

Рассмотрение встроенного инструмента DB по работе с базами данных в PHP фреймворке Yii2

Кочитов Михаил Евгеньевич

*Приамурский государственный университет им. Шолом-Алейхема
студент*

Аннотация

В данной статье рассматривается инструмент DB, который встроен в PHP фреймворк Yii2 и позволяет работать с базами данных, то есть добавлять, обновлять и удалять записи в таблицах. Также в статье будет разработан собственный пример с использованием данного инструмента DB в PHP фреймворке Yii2 и применением его возможностей на редактирование таблицы базы данных.

Ключевые слова: встроенный инструмент, DB, PHP фреймворк, Yii2, база данных

Examining the built-in DB tool for working with databases in PHP framework Yii2

Kochitov Mikhail Evgenevich

*Sholom-Aleichem Priamursky State University
student*

Abstract

This article discusses the DB tool, which is built into the Yii2 PHP framework and allows you to work with databases, that is, add, update and delete records in tables. Also, the article will develop its own example using this DB tool in the Yii2 PHP framework and using its capabilities to edit a database table.

Keywords: embedded tool, DB, PHP framework, Yii2, database

Каждый PHP фреймворк будет использоваться для создания различных веб-сайтов, но для хранения значительного объема данных понадобятся базы данных, поэтому каждый PHP фреймворк должен быть наделен встроенными инструментами по работе с таблицами в базах данных. Рассматриваемый в статье PHP фреймворк Yii2 обладает своим встроенным инструментом DB, позволяющий работать с различными базами данных.

В статье А.В. Оленникова рассматривается разработка панели администратора сайта на фреймворке Yii2 [1]. Рассматривая статью И.В. Филимонова, И.Б. Разина можно увидеть разработку платформы для создания сайта типа интернет-магазин на базе Yii2 framework [2]. Д.Р. Салихов, А.Р. Халиков в своей статье рассмотрели разработку веб-приложения для складского учета на базе фреймворка Yii2 [3]. В статье М.Е.

Кочитова рассматривается использование плагинов ActiveRecord и Query для управления базами данных в PHP фреймворке Yii2 [4]. Д.С. Жаворонков, А.А. Бабкина, Е.Ю. Довгий в своей статье рассмотрели достоинства фреймворка Yii2 в разработке REST систем [5].

Целью данной статьи является использование инструмента DB, встроенного в PHP фреймворк Yii2 и применение его в работе с таблицами базы данных. Также будет разработан пример с использованием данного инструмента на редактирование тестируемой таблицы базы данных.

Базы данных всегда для корректной работы требуется рабочий сервер, поэтому в качестве примера будет использоваться локальный веб-сервер OpenServer [6], распространяющей в свободном доступе во всемирной паутине. Также потребуется временно программа по работе с базами данных Heidi SQL [7] для того, чтобы создать базу данных и таблицу, которая будет редактироваться встроенным инструментом DB в PHP фреймворке Yii2.

Для начала необходимо в программе Heidi SQL создать базу данных под названием «test» и в ней таблицу с названием «users» (пользователи). Структура таблицы «users» будет продемонстрирована ниже

#	Имя	Тип данных	Длина/Знач...	Беззна...	Разреш...	Zerofill	По умолчанию
1	id	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT...
2	name	VARCHAR	50	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	'0'
3	age	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

Рисунок 1 – Структура таблицы «users»

На рисунке 1 изображена структура таблицы «users». В ней имеются столбцы: id – идентификатор пользователя, name – имя пользователя, age – возраст пользователя. Первый столбец обладает параметром авто инкремента и содержит первичный ключ. Далее перейдем к использованию PHP фреймворка Yii2 [8] на локальном сервере. Сначала нам надо настроить подключение к базе данных, для этого перейдем в корневую директорию PHP фреймворка Yii2, далее откроем папку «config» и в ней будет файл «db.php», в котором и содержатся настройки подключения к базе данных, данный файл надо открыть и изменить на следующее

```
<?php
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=test',
    'username' => 'root',
    'password' => 'root',
    'charset' => 'utf8',
];
```

Рисунок 2 – Настройки подключения к базе данных

На рисунке 2 изображены настройки соединения к базе данных. Теперь рассмотрим каждый параметр более подробно. Параметр «class» отвечает за название компонента, который будет отвечать за соединение к базе данных. Параметр «dsn» содержит настройки хоста «host=localhost» и название базы данных «dbname=test». Параметры «username» и «root» собственно являются именем пользователя и паролем, а параметр «charset» отвечает за кодировку базы данных в «utf-8».

