

Средства PHP для работы с MySQL

Зорина И.Г.

Расширение PDO

Расширение PDO предоставляет интерфейс для доступа к базам данных и PHP. Механизму PDO необходим драйвер конкретной базы данных – PDO_MYSQL

Новая объектно-ориентированная библиотека PDO предоставляет одинаковый интерфейс для всех типов баз данных, облегчая переход от одной базы данных к другой.



Установка соединения с базой данных

Для соединения с базой данных необходимо создать объект класса PDO.

```
PDO:  __construct(  
    string $dsn [,  
    string $username [,  
    string $password [,  
    array $options]])
```

Пример:

```
$pdo = new PDO('mysql:host=localhost;  
dbname=sales', 'root', '');
```



Выполнение SQL-запросов

Для выполнения запроса можно использовать метод `exec()` класса `PDO`. Метод принимает в качестве параметра строку `$statement` с запросом и возвращает количество затронутых в ходе его выполнения записей.

```
public int PDO::exec(string $statement )
```



Установка соединения с базой данных

connect_db.php

```
1  <?php
2  // Соединение с базой данных
3  try
4  {
5      $pdo = new PDO('mysql:host=localhost; dbname=sales', 'root', '');
6      $pdo->exec('SET NAMES "utf8"');
7  }
8  catch (PDOException $e)
9  {
10     echo "Невозможно установить соединение с базой данных";
11     include 'error.html.php';
12     exit();
13 }
14 ?>
15
```

Использование шаблона



error.html.php



```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>Ошибка скрипта</title>
6  </head>
7  <body>
8  <p>
9  <?php echo $error; ?>
10 </p>
11 </body>
12 </html>
```

Извлечение версии MySQL -сервера

version.php

```
1  <?php ## Извлечение текущей версии
2      require_once('connect_db.php');
3
4      // Выполняем запрос
5      $query = "SELECT VERSION() AS version";
6      $ver = $pdo->query($query);
7      // Извлекаем результат
8      $version = $ver->fetch();
9      echo $version['version']; // 5.6.37
10  ?>
```

Выполнение SQL запросов

```
exes.php x
1 <?php ## Использование метода PDO::exec()
2 require_once('connect_db.php');
3
4 // Формируем и выполняем SQL-запрос
5 $query = "CREATE TABLE customers (
6         id serial,
7         FIO varchar(100) NOT NULL,
8         address varchar(100),
9         phone varchar(15),
10        rating int,
11        PRIMARY KEY (id))";
12 $count = $pdo->exec($query);
13 if ($count !== false)
14     echo "Таблица создана успешно";
15 else {
16     echo "Не удалось создать таблицу";
17     echo "<pre>";
18     print_r($pdo->errorInfo());
19     echo "<pre>";
20 }
21 ?>
```


Извлечение данных

```
index.php x
1 <?php ## Вывод содержимого таблицы product
2 require_once("connect_db.php");
3 try
4 {
5     $query = "SELECT pname, description, price, qty FROM product";
6     $result = $pdo->query($query);
7 }
8 catch (PDOException $e)
9 {
10     echo "Ошибка выполнения запроса: " . $e->getMessage();
11     include 'error.html.php';
12     exit();
13 }
14 while ($row = $result->fetch())
15 {
16     $prod[] = array(
17         'pname' => $row['pname'],
18         'description' => $row['description'],
19         'price' => $row['price'],
20         'qty' => $row['qty']
21     );
22 }
23 include 'product.html.php';
24 ?>
```

Шаблон для вывода списка товаров

```
product.html.php x
1 <!DOCTYPE html>
2 <head>
3 <meta charset="utf-8">
4 <title>Вывод данных из базы данных</title>
5 </head>
6 <body>
7     <h1>Товары</h1>
8     <table>
9         <tr>
10             <td>Название</td>
11             <td>Описание</td>
12             <td>Цена</td>
13             <td>Количество</td>
14         </tr>
15         <?php foreach($prod as $pro): ?>
16             <tr>
17                 <td>
18                     <?php echo htmlspecialchars($pro['pname'], ENT_QUOTES, 'UTF-8'); ?></td>
19                 <td>
20                     <?php echo htmlspecialchars($pro['description'], ENT_QUOTES, 'UTF-8'); ?></td>
21                 <td>
22                     <?php echo htmlspecialchars($pro['price'], ENT_QUOTES, 'UTF-8'); ?></td>
23                 <td>
24                     <?php echo htmlspecialchars($pro['qty'], ENT_QUOTES, 'UTF-8'); ?></td>
25             </tr>
26         <?php endforeach; ?>
27     </table>
28 </body>
29 </html>
```

