

## Взаимодействие PHP и SQL. Способы взаимодействия PHP и СУБД MySQL. Установка соединения с базой данных, функции отправки запросов и обработка ответов

Рассмотрим описания наиболее часто используемых функций, позволяющих работать с MySQL средствами PHP.

*SQL может применяться в прикладных программах двумя способами: в виде встроенного SQL и интерфейса программирования приложений (Application Program Interface, API). Первый способ напоминает использование PHP - инструкции SQL размещаются среди кода прикладной программы. В настоящий момент такой стиль не поддерживает ни MySQL ни PHP. Второй подход заключается в том, что программа взаимодействует с СУБД посредством совокупности функций. Именно такой подход используется при взаимодействии PHP и MySQL.*

### mysql\_connect

```
resource mysql_connect ([string server [, string username [, string password]])
```

Эта функция устанавливает соединение с сервером `server` MySQL и возвращает дескриптор соединения с базой данных, по которому все другие функции, принимающие этот дескриптор в качестве аргумента, будут однозначно определять выбранную базу данных. Вторым и третьим аргументами этой функции являются имя пользователя `username` и его пароль `password` соответственно:

```
<?php
$dblocation = "localhost"; // Имя сервера
$dbuser = "root";         // Имя пользователя
$dbpasswd = "";           // Пароль
$dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
if (!$dbcnx) // Если дескриптор равен 0 соединение не установлено
{
    echo("<P>В настоящий момент сервер базы данных не доступен, поэтому
        корректное отображение страницы невозможно.</P>");
    exit();
}
?>
```

Переменные `$dblocation`, `$dbuser` и `$dbpasswd` хранят имя сервера, имя пользователя и пароль, и, как правило, прописываются в отдельном файле (к примеру, `config.php`), который потом вставляется в каждый PHP-файл, в котором имеется код для работы с MySQL:

```
<?
$dblocation = "localhost";
$dbname = "forum";
$dbuser = "root";
```

```
$dbpasswd = "";
?>
```

## **mysql\_close**

bool mysql\_close ([resource link\_identifier])

Эта функция разрывает соединение с сервером MySQL, и возвращает `true` при успешном выполнении операции и `false` в противном случае. Функция принимает в качестве аргумента дескриптор соединения с базой данных, возвращаемый функцией `mysql_connect`.

К примеру:

```
// устанавливаем соединение с базой данных
$dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
if (!$dbcnx)
{
    // Выводим предупреждение
    echo("<P>В настоящий момент сервер базы данных не доступен, поэтому
корректное отображение страницы невозможно.</P>");
    // Завершаем работу в случае неудачи
    exit();
}
if(mysql_close($dbcnx)) // разрываем соединение
{
    echo("Соединение с базой данных прекращено");
}
else
{
    echo("Не удалось завершить соединение");
}
}
```

## **mysql\_select\_db**

bool mysql\_select\_db (string database\_name [, resource link\_identifier])

Использование этой функции эквивалентно вызову команды `USE` в SQL-запросе, т. е. функция `mysql_select_db` выбирает базу данных для дальнейшей работы, и все последующие SQL-запросы применяются к выбранной базе данных. Функция принимает в качестве аргументов название выбираемой базы данных `database_name` и дескриптор соединения `resource`. Функция возвращает `true` при успешном выполнении операции и `false` в противном случае. К примеру:

```
<?php
// Код соединения с базой данных
if (!@mysql_select_db($dbname, $dbcnx))
{
    echo("<P>В настоящий момент база данных не доступна, поэтому
корректное отображение страницы невозможно.</P>");
    exit();
}
?>
```

Имеет смысл помещать функции для соединения и выбора базы данных в тот же файл (config.php), где объявлены переменные с именами сервера, пользователя и паролем:

```
<?php
$dblocation = "localhost";
$dbname = "softtime";
$dbuser = "root";
$dbpasswd = "";
$dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
if (!$dbcnx)
{
    echo "<P>В настоящий момент сервер базы данных не доступен, поэтому
        корректное отображение страницы невозможно.</P>" );
    exit();
}
if (!@mysql_select_db($dbname, $dbcnx))
{
    echo "<P>В настоящий момент база данных не доступна, поэтому
        корректное отображение страницы невозможно.</P>" );
    exit();
}
?>
```