После того, как был правильно настроен файл «db.php», то перейдем уже к использованию самого встроенного инструмента DB, для этого снова нужно перейти в корневую директорию PHP фреймворка Yii2 и перейти по пути: views/site и открыть файл «index.php» и все его содержимое заменить на следующий код

```
<?php
/* @var $this yii\web\View */
$this->title = 'My Yii Application';
?>
<div class="site-index">
  <h2>Таблица пользователей "users"</h2>
  <table border="1">
    <tr>
      <th>ID</th>
      <th>Имя</th>
      <th>Возраст</th>
    </tr>
    <tr>
      <td><?php
        Yii::$app->db->createCommand()->batchInsert('users', ['name', 'age'], [
          ['Михаил', 23],
          ['Евгений', 45],
          ['Светлана', 43],
          ['Борис', 37],
          ['Александра', 22],
          ['Олег', 38],
          ['Георгий', 33],
          ['Вероника', 25],
          ['Анастасия', 18],
          ['Дмитрий', 27],
          ['Марина', 47],
          ['Екатерина', 44],
          ['Ольга', 40],
          ['Андрей', 39],
          ['Николай', 58],
        ])->execute();
      </td>
    </tr>
  </table>
</div>
```

Рисунок 3 – PHP код добавления данных пользователей в таблицу «users»

На рисунке 3 показан PHP код, в котором используется в инструменте DB метод «createCommand()», который позволяет формировать SQL запросы с передачей активного метода «batchInsert()», позволяющий добавить в таблицу «users» данные пользователей их имена и возраст. В конце метод «execute()» вызывает выполнение данного SQL запроса и занесение данных в таблицу «users» базы данных «test».

Теперь нужно вернуться в программу Heidi SQL и посмотреть содержимое таблицы «users»

id	name	age
1	Михаил	23
2	Евгений	45
3	Светлана	43
4	Борис	37
5	Александра	22
6	Олег	38
7	Георгий	33
8	Вероника	25
9	Анастасия	18
10	Дмитрий	27
11	Марина	47
12	Екатерина	44
13	Ольга	40
14	Андрей	39
15	Николай	58

Рисунок 4 – Содержимое добавленных пользователей в таблице «users»

Как видно на рисунке 4 представлена таблица «users» с добавленными пользователями, что были указаны в запросе на рисунке 3. Далее теперь предстоит использовать в инструменте DB возможность отображать содержимое таблицы на самой веб-странице, для этого нужно в том же файле «index.php» вписать следующий код

```
<?php
/* @var $this yii\web\View */
$this->title = 'My Yii Application';
?>
<div class="site-index">
    <h2>Таблица пользователей "users"</h3>
    <table border=1>
        <tr>
            <th>ID</th>
            <th>Имя</th>
            <th>Возраст</th>
        </tr>

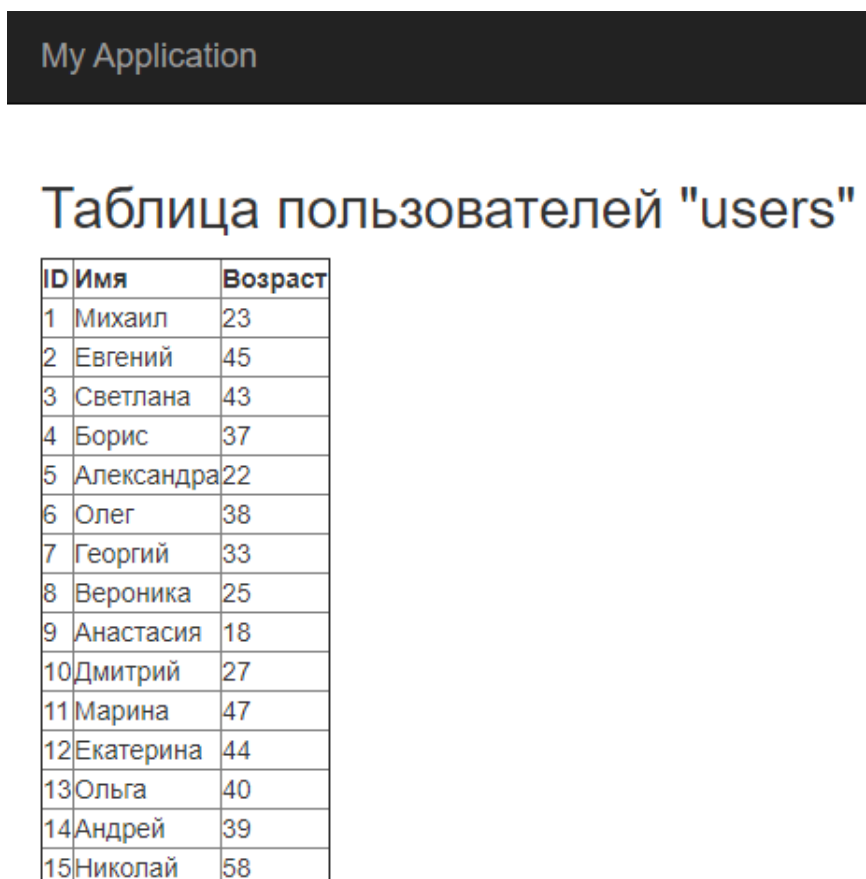
        <?php
        $count = Yii::$app->db->createCommand('SELECT COUNT(*) FROM users')
            ->queryScalar();
        $users = Yii::$app->db->createCommand('SELECT * FROM users')
            ->queryAll();

        for ($i = 0; $i < $count; $i++) {
            echo '<tr>
                <td>'.$users[$i]['id'].'</td>
                <td>'.$users[$i]['name'].'</td>
                <td>'.$users[$i]['age'].'</td>
            </tr>';
        }
        ?>
    </table>
</div>
```

