

PHP 连接 MySQL

PHP 5 及以上版本建议使用以下方式连接 MySQL :

- **MySQLi extension** ("i" 意为 improved)
- **PDO (PHP Data Objects)**

在 PHP 早期版本中我们使用 MySQL 扩展。但该扩展在 2012 年开始不建议使用。

我是该用 MySQLi , 还是 PDO?

如果你需要一个简短的回答, 即 "你习惯哪个就用哪个"。

MySQLi 和 PDO 有它们自己的优势 :

PDO 应用在 12 种不同数据库中 , MySQLi 只针对 MySQL 数据库。

所以, 如果你的项目需要在多种数据库中切换, 建议使用 PDO , 这样你只需要修改连接字符串和部分查询语句即可。使用 MySQLi, 如果不同数据库, 你需要重新编写所有代码, 包括查询。

两者都是面向对象, 但 MySQLi 还提供了 API 接口。

两者都支持预处理语句。预处理语句可以防止 SQL 注入, 对于 web 项目的安全性是非常重要的。

MySQLi 和 PDO 连接 MySQL 实例

在本章节及接下来的章节中, 我们会使用以下三种方式来演示 PHP 操作 MySQL:

- MySQLi (面向对象)
- MySQLi (面向过程)
- PDO

MySQLi 安装

Linux 和 Windows: 在 php5 mysql 包安装时 MySQLi 扩展多数情况下是自动安装的。

安装详细信息, 请查看 : <http://php.net/manual/en/mysql.installation.php>

可以通过 phpinfo() 查看是否安装成功 :

mysqli		
Mysql Support	enabled	
Client API library version	mysqlnd 5.0.11-dev - 20120503 - \$Id: f373ea5dd5538761406a8022a4b8a374418b240e \$	
Active Persistent Links	0	
Inactive Persistent Links	0	
Active Links	0	
Directive	Local Value	Master Value
mysqli.allow_local_infile	On	On
mysqli.allow_persistent	On	On
mysqli.default_host		

PDO 安装

For 安装详细信息，请查看：<http://php.net/manual/en/pdo.installation.php>

可以通过 phpinfo() 查看是否安装成功：

PDO	
PDO support	enabled
PDO drivers	mysql, pgsql, sqlite
pdo_mysql	
PDO Driver for MySQL	enabled
Client API version	mysqlnd 5.0.11-dev - 20120503 - \$Id: f373ea5dd5538761406a8022a4b8a374418b240e \$

连接 MySQL

在我们访问 MySQL 数据库前，我们需要先连接到数据库服务器：

实例 (MySQLi - 面向对象)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// 创建连接
$conn = new mysqli($servername, $username, $password);
// 检测连接
if ($conn->connect_error) {
    die("连接失败: " . $conn->connect_error);
}
echo "连接成功";
?>
```

注意在以上面向对象的实例中 \$connect_error 是在 PHP 5.2.9 和 5.3.0 中添加的。如果你需要兼容更早版本 请使用以下代码替换：



```
// 检测连接
if (mysqli_connect_error()) {
    die("数据库连接失败: " . mysqli_connect_error());
}
```

实例 (MySQLi - 面向过程)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// 创建连接
$conn = mysqli_connect($servername, $username, $password);
// 检测连接
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
```

```
echo "连接成功";  
?>
```

实例 (PDO)

```
<?php  
$servername = "localhost";  
$username = "username";  
$password = "password";  
try {  
    $conn = new PDO("mysql:host=$servername;", $username, $password);  
    echo "连接成功";  
}  
catch(PDOException $e)  
{  
    echo $e->getMessage();  
}  
?>
```



注意在以上 PDO 实例中我们已经指定了数据库 (myDB)。PDO 在连接过程需要设置数据库名。如果没有指定，则会抛出异常。

关闭连接

连接在脚本执行完后会自动关闭。你也可以使用以下代码来关闭连接：

实例 (MySQLi - 面向对象)

```
$conn->close();
```

实例 (MySQLi - 面向过程)

```
mysqli_close($conn);
```

实例 (PDO)

```
$conn = null;
```

[← PHP MySQL 简介](#)[PHP MySQL 创建数据库 →](#)**1 篇笔记****写笔记**



我们可以用一下办法来测试PHP的MySQL数据库的连接。

使用函数 `parse_ini_file()` 解析配置文件 `config.ini` 来获得数据库连接参数，然后使用 `new` 关键字对 `mysqli` 类进行实例化，最后使用函数 `mysqli_connect_errno()` 来判断是否成功连接上了 MySQL 数据库，实现该过程的代码如下所示：

```
try{
    //解析config.ini文件
    $config = parse_ini_file(realpath(dirname(__FILE__) . '/config/config.ini'));
    //对mysqli类进行实例化
    $mysqli = new mysqli($config['host'], $config['username'], $config['password'], $config['dbname']);
    if(mysqli_connect_errno()){    //判断是否成功连接上MySQL数据库
        throw new Exception('数据库连接错误! ');    //如果连接错误，则抛出异常
    }else{
        echo '数据库连接成功! ';    //打印连接成功的提示
    }
}catch (Exception $e){    //捕获异常
    echo $e->getMessage();    //打印异常信息
}
```

kindyear 1年前 (2017-12-15)