**一、简介**

前面讲解的功能开发都是简单的调用API 完成的，没有对数据库进行操作。在接下来的高级功能开发中，需要使用到数据库，所以在这一篇中，将对MySQL 数据库的操作做一下简单的介绍，以供读者参考。

**二、思路分析**

百度开发者中心提供了强大的云数据库（包括MySQL, MongoDB, Redis），在这一节教程中，我们将对大家比较熟悉的MySQL 数据库进行操作演示，实现微信与数据库的交互。

在BAE应用中使用云数据库十分简单，数据库列表中的名称即是连接数据库时的dbname。用户名、密码、连接地址和端口在应用中通过环境变量取出。

可使用标准的PHP Mysql 或PHP Mysqli 扩展访问数据库，BAE的PHP中已提供这两个扩展，应用可直接使用。

官方文档，请参考：<http://developer.baidu.com/wiki/index.php?title=docs/cplat/rt/mysql>

**三、创建BAE MySQL数据库**

3.1 登陆百度开发者中心 -> 管理中心 -> 选择应用 -> 云环境 -> 服务管理 -> MySQL(云数据库) -> 创建数据库



3.2 创建数据库



**注意：**每个应用有且只有一个数据库享受1G免费配额，其余数据库均不享受免费配额优惠。只有将已使用免费配额的数据库删除，才能再次使用此项优惠。

3.3 创建成功

在这里可以看到数据库的名称，也就是dbname，后面会使用到。

点击 “phpMyadmin” 访问数据库。



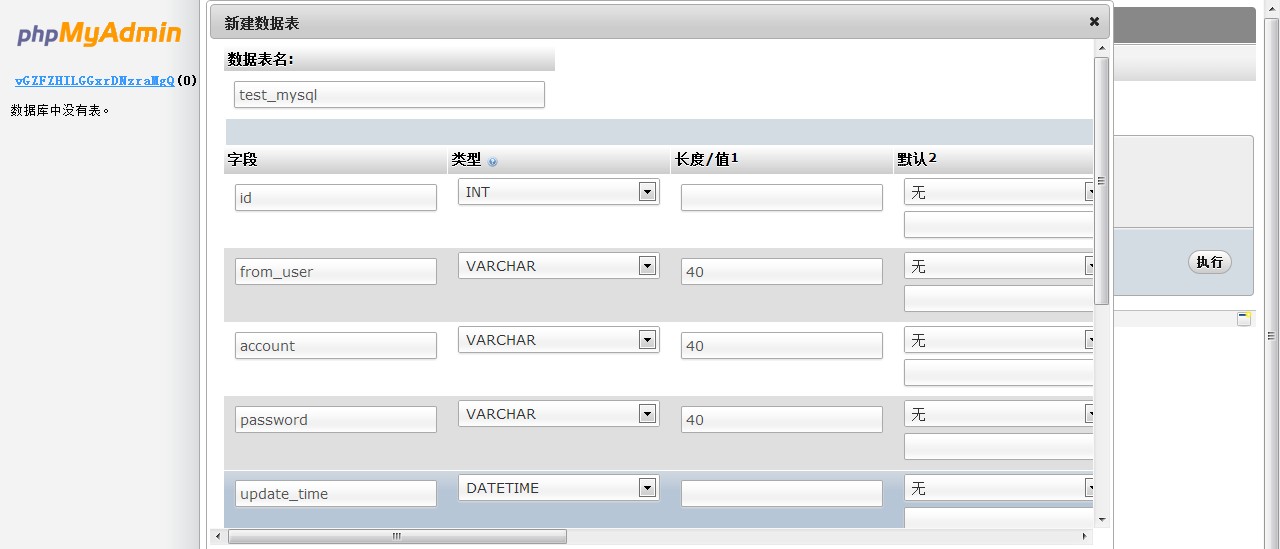
3.4 phpMyadmin界面

新建数据表，输入表名及字段数，点击 “执行” 创建表。



3.5 创建表

输入字段名及字段类型，输入完毕后，点击下面的“保存”，完成表的创建。



3.6 创建完成

修改id 字段为主键并添加AUTO\_INCREMENT；修改from\_user 字段为唯一（UNIQUE），完成表的修改。



建表操作也可以使用以下SQL语句完成：

[复制代码](javascript:void(0);)

CREATE TABLE IF NOT EXISTS `test\_mysql` (

`id` int(**11**) NOT NULL AUTO\_INCREMENT,

`from\_user` varchar(**40**) DEFAULT NULL,

`account` varchar(**40**) DEFAULT NULL,

`password` varchar(**40**) DEFAULT NULL,

`update\_time` datetime DEFAULT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `from\_user` (`from\_user`)

);

[复制代码](javascript:void(0);)

phpMyAdmin 操作



数据库及数据表的创建到此结束，下面将编写代码对数据库及数据表的使用做详细讲解。

**四、官方示例（PHP MySQL）**

BAE 官方提供的demo（PHP MySQL）示例如下：

mysql/basic.php 文件内容

[复制代码](javascript:void(0);)

<?php

/\*\*

\* MySQL示例，通过该示例可熟悉BAE平台MySQL的使用（CRUD）

\*/

require\_once("../configure.php");

/\*替换为你自己的数据库名（可从管理中心查看到）\*/

$dbname = MYSQLNAME;

/\*从环境变量里取出数据库连接需要的参数\*/

$host = getenv('HTTP\_BAE\_ENV\_ADDR\_SQL\_IP');

$port = getenv('HTTP\_BAE\_ENV\_ADDR\_SQL\_PORT');

$user = getenv('HTTP\_BAE\_ENV\_AK');

$pwd = getenv('HTTP\_BAE\_ENV\_SK');

/\*接着调用mysql\_connect()连接服务器\*/

$link = @mysql\_connect("{$host}:{$port}",$user,$pwd,true);

if(!$link) {

die("Connect Server Failed: " . mysql\_error());

}

/\*连接成功后立即调用mysql\_select\_db()选中需要连接的数据库\*/

if(!mysql\_select\_db($dbname,$link)) {

die("Select Database Failed: " . mysql\_error($link));

}

/\*至此连接已完全建立，就可对当前数据库进行相应的操作了\*/

/\*！！！注意，无法再通过本次连接调用mysql\_select\_db来切换到其它数据库了！！！\*/

/\* 需要再连接其它数据库，请再使用mysql\_connect+mysql\_select\_db启动另一个连接\*/

/\*\*

\* 接下来就可以使用其它标准php mysql函数操作进行数据库操作

\*/

//创建一个数据库表

$sql = "create table if not exists test\_mysql(

id int primary key auto\_increment,

no int,

name varchar(1024),

key idx\_no(no))";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Create Table Failed: " . mysql\_error($link));

} else {

echo "Create Table Succeed<br />";

}

//插入数据

$sql = "insert into test\_mysql(no, name) values(2007,'this is a test message'),

(2008,'this is another test message'),

(2009,'xxxxxxxxxxxxxx')";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Insert Failed: " . mysql\_error($link));