Рисунок 5 – PHP код отображения таблицы «users» на веб-странице

На рисунке 5 показан PHP код, который позволяет на веб-страницу отобразить саму таблицу «users». В самом коде используются две переменные: \$count – количество строк в таблице «users» (метод «queryScalar()»), \$users – массив всех пользователей таблицы «users» (метод «queryAll()»). Счетчик «for» позволяет перебрать полностью весь массив переменной \$users и построчно отобразить в таблице все данные на веб-странице.

Теперь откроем данную веб-страницу с помощью веб-браузера и посмотрим результат



The screenshot shows a web browser window with a dark header bar containing the text "My Application". Below the header, the title "Таблица пользователей "users"" is displayed in a large, bold font. Underneath the title is a table with three columns: "ID", "Имя", and "Возраст". The table contains 15 rows of user data.

ID	Имя	Возраст
1	Михаил	23
2	Евгений	45
3	Светлана	43
4	Борис	37
5	Александра	22
6	Олег	38
7	Георгий	33
8	Вероника	25
9	Анастасия	18
10	Дмитрий	27
11	Марина	47
12	Екатерина	44
13	Ольга	40
14	Андрей	39
15	Николай	58

Рисунок 6 – Таблица «users» на веб-странице PHP фреймворка Yii2

На рисунке 6 показан результат работы инструмента DB, который отобразил на веб-странице саму таблицу «users» и ее содержимое полностью совпадает с содержимым, что в таблице «users» на рисунке 4.

Далее рассмотрим у DB инструмента возможности обновления и удаления данных в таблице, для этого снова перейдем к файлу «index.php» и напишем следующий код

```

<?php
/* @var $this yii\web\View */
$this->title = 'My Yii Application';
?>
<div class="site-index">
  <h2>Таблица пользователей "users"</h3>
  <table border=1>
    <tr>
      <th>ID</th>
      <th>Имя</th>
      <th>Возраст</th>
    </tr>

    <?php
    Yii::$app->db->createCommand()->update('users', ['age' => 50], ['name' => "Михаил"])->execute();
    Yii::$app->db->createCommand()->update('users', ['age' => 50], ['name' => "Борис"])->execute();
    Yii::$app->db->createCommand()->update('users', ['age' => 50], ['name' => "Дмитрий"])->execute();
    Yii::$app->db->createCommand()->update('users', ['age' => 50], ['name' => "Екатерина"])->execute();
    Yii::$app->db->createCommand()->update('users', ['age' => 50], ['name' => "Вероника"])->execute();

    $count = Yii::$app->db->createCommand('SELECT COUNT(*) FROM users')
      ->queryScalar();
    $users = Yii::$app->db->createCommand('SELECT * FROM users')
      ->queryAll();
    for ($i = 0; $i < $count; $i++) {
      echo '<tr>
        <td>'.$users[$i]['id'].'</td>
        <td>'.$users[$i]['name'].'</td>
        <td>'.$users[$i]['age'].'</td>
      </tr>';
    }
  </table>
</div>

```

Рисунок 7 – PHP код обновления данных пользователей в таблице «users»

На рисунке 7 продемонстрирован PHP код, в котором используется метод «update()», позволяющий в таблице «users» обновить строки определенных пользователей и установить их возраст равный 50 годам. Теперь снова обновив веб-страницу, увидим следующий результат

My Application

Таблица пользователей "users"