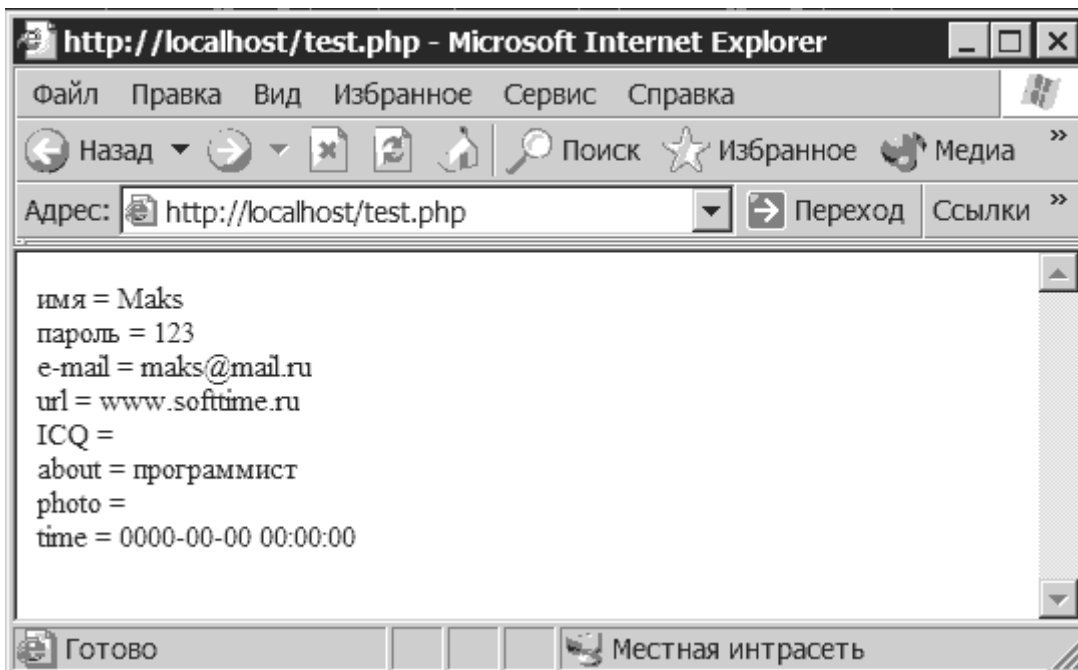
## mysql\_query

resource mysql\_query (string query)

Эта функция применяется для отправки серверу SQL-запросов. Функция возвращает дескриптор запроса в случае успеха и `false` в случае неудачного выполнения запроса. В листинге 13.32 показан код, с помощью которого извлекается одна строка из таблицы `authors` базы данных `forum`.

```
<?php
include "config.php";
$ath = mysql_query("select * from authors;");
if($ath)
{
    $author = mysql_fetch_array($ath);
    echo "<br>имя = ".$author['name']."<br>";
    echo "пароль = ".$author['passwd']."<br>";
    echo "e-mail = ".$author['email']."<br>";
    echo "url = ".$author['url']."<br>";
    echo "ICQ = ".$author['icq']."<br>";
    echo "about = ".$author['about']."<br>";
    echo "photo = ".$author['photo']."<br>";
    echo "time = ".$author['time'];
}
else
{
    echo "<p><b>Error: ".mysql_error()."</b></p>";
    exit();
}
?>
```

Результат выполнения показан на следующем рисунке:



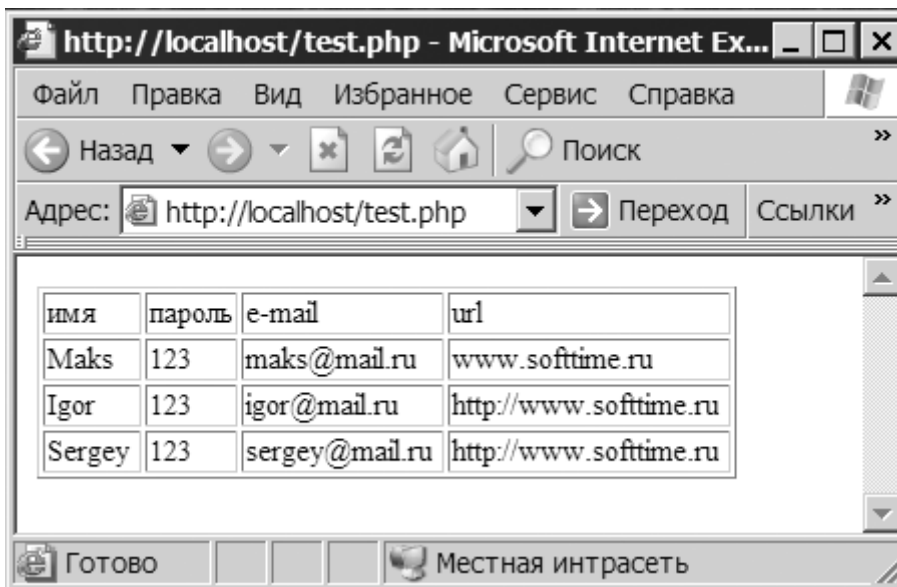
## mysql\_fetch\_array

array mysql\_fetch\_array (resource result)