} else {

echo "Insert Succeed<br />";

}

//删除数据

$sql = "delete from test\_mysql where no = 2008";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Delete Failed: " . mysql\_error($link));

} else {

echo "Delete Succeed<br />";

}

//修改数据

$sql = "update test\_mysql set name = 'yyyyyy' where no = 2009";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Update Failed: " . mysql\_error($link));

} else {

echo "Update Succeed<br />";

}

//检索数据

$sql = "select id,no,name from test\_mysql";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Select Failed: " . mysql\_error($link));

} else {

echo "Select Succeed<br />";

while ($row = mysql\_fetch\_assoc($ret)) {

echo "{$row['id']} {$row['no']} {$row['name']}<br />";

}

}

//删除表

$sql = "drop table if exists test\_mysql";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Drop Table Failed: " . mysql\_error($link));

} else {

echo "Drop Table Succeed<br />";

}

?>

[复制代码](javascript:void(0);)

configure.php 文件内容

[复制代码](javascript:void(0);)

<?php

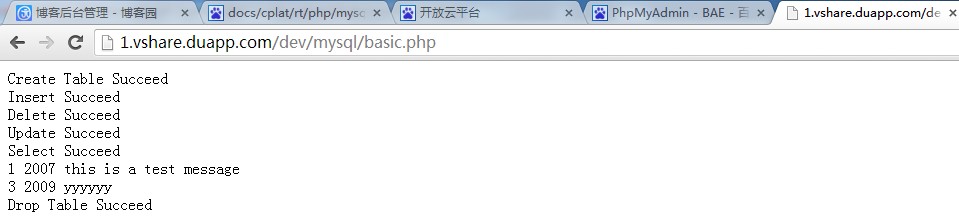
/\*\*\*配置数据库名称\*\*\*/

define("MYSQLNAME", "qzMlSkByflhScPCOFtax");

?>

[复制代码](javascript:void(0);)

测试使用：



执行成功。

**五、修改成可调用的函数形式（PHP MySQL）**

5.1 创建数据表

//创建一个数据库表

function \_create\_table($sql){

mysql\_query($sql) or die('创建表失败，错误信息：'.mysql\_error());

return "创建表成功";

}

5.2 插入数据

[复制代码](javascript:void(0);)

//插入数据

function \_insert\_data($sql){

if(!mysql\_query($sql)){

return 0; //插入数据失败

}else{

if(mysql\_affected\_rows()>0){

return 1; //插入成功

}else{

return 2; //没有行受到影响

}

}

}

[复制代码](javascript:void(0);)

5.3 删除数据

[复制代码](javascript:void(0);)

//删除数据

function \_delete\_data($sql){

if(!mysql\_query($sql)){

return 0; //删除失败

}else{

if(mysql\_affected\_rows()>0){

return 1; //删除成功

}else{

return 2; //没有行受到影响

}

}

}

[复制代码](javascript:void(0);)

5.4 修改数据

[复制代码](javascript:void(0);)

//修改数据

function \_update\_data($sql){

if(!mysql\_query($sql)){

return 0; //更新数据失败

}else{

if(mysql\_affected\_rows()>0){

return 1; //更新成功;

}else{

return 2; //没有行受到影响

}

}

}

[复制代码](javascript:void(0);)

5.5 检索数据

//检索数据

function \_select\_data($sql){

$ret = mysql\_query($sql) or die('SQL语句有错误，错误信息：'.mysql\_error());

return $ret;

}

5.6 删除数据表

//删除表

function \_drop\_table($sql){

mysql\_query($sql) or die('删除表失败，错误信息：'.mysql\_error());

return "删除表成功";

}

将以上函数和连接数据库的代码结合起来，生成mysql\_bae.func.php 文件，供下面测试使用。

**六、测试MySQL 函数使用**

6.1 新建文件dev\_mysql.php 在同一目录下并引入mysql\_bae.func.php 文件

require\_once './mysql\_bae.func.php';

6.2 测试创建表

将上面使用phpMyAdmin 创建的test\_mysql 表删除，测试语句如下：

[复制代码](javascript:void(0);)

//创建表

$create\_sql = "CREATE TABLE IF NOT EXISTS `test\_mysql` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`from\_user` varchar(40) DEFAULT NULL,

`account` varchar(40) DEFAULT NULL,

`password` varchar(40) DEFAULT NULL,

`update\_time` datetime DEFAULT NULL,

PRIMARY KEY (`id`),

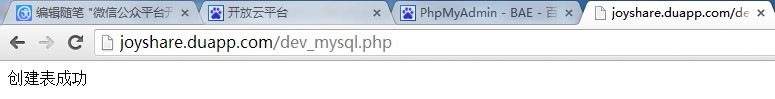
UNIQUE KEY `from\_user` (`from\_user`)

)";

echo \_create\_table($create\_sql);

[复制代码](javascript:void(0);)

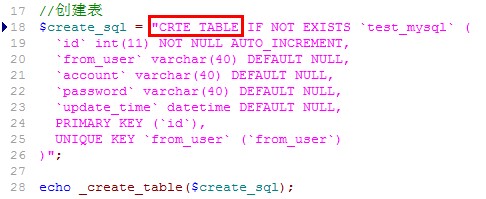
测试正确结果：



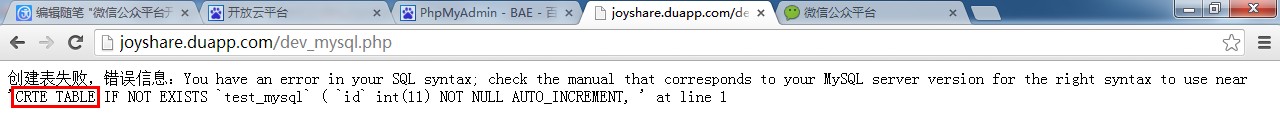
到phpMyAdmin中查看



故意将SQL语句写错



测试错误结果：



6.3 测试插入数据

测试语句如下：

[复制代码](javascript:void(0);)

//插入数据

$insert\_sql = "insert into test\_mysql(from\_user, account, password, update\_time) values('David','860510', 'abcabc', '2013-09-29 17:14:28')";

$res = \_insert\_data($insert\_sql);

if($res == 1){

echo "插入成功";

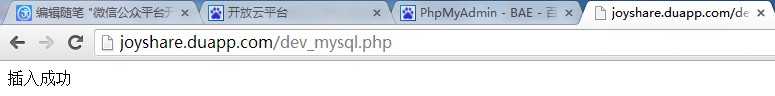
}else{

echo "插入失败";

}

[复制代码](javascript:void(0);)

测试结果：



6.4 测试更新数据

测试语句如下：

[复制代码](javascript:void(0);)

//更新数据

$update\_sql = "update test\_mysql set account = 860512 where account = 860510";

$res = \_update\_data($update\_sql);

if($res == 1){

echo "更新成功";

}elseif($res == 0){

echo "更新失败";

