

 **Краткое содержание**

# PHPMailer - полнофункциональный класс для создания и отправки электронных писем на PHP.



 Tests falling

stable v7.1.1 downloads 108.27 M license LGPL-2.1-only

 Docs passing

openssf scorecard 6.4

## Функции



- Вероятно, самый популярный в мире код для отправки электронных писем из PHP!
- Используется во многих проектах с открытым исходным кодом: WordPress, Drupal, 1CRM, SugarCRM, Yii, Joomla! и многих других.
- Встроенная поддержка SMTP — отправка без локального почтового сервера.
- Отправляйте электронные письма с несколькими адресами в полях «Кому», «Копия», «Скрытая копия» и «Ответить».
- Многокомпонентные/альтернативные электронные письма для почтовых клиентов, которые не читают HTML-письма.
- Добавляйте вложения, в том числе встроенные.
- Поддержка содержимого в кодировке UTF-8, а также 8-битной, base64, двоичной и quoted-printable кодировок.
- Полная поддержка UTF-8 при использовании серверов, поддерживающих эту кодировку SMTPUTF8.
- Поддержка событий iCal в многокомпонентных файлах и вложениях.
- Аутентификация SMTP с использованием механизмов LOGIN, PLAIN, CRAM-MD5, и XOAUTH2 поверх транспортных протоколов SMTPS и SMTP+STARTTLS.
- Автоматически проверяет адреса электронной почты
- Защита от атак с внедрением заголовков.
- Сообщения об ошибках на более чем 50 языках!
- Поддержка подписания DKIM и S/MIME
- Совместимо с PHP 5.5 и более поздними версиями, включая PHP 8.5.
- Использование пространств имен для предотвращения конфликтов имен.

- И многое другое!

## Зачем это может вам понадобится



Многим PHP-разработчикам необходимо отправлять электронные письма из своего кода. Единственная функция PHP, которая напрямую поддерживает это, — это `mail()`. Однако она не предоставляет никакой помощи в использовании таких популярных функций, как аутентификация, HTML-сообщения и вложения.

Правильно отформатировать электронное письмо на удивление сложно. Существует множество пересекающихся (и противоречащих друг другу) стандартов, требующих строгого соблюдения ужасно сложных правил форматирования и кодирования – подавляющее большинство кода, который вы найдете в интернете и который использует эту `mail()` функцию напрямую, просто неверен, если не небезопасен!

Функция PHP `mail()` обычно отправляет письма через локальный почтовый сервер, как правило, управляемый исполняемым `sendmail` файлом на платформах Linux, BSD и macOS. Однако Windows обычно не включает локальный почтовый сервер; встроенный SMTP-клиент PHPMailer позволяет отправлять электронные письма на всех платформах без необходимости использования локального почтового сервера. Следует помнить, что этой `mail()` функции следует избегать по возможности; использование SMTP для отправки писем на локальный сервер быстрее и **безопаснее**.

*Пожалуйста*, не пытайтесь сделать это самостоятельно — если вы не используете PHPMailer, существует множество других отличных библиотек, которые стоит изучить, прежде чем создавать свою собственную. Попробуйте [Symfony Mailer](#), [Laminas/Mail](#), [ZetaComponents](#) и т. д.

## Лицензия



Данное программное обеспечение распространяется под лицензией [LGPL 2.1](#), а также в соответствии с [обязательствами по сотрудничеству GPL](#). Пожалуйста, ознакомьтесь с [файлом LICENSE](#) для получения информации о доступности и распространении программного обеспечения.

## Монтаж и погрузка



PHPMailer доступен на [Packagist](#) (с использованием семантического версионирования), и рекомендуемый способ установки — через `composer.json`. Просто добавьте эту строку в свой файл:

```
"phpmailer/phpmailer": "^7.0.0"
```



или бежать

```
composer require phpmailer/phpmailer
```



Обратите внимание, что `vendor` папка и `vendor/autoload.php` скрипт генерируются Composer; они не являются частью PHPMailer.

Если вы хотите использовать аутентификацию XOAUTH2, вам также потребуется добавить зависимость от `league/oauth2-client` соответствующего пакета адаптеров служб в ваш файл конфигурации `composer.json`, или же ознакомиться с [оберткой SendOauth2](#) от @decomplexity, особенно если вы используете службы Microsoft.

В качестве альтернативы, если вы не используете Composer, вы можете [загрузить PHPMailer в виде ZIP-архива](#) (обратите внимание, что документация и примеры в ZIP-архив не включены), затем скопировать содержимое папки PHPMailer в один из `include_path` каталогов, указанных в вашей конфигурации PHP, и загрузить каждый файл класса вручную:

```
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'path/to/PHPMailer/src/Exception.php';
require 'path/to/PHPMailer/src/PHPMailer.php';
require 'path/to/PHPMailer/src/SMTP.php';
```



Если вы не используете SMTP класс явно (что, вероятно, так и есть), вам не нужно добавлять `use` для этого отдельную строку. Даже если вы не используете исключения, вам все равно необходимо загрузить Exception класс, поскольку он используется внутри системы.

