

Интеграция API в веб-приложение на базе фреймворка Yii2

Андриенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается применение API OpenWeatherMap в рамках веб-приложения, созданного с использованием фреймворка Yii2. В работе подробно описан процесс настройки взаимодействия с внешним API, включая создание необходимых компонентов приложения: контроллера, модели и представления. Особое внимание уделено методам обработки и визуализации данных о погоде, что позволяет легко адаптировать предложенное решение для своих проектов. Статья будет полезна специалистам, занимающимся разработкой веб-приложений с интеграцией внешних сервисов.

Ключевые слова: Yii2, OpenWeatherMap, веб-разработка, API, PHP, метеорологические данные, RESTful API, фреймворк, интеграция сервисов.

API integration into a web application based on the Yii2 framework

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the use of the OpenWeatherMap API in a web application created using the Yii2 framework. The paper describes in detail the process of configuring interaction with the external API, including creating the necessary application components: controller, model, and view. Special attention is paid to methods of processing and visualizing weather data, which makes it easy to adapt the proposed solution for your projects. This article will be useful for specialists involved in the development of web applications with the integration of external services.

Keywords: Yii2, OpenWeatherMap, web development, API, PHP, meteorological data, RESTful API, framework, service integration.

1 Введение

1.1 Актуальность

Современные веб-приложения активно используют внешние API для расширения своей функциональности и предоставления пользователям актуальной информации. Одним из востребованных направлений является интеграция сервисов, предоставляющих данные о погоде, таких как OpenWeatherMap. Фреймворк Yii2, благодаря своей производительности и

удобству, позволяет эффективно реализовывать подобные решения. Умение работать с внешними API становится важным навыком для разработчиков, так как это открывает возможности для создания более сложных и полезных приложений. Данная статья предлагает пошаговое руководство по интеграции OpenWeatherMap API в Yii2, что делает её актуальной для специалистов в области веб-разработки.

1.2 Обзор исследований

В исследовании Ю. М. Закировой рассматриваются особенности разработки REST API сервиса с использованием фреймворка Yii2, где особое внимание уделяется его применению для создания масштабируемых и производительных API. В работе показано, как Yii2 упрощает процесс разработки RESTful-сервисов благодаря встроенным инструментам для маршрутизации, аутентификации и сериализации данных [1]. Т. Н. Филимонова исследует архитектурные особенности проектирования и разработки веб-приложений, рассматривая Yii2 как один из ключевых фреймворков для создания сложных веб-приложений. В работе отмечается, что Yii2 предоставляет мощные инструменты для интеграции с внешними API, что делает его универсальным решением для современных проектов [2]. М. Ф. Гайнутдинов, А. С. Шкуркин и А. А. Пичугина в своей работе описывают разработку back-end для сервисов, ориентированных на интернационализацию веб-приложений. Авторы показывают, как Yii2 упрощает работу с многоязычными приложениями, предоставляя встроенные механизмы для локализации и перевода, что делает его эффективным инструментом для создания приложений, ориентированных на международную аудиторию [3]. А. А. Беляев и В. А. Глаголев рассматривают практический пример интеграции внешнего API в Yii2, а именно использование Яндекс API для добавления функциональности "Поделиться" в веб-приложение. В работе демонстрируется, как Yii2 упрощает процесс интеграции внешних сервисов благодаря своей модульной архитектуре и поддержке RESTful-запросов [4]. И. В. Дашенко и Е. В. Фешина анализируют современные средства разработки веб-ориентированных информационных систем, рассматривая Yii2 как один из наиболее подходящих фреймворков для создания информационных систем. В работе подчеркивается, что Yii2 позволяет разработчикам создавать гибкие и масштабируемые решения благодаря поддержке интеграции с внешними API и современными стандартами веб-разработки [5].