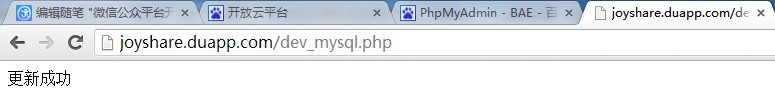
}elseif($res == 2){

echo "没有行受到影响";

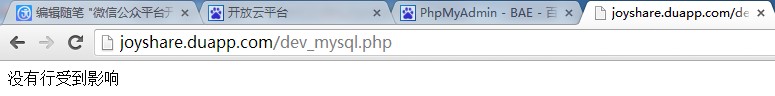
}

[复制代码](javascript:void(0);)

测试结果：



再次更新：



6.5 测试删除数据

测试语句如下：

[复制代码](javascript:void(0);)

//删除数据

$delete\_sql = "delete from test\_mysql where account = 860512";

$res = \_delete\_data($delete\_sql);

if($res == 1){

echo "删除成功";

}elseif($res == 0){

echo "删除失败";

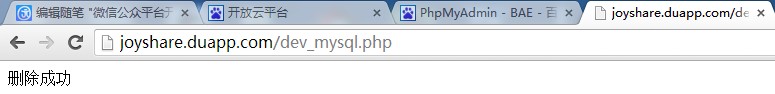
}elseif($res == 2){

echo "没有该条记录";

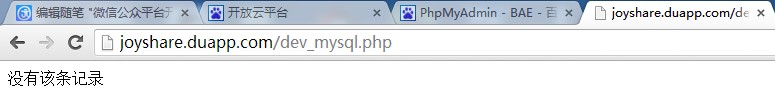
}

[复制代码](javascript:void(0);)

测试结果：



再次删除：



6.6 测试检索数据

再次执行上面的插入操作做检索测试，测试语句如下：

[复制代码](javascript:void(0);)

//检索数据

$select\_sql = "select \* from test\_mysql";

$result = \_select\_data($select\_sql);

while($rows = mysql\_fetch\_array($result,MYSQL\_ASSOC)){

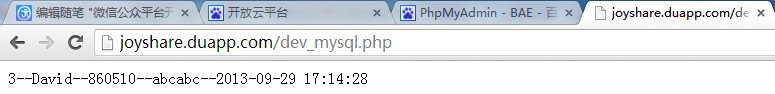
echo $rows[id]."--".$rows[from\_user]."--".$rows[account]."--".$rows[password]."--".$rows[update\_time];

echo "<br />";

}

[复制代码](javascript:void(0);)

测试结果：



6.7 测试删除表

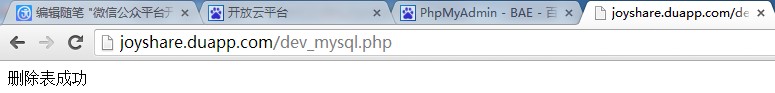
测试语句如下：

//删除表

$drop\_sql = "drop table if exists test\_mysql";

echo \_drop\_table($drop\_sql);

测试结果：



MySQL 函数测试全部成功。

**七、实现与微信的交互（Mysql 扩展）**

保证数据库中存在test\_msyql表，这里测试微信对MySQL数据库的增删改查操作，不考虑特殊情况，只按照下面的方法测试：

[复制代码](javascript:void(0);)

1. 绑定+账户+密码

如：绑定+860512+abc123

2. 查询

如：查询

3. 修改+旧密码+新密码

如：修改+abc123+123456

4. 删除

如：删除

[复制代码](javascript:void(0);)

7.1 引入mysql\_bae.func.php 文件

//引入数据库函数文件

require\_once 'mysql\_bae.func.php';

7.2 前置操作

A. 将输入的语句拆分成数组，以“+”号分隔

$keywords = explode("+",$keyword);

B. 获取当前时间

//获取当前时间

$nowtime=date("Y-m-d G:i:s");

C. 判断用户是否已经绑定

[复制代码](javascript:void(0);)

//判断是否已经绑定

$select\_sql="SELECT id from test\_mysql WHERE from\_user='$fromUsername'";

$res=\_select\_data($select\_sql);

$rows=mysql\_fetch\_array($res, MYSQL\_ASSOC);

if($rows[id] <> ''){

$user\_flag='y';

}

[复制代码](javascript:void(0);)

7.3 测试插入操作

测试代码：

[复制代码](javascript:void(0);)

if(trim($keywords[0] == '绑定')){

if($user\_flag <> 'y'){

$insert\_sql="INSERT INTO test\_mysql(from\_user, account, password, update\_time) VALUES('$fromUsername','$keywords[1]','$keywords[2]','$nowtime')";

$res = \_insert\_data($insert\_sql);

if($res == 1){

$contentStr = "绑定成功";

}elseif($res == 0){

$contentStr = "绑定失败";

}

}else{

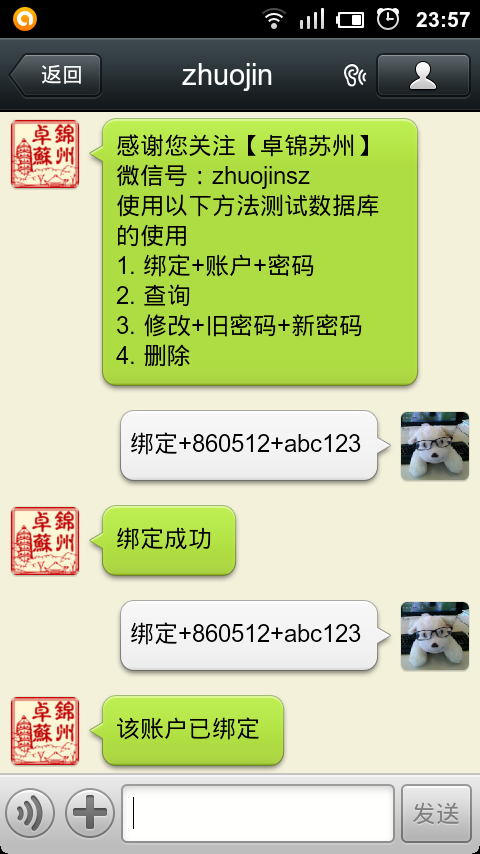
$contentStr = "该账户已绑定";

}

}

[复制代码](javascript:void(0);)

测试结果：



7.4 测试查询操作

测试代码：

[复制代码](javascript:void(0);)

if(trim($keywords[0] == '查询')){

$select\_sql="SELECT \* FROM test\_mysql WHERE from\_user='$fromUsername'";

$select\_res=\_select\_data($select\_sql);

$rows=mysql\_fetch\_assoc($select\_res);

if($rows[id] <> ''){

$contentStr="账户:$rows[account]\n"."密码：$rows[password]\n"."From\_user：$rows[from\_user]\n"."更新时间：$rows[update\_time]";

}else{

$contentStr="您还未绑定账户，查询不到相关信息，请先绑定，谢谢！";

}

}

[复制代码](javascript:void(0);)

