

## Использование плагинов ActiveRecord и Query для управления базами данных в PHP фреймворке Yii2

*Кочитов Михаил Евгеньевич*

*Приамурский государственный университет им. Шолом-Алейхема  
студент*

### Аннотация

В данной статье рассматривается работа двух плагинов ActiveRecord и Query, позволяющих в PHP фреймворке Yii2 работать с базами данных. Также будет показан пример работы этих двух плагинов на локальной базе данных с добавлением, обновлением и удалением в ней данных, используя PHP фреймворк Yii2.

**Ключевые слова:** плагины, ActiveRecord, Query, база данных, PHP фреймворк, Yii2

## Using ActiveRecord and Query plugins to manage databases in the PHP Yii2 framework

*Kochitov Mikhail Evgenevich*

*Sholom-Aleichem Priamursky State University  
student*

### Abstract

This article discusses the work of two ActiveRecord and Query plugins that allow you to work with databases in the Yii2 PHP framework. An example of how these two plug-ins work on a local database with adding, updating and deleting data in it using the PHP Yii2 framework will also be shown.

**Keywords:** plugins, ActiveRecord, Query, database, PHP framework, Yii2

В PHP Фреймворке Yii2 имеются все инструменты разработки веб-сайта любой сложности. Такими веб-сайтами могут быть, например сайт новостей, онлайн магазин, сайт ведения блогов, социальная сеть и так далее. Однако выбирая для разработки веб-проекта данный фреймворк, то необходимо знать хорошо язык программирования PHP и немного JavaScript, так как PHP фреймворки не имеют встроенного визуального конструктора сайтов, все создается с нуля кодом. Также данный PHP фреймворк Yii2 имеет инструменты работы с базами данными, такими являются два плагина ActiveRecord и Query, которые будут рассмотрены в данной статье с примерами данных на локальной базе данных.

В статье Пороховой М.В. и Дударевой О.В. проводится обзор фреймворка Yii2 как основного инструмента для разработки интернет-магазина [1]. Рассматривая статью Оленникова А.В. можно увидеть

разработку панели администратора сайта на фреймворке Yii2 [2]. Филимонов И.В. и Разин И.Б. в своей статье рассмотрели разработку платформы для создания сайта типа интернет-магазин на базе Yii2 Framework [3]. В статье Масолюкина П.Ю. и Александрова С.Ю. рассматривается разработка серверного приложения на базе фреймворка Yii2 [4]. Салихов Д.Р. и Халиков А.Р. в своей статье рассмотрели разработку веб-приложения для складского учета на базе фреймворка Yii2 [5].

При выборе PHP фреймворка Yii2 [6] нужно учитывать, что он работает только на сервере, поэтому будет использован локальный сервер «OpenServer» [7], предоставляемый в свободном доступе и также будет использован инструмент «HeidiSQL» [8], позволяющий работать с базами данных. Теперь перейдем к программному обеспечению «HeidiSQL» и создадим тестовую базу данных с таблицей под названием «employees», в которой будут находиться данные о сотрудниках

#	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_INCREME...
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
4	profession	VARCHAR	50	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0

Рисунок 1 – Таблица «employees»

На рисунке 1 представлена таблица «employees», в которой имеются 4 столбца: id – уникальный идентификатор, name – имя сотрудника, age – возраст сотрудника и profession – профессия сотрудника. Столбец id должен индексироваться первичным ключом и обладать параметром «Auto\_Increment», позволяющим увеличивать каждый раз на один идентификатор при добавлении в эту таблицу записей.

Далее перейдем непосредственно к самому PHP фреймворку Yii2, в котором необходимо создать два файла с расширением PHP по пути: «localhost/basic/models/Employees.php» и «localhost/basic/views/site/Employees.php». Теперь нужно открыть файл контроллера «localhost/basic/controllers/SiteController.php» и нужно в нем написать выделенный следующий код, изображенный на рисунках ниже

```
<?php
namespace app\controllers;

use Yii;
use yii\filters\AccessControl;
use yii\web\Controller;
use yii\web\Response;
use yii\filters\VerbFilter;
use app\models\LoginForm;
use app\models\ContactForm;
use app\models\Employees;

class SiteController extends Controller
```

Рисунок 2 – Добавление модели «Employees»

```
public function actionEmployees ()  
{  
    return $this->render('employees');  
}
```

Рисунок 3 – Добавление функции «actionEmployees»

На рисунке 2 показана выделенная строчка, позволяющая в контроллер добавить модель «Employees», которая необходимо в дальнейшем для управления таблицей «employees» в базе данных. На рисунке 3 изображена выделенная функция, которая дает возможность фреймворку отобразить содержимое страницы «Employees» и выполнять в ней различные операции, связанные в нашем случае с управлением базой данных и таблицей «Employees».

