

Отправка писем на электронную почту с помощью PHPMailer

Кочитов Михаил Евгеньевич

Приамурский государственный университет им. Шолом-Алейхема

студент

Аннотация

В данной статье рассматривается возможность отправлять любые письма на различные адреса электронной почты с помощью языка программирования PHP и подключенному готовому скрипту PHPMailer, который и дает возможность отправлять письма на электронную почту от почтового сервера. В скрипте будет продемонстрирована собственная форма отправки письма и использование PHPMailer для получения нужного результата.

Ключевые слова: отправка писем, письмо, электронная почта, PHP, PHPMailer, отправка

Sending emails using PHPMailer

Kochitov Mikhail Evgenevich

Sholom-Aleichem Priamursky State University

student

Abstract

This article discusses the ability to send any emails to different email addresses using the PHP programming language and the connected PHPMailer script, which makes it possible to send letters to e-mail from the mail server. The script will show its own form of sending the message and use PHPMailer to get the desired result.

Keywords: sending letters, writing, e-mail, PHP, PHPMailer, sending

В настоящее время каждый раз у многих пользователей электронной почты каждый день приходят письма от сайтов, где в письмах пишутся различные новости, ответ администратора после использования обратной связи, ссылка с подтверждением регистрации новой учетной записи, аккаунта или личного кабинета, подтверждение покупки определенных товаров, новые сообщения от социальных сетей и многое другое.

Целью данной статьи является возможность отправить письмо на адреса электронной почты, использовав скрипт PHPMailer с помощью языка программирования PHP.

В статье А.А. Колосова рассматривается защищенная система обмена электронной почтой на основе сертификатов открытых ключей [1]. V. Sinivee в статье рассмотрел устройство для дистанционного управления экспериментом через электронную почту [2]. Рассматривая статью И.

Тетерина можно увидеть автоматизацию веб-проектов через электронную почту [3]. В статье Д.Д. Куликова и С.А. Медведевой рассматривается этикет пользования электронной почтой [4]. К.А. Андреева в статье рассмотрела особенности функционирования антропонимов в деловой переписке по электронной почте [5].

Теперь приступим к написанию собственной формы отправки письма с помощью языка HTML в файле index.php

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Отправка письма с помощью PHP mailer</title>
  </head>
  <body>
    <h3>Отправка письма с помощью PHP mailer</h3>
    <form action="index.php" method="post" enctype="multipart/form-data">
      Кому: <input type="text" name="to" value="<?php echo $_POST['to'];?>"><br><br>
      Тема письма: <input type="text" name="subject" value="<?php echo $_POST['subject'];?>"><br><br>
      Текст письма: <textarea name="body"><?php echo $_POST['body'];?></textarea><br><br>
      <input type="file" name="file"><br><br>
      <input type="hidden" name="sending" value="true">
      <input type="submit" value="Отправить письмо">
    </form>
  </body>
</html>
```

Рис. 1. Код HTML формы отправки письма

На рисунке 1 представлен код HTML формы отправки письма. В нем присутствуют три текстовых поля ввода: кому отправлять письмо, тема самого письма и текст письма (тело или содержимое письма). В поле «Кому» пишется адрес электронной почты, куда потом получателю и придет само письмо от этой HTML формы. В поле «Тема письма» пишется само название письма (тема), которое уточняет что за письмо и для чего оно. В поле «Текст письма» пишется уже само содержимое письма с использованием HTML тегов или без них обычный текст, чтобы проинформировать получателя этого письма на его электронном ящике. Также есть кнопка добавления файла и отправки уже самого письма. При добавлении файла в эту форму, к письму перед отправкой будет прикреплен сам файл, который был выбран в самой форме. При нажатии кнопки «Отправить письмо» происходит уже процесс отправки письма, обращаясь к удаленному серверу в виде файла index.php, того же файла в котором и написана HTML форма отправки письма.

