

Запись на курсы по HTML, CSS, JavaScript, PHP, фреймворкам и CMS,
а также: помощь в поиске работы и заказов, стажировка на реальных проектах→

урок 107 из 107

[Верстка](#) [JavaScript](#) [PHP](#) [NodeJs](#) [Vue](#) [React](#) [Laravel](#) [WordPress](#) [AJAX](#) [Парсинг](#)

Бесплатный тренинг "Работа с узлами DOM в JavaScript". Начало 10-го декабря→
Занимательные задачи JavaScript. Перезапуск! Начнем, когда наберется более 100 желающих→

Реализация моделей в своем MVC фреймворке на PHP

Последнее, что нам осталось реализовать, это работа с моделями.

Для начала давайте в файле index.php подключим файл с настройками подключения к базе данных (разместим это подключение вверху файла):

```
1  <?php
2  namespace Core;
3
4  error_reporting(E_ALL);
5  ini_set('display_errors', 'on');
6
7  require_once $_SERVER['DOCUMENT_ROOT'] . '
   /project/config/connection.php';
8
9  ...
10 ?>
```

Файл настроек содержит константы PHP. После подключения файла эти константы будут доступны во всех вызываемых нами классах.

Родитель всех моделей

Давайте реализуем родительский класс всех моделей в файле `/core/Model.php`:

```
1  <?php
2  namespace Core;
3
4  class Model
5  {
6      private $link;
7
8      public function __construct()
9      {
10         $this->link = mysqli_connect(DB_HOST, DB_USER, DB_
            PASS, DB_NAME);
11         mysqli_query($this->link, "SET NAMES 'utf8'");
12     }
13
14     protected function findOne($query)
15     {
16         // тут будет какой-то код
17     }
18
19     protected function findMany($query)
20     {
21         // тут будет какой-то код
22     }
23 }
24 ?>
```



Как вы видите, в конструкторе класса мы осуществляем подключение к базе данных. Есть, однако, проблема: каждая создаваемая модель (если внутри одного действия контроллера будет несколько моделей) будет делать свое подключение к базе данных, а это не оптимально.

Давайте сделаем так, чтобы подключение к базе создавалось только для первой созданной модели, а остальные модели использовали уже созданное подключение. Для этого свойство `link` сделаем статическим:

```
1  <?php
2  namespace Core;
3
4  class Model
5  {
6      private static $link;
7
8      public function __construct()
9      {
10         if (!self::$link) { // если свойство не
11                               задано, то подключаемся
12             self::$link = mysqli_connect(DB_HOST, DB_USER, DB_
13                                     PASS, DB_NAME);
14             mysqli_query(self::$link, "SET NAMES 'utf
15                                     8'");
16         }
17     }
18
19     protected function findOne($query)
20     {
21     }
22
23     protected function findMany($query)
24     {
25     }
26 }
27 ?>
```

Задача 107.1

Скопируйте приведенный код класса `Model` и разместите его в файле `/core/Model.php`.

Задача 107.2

Реализуйте метод `findOne` в этом классе. Проверьте его работу.

Задача 107.3

Реализуйте метод `findMany` в этом классе. Проверьте его работу.