Далее необходимо в модель «Employees» (файл по пути: «localhost/basic/models/Employees.php») написать следующий стартовый код

```
<?php  
namespace app\models;  
use yii\db\ActiveRecord;  
  
class Employees extends ActiveRecord  
{  
    public static function tableName()  
    {  
        return '{{employees}}';  
    }  
}
```

Рисунок 4 – Начальный код в модели «Employees» с подключенным плагином ActiveRecord

На рисунке 4 продемонстрирован начальный код модели «Employees», в котором подключается плагин ActiveRecord, который позволяет управлять данными в таблице «employees». Эта таблица также указана в функции «tableName()» после оператора return. Далее потребуются в этот же файл написать следующий код, который позволяет добавлять записи в таблицу (см. рис. 5)

```
public function saveEmployee($name, $age, $profession) {  
  
    $new_employee = new Employees();  
    $new_employee->name = $name;  
    $new_employee->age = $age;  
    $new_employee->profession = $profession;  
    $new_employee->save();  
}
```

Рисунок 5 – Функция «saveEmployee», позволяющая добавлять новые записи в таблицу «employees»

Далее нужно добавить код, выполняющий работу обновление записей сотрудников (см. рис. 6)

```
public function updateEmployee($id_employee, $name, $age, $profession) {  
  
    $new_employee = Employees::findOne(['id' => $id_employee]);  
    $new_employee->name = $name;  
    $new_employee->age = $age;  
    $new_employee->profession = $profession;  
    $new_employee->save();  
  
}
```

Рисунок 6 – Функция «updateEmployee», позволяющая обновлять записи в таблице «employees»

Теперь осталось добавить код с функциями, которые предназначены для удаления записей сотрудников из таблицы «employees»

```
public function deleteEmployee($id_employee) {  
  
    $new_employee = Employees::findOne(['id' => $id_employee]);  
    $new_employee->delete();  
  
}  
  
public function deleteEmployeesName($name) {  
  
    Employees::deleteAll(['name' => $name]);  
  
}  
  
public function deleteEmployeesProfession($profession) {  
  
    Employees::deleteAll(['profession' => $profession]);  
  
}  
  
public function deleteAllEmployees() {  
  
    Employees::deleteAll();  
  
}
```

Рисунок 7 – Функции удаления записей из таблицы «employees»

На рисунке 7 изображены четыре функции, которые должны удалять записи сотрудников из таблицы «employees». Теперь рассмотрим, что делает каждая функция. Функция «deleteEmployee» удаляет одного сотрудника, указав в аргументе функции его идентификатор. «deleteEmployeesName» - удаляет всех сотрудников, у которых имеется определенное имя. «deleteEmployeesProfession» - удаляет всех сотрудников, у которых имеется определенная профессия. Функция «deleteAllEmployees» удаляет всех сотрудников, оставляя таблицу «employees» полностью пустой.

Теперь перейдем к файлу контроллера и к функции «actionEmployee» (см. рис. 3) и напишем в нем следующий код

```

public function actionEmployees ()
{
    $employee = new Employees ();
    $employee->saveEmployee ("Никита", 35, "Машинист");
    $employee->saveEmployee ("Александр", 28, "Программист");
    $employee->saveEmployee ("Андрей", 32, "Учитель");
    $employee->saveEmployee ("Денис", 44, "Инженер");
    $employee->saveEmployee ("Николай", 24, "Художник");
    $employee->saveEmployee ("Михаил", 27, "Программист");
    $employee->saveEmployee ("Алексей", 38, "Учитель");
    $employee->saveEmployee ("Владимир", 54, "Инженер");
    $employee->saveEmployee ("Никита", 36, "Программист");
    $employee->saveEmployee ("Денис", 51, "Машинист");
    $employee->saveEmployee ("Михаил", 29, "Учитель");

    return $this->render ('employees');
}

```

Рисунок 8 – Добавление сотрудников в таблицу «employees»