ID	Имя	Возраст
1	Михаил	50
2	Евгений	45
3	Светлана	43
4	Борис	50
5	Александра	22
6	Олег	38
7	Георгий	33
8	Вероника	50
9	Анастасия	18
10	Дмитрий	50
11	Марина	47
12	Екатерина	50
13	Ольга	40
14	Андрей	39
15	Николай	58

Рисунок 8 – Измененная таблица «users» с обновленным возрастом пользователей

Рассматривая рисунок 8, можно увидеть результат работы обновления данных. У пользователей с ID - 1, 4, 8, 10 и 12 возраст теперь равен 50 годам. Далее осталось рассмотреть возможности удаления строк в таблице, используя встроенный инструмент DB. Снова откроем файл «index.php» и впишем следующий код

```
<?php

/* @var $this yii\web\View */

$this->title = 'My Yii Application';
?>
<div class="site-index">
    <h2>Таблица пользователей "users"</h3>
    <table border=1>
        <tr>
            <th>ID</th>
            <th>Имя</th>
            <th>Возраст</th>
        </tr>

        <?php
        Yii::$app->db->createCommand()->delete('users', 'age < 30')->execute();

        $count = Yii::$app->db->createCommand('SELECT COUNT(*) FROM users')
            ->queryScalar();

        $users = Yii::$app->db->createCommand('SELECT * FROM users')
            ->queryAll();

        for ($i = 0; $i < $count; $i++) {
            echo '<tr>
                <td>'. $users[$i]['id']. '</td>
                <td>'. $users[$i]['name']. '</td>
                <td>'. $users[$i]['age']. '</td>
            </tr>';
        }
    ?>
    </table>
</div>
```

Рисунок 9 – PHP код, удаляющий в таблице «users» пользователей, которые младше 30 лет

На рисунке 9 показан метод «delete()», который позволяет удалить пользователей из таблицы «users», которые младше 30 лет. Напоследок осталось обновить страницу и посмотреть окончательный результат измененной таблицы «users»

Таблица пользователей "users"

ID	Имя	Возраст
1	Михаил	50
2	Евгений	45
3	Светлана	43
4	Борис	50
6	Олег	38
7	Георгий	33
8	Вероника	50
10	Дмитрий	50
11	Марина	47
12	Екатерина	50
13	Ольга	40
14	Андрей	39
15	Николай	58

Рисунок 10 – Таблица «users» с удаленными пользователями, которые младше 30 лет

Рисунок 10 показывает окончательный результат работы инструмента DB и в таблице видно, что удалены пользователи с ID – 5 и 9, потому что у них возраст был меньше 30.

Таким образом, был рассмотрен в PHP фреймворке Yii2 встроенный инструмент DB, позволяющий работать с таблицами базы данных, а именно их редактировать, добавлять, обновлять или удалять данные в них. Также был разработан собственный пример, в котором наглядно показана работа инструмента DB и он полностью выполнил свою задачу по работе с таблицами в базе данных.

Библиографический список

1. Оленников А.В. Разработка панели администратора сайта на фреймворке Yii2 // В сборнике: Молодые ученые в решении актуальных проблем науки. Сборник материалов Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых. 2019. С. 510-512.
2. Филимонов И.В., Разин И.Б. Разработка платформы для создания сайта типа интернет-магазин на базе Yii2 framework // В сборнике: Инновационное развитие легкой и текстильной промышленности" (Интекс-2018). Сборник материалов Международной научной студенческой конференции. 2018. С. 111-113.
3. Салихов Д.Р., Халиков А.Р. Разработка веб-приложения для складского учета на базе фреймворка Yii2 // В сборнике: Инновации в науке и практике. Сборник статей по материалам X международной научно-практической конференции. В 4 частях. Ответственный редактор Халиков

- А.Р.. 2018. С. 192-196.
4. Кочитов М.Е. Использование плагинов ActiveRecord и Query для управления базами данных в PHP фреймворке Yii2 // Постулат. 2020. № 1 (51). С. 40.
 5. Жаворонков Д.С., Бабкина А.А., Довгий Е.Ю. Достоинства фреймворка Yii2 в разработке REST систем // В сборнике: "Чистая наука" на службе научно-технического прогресса. Сборник статей по итогам Международной научно-практической конференции. 2018. С. 9-11.
 6. Open Server Panel – Локальный веб-сервер для Windows URL: <https://ospanel.io/> (дата обращения 24.08.2020)
 7. Heidi SQL // URL: <https://www.heidisql.com/> (дата обращения 24.08.2020)
 8. PHP framework Yii2 URL: <https://www.yiiframework.com/> (дата обращения 24.08.2020)
 9. Работа с базами данных – Yii2 URL: <https://www.yiiframework.com/doc/guide/2.0/ru/db-dao> (дата обращения 24.08.2020)