

Использование библиотеки SwiftMailer для отправления писем на электронную почту

Кочитов Михаил Евгеньевич

*Приамурский государственный университет им. Шолом-Алейхема
студент*

Аннотация

В данной статье рассматривается библиотека SwiftMailer, встроенная в PHP фреймворк Yii2, которая позволяет отправлять письма на адреса электронной почты. Также будет разработан собственный пример со страницей интерактивной формы письма, настройкой библиотеки SwiftMailer и отображением результата, что письмо было успешно отправлено на адрес электронной почты.

Ключевые слова: SwiftMailer, отправление писем, электронная почта, PHP фреймворк, Yii2

Using SwiftMailer Library to Send Emails

Kochitov Mikhail Evgenevich

*Sholom-Aleichem Priamursky State University
student*

Abstract

This article discusses the SwiftMailer library built into the Yii2 PHP framework that allows you to send emails to email addresses. We will also develop our own example with an interactive letter form page, setting up the SwiftMailer library and displaying the result that the letter was successfully sent to the email address.

Keywords: SwiftMailer, mailing, email, PHP framework, Yii2

В настоящее время большинство сайтов снабжены технологиями отправки писем на электронные почты ящиков с целью уведомить о новых скидках на товары, о подтверждении активации нового зарегистрированного аккаунта и так далее. Сайты, в которых имеется возможность зарегистрировать пользователей должны обладать инструментами отправления писем на электронную почту для того, чтобы подтвердить почту нового пользователя. В статье будет использована библиотека SwiftMailer, которая предоставляется PHP фреймворком Yii2. Сама библиотека будет делать подобную задачу формирования письма и отправления его на адрес электронной почты.

В статье М.С. Шевчука рассматривается электронная почта как способ электронного взаимодействия участников гражданского оборота [1]. Рассматривая статью А.А. Львова можно увидеть обнаружение массовых

рассылок электронной почты методом кэширования заголовков письма [2]. И.А. Тарханов в своей статье рассмотрел построение систем документооборота на основе электронной почты [3]. В статье Е.В. Лебедеенко, Д.А. Николаев рассматривается применение модели защиты с полным перекрытием для исследования защищенности системы электронной почты [4]. Рассматривая статью А.А. Решта можно заметить защиту от переполнения частотного словаря статистического фильтра электронной почты [5].

Целью данной статьи является использование библиотеки SwiftMailer, встроенной в PHP фреймворк Yii2, которая позволяет отправлять письма на электронные ящики. Также в статье будет разработан собственный пример с созданием на PHP фреймворке Yii2 интерактивной веб-формы оформления письма, настройки SwiftMailer и отображением результата отправленного письма на адрес получателя.

Библиотека SwiftMailer [6] работает только на сервере, поэтому нужно использовать локальный веб-сервер OpenServer [7], который предоставляется в свободном доступе и установить на него PHP фреймворк Yii2 [8], позволяющий делать веб-сайты гораздо удобнее с предоставлением большого количества библиотек, функций и методов. После установки PHP фреймворка Yii2, необходимо создать новую веб-страницу в ней, но для таких целей понадобится создать модель формы, страницу представления и контроллер обработки заполненной формы с функцией отправки письма.

Для начала нужно создать модель формы, для этого зайдём в папку корневой директории PHP фреймворка Yii2 и откроем папку «models» и в ней создадим PHP файл под названием «MailForm.php» и в него впишем следующий код

```
<?php
namespace app\models;

use Yii;
use yii\base\Model;

class MailForm extends Model
{
    public $to;
    public $subject;
    public $textbody;

    public function rules()
    {
        return [
            [['to', 'subject', 'textbody'], 'required'],
            [['to'], 'email'],
        ];
    }
}
```

Рисунок 1 – PHP код файла «MailForm.php»

На рисунке 1 изображен PHP код файла «MailForm.php», в котором добавляется новая модель формы под названием «MailForm». Для модели будут использоваться три переменные: \$to – адрес получателя, \$subject – тема письма, \$textbody – содержимое письма. В публичной функции «rules» указываются правила для данных переменных. Правило «required» обозначает, что переменные не должны быть пустыми, а правило «email» проверяет, чтобы переменная \$to содержала правильно заполненный адрес электронной почты.