На рисунке 8 показан код, в котором задействована функция «saveEmployee», выполняющая работу добавления нового сотрудника в таблицу «employees», указав его имя, возраст и профессию. Теперь глянем в программе «HeidiSQL» содержимое таблицы «employees»

id	name	age	profession
1	Никита	35	Машинист
2	Александр	28	Программист
3	Андрей	32	Учитель
4	Денис	44	Инженер
5	Николай	24	Художник
6	Михаил	27	Программист
7	Алексей	38	Учитель
8	Владимир	54	Инженер
9	Никита	36	Программист
10	Денис	51	Машинист
11	Михаил	29	Учитель

Рисунок 9 – Содержимое таблицы «employees» с добавленными сотрудниками

Как можно заметить, на рисунке 9 представлен результат работы функции «saveEmployee», которая добавила в таблицу «employees» всех сотрудников. Далее рассмотрим использование функции «updateEmployee», которая дает возможность обновлять записи у сотрудников в таблице

```

public function actionEmployees ()
{
    $employee = new Employees ();
    $employee->updateEmployee (6, "Ярослав", 43, "Повар");
    $employee->updateEmployee (3, "Сергей", 27, "Фермер");
    $employee->updateEmployee (7, "Олег", 27, "Врач");

    return $this->render ('employees');
}

```

Рисунок 10 – Обновление сотрудников в таблице «employees»

На рисунке 10 изображен код с функциями «updateEmployee», выполняющие работу обновление сотрудников, указав первым параметром идентификаторы сотрудников, которые подлежат замене. После рассмотрения данной функции перейдем в программу «HeidiSQL» и посмотрим изменения в таблице «employees»

🔑 id	name	age	profession
1	Никита	35	Машинист
2	Александр	28	Программист
3	Сергей	27	Фермер
4	Денис	44	Инженер
5	Николай	24	Художник
6	Ярослав	43	Повар
7	Олег	27	Врач
8	Владимир	54	Инженер
9	Никита	36	Программист
10	Денис	51	Машинист
11	Михаил	29	Учитель

Рисунок 11 – Содержимое таблицы «employees» с обновленными тремя сотрудниками

Как можно заметить на рисунке 11 в таблице были заменены сотрудниками с идентификаторами 3, 6, и 7, которым присвоено новое имя и новая профессия сотрудника. Теперь перейдем к рассмотрению четырех функций удаления сотрудников и разберем каждую из них на примере

```
public function actionEmployees()
{
    $employee = new Employees();
    $employee->deleteEmployee(11);
    $employee->deleteEmployee(10);

    return $this->render('employees');
}
```

Рисунок 12 – Удаление сотрудников с указанными идентификаторами

🔑 id	name	age	profession
1	Никита	35	Машинист
2	Александр	28	Программист
3	Сергей	27	Фермер
4	Денис	44	Инженер
5	Николай	24	Художник
6	Ярослав	43	Повар
7	Олег	27	Врач
8	Владимир	54	Инженер
9	Никита	36	Программист

Рисунок 13 – Содержимое таблицы «employees» с удаленными двумя последними сотрудниками

```
public function actionEmployees ()
{
    $employee = new Employees ();
    $employee->deleteEmployeesName ("Никита");

    return $this->render ('employees');
}
```

Рисунок 14 – Удаление сотрудников с именем «Никита»

id	name	age	profession
2	Александр	28	Программист
3	Сергей	27	Фермер
4	Денис	44	Инженер
5	Николай	24	Художник
6	Ярослав	43	Повар
7	Олег	27	Врач
8	Владимир	54	Инженер

Рисунок 15 - Содержимое таблицы «employees» с удаленными сотрудниками, у которых имя «Никита»

```
public function actionEmployees ()
{
    $employee = new Employees ();
    $employee->deleteEmployeesProfession ("Инженер");

    return $this->render ('employees');
}
```

Рисунок 16 – Удаление сотрудников с профессией «Инженер»

id	name	age	profession
2	Александр	28	Программист
3	Сергей	27	Фермер
5	Николай	24	Художник
6	Ярослав	43	Повар
7	Олег	27	Врач

Рисунок 17 - Содержимое таблицы «employees» с удаленными сотрудниками, у которых профессия «Инженер»

```
public function actionEmployees ()
{
    $employee = new Employees ();
    $employee->deleteAllEmployees ();

    return $this->render ('employees');
}
```

Рисунок 18 – Удаление всех сотрудников



id	name	age	profession
----	------	-----	------------

Рисунок 19 – Пустая таблица «employees» с удаленными всеми сотрудниками

На рисунках 12-19 показан полностью процесс работы четырех функций удаления сотрудников из таблицы «employees». Сначала было удаление сотрудников по идентификаторам, затем удаление по имени, после удаление по профессии и напоследок удаление всех сотрудников из самой таблицы.