```

<?php
if ($_POST['sending'] == true) {
    $to = $_POST['to'];
    $subject = $_POST['subject'];
    $body = $_POST['body'];

    $from = "pgusa.addresses@gmail.com";
    $from_name = "Pgusa Addresses";

    require 'PHPMailer/src/Exception.php';
    require 'PHPMailer/src/PHPMailer.php';
    require 'PHPMailer/src/SMTP.php';

    $mail = new PHPMailer\PHPMailer\PHPMailer();

    $mail->isSMTP(); // Указываем режим работы с SMTP сервером
    $mail->Host = 'smtp.gmail.com'; // Host SMTP сервера: ip или доменное имя
    $mail->SMTPDebug = 0; // Уровень журнализации работы SMTP клиента PHPMailer
    $mail->SMTPAuth = true; // Наличие авторизации на SMTP сервере
    $mail->Port = 465; // Порт SMTP сервера
    $mail->SMTPSecure = 'ssl'; // Тип шифрования. Например ssl или tls
    $mail->Username = 'pgusa.addresses@gmail.com'; // Имя пользователя на SMTP сервере
    $mail->Password = ' '; // Пароль от учетной записи на SMTP сервере
    $mail->CharSet = 'UTF-8';

    $mail->SetFrom($from, $from_name);
    $mail->AddReplyTo($from, $from_name);
    $mail->AddAddress($to, $to);

    $mail->isHTML(true);
    $mail->Subject = $subject;
    $mail->Body = $body;
    $mail->AltBody = $body;

    if (isset($_FILES['file'])) {
        $mail->addAttachment($_FILES['file']['tmp_name'], $_FILES['file']['name']); // Прикрепление файла перед отправкой письма
    }