Результат выполнения

Выход данных из базы X

← → ↻ 🏠 ⓘ localhost:81/sales/index.php

Товары

Название	Описание	Цена	Количество
Утюг BOSH	Паровой удар, мощность 1000Вт	3500.00	6
Холодильник INDESIT	Две камеры	15600.00	14

Form.html.php

```
<?php include_once 'helpers.php'; ?>
<!DOCTYPE html>
<html>
<head>
    <title><?php htmlout($pageTitle); ?></title>
</head>
<body>
    <h1>
        <?php htmlout($pageTitle); ?>
    </h1>
    <form action="<?php htmlout($action); ?>" method="POST">
        <div>
            <label for="name">
                Название:
                <input type="text" name="name" id="name" value="<?php htmlout($name); ?>">
            </label>
        </div>
        <div>
            <label for="price">
                Цена:
                <input type="text" name="price" id="price" value="<?php htmlout($price); ?>">
            </label>
        </div>
        <div>
            <label for="qty">
                Количество:
                <input type="text" name="qty" id="qty" value="<?php htmlout($qty); ?>">
            </label>
        </div>
        <div>
            <input type="hidden" name="id" value="<?php htmlout($id); ?>">
            <input type="submit" value="<?php htmlout($button); ?>">
        </div>
    </form>
</body>
</html>
```

Helpers.php

A screenshot of a code editor interface. At the top, there are four tabs: 'db.php', 'error.html.php', 'helpers.php', and 'form.html.php'. The 'helpers.php' tab is active. Below the tabs, the code is displayed with line numbers 1 through 9 on the left. The code defines two PHP functions: 'html' and 'htmlout'. The 'html' function takes a parameter '\$text' and returns the result of 'htmlspecialchars' with 'ENT_QUOTES' and 'UTF-8' encoding. The 'htmlout' function takes a parameter '\$text' and echoes the result of 'html'.

Index.php

```
1  <?php
2  if(isset($_GET['add']))
3  {
4      $pageTitle = 'Новый товар';
5      $action = 'addform';
6      $name = '';
7      $price = '';
8      $qty = '';
9      $id = '';
10     $button = 'Добавить товар';
11     include 'form.html.php';
12     exit();
13 }
14 //Добавление нового товара
15 if (isset($_GET['addform']))
16 {
17     include "db.php";
18     try
19     {
20         $sql = 'INSERT into product SET
21         name = :name,
22         price = :price,
23         qty = :qty';
24         $s = $pdo->prepare($sql);
25         $s->bindValue(':name', $_POST['name']);
26         $s->bindValue(':price', $_POST['price']);
27         $s->bindValue(':qty', $_POST['qty']);
28         $s->execute();
29     }
30     catch(PDOException $e)
31     {
32         $error = 'Ошибка при добавлении товара';
33         include 'error.html.php';
34         exit();
35     }
36     header('Location: .');
37     exit();
38 }
```

Продолжение Index.php

```
39
40 //Вывод всех товаров
41 include "db.php";
42 try
43 {
44     $query = "SELECT id, name, price, qty FROM product";
45     $result = $pdo->query($query);
46 }
47 catch(PDOException $e)
48 {
49     echo "Ошибка выполнения запроса " . $e->getMessage();
50     include 'error.html.php';
51     exit();
52 }
53 while ($row = $result->fetch())
54 {
55     $prod[]=array(
56         'id'    => $row['id'],
57         'name'   => $row['name'],
58         'price'  => $row['price'],
59         'qty'    => $row['qty']
60     );
61 }
62 include 'product.html.php';
```