1.3 Цель исследования

Цель исследования – разработка и реализация веб-приложения на базе фреймворка Yii2 с интеграцией API OpenWeatherMap для получения и отображения данных о погоде. В рамках исследования планируется продемонстрировать процесс подключения внешнего API к Yii2, настройки взаимодействия с ним, а также создания пользовательского интерфейса для отображения метеорологических данных.

2 Материалы и методы

Для выполнения работы использовались следующие инструменты и технологии:

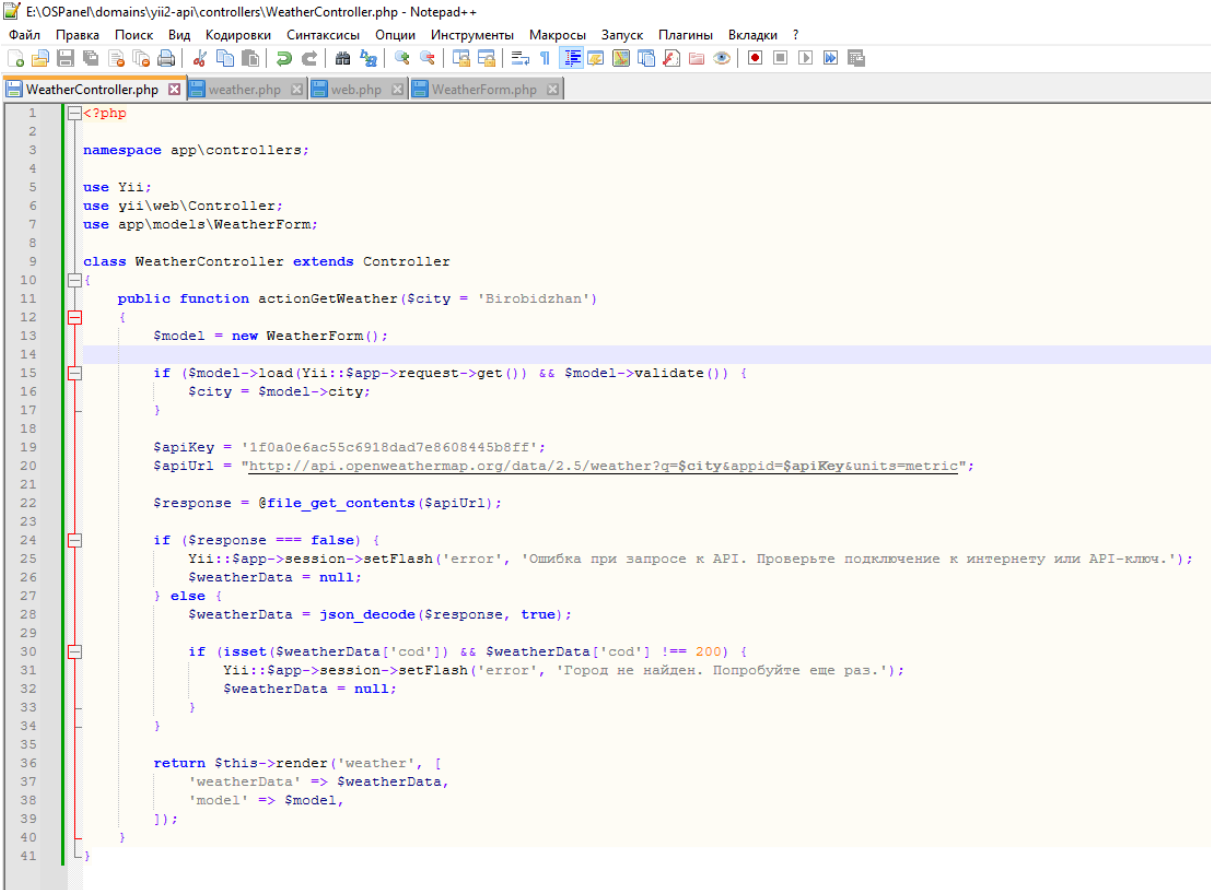
1. Фреймворк Yii2 для разработки веб-приложения.
2. API OpenWeatherMap для получения данных о погоде.
3. Язык программирования PHP для реализации логики приложения.
4. Локальный сервер на базе OpenServer для разработки и тестирования.