测试结果：



7.5 测试更新操作

测试代码：

[复制代码](javascript:void(0);)

if(trim($keywords[0] == "修改")){

$old\_password=$keywords[1];

$new\_password=$keywords[2];

$select\_password\_sql="SELECT \* FROM test\_mysql WHERE from\_user='$fromUsername'";

$select\_res=\_select\_data($select\_password\_sql);

$rows=mysql\_fetch\_assoc($select\_res);

if($old\_password == $rows[password]){

$update\_sql="UPDATE test\_mysql SET password='$new\_password' WHERE from\_user='$fromUsername'";

$res = \_update\_data($update\_sql);

if($res == 1){

$contentStr = "修改成功";

}elseif($res == 0){

$contentStr = "修改失败";

}

}else{

$contentStr = "原密码有误，请确认后重试";

}

}

[复制代码](javascript:void(0);)

测试结果：



7.6 测试删除操作

测试代码：

[复制代码](javascript:void(0);)

if(trim($keywords[0] == "删除")){

$delete\_sql="DELETE FROM test\_mysql WHERE from\_user='$fromUsername'";

$res = \_delete\_data($delete\_sql);

if($res == 1){

$contentStr = "删除成功";

}elseif($res == 0){

$contentStr = "删除失败";

}

}

[复制代码](javascript:void(0);)

测试结果：



与微信的交互测试成功。

**八、PHP Mysqli 扩展，封装成类**

将Mysqli 扩展封装成类使用，代码如下：

[复制代码](javascript:void(0);)

<?php

require\_once 'includes/configure.php';

class MySQLi\_BAE{

private $mysqli;

private $host;

private $user;

private $password;

private $port;

private $database;

//在类之外访问私有变量时使用

function \_\_get($property\_name){

if(isset($this->$property\_name)){

return($this->$property\_name);

}else{

return(NULL);

}

}

function \_\_set($property\_name, $value){

$this->$property\_name=$value;

}

function \_\_construct(){

/\*从平台获取查询要连接的数据库名称\*/

$this->database = MYSQLNAME;

/\*从环境变量里取出数据库连接需要的参数\*/

$this->host = getenv('HTTP\_BAE\_ENV\_ADDR\_SQL\_IP');

$this->user = getenv('HTTP\_BAE\_ENV\_AK');

$this->password = getenv('HTTP\_BAE\_ENV\_SK');

$this->port = getenv('HTTP\_BAE\_ENV\_ADDR\_SQL\_PORT');

$this->mysqli = new mysqli($this->host, $this->user, $this->password, $this->database, $this->port);

if($this->mysqli->connect\_error){

die("Connect Server Failed:".$this->mysqli->error);

}

$this->mysqli->query("set names utf8");

}

//dql statement

function execute\_dql($query){

$res = $this->mysqli->query($query) or die("操作失败".$this->mysqli->error);

return $res;

//$this->mysqli->close();

}

//dml statement

function execute\_dml($query){

$res = $this->mysqli->query($query) or die("操作失败".$this->mysqli->error);

if(!$res){

return 0;//失败

}else{

if($this->mysqli->affected\_rows > 0){

return 1;//执行成功

}else{

return 2;//没有行受影响

}

}

//$this->mysqli->close();

}

}

?>

[复制代码](javascript:void(0);)

**九、测试类的使用**

9.1 测试DML操作

测试代码：

[复制代码](javascript:void(0);)

<?php

require\_once "MySQLi\_BAE.class.php";

$mysqli\_BAE=new MySQLi\_BAE();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*dml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$sql="insert into test\_mysql (from\_user, account, password, update\_time) values('David','860510', 'abcabc', '2013-09-27 17:14:28')";

//$sql="update test\_mysql set account = 860512 where account = 860510";

//$sql="delete from test\_mysql where account = 860512";

$res=$mysqli\_BAE->execute\_dml($sql);

if($res==0){

echo "执行失败";

}elseif($res==1){

echo "执行成功";

}else{

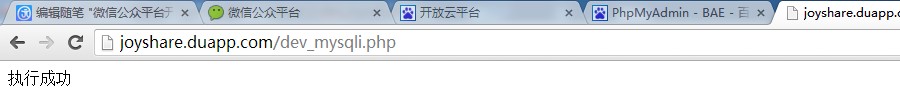
echo "没有行数影响";

}

?>

[复制代码](javascript:void(0);)

测试结果：



9.2 测试DQL操作

测试代码：

[复制代码](javascript:void(0);)

<?php

require\_once "MySQLi\_BAE.class.php";

$mysqli\_BAE=new MySQLi\_BAE();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*dql\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$sql="select \* from test\_mysql";

$res=$mysqli\_BAE->execute\_dql($sql);

while($row=$res->fetch\_row()){

foreach($row as $key=>$val){

echo "$val--";

}

echo '<br/>';

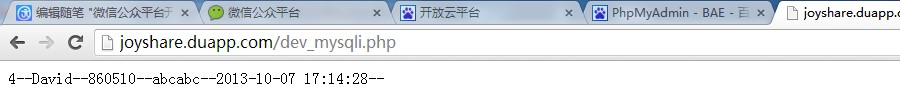
}

$res->free();

?>

[复制代码](javascript:void(0);)

测试结果：



**十、实现与微信的交互（Mysqli 扩展）**

10.1 前置操作

A. 引入MySQLi\_BAE.class.php 文件

//引入数据库函数文件

require\_once "MySQLi\_BAE.class.php";

B. 实例化对象

public function \_\_construct()

{

$this->mysqli\_BAE=new MySQLi\_BAE();

}

10.2 测试插入操作

测试代码：

$insert\_sql="INSERT INTO test\_mysql(from\_user, account, password, update\_time) VALUES('$fromUsername','$keywords[1]','$keywords[2]','$nowtime')";

$res = $this->mysqli\_BAE->execute\_dml($insert\_sql);

测试结果：



10.3 测试查询操作

测试代码：

$select\_sql="SELECT \* FROM test\_mysql WHERE from\_user='$fromUsername'";

$select\_res=$this->mysqli\_BAE->execute\_dql($select\_sql);

$rows=$select\_res->fetch\_array(MYSQLI\_ASSOC);

测试结果：



10.4 测试更新操作

测试代码：

$update\_sql="UPDATE test\_mysql SET password='$new\_password' WHERE from\_user='$fromUsername'";   
$res = $this->mysqli\_BAE->execute\_dml($update\_sql);

测试结果：



10.5 测试删除操作

测试代码：

$delete\_sql="DELETE FROM test\_mysql WHERE from\_user='$fromUsername'";

$res = $this->mysqli\_BAE->execute\_dml($delete\_sql);

测试结果：



与微信交互测试成功。

**十一、完整代码获取**

请访问 [**乐思乐享**](http://pan.baidu.com/s/1c0s3Jby) 官方论坛

**URL**：<http://pan.baidu.com/s/1c0s3Jby>

**一、简介**

前面讲解的功能开发都是简单的调用API 完成的，没有对数据库进行操作。在接下来的高级功能开发中，需要使用到数据库，所以在这一篇中，将对MySQL 数据库的操作做一下简单的介绍，以供读者参考。