После того как создали файл модели формы «MailForm.php», то необходимо создать страницу представления, которая будет использоваться как веб-страница отображения интерактивной формы заполнения письма. Чтобы это сделать, нужно вернуться обратно в корневую директорию PHP фреймворка и перейти по пути views/site и создать новый PHP файл под названием «mail.php» и в него вписать следующий код

```
<?php
/* @var $this yii\web\View */
/* @var $form yii\bootstrap\ActiveForm */
/* @var $model app\models\LoginForm */

use yii\helpers\Html;
use yii\bootstrap\ActiveForm;

$this->title = 'Отправка письма';
?>
<h1>= Html::encode($this-&gt;title) ?&gt;&lt;/h1&gt;

&lt;?php $form = ActiveForm::begin([
    'id' =&gt; 'mail-form',
]); ?&gt;

&lt;?= $form-&gt;field($model, 'to')-&gt;textInput()-&gt;label('Кому') ?&gt;
&lt;?= $form-&gt;field($model, 'subject')-&gt;textInput()-&gt;label('Тема письма') ?&gt;
&lt;?= $form-&gt;field($model, 'textbody')-&gt;textarea(['rows' =&gt; 5, 'cols' =&gt; 5])-&gt;label('Текст письма') ?&gt;

&lt;?= Html::submitButton('Отправить письмо', ['class' =&gt; 'btn btn-primary', 'name' =&gt; 'login-button']) ?&gt;
&lt;?php ActiveForm::end(); ?&gt;</pre
```

Рисунок 2 – PHP код файла «mail.php»

На рисунке 2 представлен PHP код файла «mail.php», который используется для отображения на веб-странице интерактивной формы заполнения письма. Теперь рассмотрим данный код более подробно. В PHP фреймворке Yii2 имеется встроенная библиотека «ActiveForm», которая упрощает создание интерактивных форм на основе готовых функций. «ActiveForm::begin()» обозначает начало интерактивной формы и в ней указывается параметры заголовка: id – название идентификатора самой формы или уникальное имя. Далее идут функции «field», которые добавляет собственно сами поля в интерактивную форму: to – кому отправлять письмо (функция textInput добавляет текстовое поле, а label добавляет описание самого поля и для чего оно предназначено), subject – тема письма, textbody – текст письма (функция textarea добавляет многострочное текстовое поле) и кнопка «Отправить письмо». Функция «ActiveForm::end()» указывает на завершение интерактивной формы.

Далее необходимо настроить саму библиотеку SwiftMailer, для этого нужно вернуться в директорию PHP фреймворка Yii2 и зайти в папку «config» и в ней открыть PHP файл «web.php» и внести некие изменения в данный файл

```
'mailer' => [
    'class' => 'yii\swiftmailer\Mailer',
    'useFileTransport' => false,
    'transport' => [
        'class' => 'Swift_SmtpTransport',
        'host' => 'smtp.gmail.com',
        'username' => 'mikhail150297@gmail.com',
        'password' => ' ',
        'port' => '465',
        'encryption' => 'ssl',
    ],
],
```

Рисунок 3 – Настройки библиотеки SwiftMailer в файле «web.php»

На рисунке 3 показана часть кода в файле «web.php», которая содержит параметры настроек библиотеки SwiftMailer. Важными параметрами для полноценной работы SwiftMailer являются настройки установки соединения с сервером исходящей почты SMTP, то есть все параметры, что находятся в блоке «transport». Параметр «class» указывает наименование метода, который будет отвечать за обработку и отправку письма на почтовый сервер. «host» - наименование хоста (в нашем случае исходящей почтой будет сервер Gmail). «username» - адрес электронной почты отправителя, «password» - действительный пароль, который позволит SwiftMailer получить доступ к почтовому серверу через указанный адрес. Параметры «port» и «encryption» отвечают за номер порта, по которому будет осуществляться отправление писем и тип шифрования (в данном случае порт будет 465, а шифрование – SSL).