В процессе работы были созданы контроллер, модель и представление для обработки и отображения данных о погоде.

3 Результаты и обсуждения

Разработка веб-приложения для получения и отображения данных о погоде с использованием фреймворка Yii2 и API OpenWeatherMap включает несколько этапов. Подробно описывается процесс создания контроллера, модели и представления.

Первым шагом в разработке приложения стало создание контроллера WeatherController, который отвечает за обработку запросов к API OpenWeatherMap и передачу данных в представление. Контроллер был создан с помощью встроенного генератора кода Yii2. Внутри контроллера было определено действие actionGetWeather, которое принимает название города в качестве параметра. Если город не указан, по умолчанию используется значение "Birobidzhan" (рис. 1).

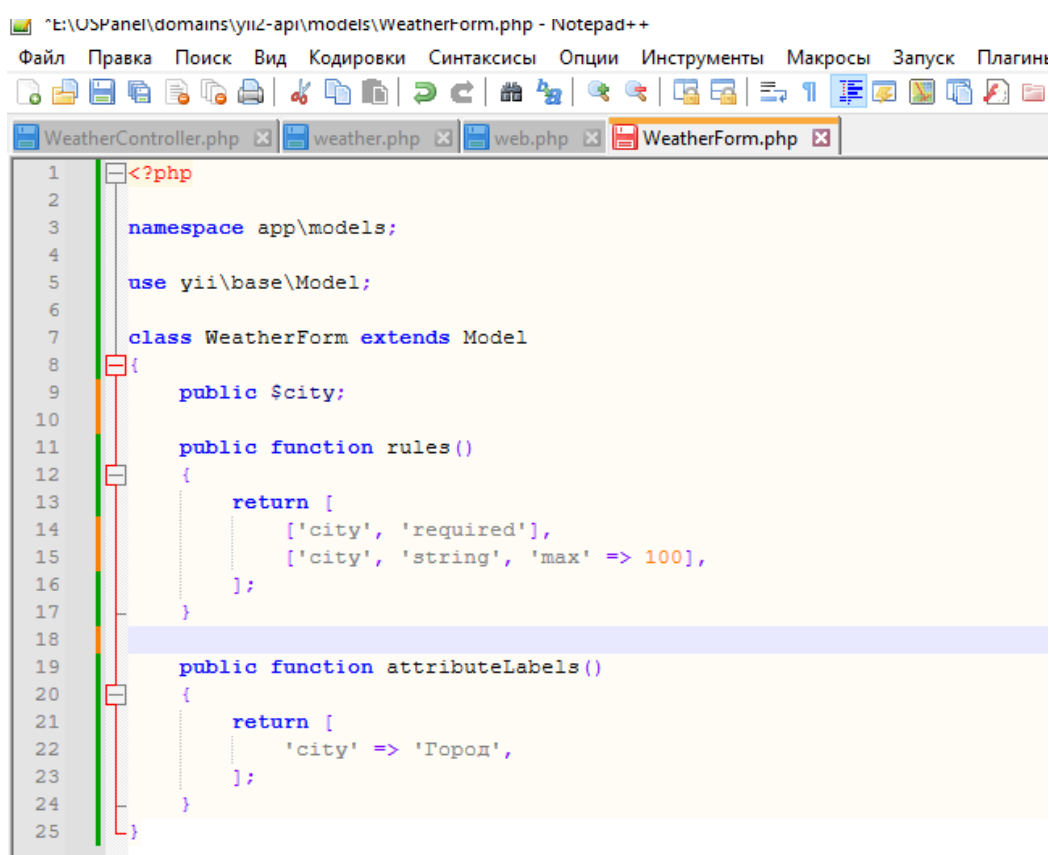


```
1 <?php
2
3 namespace app\controllers;
4
5 use Yii;
6 use yii\web\Controller;
7 use app\models\WeatherForm;
8
9 class WeatherController extends Controller
10 {
11     public function actionGetWeather($city = 'Birobidzhan')
12     {
13         $model = new WeatherForm();
14
15         if ($model->load(Yii::$app->request->get()) && $model->validate()) {
16             $city = $model->city;
17         }
18
19         $apiKey = '1f0a0e6ac55c6918dad7e8608445b8ff';
20         $apiUrl = "http://api.openweathermap.org/data/2.5/weather?q=$city&appid=$apiKey&units=metric";
21
22         $response = @file_get_contents($apiUrl);
23
24         if ($response === false) {
25             Yii::$app->session->setFlash('error', 'Ошибка при запросе к API. Проверьте подключение к интернету или API-ключ.');
```