**二、思路分析**

百度开发者中心提供了强大的云数据库（包括MySQL, MongoDB, Redis），在这一节教程中，我们将对大家比较熟悉的MySQL 数据库进行操作演示，实现微信与数据库的交互。

在BAE应用中使用云数据库十分简单，数据库列表中的名称即是连接数据库时的dbname。用户名、密码、连接地址和端口在应用中通过环境变量取出。

可使用标准的PHP Mysql 或PHP Mysqli 扩展访问数据库，BAE的PHP中已提供这两个扩展，应用可直接使用。

官方文档，请参考：<http://developer.baidu.com/wiki/index.php?title=docs/cplat/rt/mysql>

**三、创建BAE MySQL数据库**

3.1 登陆百度开发者中心 -> 管理中心 -> 选择应用 -> 云环境 -> 服务管理 -> MySQL(云数据库) -> 创建数据库



3.2 创建数据库



**注意：**每个应用有且只有一个数据库享受1G免费配额，其余数据库均不享受免费配额优惠。只有将已使用免费配额的数据库删除，才能再次使用此项优惠。

3.3 创建成功

在这里可以看到数据库的名称，也就是dbname，后面会使用到。

点击 “phpMyadmin” 访问数据库。



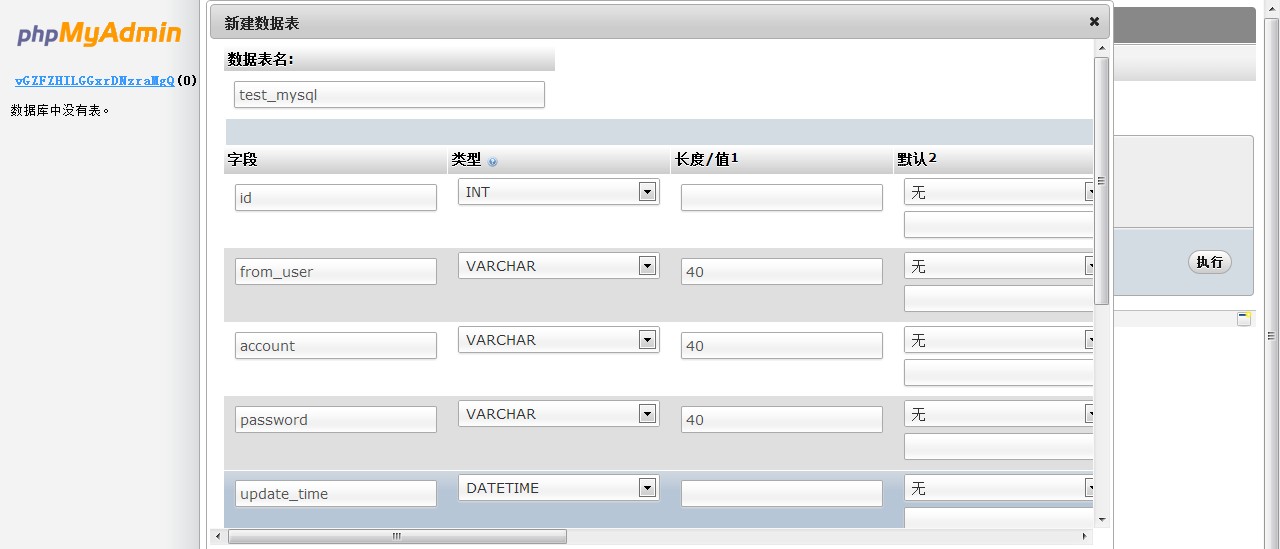
3.4 phpMyadmin界面

新建数据表，输入表名及字段数，点击 “执行” 创建表。



3.5 创建表

输入字段名及字段类型，输入完毕后，点击下面的“保存”，完成表的创建。



3.6 创建完成

修改id 字段为主键并添加AUTO\_INCREMENT；修改from\_user 字段为唯一（UNIQUE），完成表的修改。



建表操作也可以使用以下SQL语句完成：

[复制代码](javascript:void(0);)

CREATE TABLE IF NOT EXISTS `test\_mysql` (

`id` int(**11**) NOT NULL AUTO\_INCREMENT,

`from\_user` varchar(**40**) DEFAULT NULL,

`account` varchar(**40**) DEFAULT NULL,

`password` varchar(**40**) DEFAULT NULL,

`update\_time` datetime DEFAULT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `from\_user` (`from\_user`)

);

[复制代码](javascript:void(0);)

phpMyAdmin 操作



数据库及数据表的创建到此结束，下面将编写代码对数据库及数据表的使用做详细讲解。

**四、官方示例（PHP MySQL）**

BAE 官方提供的demo（PHP MySQL）示例如下：

mysql/basic.php 文件内容

[复制代码](javascript:void(0);)

<?php

/\*\*

\* MySQL示例，通过该示例可熟悉BAE平台MySQL的使用（CRUD）

\*/

require\_once("../configure.php");

/\*替换为你自己的数据库名（可从管理中心查看到）\*/

$dbname = MYSQLNAME;

/\*从环境变量里取出数据库连接需要的参数\*/

$host = getenv('HTTP\_BAE\_ENV\_ADDR\_SQL\_IP');

$port = getenv('HTTP\_BAE\_ENV\_ADDR\_SQL\_PORT');

$user = getenv('HTTP\_BAE\_ENV\_AK');

$pwd = getenv('HTTP\_BAE\_ENV\_SK');

/\*接着调用mysql\_connect()连接服务器\*/

$link = @mysql\_connect("{$host}:{$port}",$user,$pwd,true);

if(!$link) {

die("Connect Server Failed: " . mysql\_error());

}

/\*连接成功后立即调用mysql\_select\_db()选中需要连接的数据库\*/

if(!mysql\_select\_db($dbname,$link)) {

die("Select Database Failed: " . mysql\_error($link));

}

/\*至此连接已完全建立，就可对当前数据库进行相应的操作了\*/

/\*！！！注意，无法再通过本次连接调用mysql\_select\_db来切换到其它数据库了！！！\*/

/\* 需要再连接其它数据库，请再使用mysql\_connect+mysql\_select\_db启动另一个连接\*/

/\*\*

\* 接下来就可以使用其它标准php mysql函数操作进行数据库操作

\*/

//创建一个数据库表

$sql = "create table if not exists test\_mysql(

id int primary key auto\_increment,

no int,

name varchar(1024),

key idx\_no(no))";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Create Table Failed: " . mysql\_error($link));

} else {

echo "Create Table Succeed<br />";

}

//插入数据

$sql = "insert into test\_mysql(no, name) values(2007,'this is a test message'),

(2008,'this is another test message'),

(2009,'xxxxxxxxxxxxxx')";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Insert Failed: " . mysql\_error($link));