После того как настроили правильно параметры почтового сервера исходящей почты SMTP в PHP файле конфигурации Yii2 «web.php», то необходимо настроить еще контроллер, который позволит отобразить новую веб-страницу с формой заполнения письма и добавить функцию обработки и отправления письма на почтовый сервер. Для этого нужно вернуться в корневую директорию PHP фреймворка Yii2 и перейти в папку «controllers». В текущей папке нужно открыть PHP файл контроллера «SiteController.php». В начале файла нужно добавить одну строчку

```
<?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\MailForm;

class SiteController extends Controller
{
```

Рисунок 4 – Добавленная строка в начале файла «SiteController.php»

На рисунке 4 добавлена выделенная строка в начале файла «SiteController.php», которая загружает в сам контроллер модель формы заполнения письма «MailForm», что была создана в файле «MailForm.php» в папке «models» (рисунок 1).

Далее нужно в конец файла «SiteController.php» добавить функцию отображения новой веб-страницы с интерактивной формой письма и возможностью отправлять с помощью библиотеки SwiftMailer заполненное письмо на адрес отправителя

```
public function actionMail()
{
    $model = new MailForm();

    if ($model->load(Yii::$app->request->post()) && $model->validate()) {
        Yii::$app->mailer->compose()
            ->setFrom('mikhail150297@gmail.com')
            ->setTo($model['to'])
            ->setSubject($model['subject'])
            ->setTextBody($model['textbody'])
            ->send();
    }

    return $this->render('mail', [
        'model' => $model,
    ]);
}
```

Рисунок 5 – PHP код функции «actionMail()», добавленный в конец файла «SiteController.php»

Как можно увидеть, на рисунке 5 продемонстрирован PHP код публичной функции «actionMail()», которая позволяет в веб-браузере отобразить страницу с формой заполнения письма (для такой задачи используется функция «render»). Теперь рассмотрим код более подробно. Изначально в переменную \$model добавляется новая модель формы письма,

которая будет использоваться для получения введенных данных с полей самой интерактивной формы. Далее идет проверка на условие, метод «load» загружает в саму переменную \$model сами введенные данные с полей формы через POST запрос, а метод «validate» осуществляет проверку на валидность введенных данных в самой форме. Функция валидности работает в форме модели (рисунок 1) в ней проверяется, что переменные \$to, \$subject и \$textbody должны быть обязательно заполненными, а переменная \$to – должна содержать правильно оформленный адрес электронной почты. Далее после условия выполняется уже сама функция отправления письма «mailer compose()». В ней указываются параметры: «setFrom» - адрес отправителя (адрес отправителя обязательно должен совпадать с адресом исходящей почты SMTP, что на рисунке 3 в параметре «username»), «setTo» - адрес получателя, «setSubject» - тема письма, «setTextBody» - содержимое или текст письма. Функция «send()» непосредственно делает главную задачу у библиотеки SwiftMailer это отправка сформированного письма на почтовый сервер и после одобрения уже доставка проверенного письма на адрес получателя.

Теперь осталось проверить работоспособность отправки письма, для этого нужно в адресную строку браузера вписать следующее: localhost/web/index.php?r=site%2Fmail и нажать Enter. После этого будет отображена веб-страница самой формы заполнения письма, в которой мы сейчас заполним все данные

Отправка письма

Кому

muzhik.miha@mail.ru

Тема письма

Проверка работы библиотеки SwiftMailer

Текст письма

Это письмо было отправлено с помощью библиотеки SwiftMailer, встроенной в PHP фреймворк Yii2

Отправить письмо

Рисунок 6 – Интерактивная форма с заполненными полями

На рисунке 6 изображена интерактивная форма с заполненными полями: кому – адрес электронной почты получателя, тема письма и текст письма, в котором заполняется содержимое самого письма. Теперь осталось

нажать внизу на синюю кнопку «Отправить письмо» и отправить результат отправки письма на самом ящике Mail.Ru получателя