Рис. 1. Код контролера WeatherController.php

В этом коде создается экземпляр модели WeatherForm, которая используется для обработки ввода пользователя. Если данные формы были отправлены и прошли валидацию, название города обновляется. Затем формируется URL для запроса к API OpenWeatherMap, где указывается название города, API-ключ и единицы измерения (в данном случае — метрические). Выполняется HTTP-запрос с помощью функции `file_get_contents()`, и полученные данные декодируются из JSON-формата в массив. Если запрос завершился ошибкой, пользователю выводится соответствующее сообщение. В случае успеха данные передаются в представление `weather` для отображения.

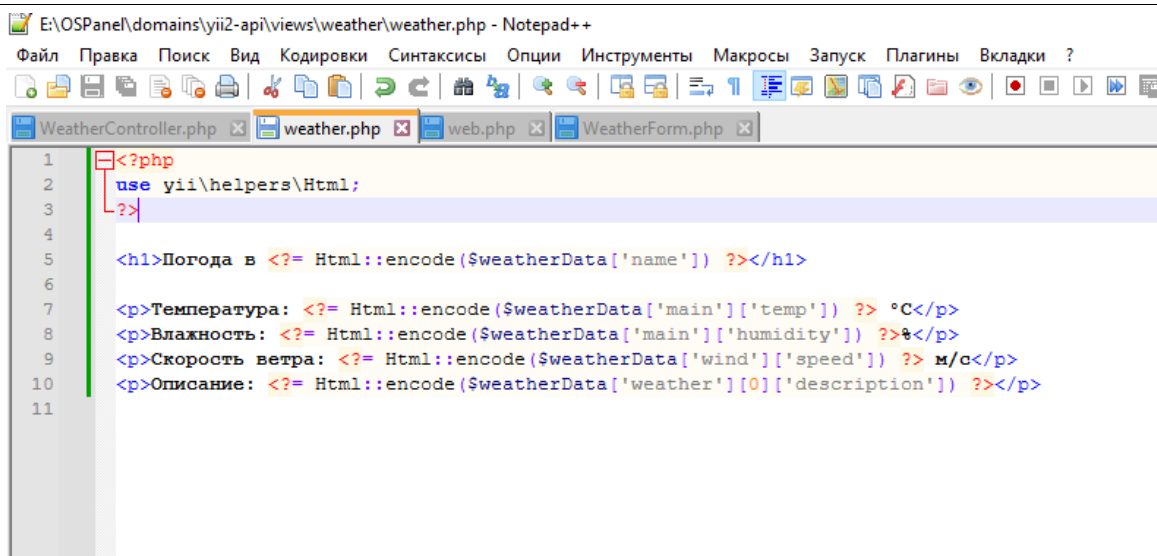
Для обработки ввода пользователя (название города) была создана модель WeatherForm. Модель включает поле `city`, правила валидации и метки для полей формы (рис. 2).



```
1 <?php
2
3 namespace app\models;
4
5 use yii\base\Model;
6
7 class WeatherForm extends Model
8 {
9     public $city;
10
11     public function rules()
12     {
13         return [
14             ['city', 'required'],
15             ['city', 'string', 'max' => 100],
16         ];
17     }
18
19     public function attributeLabels()
20     {
21         return [
22             'city' => 'Город',
23         ];
24     }
25 }
```