## Устаревшие версии



PHPMailer 5.2 (совместимый с PHP 5.0 — 7.0) больше не поддерживается, даже обновлениями безопасности. Последнюю версию 5.2 можно найти в [ветке 5.2-stable](#). Если вы используете PHP 5.5 или более позднюю версию (что и следует делать), переключитесь на версии 6.x.

## Обновление с версии 5.2



Самые значительные изменения заключаются в том, что исходные файлы теперь находятся в `src/` папке, и PHPMailer теперь объявляет пространство имен `PHPMailer\PHPMailer`. Это имеет несколько важных последствий — подробнее [см. в руководстве по обновлению](#).

## Минимальная установка



Хотя установка всего пакета вручную или с помощью Composer проста, удобна и надежна, вы можете захотеть включить в свой проект только самые необходимые файлы. Как минимум, вам понадобится [src/PHPMailer.php](#) . Если вы используете SMTP, вам понадобится [src/SMTP.php](#) , а если вы используете POP-before SMTP ( что крайне маловероятно!), вам понадобится [src/POP3.php](#) . Вы можете пропустить папку [language](#), если не показываете пользователям ошибки и можете обойтись только английскими сообщениями об ошибках. Если вы используете XOAUTH2, вам понадобится [src/OAuth.php](#), а также зависимости Composer для сервисов, с которыми вы хотите аутентифицироваться. На самом деле, использовать Composer гораздо проще!

## Простой пример



```
<?php
//Import PHPMailer classes into the global namespace
//These must be at the top of your script, not inside a function
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

//Load Composer's autoloader (created by composer, not included with
PHPMailer)
require 'vendor/autoload.php';

//Create an instance; passing `true` enables exceptions
$mail = new PHPMailer(true);

try {
    //Server settings
    $mail->SMTPDebug = SMTP::DEBUG_SERVER;           //Enable
    verbose debug output
    $mail->isSMTP();                                //Send using
    SMTP
    $mail->Host      = 'smtp.example.com';           //Set the
    SMTP server to send through
    $mail->SMTPAuth  = true;                         //Enable
    SMTP authentication
    $mail->Username  = 'user@example.com';           //SMTP
    username
    $mail->Password  = 'secret';                     //SMTP
    password
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS; //Enable
    implicit TLS encryption
    $mail->Port      = 465;                           //TCP port
    to connect to; use 587 if you have set `SMTPSecure =
```

```
PHPMailer::ENCRYPTION_STARTTLS`
```

```
//Recipients
$mail->setFrom('from@example.com', 'Mailer');
$mail->addAddress('joe@example.net', 'Joe User');      //Add a recipient
$mail->addAddress('ellen@example.com');                //Name is optional
$mail->addReplyTo('info@example.com', 'Information');
$mail->addCC('cc@example.com');
$mail->addBCC('bcc@example.com');

//Attachments
$mail->addAttachment('/var/tmp/file.tar.gz');          //Add attachments
$mail->addAttachment('/tmp/image.jpg', 'new.jpg');     //Optional name

//Content
$mail->isHTML(true);                                  //Set email format
to HTML
$mail->Subject = 'Here is the subject';
$mail->Body    = 'This is the HTML message body <b>in bold!</b>';
$mail->AltBody = 'This is the body in plain text for non-HTML mail
clients';

$mail->send();
echo 'Message has been sent';
} catch (Exception $e) {
    echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}
```



В папке [с примерами](#) вы найдете множество интересных примеров, охватывающих различные распространенные сценарии, включая отправку писем через Gmail, создание контактных форм, отправку писем по спискам рассылки и многое другое.

Если вы повторно используете экземпляр (например, при отправке сообщений в список рассылки), вам может потребоваться очистить список получателей, чтобы избежать отправки дубликатов сообщений. Дополнительные рекомендации см. в [примере с рассылкой](#).

Вот и всё. Теперь вы готовы использовать PHPMailer!

## Локализация



По умолчанию PHPMailer использует английский язык, но в папке [language](#) вы найдете множество переводов сообщений об ошибках PHPMailer, с которыми вы можете столкнуться. Имена файлов содержат код языка [ISO 639-1](#) для переводов, например, `fr` для французского. Чтобы указать язык, вам нужно сообщить PHPMailer, какой язык использовать, следующим образом:

```
//To load the French version
$mail->setLanguage('fr', '/optional/path/to/language/directory/');
```



Мы приветствуем исправления и добавление новых языков — если вы ищете исправления, запустите скрипт [Language/TranslationCompletenessTest.php](#) в папке tests, и он покажет все отсутствующие переводы.

## Документация



Начните чтение с [вики GitHub](#) . Если у вас возникнут проблемы, обратитесь к [руководству по устранению неполадок](#), так как оно часто обновляется.

Примеры использования RHPMailer в распространенных сценариях можно найти в папке [examples](#) . Если вы ищете хорошую отправную точку, мы рекомендуем начать с [примера Gmail](#) .