} else {

echo "Insert Succeed<br />";

}

//删除数据

$sql = "delete from test\_mysql where no = 2008";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Delete Failed: " . mysql\_error($link));

} else {

echo "Delete Succeed<br />";

}

//修改数据

$sql = "update test\_mysql set name = 'yyyyyy' where no = 2009";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Update Failed: " . mysql\_error($link));

} else {

echo "Update Succeed<br />";

}

//检索数据

$sql = "select id,no,name from test\_mysql";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Select Failed: " . mysql\_error($link));

} else {

echo "Select Succeed<br />";

while ($row = mysql\_fetch\_assoc($ret)) {

echo "{$row['id']} {$row['no']} {$row['name']}<br />";

}

}

//删除表

$sql = "drop table if exists test\_mysql";

$ret = mysql\_query($sql, $link);

if ($ret === false) {

die("Drop Table Failed: " . mysql\_error($link));

} else {

echo "Drop Table Succeed<br />";

}

?>

[复制代码](javascript:void(0);)

configure.php 文件内容

[复制代码](javascript:void(0);)

<?php

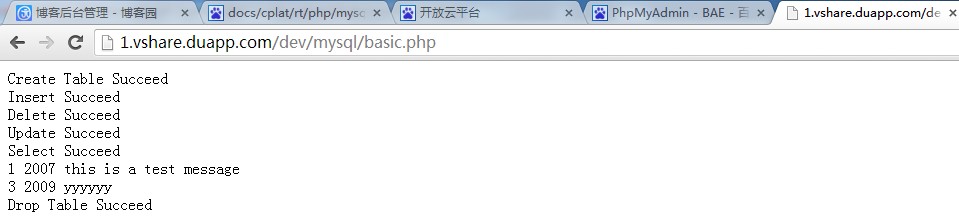
/\*\*\*配置数据库名称\*\*\*/

define("MYSQLNAME", "qzMlSkByflhScPCOFtax");

?>

[复制代码](javascript:void(0);)

测试使用：



执行成功。

**五、修改成可调用的函数形式（PHP MySQL）**

5.1 创建数据表

//创建一个数据库表

function \_create\_table($sql){

mysql\_query($sql) or die('创建表失败，错误信息：'.mysql\_error());

return "创建表成功";

}

5.2 插入数据

[复制代码](javascript:void(0);)

//插入数据

function \_insert\_data($sql){

if(!mysql\_query($sql)){

return 0; //插入数据失败

}else{

if(mysql\_affected\_rows()>0){

return 1; //插入成功

}else{

return 2; //没有行受到影响

}

}

}

[复制代码](javascript:void(0);)

5.3 删除数据

[复制代码](javascript:void(0);)

//删除数据

function \_delete\_data($sql){

if(!mysql\_query($sql)){

return 0; //删除失败

}else{

if(mysql\_affected\_rows()>0){

return 1; //删除成功

}else{

return 2; //没有行受到影响

}

}

}

[复制代码](javascript:void(0);)

5.4 修改数据

[复制代码](javascript:void(0);)

//修改数据

function \_update\_data($sql){

if(!mysql\_query($sql)){

return 0; //更新数据失败

}else{

if(mysql\_affected\_rows()>0){

return 1; //更新成功;

}else{

return 2; //没有行受到影响

}

}

}

[复制代码](javascript:void(0);)

5.5 检索数据

//检索数据

function \_select\_data($sql){

$ret = mysql\_query($sql) or die('SQL语句有错误，错误信息：'.mysql\_error());

return $ret;

}

5.6 删除数据表

//删除表

function \_drop\_table($sql){

mysql\_query($sql) or die('删除表失败，错误信息：'.mysql\_error());

return "删除表成功";

}

将以上函数和连接数据库的代码结合起来，生成mysql\_bae.func.php 文件，供下面测试使用。

**六、测试MySQL 函数使用**

6.1 新建文件dev\_mysql.php 在同一目录下并引入mysql\_bae.func.php 文件

require\_once './mysql\_bae.func.php';

6.2 测试创建表

将上面使用phpMyAdmin 创建的test\_mysql 表删除，测试语句如下：

[复制代码](javascript:void(0);)

//创建表

$create\_sql = "CREATE TABLE IF NOT EXISTS `test\_mysql` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`from\_user` varchar(40) DEFAULT NULL,

`account` varchar(40) DEFAULT NULL,

`password` varchar(40) DEFAULT NULL,

`update\_time` datetime DEFAULT NULL,

PRIMARY KEY (`id`),

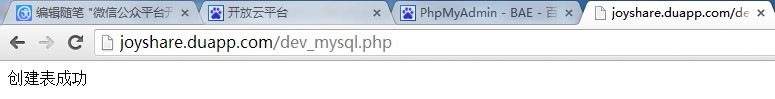
UNIQUE KEY `from\_user` (`from\_user`)

)";

echo \_create\_table($create\_sql);

[复制代码](javascript:void(0);)

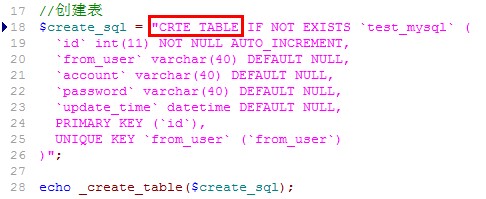
测试正确结果：



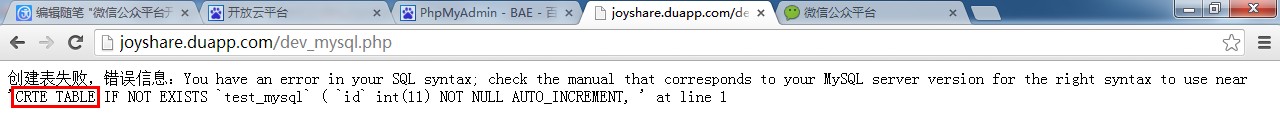
到phpMyAdmin中查看



故意将SQL语句写错



测试错误结果：



6.3 测试插入数据

测试语句如下：

[复制代码](javascript:void(0);)

//插入数据

$insert\_sql = "insert into test\_mysql(from\_user, account, password, update\_time) values('David','860510', 'abcabc', '2013-09-29 17:14:28')";

$res = \_insert\_data($insert\_sql);

if($res == 1){

echo "插入成功";

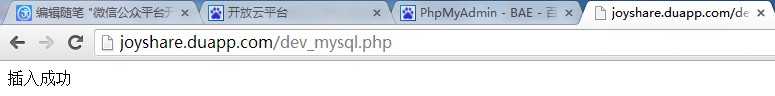
}else{

echo "插入失败";

}

[复制代码](javascript:void(0);)

测试结果：



6.4 测试更新数据

测试语句如下：

[复制代码](javascript:void(0);)

//更新数据

$update\_sql = "update test\_mysql set account = 860512 where account = 860510";

$res = \_update\_data($update\_sql);

if($res == 1){

echo "更新成功";