Рис. 2. Модель WeatherFrom.php

Для отображения данных о погоде было создано представление `weather.php`. В представлении используется форма для ввода названия города и выводятся данные о погоде, если они были получены. В этом коде создается форма для ввода названия города с использованием виджета ActiveForm. Если данные о погоде получены, они отображаются в виде HTML-элементов. Если данные недоступны, выводится сообщение об ошибке (рис. 3).



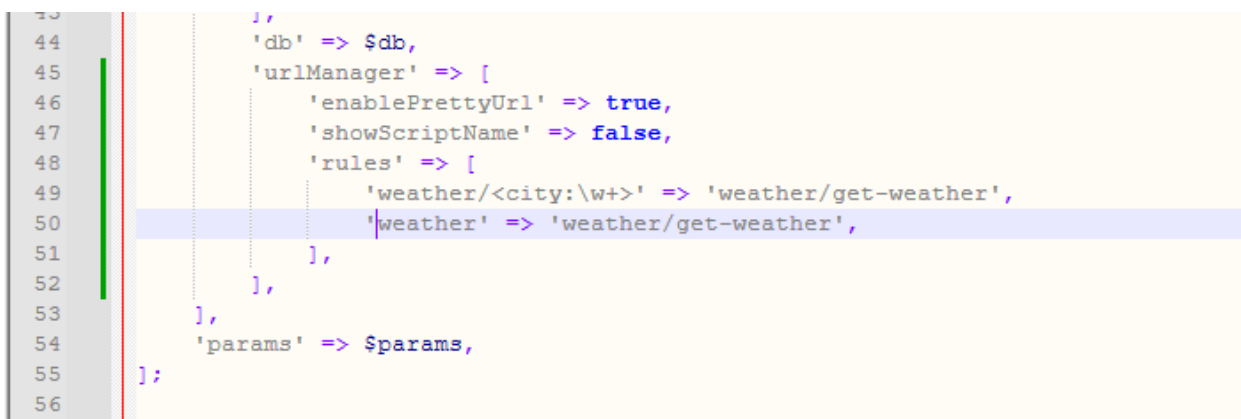
```

1 <?php
2 use yii\helpers\Html;
3 ?>
4
5 <h1>Погода в <?= Html::encode($weatherData['name']) ?></h1>
6
7 <p>Температура: <?= Html::encode($weatherData['main']['temp']) ?> °C</p>
8 <p>Влажность: <?= Html::encode($weatherData['main']['humidity']) ?>%</p>
9 <p>Скорость ветра: <?= Html::encode($weatherData['wind']['speed']) ?> м/с</p>
10 <p>Описание: <?= Html::encode($weatherData['weather'][0]['description']) ?></p>
11

```

Рис. 3. Создание представления weather.php

Для удобства использования приложения были настроены маршруты в config/web.php. Это позволяет обращаться к действию getWeather через удобные URL (рис. 4).



```

44 'db' => $db,
45 'urlManager' => [
46     'enablePrettyUrl' => true,
47     'showScriptName' => false,
48     'rules' => [
49         'weather/<city:\w+>' => 'weather/get-weather',
50         'weather' => 'weather/get-weather',
51     ],
52 ],
53 ],
54 'params' => $params,
55 ];
56

```

Рис. 4. Настройка маршрутов

После настройки всех компонентов приложение было протестировано. Проверялась корректность работы API, обработка данных и их отображение на веб-странице. Теперь пользователь может получить данные о погоде, перейдя по следующим URL: <http://yii2-api/web/weather> (по умолчанию — Биробиджан), <http://yii2-api/web/weather/Moscow> (для получения погоды в Москве). Пример вывода данных о погоде в Биробиджане (рис. 5).