Далее осталось рассмотреть плагин Query, позволяющий на веб-странице PHP фреймворка Yii2 отображать данные из таблицы базы данных. Для этого необходимо открыть созданный ранее файл по пути «localhost/basic/views/site/Employes.php» и вписать в него следующий код

```
<?php
use yii\db\Query;

$total_employees = (new Query())
    ->from('employees')
    ->count();

$employees = (new Query())
    ->from('employees')
    ->all();

echo "<table border=1>";
echo "<tr>";
echo "<th>id</th><th>name</th><th>age</th><th>profession</th>";
echo "</tr>";
for ($i = 0; $i < $total_employees; $i++) {
    echo "<tr>";
    echo "<td>" . $employees[$i]['id'] . "</td>";
    echo "<td>" . $employees[$i]['name'] . "</td>";
    echo "<td>" . $employees[$i]['age'] . "</td>";
    echo "<td>" . $employees[$i]['profession'] . "</td>";
    echo "</tr>";
}
echo "</table>";
?>
```

Рисунок 20 – PHP код подключения плагина Query и вывод содержимого таблицы «employees» на веб-страницу

Стоит заметить, что на данный момент таблица пустая, поэтому следует в нее обратно добавить данные о сотрудниках, используя код на рисунке 8. После возвращения в таблицу «employees» исходных сотрудников можно рассмотреть рисунок 20 и изображенный в нем PHP код. Изначально загружается плагин Query, затем в переменную \$total\_employees заносится число всех сотрудников в таблице, далее в переменную \$employees заносится полностью сформированный массив данных всех сотрудников из таблицы

«employees». После используется цикл for и функция echo для отображения данных, оформленных в таблицу (см. рис. 21)



The image shows a screenshot of a web application with a black header bar containing the text "My Application" in white. Below the header is a table with 11 rows and 4 columns. The columns are labeled "id", "name", "age", and "profession". The data in the table is as follows:

id	name	age	profession
1	Никита	35	Машинист
2	Александр	28	Программист
3	Андрей	32	Учитель
4	Денис	44	Инженер
5	Николай	24	Художник
6	Михаил	27	Программист
7	Алексей	38	Учитель
8	Владимир	54	Инженер
9	Никита	36	Программист
10	Денис	51	Машинист
11	Михаил	29	Учитель

Рисунок 21 – Таблица сотрудников на веб-странице PHP фреймворка Yii2 с базовым названием «My Application»

Таким образом, были рассмотрены два плагина ActiveRecord и Query, которые позволяли управлять базами данных, используя PHP фреймворк Yii2. Также был приведен пример работы этих плагинов на локальной базе данных, используя таблицу сотрудников и изменения в них данных. Можно предположить, что PHP фреймворк Yii2 имеет огромный инструментальный набор для разработки многофункциональных веб-сайтов и встроенные в него плагины ActiveRecord и Query действительно дают колоссальные возможности для работы с данными в базе данных.

### Библиографический список

1. Порохова М.В., Дударева О.В. Обзор фреймворка Yii2 как основного инструмента для разработки интернет-магазина // В сборнике: Актуальные проблемы химии, биотехнологии и сферы услуг Филатова Е.Г. III Всероссийская научно-практическая конференция с международным участием. Главный редактор: Анциферов Е.А. Технический редактор: Степанова М.В.. 2019. С. 251-255.
2. 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.
3. 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.
4. Масолькин П.Ю., Александров С.Ю. Разработка серверного приложения на базе фреймворка Yii2 // В сборнике: Научно-техническое творчество аспирантов и студентов Материалы всероссийской научно-технической конференции студентов и аспирантов: в 2 частях. 2018. С. 214-216.
  5. 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.
  6. Yii2 Framework URL: <https://www.yiiframework.com/> (дата обращения 15.01.2020)
  7. Open Sever Panel – Локальный веб сервер для Windows URL: <https://ospanel.io/> (дата обращения 15.01.2020)
  8. HeidiSQL URL: <https://www.heidisql.com/> (дата обращения 15.01.2020)