Эта функция возвращает значения полей в виде ассоциативного массива, в качестве аргумента принимает дескриптор запроса возвращаемый функцией `mysql_query`. Вот как с помощью этой функции можно вывести все строки таблицы `authors`:

```
<?php
include "config.php";
$ath = mysql_query("select * from authors;");
if($ath)
{
    // Определяем таблицу и заголовок
    echo "<table border=1>";
    echo "<tr><td>имя</td><td>пароль</td><td>e-mail</td><td>url</td></tr>";
    // Так как запрос возвращает несколько строк, применяем цикл
    while($author = mysql_fetch_array($ath))
    {
        echo "<tr><td>". $author['name']. "&nbsp;</td><td>". $author['passw']. "
        &nbsp;</td><td>". $author['email']. "&nbsp;</td><td>".
        $author['url']. "&nbsp;</td></tr>";
    }
    echo "</table>";
}
else
{
    echo "<p><b>Error: ".mysql_error()."</b><p>";
    exit();
}
?>
```

Результат показан на следующем рисунке:



## mysql\_result

mixed mysql\_result (resource result, int row)

С помощью этой функции можно получить доступ к отдельному полю записи. Допустим, нам нужно вывести имя автора, который первым найдется в базе данных. Сделать это можно следующим образом:

```
<?php
include "config.php";
$sath = mysql_query("select name from authors;");
if($sath)
{
    echo mysql_result($sath,0,'name');
}
else
{
    echo "<p><b>Error: ".mysql_error()."</b><p>";
    exit();
}
?>
```

## mysql\_fetch\_object

object mysql\_fetch\_object (resource result)

Эта функция возвращает поля записи данных в виде объекта. Ниже приведен пример, в котором с помощью этой функции из таблицы `authors` выводятся имя, URL и e-mail авторов.

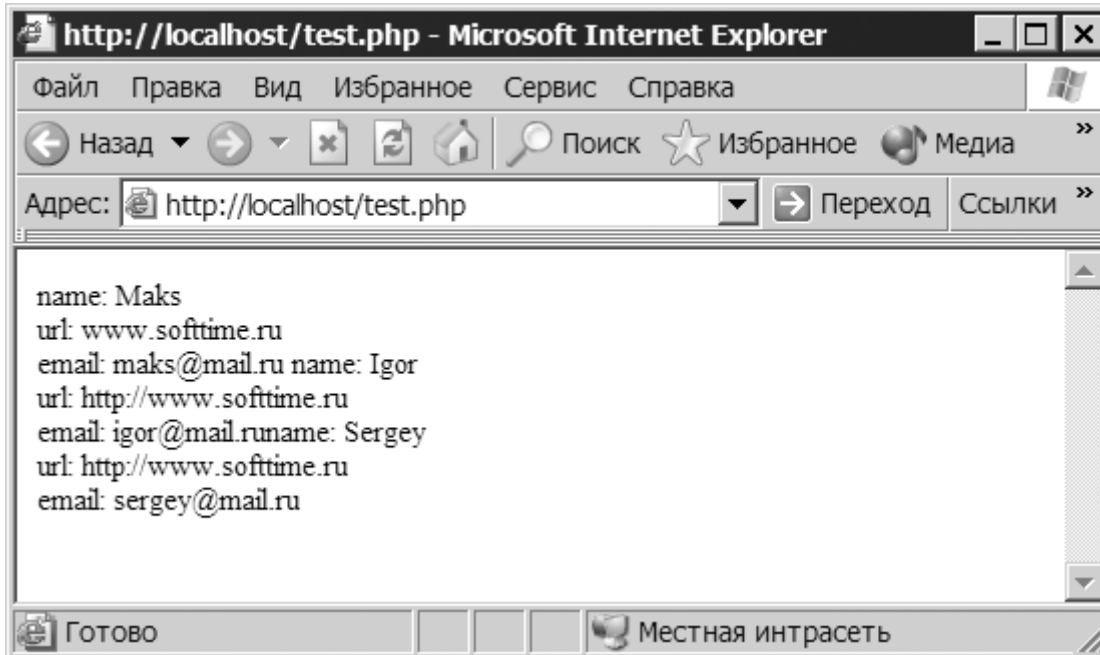
```
<?php
include "config.php";
$sath = mysql_query("select * from authors;");
if($sath)
{
    while($row = mysql_fetch_object($sath))
    {
        echo "<p>name: ".$row->name."</p>";
        echo "<p>url: ".$row->url."</p>";
        echo "<p>email: ".$row->email."</p>";
    }
}
```

```

}
else
{
    echo "<p><b>Error: ".mysql_error()."</b><p>";
    exit();
}
?>

```

Результат выполнения показан на следующем рисунке:



## mysql\_fetch\_row

array mysql\_fetch\_row (resource result)

В отличие от функции `mysql_fetch_object`, эта функция возвращает не объект, а массив, в котором содержатся значения полей:

```

<?php
include "config.php";
$ath = mysql_query("select * from authors;");
if($ath)
{
    while($row = mysql_fetch_row($ath))
    {
        echo "<p>name: ".$row[1]."</p>";
        echo "<p>url: ".$row[4]."</p>";
        echo "<p>email: ".$row[3]."</p>";
    }
}
else
{
    echo "<p><b>Error: ".mysql_error()."</b><p>";
    exit();
}
?>

```

Результат выполнения этого кода в точности совпадает с приведённым выше.