

Представления в MVC на PHP

Давайте теперь добавим нашим действиям представления. По правилам нашего фреймворка представления хранятся в папке `/project/views/`.

Кроме того, в нашем фреймворке действует соглашение: имя файла с представлением должно совпадать с именем действия и размещаться в папке с таким же названием, как имя контроллера. Например, если у нас есть контроллер `PageController`, то все его представления следует хранить в папке `/project/views/page/`. Сам файл представления должен иметь расширение **.php**. К примеру, если в контроллере `PageController` будет действие `act`, то его представление следует размещать в `/project/views/page/act.php`.

Для того, чтобы задать представление какому-нибудь действию, это действие должно вызвать метод **render**, передав ему параметром имя представления. Имя представления должно состоять из имени контроллера с маленькой буквы, после которого через `/` будет записано имя действия

Метод **render** не нужно реализовывать в контроллере - он наследуется от родительского класса `Core\Controller`. Кроме того, для корректной работы результат работы метода следует вернуть через **return**.

Итак, пусть у нас есть контроллер `Page` с действием `act`. Давайте привяжем к этому действию представление:

```
1  <?php
2      namespace Project\Controllers;
3      use Core\Controller;
4
5      class PageController extends Controller
6      {
7          public function act()
8          {
9              return $this->render('page/act');
10         }
11     }
12  ?>
```

Сделаем теперь файл `/project/views/page/act.php` с HTML кодом нашего представления:

```
1  <div>
2      это представление
3      действия act контроллера page
4  </div>
```

Задача 95.1

Прodelайте описанные манипуляции, а затем обратитесь к нашему действию через адресную строку. Убедитесь, что в браузере появится текст из созданного вами представления.

Передача данных в представление

С помощью второго параметра метода `render` можно передавать данные из контроллера в представление. Данные следует передавать в виде ассоциативного массива. При этом в представлении ключи этого массива станут переменными с соответствующими значениями.

Давайте посмотрим на практике. Передадим в наше представление массив с тремя элементами:

```
1  <?php
2  namespace Project\Controllers;
3  use Core\Controller;
4
5  class PageController extends Controller
6  {
7      public function act()
8      {
9          return $this->render('page/act', [
10             'var1' => 'eee',
11             'var2' => 'bbb',
12             'var3' => 'kkk',
13         ]);
14     }
15 }
16 ?>
```

Как вы видите, ключи нашего массива - var1, var2 и var3. Это значит, что в представлении будут доступны такие переменные и мы можем вывести их значения в нужных местах HTML кода.

Сделаем это:

```
1  <div>
2      это представление
3      действия act контроллера test
4  </div>
5  <ul>
6      <li><?php echo $var1; ?></li>
7      <li><?php echo $var2; ?></li>
8      <li><?php echo $var3; ?></li>
9  </ul>
```

Задача 95.2

Проделайте описанные манипуляции, а затем обратитесь к нашему действию через адресную строку. Убедитесь, что в браузере появится текст из созданного вами представления.

Передача массивов

В представление можно также передавать и массивы. Давайте, например, передадим массив юзеров:

```
1  <?php
2  namespace Project\Controllers;
3  use Core\Controller;
4
5  class PageController extends Controller
6  {
7      public function act()
8      {
9          return $this->render('page/act', [
10             'header' => 'список юзеров',
11             'users'  => ['user1', 'user2', 'user3'],
12         ]);
13     }
14 }
15 ?>
```

Можно обратиться к каждому элементу массива:

```
1  <h1><?= $header ?></h1>
2  <ul>
3      <li><?= $users[0]; ?></li>
4      <li><?= $users[1]; ?></li>
5  </ul>
```

```
        <li><?= $users[2]; ?></li>
6 | </ul>
```

А можно перебрать массив циклом и сформировать нужный HTML код:

```
1 | <h1><?= $header ?></h1>
2 | <ul>
3 |     <?php foreach ($users as $user): ?>
4 |         <li><?= $user; ?></li>
5 |     <?php endforeach; ?>
6 | </ul>
```

Задача 95.3

Прodelайте описанные манипуляции, а затем обратитесь к нашему действию через адресную строку. Убедитесь, что в браузере появится текст из созданного вами представления.