    if ($mail->Send()) echo "Письмо успешно отправлено!";
    else echo "Ошибка отправки письма!";
}
?>

```

Рис. 2. PHP код отправки письма с помощью PHPMailer

На рисунке 2 изображен код, который полученные данные из HTML формы обрабатывает и с помощью PHPMailer [6] формирует письмо и уже отправляет на электронную почту получателя. Теперь рассмотрим код немного подробнее.

В начале видно проверку того, что данные были отправлены от формы или нет, этим свидетельствует переменная `$_POST['sending']`, которая и должна проверить значение, если `true` значит форма была отправлена для формирования и отправки письма и `false`, если форма не отправлялась с данными. Ниже видно три переменные `$_POST['to']`, `$_POST['subject']` и `$_POST['body']`, которые хранят значение адреса электронной почты получателя, тему письма и содержимое письма соответственно. Далее для проверки работоспособности скрипта PHPMailer было необходимо создать электронный ящик на Gmail, который будет являться отправителем данного письма из формы – это «`pgusa.addresses@gmail.com`».

Отправка письма работает по принципу почтового сервера исходящей почты под названием SMTP в этом и заключается работа скрипта PHPMailer. Для успешной работы отправки письма от SMTP сервера должны указываться обязательные параметры это: `Host` – хост SMTP сервера, `SMTPAuth` – наличие авторизации на SMTP сервере, `Port` – порт SMTP сервера, `SMTPSecure` – Тип шифрования, `Username` – имя пользователя SMTP сервера (адрес электронной почты отправителя), `Password` – пароль SMTP

сервера (правильный пароль отправителя) на рисунке 2 он замазан для безопасности и защиты данного электронного ящика отправителя. Charset – кодировка обработки и формирования письма. Все эти параметры дадут PHPMailer скрипту обратиться к существующему SMTP серверу и отправить валидные данные авторизации, а также установить кодировку, чтобы при получении письма получатель в нем не увидел непонятные символы (каракули).

После указания параметров авторизации SMTP сервера нужно теперь указать функции SetFrom – откуда отправляется письмо (адрес электронной почты отправителя) должен обязательно совпадать с Username, иначе выдаст соответствующую ошибку сам PHPMailer. AddReplyTo – это возможность при нажатии кнопки «Ответить» на письме уже в электронной почте автоматически вбить в поле «Кому» сам адрес электронной почты отправителя. AddAddress – добавляет адреса электронной почты получателей, можно эту функцию писать много раз, указывая другие ящики получателей, письмо придет всем за одну отправку письма скриптом PHPMailer.

Теперь указываются уже функции формирования самого письма перед отправкой. Параметр isHTML – дает возможность обрабатывать HTML теги в содержимом письма при обработке его SMTP сервером. Subject – собственно в нем указывается тема письма и Body – само тело (содержимое) письма. Параметр AltBody предназначен для альтернативного содержимого (тела) письма без HTML тегов, в том случае если электронная почта получателя не обрабатывает HTML теги письма, а только отобразит текст.

