linguagem, com isso surgiu o PHP Data Object (PDO), que realiza também a abstração do banco de dados.

A sua **vantagem** está no objetivo de fornecer uma biblioteca limpa e consistente, para deixar unificadas as características das extensões que acessam os bancos de dados. Porém, o PDO possui também algumas desvantagens, por exemplo, não efetua a leitura e tradução das instruções SQL, é apenas realizada uma fusão dos métodos mandados para as extensões respectivas.

Características PDO

Flexibilidade – Como o PDO carrega o driver específico do banco de dados em tempo de execução, não é preciso reconfigurar o PHP sempre que um banco de dados diferente for usado.

Desempenho – O PDO está escrito em C e compilado no PHP, o que lhe garante um aumento considerável no desempenho em relação a soluções escritas em PHP.

Consistência de código – No PDO não existe a inconsistência de código, pois é oferecida apenas uma interface unificada que é está disponível para qualquer banco de dados.

Características de orientação de objetos – Possui recursos de orientação de objetos, o que resulta em uma comunicação mais poderosa e eficiente com banco de dados.

Instalação do PDO

O PDO vem junto com a versão 5.1 do PHP ou mais recente, por isso quem possuir essa versão terá apenas que efetuar uma configuração, adicionando a linha abaixo no arquivo php.ini.

Listagem 1: Habilitar suporte ao MySQL.

```
extension=php_pdo.dll  
extension=php_pdo_mysql.dll
```

Conexão para diversos bancos de dados

Para realizar a conexão com os bancos de dados, muda-se apenas a string da conexão. Abaixo alguns exemplos de conexão para alguns bancos de dados estão sendo demonstrados:

```
SQLite – new PDO('sqlite:nomeConexao.db');  
FireBird – new PDO("firebird:dbname=C:\\\\nomeBase.DB", "SYSDBA", "masterkey");  
MySQL – new PDO('mysql:host=localhost:port=3307;dbname=nomeBase', 'usuario', 'senha');  
Postgress – new PDO('pgsql:dbname=example;user=usuario;password=senha;host=localhost');
```

Opções relacionadas à conexão do PDO

Existem várias opções relacionadas à conexão com o banco no PDO, abaixo estão algumas:

PDO_ATTR_PERSISTENT: determina se a conexão é persistente, vem como padrão configurado como FALSE.

PDO_ATTR_PREFETCH: é um recurso do banco de dados que retorna várias linhas da consulta realizada, mesmo que o usuário tenha requisitado apenas uma linha, são automaticamente retornados mais linhas.

PDO_ATTR_CASE: pode converter as letras das colunas para maiúsculas, minúsculas ou estado natural (como vem do banco), podendo ser configurado os seguintes valores:

PDO_CASE_UPPER, PDO_CASE_LOWER ou PDO_CASE_NATURAL.

PDO_ATTR_EMULATE_PREPARES: essa função habilitada fornece a possibilidade que as prepared statements tirem vantagem do cache de query do MySQL.

PDO_DEFAULT_FETCH_MODE: configura o modo de fetching default(arrays associativos, arrays indexados ou objetos), o que pode facilitar na implementação de um método em particular.

PDO_ATTR_AUTOCOMMIT: determina se o PDO faz commit de cada query quando for executada, ou irá esperar que o método commit() seja executado antes de tornar as alterações efetivas.

PDO_ATTR_CONNECTION_STATUS: contém informações do banco de dados sobre o status da conexão, se foi bem sucedida ou não.

https://www.php.net/manual/pt_BR/book.pdo.php

<https://www.devmedia.com.br/introducao-ao-php-data-objects-pdo/25318>