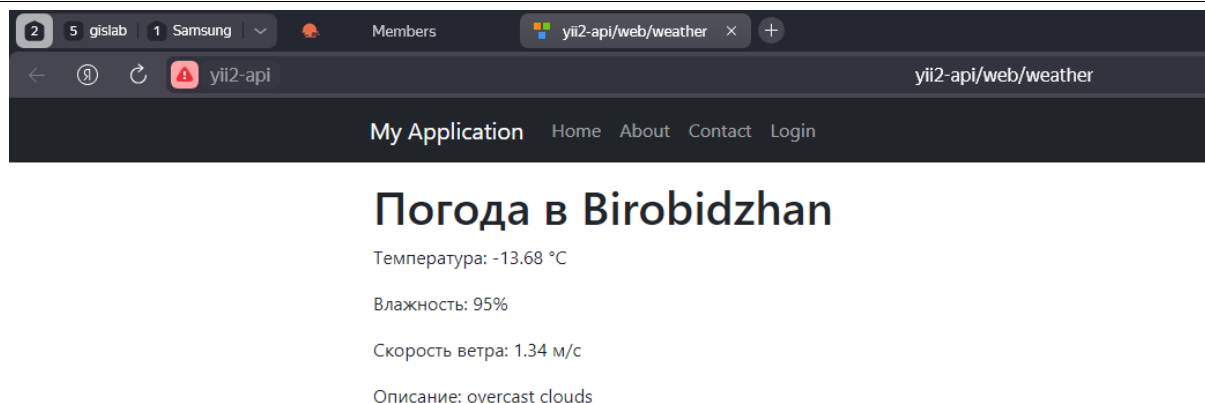


Рис. 5. Вывод погоды через API

Выводы

В данной работе было разработано веб-приложение на базе фреймворка Yii2, интегрированное с API OpenWeatherMap для получения и отображения данных о погоде. В процессе разработки были выполнены следующие задачи: создан контроллер для обработки запросов к API, разработана модель для валидации пользовательского ввода, реализовано представление для отображения данных о погоде, а также настроены маршруты для удобного доступа к функциональности приложения.

Приложение успешно демонстрирует возможности Yii2 для интеграции с внешними API и создания современных веб-приложений, ориентированных на работу с данными в реальном времени. В ходе тестирования была подтверждена корректность работы приложения, включая обработку ошибок и отображение актуальных данных о погоде. Результаты работы показывают, что Yii2 является мощным инструментом для разработки веб-приложений, позволяющим быстро и эффективно реализовывать сложные задачи, такие как интеграция с внешними API.

Библиографический список

1. Закирова Ю. М. Разработка REST API сервиса на базе PHP фреймворка Yii2 // Научные исследования в современном мире: опыт, проблемы и перспективы развития. Сборник научных статей по материалам III Международной научно-практической конференции. Уфа, 2020. С. 36–40.
2. Филимоненкова Т. Н. Архитектурные особенности проектирования и разработки веб-приложений // Дистанционные образовательные технологии. Материалы III Всероссийской научно-практической конференции. Ответственный редактор В. Н. Таран. 2018. С. 253–258.
3. Gainutdinov M. F., Shkurkin A. S., Pichugina A. A. Development of Back-end of the Service for Internationalization of Web-applications // Information technologies and mathematical modelling (ITMM-2020). Материалы XIX

- Международной конференции имени А. Ф. Терпугова. Томск, 2021. С. 265–270.
4. Беляев А. А., Глаголев В. А. Использование Яндекс API для добавления блока "Поделиться" в фреймворке Yii2 // Постулат. 2019. № 1-1 (39). С. 36.
 5. Дашенко И. В., Фешина Е. В. Средства разработки веб-ориентированных информационных систем // Наука XXI века: проблемы, перспективы и актуальные вопросы развития общества, образования и науки. Материалы XII международной межвузовской научно-практической конференции. Краснодар, 2023. С. 46–52.