Если был добавлен файл в форму, то функция AddAttachment прикрепляет к письму сам файл, эту функцию можно писать много раз, если надо в письме прикрепить более одного файла.

Последняя функция Send собственно уже отправляет сформированное письмо указанному SMTP серверу для его обработки и после уже отправки этого письма на адреса электронной почты получателей. Данную функцию лучше писать в условии, чтобы убедиться, что письмо было успешно отправлено, либо возникла какая-то ошибка во время письма с указанием самой ошибки, чтобы можно было ее потом предотвратить.

Теперь остается уже проверить работоспособность скрипта PHPMailer на деле, попробуем отправить письмо на адрес электронной почты Mail.ru и Google Mail.

Отправка письма с помощью PHP mailer

Кому:

Тема письма:

Текст письма:

Письмо успешно отправлено!

Рис. 3. Форма отправки письма с указанием адреса получателя Mail.ru

Отправка письма с помощью PHP mailer

Кому:

Тема письма:

Текст письма:

Письмо успешно отправлено!

Рис. 4. Форма отправки письма с указанием адреса получателя Gmail

На рисунках 3 и 4 изображены сами формы отправки писем, разница в них в том, что разные адреса электронной почты получателя. Теперь при нажатии у обеих форм кнопки «Отправить письмо» получаем успешное уведомление, что письмо было успешно отправлено и теперь проверяем электронные ящики получателей.

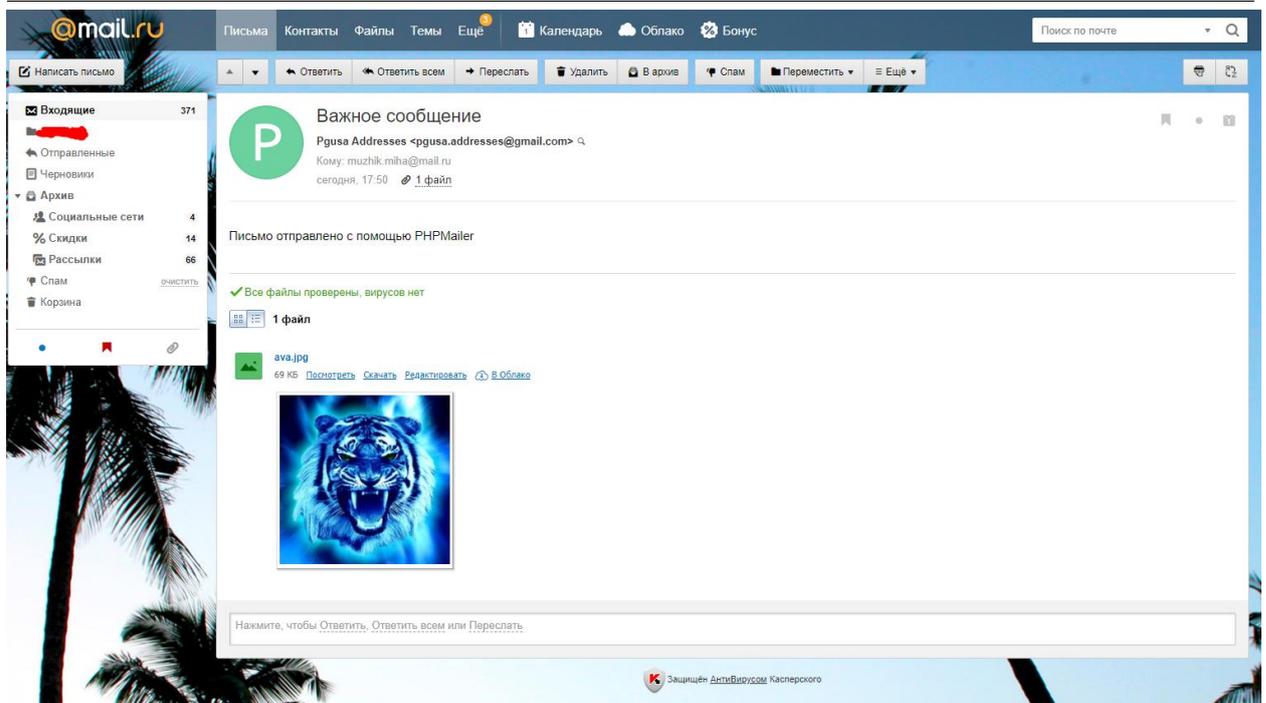


Рис. 5. Полученное письмо на Mail.ru

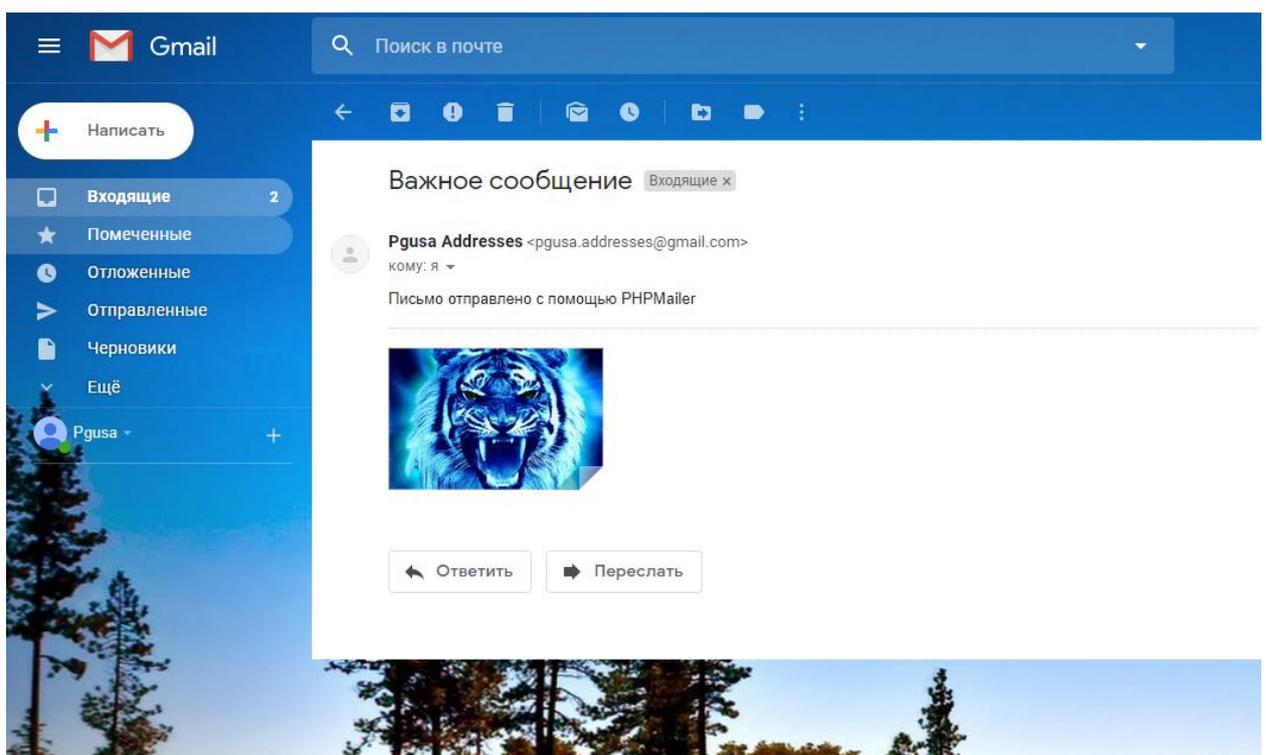


Рис. 6. Полученное письмо на Gmail

Как видно на рисунках 5 и 6 уже представлены полученные письма на электронных ящиках двух получателей: Mail.ru и Gmail. Скрипт RHPMailer успешно выполнил свою работу и отправил письмо двум получателем без всяких ошибок.

Таким образом, была рассмотрена возможность отправки писем на электронную почту с помощью скрипта RHPMailer используя язык программирования PHP. Можно предположить, что большинство сайтов с

автоматической отправкой писем на электронную почту используют именно этот скрипт PHPMailer или без него используют стандартную функцию в php под названием mail, но с ней формировать письмо немного труднее, чем с использованием PHPMailer, который делает это все сам. Собственный пример с возможностью отправки письма успешно был выполнен и письма получили получатели.

Библиографический список

1. Колосов А.А. Защищенная система обмена электронной почтой на основе сертификатов открытых ключей // Безопасность информационных технологий. 2009. № 1. С. 112-113.
2. Sinivee V. Устройство для дистанционного управления экспериментом через электронную почту // Приборы и техника эксперимента. 2007. № 4. С. 73-77.
3. Тетерин И. Автоматизация веб-проектов через электронную почту // Системный администратор. 2004. № 4 (17). С. 76-79.
4. Куликов Д.Д., Медведева С.А. Этикет пользования электронной почтой // Вестник образовательного консорциума Среднерусский университет. Информационные технологии. 2015. № 6. С. 28-30.
5. Андреева К.А. Особенности функционирования антропонимов в деловой переписке по электронной почте // Международный аспирантский вестник. Русский язык за рубежом. 2009. № 1-2. С. 4-9.
6. GitHub PHPMailer URL: <https://github.com/PHPMailer/PHPMailer> (дата обращения 01.07.2018)