Чтобы уменьшить объем кода, используемого RHPMailer, примеры не включены, если вы загружаете RHPMailer через Composer или через [ZIP-архив на GitHub](#) . Поэтому вам потребуется либо клонировать репозиторий Git, либо использовать указанные выше ссылки для прямого доступа к примерам.

Полная сгенерированная документация по API [доступна в режиме онлайн](#) .

Полную документацию на уровне API можно сгенерировать, запустив команду `phpdoc` корневой папке; документация появится в этой `doc` папке, хотя для этого потребуется установленный [PHPDocumentor](#) . [Модульные тесты](#) могут послужить хорошим примером выполнения различных операций, таких как шифрование.

Если в документации нет нужной вам информации, поищите ответы на [многочисленные вопросы на Stack Overflow](#) , и прежде чем задавать вопрос об ошибке "SMTP Error: Could not connect to SMTP host", [ознакомьтесь с руководством по устранению неполадок](#) .

## Тесты



В [тестах RHPMailer](#) используется PHPUnit 9, а также [полифил](#) , позволяющий запускать тесты в стиле 9 на более старых версиях PHPUnit и PHP.



Если это не пройдет, можете ли вы чем-нибудь помочь?

## Безопасность



Пожалуйста, сообщайте обо всех обнаруженных уязвимостях ответственно – о проблемах безопасности следует сообщать разработчикам в частном порядке.

См. [раздел БЕЗОПАСНОСТЬ](#) и [уведомления о безопасности PHPMailer на GitHub](#) .

## Вклад



Пожалуйста, отправляйте сообщения об ошибках, предложения и запросы на слияние в [систему отслеживания проблем GitHub](#) .

Нас особенно интересует исправление частных случаев, расширение тестового покрытия и обновление переводов.

Если вы обнаружили ошибку в документации или хотите что-то добавить, смело вносите изменения в вики — редактировать её может любой желающий.

Если у вас есть клоны репозитория Git, созданные до перехода в организацию PHPMailer на GitHub, вам потребуется обновить все удаленные URL-адреса, ссылающиеся на старое местоположение GitHub, с помощью команды, подобной этой, внутри вашего клонированного репозитория:

```
git remote set-url upstream https://github.com/PHPMailer/PHPMailer.git
```



Пожалуйста, больше не используйте проекты на SourceForge или Google Code; они устарели и больше не поддерживаются.

## Спонсорство



Время и ресурсы для разработки PHPMailer предоставляются компанией [Smartmessages.net](#) , единственной в мире системой email-маркетинга, в которой приоритет отдается конфиденциальности.

# SMART ESSAGES

Мы будем очень рады пожертвованиям, будь то пиво 🍺, футболки 📦 или наличные 💰. Спонсорство через GitHub — это простой и удобный способ сказать «спасибо» разработчикам и участникам проекта PHPMailer — просто нажмите кнопку «Спонсор» [на странице проекта](#) . Если ваша компания использует PHPMailer, рассмотрите возможность участия в программе

корпоративной поддержки Tidelift.

## PHPMailer для предприятий



Доступно в рамках подписки Tidelift.

Разработчики PHPMailer и тысяч других пакетов сотрудничают с Tidelift, чтобы обеспечить коммерческую поддержку и обслуживание пакетов с открытым исходным кодом, которые вы используете для создания своих приложений. Экономьте время, снижайте риски и улучшайте качество кода, одновременно оплачивая работу разработчиков именно тех пакетов, которые вы используете. [Узнайте больше](#).

## Список изменений



См. [список изменений](#) .

## История



- PHPMailer был первоначально написан в 2001 году Brentом Р. Матцеллем в рамках [проекта на SourceForge](#) .
- [Маркус Бойнтон](#) ( соo1bruавтор научно-фантастического произведения) и Энди Превост (автор литературного произведения codeworxtech) возглавили проект в 2004 году.
- В 2010 году проект стал участником инкубатора Apache на платформе Google Code, которым руководит Джим Ягельски.
- Маркус создал [свой форк на GitHub](#) в 2008 году.
- В 2013 году Джим и Маркус решили объединить усилия и использовать GitHub в качестве канонического и официального репозитория для PHPMailer.
- В 2013 году PHPMailer перешёл в [организацию PHPMailer](#) на GitHub.

## Что изменилось после перехода с SourceForge?



- Официальный преемник проектов SourceForge и Google Code.
- Набор тестов.
- Непрерывная интеграция с GitHub Actions.
- Поддержка композитора.
- Общественное развитие.
- Дополнительные языки и языковые строки.
- Поддержка аутентификации CRAM-MD5.
- Сохраняет полную историю репозитория, включая авторов, коммиты и ветки, из оригинального проекта SourceForge.



## Дополнения и Файлы

phpmailer-master.zip

From:

<https://wwoss.direct.quickconnect.to/> - worldwide open-source software

Permanent link:

[https://wwoss.direct.quickconnect.to/doku.php?id=software:development:demo:cms:ucms:appendix:appendix\\_php\\_mailer](https://wwoss.direct.quickconnect.to/doku.php?id=software:development:demo:cms:ucms:appendix:appendix_php_mailer)

Last update: **2026/02/11 20:53**

