← PHP 命名空间(namespace) PHP PDO预定义常量 →

PHP PDO

PHP 数据对象 （ PDO ） 扩展为PHP访问数据库定义了一个轻量级的一致接口。

PDO 提供了一个数据访问抽象层，这意味着，不管使用哪种数据库，都可以用相同的函数（方法）来查询和获取数据。

PDO随PHP5.1发行，在PHP5.0的PECL扩展中也可以使用，无法运行于之前的PHP版本。

PDO 安装

你可以通过 PHP 的 phpinfo() 函数来查看是否安装了PDO扩展。

在 Unix 系统上安装 PDO

在Unix上或Linux上你需要添加以下扩展：

```
extension=pdo.so
```

Windows 用户

PDO 和所有主要的驱动作为共享扩展随 PHP 一起发布，要激活它们只需简单地编辑 php.ini 文件，并添加以下扩展：

```
extension=php_pdo.dll
```

除此之外还有以下对应的各种数据库扩展：

```
;extension=php_pdo_firebird.dll
;extension=php_pdo_informix.dll
;extension=php_pdo_mssql.dll
;extension=php_pdo_mysql.dll
;extension=php_pdo_oci.dll
;extension=php_pdo_oci8.dll
;extension=php_pdo_odbc.dll
;extension=php_pdo_pgsql.dll
;extension=php_pdo_sqlite.dll
```

在设定好这些配置后，我们需要重启PHP 或 Web服务器。

接下来我们们来看下具体的实例，以下为使用PDO连接MySql数据库的实例：

```
<?php
$dbms='mysql';     //数据库类型
$host='localhost'; //数据库主机名
$dbName='test';    //使用的数据库
$user='root';      //数据库连接用户名
$pass='';          //对应的密码
```

```
$dsn="$dbms:host=$host;dbname=$dbName";


try {
    $dbh = new PDO($dsn, $user, $pass); //初始化一个PDO对象
    echo "连接成功<br/>";
    /*你还可以进行一次搜索操作
    foreach ($dbh->query('SELECT * from FOO') as $row) {
        print_r($row); //你可以用 echo($GLOBAL); 来看到这些值
    }
    */
    $dbh = null;
} catch (PDOException $e) {
    die ("Error!: " . $e->getMessage() . "<br/>");
}
//默认这个不是长连接，如果需要数据库长连接，需要最后加一个参数：array(PDO::ATTR_PERSISTENT => true) 变成这样：
$db = new PDO($dsn, $user, $pass, array(PDO::ATTR_PERSISTENT => true));

?>
```

很简单吧，接下来就让我们来具体看下PHP PDO具体说明：

- 预定义常量
- PHP PDO连接连接管理
- PHP PDO 事务与自动提交
- PHP PDO 预处理语句与存储过程
- PHP PDO 错误与错误处理
- PHP PDO 大对象 (LOBs)
- PDO 类：
    - PDO::beginTransaction — 启动一个事务
    - PDO::commit — 提交一个事务
    - PDO::__construct — 创建一个表示数据库连接的 PDO 实例
    - PDO::errorCode — 获取跟数据库句柄上一次操作相关的 SQLSTATE
    - PDO::errorInfo — 返回最后一次操作数据库的错误信息
    - PDO::exec — 执行一条 SQL 语句，并返回受影响的行数
    - PDO::getAttribute — 取回一个数据库连接的属性
    - PDO::getAvailableDrivers — 返回一个可用驱动的数组
    - PDO::inTransaction — 检查是否在一个事务内
    - PDO::lastInsertId — 返回最后插入行的ID或序列值

- PDO::prepare — 备要执行的SQL语句并返回一个 PDOStatement 对象
- PDO::query — 执行 SQL 语句，返回PDOStatement对象,可以理解为结果集
- PDO::quote — 为SQL语句中的字符串添加引号。
- PDO::rollBack — 回滚一个事务
- PDO::setAttribute — 设置属性

- PDOStatement 类：
  - PDOStatement::bindColumn — 绑定一列到一个 PHP 变量
  - PDOStatement::bindParam — 绑定一个参数到指定的变量名
  - PDOStatement::bindValue — 把一个值绑定到一个参数
  - PDOStatement::closeCursor — 关闭游标，使语句能再次被执行。
  - PDOStatement::columnCount — 返回结果集中的列数
  - PDOStatement::debugDumpParams — 打印一条 SQL 预处理命令
  - PDOStatement::errorCode — 获取跟上一次语句句柄操作相关的 SQLSTATE
  - PDOStatement::errorInfo — 获取跟上一次语句句柄操作相关的扩展错误信息
  - PDOStatement::execute — 执行一条预处理语句
  - PDOStatement::fetch — 从结果集中获取下一行
  - PDOStatement::fetchAll — 返回一个包含结果集中所有行的数组
  - PDOStatement::fetchColumn — 从结果集中的下一行返回单独的一列。
  - PDOStatement::fetchObject — 获取下一行并作为一个对象返回。
  - PDOStatement::getAttribute — 检索一个语句属性
  - PDOStatement::getColumnMeta — 返回结果集中一列的元数据
  - PDOStatement::nextRowset — 在一个多行集语句句柄中推进到下一个行集
  - PDOStatement::rowCount — 返回受上一个 SQL 语句影响的行数
  - PDOStatement::setAttribute — 设置一个语句属性
  - PDOStatement::setFetchMode — 为语句设置默认的获取模式。

← PHP 命名空间(namespace) PHP PDO预定义常量 →

点我分享笔记