}elseif($res == 0){

echo "更新失败";

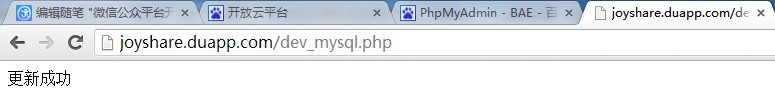
}elseif($res == 2){

echo "没有行受到影响";

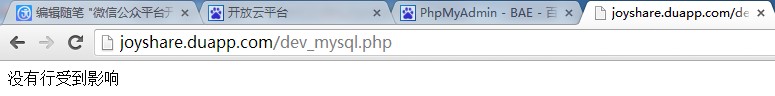
}

[复制代码](javascript:void(0);)

测试结果：



再次更新：



6.5 测试删除数据

测试语句如下：

[复制代码](javascript:void(0);)

//删除数据

$delete\_sql = "delete from test\_mysql where account = 860512";

$res = \_delete\_data($delete\_sql);

if($res == 1){

echo "删除成功";

}elseif($res == 0){

echo "删除失败";

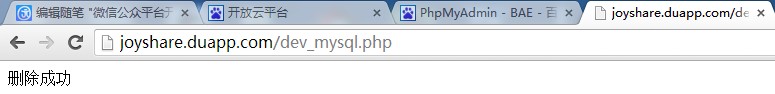
}elseif($res == 2){

echo "没有该条记录";

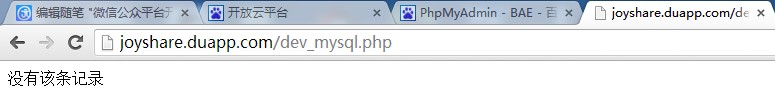
}

[复制代码](javascript:void(0);)

测试结果：



再次删除：



6.6 测试检索数据

再次执行上面的插入操作做检索测试，测试语句如下：

[复制代码](javascript:void(0);)

//检索数据

$select\_sql = "select \* from test\_mysql";

$result = \_select\_data($select\_sql);

while($rows = mysql\_fetch\_array($result,MYSQL\_ASSOC)){

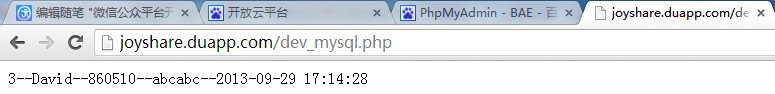
echo $rows[id]."--".$rows[from\_user]."--".$rows[account]."--".$rows[password]."--".$rows[update\_time];

echo "<br />";

}

[复制代码](javascript:void(0);)

测试结果：



6.7 测试删除表

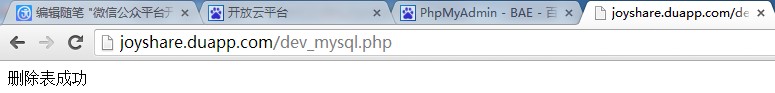
测试语句如下：

//删除表

$drop\_sql = "drop table if exists test\_mysql";

echo \_drop\_table($drop\_sql);

测试结果：



MySQL 函数测试全部成功。

**七、实现与微信的交互（Mysql 扩展）**

保证数据库中存在test\_msyql表，这里测试微信对MySQL数据库的增删改查操作，不考虑特殊情况，只按照下面的方法测试：

[复制代码](javascript:void(0);)

1. 绑定+账户+密码

如：绑定+860512+abc123

2. 查询

如：查询

3. 修改+旧密码+新密码

如：修改+abc123+123456

4. 删除

如：删除

[复制代码](javascript:void(0);)

7.1 引入mysql\_bae.func.php 文件

//引入数据库函数文件

require\_once 'mysql\_bae.func.php';

7.2 前置操作

A. 将输入的语句拆分成数组，以“+”号分隔

$keywords = explode("+",$keyword);

B. 获取当前时间

//获取当前时间

$nowtime=date("Y-m-d G:i:s");

C. 判断用户是否已经绑定

[复制代码](javascript:void(0);)

//判断是否已经绑定

$select\_sql="SELECT id from test\_mysql WHERE from\_user='$fromUsername'";

$res=\_select\_data($select\_sql);

$rows=mysql\_fetch\_array($res, MYSQL\_ASSOC);

if($rows[id] <> ''){

$user\_flag='y';

}

[复制代码](javascript:void(0);)

7.3 测试插入操作

测试代码：

[复制代码](javascript:void(0);)

if(trim($keywords[0] == '绑定')){

if($user\_flag <> 'y'){

$insert\_sql="INSERT INTO test\_mysql(from\_user, account, password, update\_time) VALUES('$fromUsername','$keywords[1]','$keywords[2]','$nowtime')";

$res = \_insert\_data($insert\_sql);

if($res == 1){

$contentStr = "绑定成功";

}elseif($res == 0){

$contentStr = "绑定失败";

}

}else{

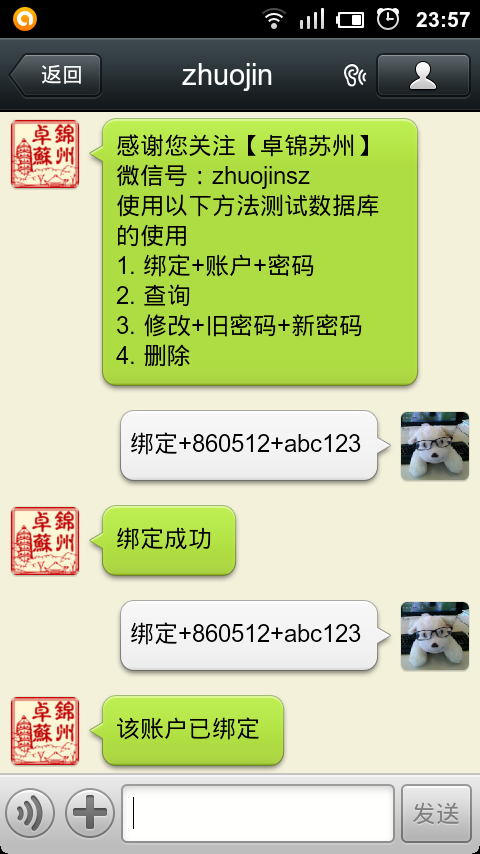
$contentStr = "该账户已绑定";

}

}

[复制代码](javascript:void(0);)

测试结果：



7.4 测试查询操作

测试代码：

[复制代码](javascript:void(0);)

if(trim($keywords[0] == '查询')){

$select\_sql="SELECT \* FROM test\_mysql WHERE from\_user='$fromUsername'";

$select\_res=\_select\_data($select\_sql);

$rows=mysql\_fetch\_assoc($select\_res);

if($rows[id] <> ''){

$contentStr="账户:$rows[account]\n"."密码：$rows[password]\n"."From\_user：$rows[from\_user]\n"."更新时间：$rows[update\_time]";

}else{

$contentStr="您还未绑定账户，查询不到相关信息，请先绑定，谢谢！";

