Контроллеры, действия и роуты в MVC на PHP

Первое, с чем мы разберемся - это контроллеры. Контроллеры обрабатывают запросы пользователя, понимают, что хотел попросить пользователь у сайта, просят соответствующие данные из модели и отправляют их во view.

Контроллеры представляют собой ООП классы. Один файл - это один класс и, соответственно, один контроллер. В нашем фреймворке контроллеры будут храниться в папке **project/controllers**.

Давайте потренируемся в создании контроллеров. Для разминки сделаем класс **PageController**, который будет управлять текстовыми страницами на нашем сайте.

Давайте сразу создадим файл для нашего контроллера. По правилам нашего фреймворка каждый класс должен хранится в файле с одноименным названием (вплоть до регистра). То есть наш класс **PageController** будет хранится в файле **PageController.php**. Создайте этот файл в папке **project/controllers**.

Создайте в этом файле наш класс:

```php
<?php
  namespace Project\Controllers;
  use \Core\Controller;

  class PageController extends Controller
  {

  }
?>
```

Как вы видите, наш класс принадлежит пространству имен **Project\Controllers**, следуя соглашению об автозагрузке файлов (то есть путь по папкам должен совпадать с пространством имен).

Кроме того, наш класс наследуется от класса Core\Controller, находящегося в ядре фреймворка. В этом не нужно искать глубокий смысл, а просто нужно принять как правило фреймворка. Вот это правило: *все создаваемые вами контроллеры должны наследоваться от класса Core\Controller, чтобы все работало, как надо.*

Действия

Добавим теперь в наш контроллер методы класса. В терминах MVC методы контроллеров называются **действиями** (*actions*).

Сделаем, например, два действия - **show1** и **show2**, и в каждом действии выведем что-нибудь на экран:

```php
<?php
  namespace Project\Controllers;
  use \Core\Controller;

  class PageController extends Controller
  {
    public function show1()
    {
      echo '1';
    }

    public function show2()
    {
      echo '2';
    }
  }
?>
```

Роутинг

Теперь вам необходимо познакомиться с таким понятием, как *роутинг*. Он представляет собой механизм, с помощью которого можно вызывать определенное действие определенного контроллера через адресную строку браузера.

Настройки роутинга хранятся в файле **/project/config/routes.php** и представляют собой массив объектов класса **\Core\Route**. Конструктор этого класса первым параметром принимает URI, по запросу которого вызовется соответствующий метод соответствующего контроллера. Имя контроллера и имя действия задаются вторым и третьим параметрами. При этом имя контроллера задается с маленькой буквы.

Давайте для примера добавим два роута (то есть маршрута): первый при обращении к адресу **/my-page1/** будет вызывать метод **show1** контроллера **page**, а второй - метод **show1** этого же контроллера:

```php
<?php
  use \Core\Route;

  return [
    new Route('/my-page1/', 'page', 'show1'),
    new Route('/my-page2/', 'page', 'show2'),
  ];
?>
```

Практические задачи

Задача 93.1

Сделайте контроллер TestController с действиями act1, act2 и act3. Сделайте 3 роута, задающие адреса, по которым можно будет обратиться к этим действиям. Проверьте работу созданного вами кода, по очереди обратившись через адресную строку к каждому из действий.