

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

урок 99 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→

Модели в MVC на PHP

Давайте теперь разберемся с моделями. Модели также, как и контроллеры, представляют собой ООП классы. Их следует размещать в папке `/project/models/`. Давайте в этой папке в файле `Page.php` сделаем модель `Page` отвечающую за данные страниц нашего сайта:

```
1 <?php
2 namespace Project\Models;
3 use \Core\Model;
4
5 class Page extends Model
6 {
7
8 }
9 ?>
```

Как вы видите, наша модель наследует от класса `\Core\Model`. От этого класса наша модель наследует protected методы `findOne` и `findMany`. Метод **`findOne`** параметром принимает SQL запрос и возвращает первую запись из БД, попавшую в результат запроса. Метод **`findMany`** также принимает SQL запрос, но возвращает массив записей.

При этом вам не нужно писать код, устанавливающий соединение с БД, а также обрабатывать результаты запроса через `mysqli_fetch_assoc`. Фреймворк делает это за вас. Вам просто нужно выбрать, ожидаете ли вы от запроса одну запись из БД или несколько и использовать соответствующую функцию, получив результатом вменяемый массив, которым можно просто взять и воспользоваться дальше.

Попробуем на практике

Пусть в вашей базе данных есть таблица `page` с какими-нибудь записями. Давайте сделаем модель `Page` с двумя методами. Первый метод будет получать одну запись по ее `id`, а второй метод - получать массив записей из диапазона `id`:

```
1 <?php
2 namespace Project\Models;
3 use \Core\Model;
4
5 class Page extends Model
6 {
7     public function getById($id)
8     {
9         return $this->findOne("SELECT * FROM page
10             WHERE id=$id");
11     }
12
13     public function getByRange($from, $to)
14     {
15         return $this->findAll("SELECT * FROM page
16             WHERE id>=$from AND id<=$to");
17     }
18 }
19 ?>
```

Давайте теперь поработаем с нашей моделью в контроллере:

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

```
4 | use \Project\Models\Page; // подключаем наш
   | у модель
5 |
6 | class PageController extends Controller
7 | {
8 |     public function test() {
9 |         $page = new Page; // создаем объект модели
10 |
11 |         $data = $page->getById(3); // получим зап
   |       ись с id=3
12 |         var_dump($data);
13 |
14 |         $data = $page->getById(5); // получим зап
   |       ись с id=5
15 |         var_dump($data);
16 |
17 |         $data = $page->getByRange(2, 5); // записи с i
   |       d от 2 до 5
18 |         var_dump($data);
19 |     }
20 | }
21 | ?>
```

Задача 99.1

Создайте в вашей базе данных таблицу page, заполните ее данными. Проверьте работу кода, описанного в теории.