}

}

[复制代码](javascript:void(0);)

测试结果：



7.5 测试更新操作

测试代码：

[复制代码](javascript:void(0);)

if(trim($keywords[0] == "修改")){

$old\_password=$keywords[1];

$new\_password=$keywords[2];

$select\_password\_sql="SELECT \* FROM test\_mysql WHERE from\_user='$fromUsername'";

$select\_res=\_select\_data($select\_password\_sql);

$rows=mysql\_fetch\_assoc($select\_res);

if($old\_password == $rows[password]){

$update\_sql="UPDATE test\_mysql SET password='$new\_password' WHERE from\_user='$fromUsername'";

$res = \_update\_data($update\_sql);

if($res == 1){

$contentStr = "修改成功";

}elseif($res == 0){

$contentStr = "修改失败";

}

}else{

$contentStr = "原密码有误，请确认后重试";

}

}

[复制代码](javascript:void(0);)

测试结果：



7.6 测试删除操作

测试代码：

[复制代码](javascript:void(0);)

if(trim($keywords[0] == "删除")){

$delete\_sql="DELETE FROM test\_mysql WHERE from\_user='$fromUsername'";

$res = \_delete\_data($delete\_sql);

if($res == 1){

$contentStr = "删除成功";

}elseif($res == 0){

$contentStr = "删除失败";

}

}

[复制代码](javascript:void(0);)

测试结果：



与微信的交互测试成功。

**八、PHP Mysqli 扩展，封装成类**

将Mysqli 扩展封装成类使用，代码如下：

[复制代码](javascript:void(0);)

<?php

require\_once 'includes/configure.php';

class MySQLi\_BAE{

private $mysqli;

private $host;

private $user;

private $password;

private $port;

private $database;

//在类之外访问私有变量时使用

function \_\_get($property\_name){

if(isset($this->$property\_name)){

return($this->$property\_name);

}else{

return(NULL);

}

}

function \_\_set($property\_name, $value){

$this->$property\_name=$value;

}

function \_\_construct(){

/\*从平台获取查询要连接的数据库名称\*/

$this->database = MYSQLNAME;

/\*从环境变量里取出数据库连接需要的参数\*/

$this->host = getenv('HTTP\_BAE\_ENV\_ADDR\_SQL\_IP');

$this->user = getenv('HTTP\_BAE\_ENV\_AK');

$this->password = getenv('HTTP\_BAE\_ENV\_SK');

$this->port = getenv('HTTP\_BAE\_ENV\_ADDR\_SQL\_PORT');

$this->mysqli = new mysqli($this->host, $this->user, $this->password, $this->database, $this->port);

if($this->mysqli->connect\_error){

die("Connect Server Failed:".$this->mysqli->error);

}

$this->mysqli->query("set names utf8");

}

//dql statement

function execute\_dql($query){

$res = $this->mysqli->query($query) or die("操作失败".$this->mysqli->error);

return $res;

//$this->mysqli->close();

}

//dml statement

function execute\_dml($query){

$res = $this->mysqli->query($query) or die("操作失败".$this->mysqli->error);

if(!$res){

return 0;//失败

}else{

if($this->mysqli->affected\_rows > 0){

return 1;//执行成功

}else{

return 2;//没有行受影响

}

}

//$this->mysqli->close();

}

}

?>

[复制代码](javascript:void(0);)

**九、测试类的使用**

9.1 测试DML操作

测试代码：

[复制代码](javascript:void(0);)

<?php

require\_once "MySQLi\_BAE.class.php";

$mysqli\_BAE=new MySQLi\_BAE();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*dml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$sql="insert into test\_mysql (from\_user, account, password, update\_time) values('David','860510', 'abcabc', '2013-09-27 17:14:28')";

//$sql="update test\_mysql set account = 860512 where account = 860510";

//$sql="delete from test\_mysql where account = 860512";

$res=$mysqli\_BAE->execute\_dml($sql);

if($res==0){

echo "执行失败";

}elseif($res==1){

echo "执行成功";

}else{

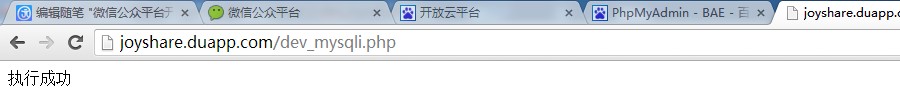
echo "没有行数影响";

}

?>

[复制代码](javascript:void(0);)

测试结果：



9.2 测试DQL操作

测试代码：

[复制代码](javascript:void(0);)

<?php

require\_once "MySQLi\_BAE.class.php";

$mysqli\_BAE=new MySQLi\_BAE();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*dql\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$sql="select \* from test\_mysql";

$res=$mysqli\_BAE->execute\_dql($sql);

while($row=$res->fetch\_row()){

foreach($row as $key=>$val){

echo "$val--";

}

echo '<br/>';

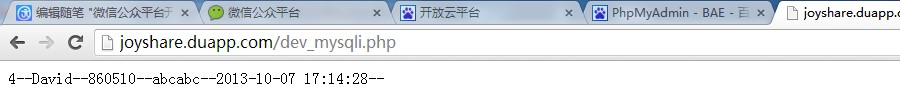
}

$res->free();

?>

[复制代码](javascript:void(0);)

测试结果：



**十、实现与微信的交互（Mysqli 扩展）**

10.1 前置操作

A. 引入MySQLi\_BAE.class.php 文件

//引入数据库函数文件

require\_once "MySQLi\_BAE.class.php";

B. 实例化对象

public function \_\_construct()

{

$this->mysqli\_BAE=new MySQLi\_BAE();

}

10.2 测试插入操作

测试代码：

$insert\_sql="INSERT INTO test\_mysql(from\_user, account, password, update\_time) VALUES('$fromUsername','$keywords[1]','$keywords[2]','$nowtime')";

$res = $this->mysqli\_BAE->execute\_dml($insert\_sql);

测试结果：



10.3 测试查询操作

测试代码：

$select\_sql="SELECT \* FROM test\_mysql WHERE from\_user='$fromUsername'";

$select\_res=$this->mysqli\_BAE->execute\_dql($select\_sql);

$rows=$select\_res->fetch\_array(MYSQLI\_ASSOC);

测试结果：



10.4 测试更新操作

测试代码：

$update\_sql="UPDATE test\_mysql SET password='$new\_password' WHERE from\_user='$fromUsername'";   
$res = $this->mysqli\_BAE->execute\_dml($update\_sql);

测试结果：



10.5 测试删除操作

测试代码：

$delete\_sql="DELETE FROM test\_mysql WHERE from\_user='$fromUsername'";

$res = $this->mysqli\_BAE->execute\_dml($delete\_sql);

测试结果：



与微信交互测试成功。

**十一、完整代码获取**

请访问 [**乐思乐享**](http://pan.baidu.com/s/1c0s3Jby) 官方论坛

**URL**：<http://pan.baidu.com/s/1c0s3Jby>