

后盾网 人人做后盾

www.houdunwang.com

MySQLi扩展使用

后盾网 2011-2013 v3.0

Mysqli

- mysqli扩展允许我们访问MySQL 4.1及以上版本提供的功能
- 字母i寓意先进、快速
- Php5.0后为了适应MySQL的新特性，PHP重新编写了MySQLi扩展库
- MySQLi提供面向过程与面向对象两种接口
- 没有定义资源类型

运行环境要求

- 需要环境 php5.0以上 MySQL 4.1 以上版本，当然这已经不是问题了
- extension=php_mysqli.dll //在 php.ini 文件中启用这一行
- 通过phpinfo()查看是否成功加载
- Linux下可以使用源码方式安装或者直接使用yum -y install php-mysqli安装

了解与加载

MySQL服务器的连接

- 通过构造函数__construct连接
- mysqli::__construct() (主机, 用户, 密码, 数据库)
- mysqli::connect_errno 连接错误代码
- mysqli::connect_error 连接错误信息
- mysqli::errno 返回最近函数调用的错误代码
- mysqli::error 返回最后发生的错误

连接代码

- `$link = @new mysqli("127.0.0.1","root","", "demo");`
- `if($link->connect_errno){`
 `echo $link->connect_error;exit;`
- `}`

连接

发送查询

- `mysqli::query()`

获得受影响的记录数

- `mysqli::affected_rows`
- 此记录包含查询的条数这与mysql的有区别的，会得到CURD受影响的条数

返回自增的主键ID

- `Mysqli::insert_id`

查询

```
1. $mysqli = @new mysqli("127.0.0.1","root","","");
2. if($mysqli->connect_errno){
3.     echo $mysqli->connect_error;exit;
4. }
5. $mysqli->query("select * from news");
6. if($mysqli->affected_rows>0){
7.     echo "ok";
8. }
```

发送查询

`mysqli_result::fetch_array` 参数为返回结果的类型

- `MYSQLI_ASSOC` 返回关联数组
- `MYSQLI_NUM` 返回索引数组
- `MYSQLI_BOTH` 返回关联与索引数组

`mysqli_result::fetch_assoc()`

- 取得一行以关联数组形式返回的数据

`mysqli_result::fetch_row()`

- 取得一行以索引数组形式返回的数据

`mysqli_result::fetch_object()`

- 取得一行以对象形式返回的数据

查询结果集

`mysqli_result::num_rows`

- 返回结果集行数

`mysqli_result::free()`

- 从内存中释放结果集

查询结果集

```
1. header("Content-type:text/html;charset=utf-8");
2. $mysqli = @new mysqli("127.0.0.1","root","","hd");
3. if($mysqli->connect_errno){
4.     echo $mysqli->connect_error;exit;
• }
• $mysqli->query("SET NAMES utf8");
• $result = $mysqli->query("select username,tel from user");
• if($mysqli->affected_rows>0){
•     $rows = array();
•     while($row = $result->fetch_assoc()){
•         $rows[]=$row;
•     }
•     foreach($rows as $f){
•         echo "姓名: ".$f['username']." 电话:{$f['tel']}<br/>";
•     }
•     $result->free();
• }
```

执行查询

`mysqli_result::free()`

- 释放内存中的结果集

`mysqli::close()`

- 关闭一个打开的数据库连接

释放资源

```
1. $link = new mysqli("127.0.0.1", "root", "", "demo");
2.     if ($link->connect_errno) {
3.         echo $link->connect_error;
4.     }
5.     $link->query("set names utf8");
6.     if (!$result = $link->query("select * from hd_demo")) {
7.         echo $link->error;
8.     } else {
9.         echo "<table border=1>";
10.        while ($row = $result->fetch_object()) {
11.            printf("<tr> <td>%s</td> </tr>", $row->name);
12.        }
13.        echo "</table>";
14.    }
15. $result->free();
16. $link->close();
```

示例

预准备语句

- 先将SQL提交到MySQL服务器，预先解析，事后可以只传递参数到MySQL服务器，减少解析次数、提高MySQL处理效率
- 同时由于参数不是随着SQL一起发送到服务器端，所以屏蔽了SQL注入等等的危险操作

绑定参数

- 可以事先在预先解析的SQL上指定占位符？号来为后面传递的值预留空间位，可以把？想象成函数的形参这样更好理解

预准备查询

`mysqli_stmt::affected_rows`

- 返回受影响的行数

`mysqli_stmt::insert_id`

- 获得最后插入的自增主键值

`mysqli_stmt::errno`

- 返回最新的错误代码

`mysqli_stmt::error`

- 返回一个字符串，描述最后语句错误

属性

示例

```
• $link = new mysqli("127.0.0.1", "root", "", "demo");
•     if ($link->connect_errno) {
•         echo $link->connect_error;
•     }
•     $link->query("set names utf8");
•     $stmt = $link->prepare("insert hd_demo (name,sendtime) values(?,?)");
•     if (!$stmt) {
•         printf("<b>发生错误:</b>%s", $link->error);
•     } else {
•         $stmt->bind_param( "ss" , $name, $sendtime); //绑定参数
•     $arr = array("name" => "王五", "sendtime" => date("y/m/d h:i:s", time()));
•         extract($arr);
•         if ($stmt->execute()) {
•             printf("影响的条数%s<br/>", $stmt->affected_rows);
•             printf("最后插入的%s<br/>", $stmt->insert_id);
•         } else {
•             printf("错误代码:%s,<br/>错误内容:%s", $stmt->errno,$stmt->error);           }
•             $stmt->close();
•             $link->close();
•         }
•     }
```

通过预准备完成用户注册

作业