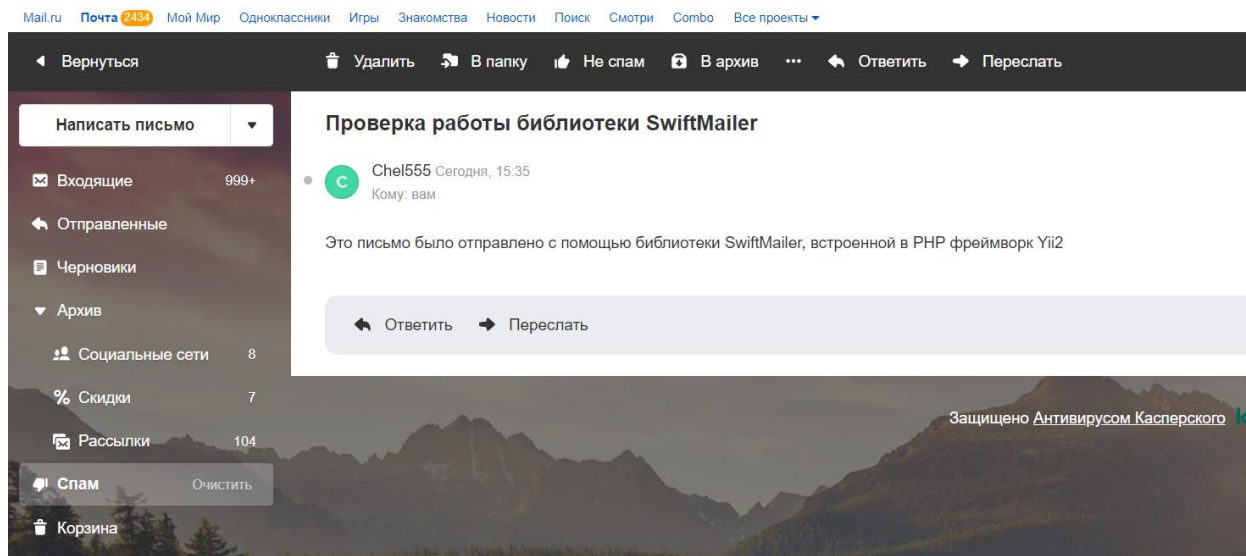


Рисунок 7 – Доставленное письмо на адрес получателя Mail.Ru

На рисунке 7 показан результат работы библиотеки SwiftMailer, которая успешно отправила письмо на адрес электронной почты получателя Mail.Ru и в ней все данные полностью совпадают с введенными данными самой формы на рисунке 6.

Таким образом, была рассмотрена библиотека SwiftMailer, позволяющая отправлять письма на адреса электронной почты. Сама библиотека SwiftMailer встроена в PHP фреймворк Yii2 и поэтому в ней можно максимально быстро создать веб-страницу с организацией возможности проводить массовую рассылку писем на различные ящики получателей.

Библиографический список

1. Шевчук М.С. Электронная почта как способ электронного взаимодействия участников гражданского оборота // Вестник Московского государственного областного университета. Серия: Юриспруденция. 2014. № 2. С. 95-100.
2. Львов А.А. Обнаружение массовых рассылок электронной почты методом кэширования заголовков письма // В сборнике: Математическое моделирование и краевые задачи. Труды Всероссийской научной конференции. Редколлегия: В. П. Радченко (отв. редактор), Э. Я. Рапопорт, Е. Н. Огородников (отв. секретарь), М. Н. Саушкин (отв. секретарь). 2004. С. 151-153.
3. Тарханов И.А. Построение систем документооборота на основе электронной почты // Труды Института системного анализа Российской академии наук. 2006. Т. 23. С. 222-237.
4. Лебеденко Е.В., Николаев Д.А. Применение модели защиты с полным

- перекрытием для исследования защищенности системы электронной почты // Информационные технологии моделирования и управления. 2010. № 1 (60). С. 60-66.
5. Решта А.А. Защита от переполнения частотного словаря статистического фильтра электронной почты // Труды Одесского политехнического университета. 2005. № 1. С. 147-149.
 6. GitHub – SwiftMailer URL: <https://github.com/swiftmailer/swiftmailer> (дата обращения 23.08.2020)
 7. Open Sever Panel – Локальный веб-сервер для Windows // URL: <https://ospanel.io/> (дата обращения 23.08.2020)
 8. Yii PHP Framework URL: <https://www.yiiframework.com/> (дата обращения 23.08.2020)
 9. Отправить электронную почту PHP, с использованием SwiftMailer URL: <https://code.tutsplus.com/ru/tutorials/send-emails-in-php-using-the-swift-mailer--cms-31218> (дата обращения 23.08.2020)
 10. SwiftMailer – Documentation URL: <https://swiftmailer.symfony.com/docs/introduction.html> (дата обращения 23.08.2020)