**Исчерпывающая инструкция по PHP Mailer – Все, что вам надо знать**

**Введение**

Рассылка сообщений посредствам PHP может быть очень простой, а может быть очень сложной, все зависит от того, что и как вы собираетесь посылать.

Стандартное электронное письмо – это обычный текст, что и используется большинством разработчиков, поэтому создание MIME заголовков для HTML писем может стать сложным процессом. Теперь трудности вам не грозят, потомучто у вас есть чудесная библиотека PHPMailer, которая, к тому же, бесплатна.

В этом обучающем курсе мы будем подробно обсуждать особенности и возможности этой библиотеки.

**Требования**

Требования у данной библиотеки очень скромны. Вам нужен только PHP и возможность отсылать письма посредствам команды mail() или через SMTP соединение.

Также вы должны понимать основы объектно-ориентированного программирования (ООП), или, по крайней мере, понимать как правильно применять данные ниже примеры.

Не переживайте! Этот курс достаточно прост!

**О PHPMailer**

Это класс, обеспечивающий полную функциональность при отправке почтовых сообщений на PHP. Я назвал бы его самым лучшим из всех, которые я когда-либо использовал. Его популярность очень быстро росла. С момента появления, 7 декабря 2004 года, его скачали уже более 100 000 раз! Я надеюсь, что вы захотите увеличить это число после прочтения данного обучающего курса, и, что еще более важно, вы узнаете, как использовать те возможности по работе с почтой, о которых вы только мечтали.

**Что может PHPMailer**

На момент написания этого обучающего курса PHP Mailer умел:

* посылать письма с множественными: адресатами (TO), копиями (CC), BCC и REPLY-TO;
* избыточные SMTP серверы;
* многослойные/альтернативные сообщения для клиентов, которые не могут читать HTML письма;
* поддержка 8 бит, base64, бинарного режима, и пригодного для печати формата;
* Использовать все теже самые методы, что и популярный AspEmail активный сервер (COM);
* авторизация SMTP;
* перенос слов (word wrap);
* сообщения в виде HTML;
* библиотека проверена на множестве SMTP серверах: Sendmail, qmail, Postfix, Imail, Exchange, Mercury, Courier;
* работает на любой win32 и \*nix платформе;
* гибкость отладки;
* определяемые вручную заголовки писем;
* совмещение нескольких сообщений и вложений;
* встроенная поддержка изображений.

**Разработчики PHPMailer**

* Brent R. Matzelle
* Patrice Fournier
* Chris Ryan
* Cem Hurturk
* Tom Klingenberg
* Jaime Bozza

Я выражаю этим людям свою признательность, потомучто они столько сделали, чтобы ускорить развитие моих проектов, включая PHPFreaks.com!

**Готовим PHPMailer для использования**

**Скачать и распаковать**

Первое, что вы должны сделать, это, конечно же, скачать PHPMailer! Вы можете найти его здесь: <http://phpmailer.sourceforge.net/>. После того, как вы скачаете файл, распакуйте его в каталог. В нашем учебнике мы предположим, что ваш сайт находится здесь:**/home/mywebsite/public\_html/**. Теперь необходимо создать несколько директорий, чтобы упорядочить структуру проекта. Я обычно помещаю библиотеки в каталог '**lib**', а затем их название. Таким образом, извлекаем PHPMailer сюда:**/home/mywebsite/public\_html/lib/phpmailer**. Вот пример из командной строки:

|  |
| --- |
| /home/mywebsite/public\_html/lib  /home/mywebsite/public\_html/lib/phpmailer  /home/mywebsite/public\_html/lib/phpmailer/docs  /home/mywebsite/public\_html/lib/phpmailer/docs/extending.html  /home/mywebsite/public\_html/lib/phpmailer/docs/faq.html  /home/mywebsite/public\_html/lib/phpmailer/docs/timeoutfix.diff  /home/mywebsite/public\_html/lib/phpmailer/language  /home/mywebsite/public\_html/lib/phpmailer/language/phpmailer.lang-br.php  /home/mywebsite/public\_html/lib/phpmailer/language/phpmailer.lang-cz.php  /home/mywebsite/public\_html/lib/phpmailer/language/phpmailer.lang-de.php  /home/mywebsite/public\_html/lib/phpmailer/language/phpmailer.lang-en.php  /home/mywebsite/public\_html/lib/phpmailer/language/phpmailer.lang-es.php  /home/mywebsite/public\_html/lib/phpmailer/language/phpmailer.lang-fr.php  /home/mywebsite/public\_html/lib/phpmailer/language/phpmailer.lang-it.php  /home/mywebsite/public\_html/lib/phpmailer/language/phpmailer.lang-nl.php  /home/mywebsite/public\_html/lib/phpmailer/language/phpmailer.lang-no.php  /home/mywebsite/public\_html/lib/phpmailer/language/phpmailer.lang-se.php  /home/mywebsite/public\_html/lib/phpmailer/language/phpmailer.lang-tr.php  /home/mywebsite/public\_html/lib/phpmailer/phpdoc  /home/mywebsite/public\_html/lib/phpmailer/phpdoc/allclasses-frame.html  /home/mywebsite/public\_html/lib/phpmailer/phpdoc/deprecated-list.html  /home/mywebsite/public\_html/lib/phpmailer/phpdoc/help-doc.html  /home/mywebsite/public\_html/lib/phpmailer/phpdoc/index-all.html  /home/mywebsite/public\_html/lib/phpmailer/phpdoc/index.html  /home/mywebsite/public\_html/lib/phpmailer/phpdoc/overview-tree.html  /home/mywebsite/public\_html/lib/phpmailer/phpdoc/packages.html  /home/mywebsite/public\_html/lib/phpmailer/phpdoc/phpmailer.html  /home/mywebsite/public\_html/lib/phpmailer/phpdoc/serialized-form.html  /home/mywebsite/public\_html/lib/phpmailer/phpdoc/stylesheet.css  /home/mywebsite/public\_html/lib/phpmailer/test  /home/mywebsite/public\_html/lib/phpmailer/test/phpmailer\_test.php  /home/mywebsite/public\_html/lib/phpmailer/test/phpunit.php  /home/mywebsite/public\_html/lib/phpmailer/test/rocks.png  /home/mywebsite/public\_html/lib/phpmailer/ChangeLog.txt  /home/mywebsite/public\_html/lib/phpmailer/class.phpmailer.php  /home/mywebsite/public\_html/lib/phpmailer/class.smtp.php  /home/mywebsite/public\_html/lib/phpmailer/LICENSE  /home/mywebsite/public\_html/lib/phpmailer/README |

Теперь, когда все файлы на своих местах, давайте создадим файл конфигурации нашего проекта.

**Создание и использование файла конфигурации**

Одна из главных вещей, которую я люблю делать перед созданием сайта – это написание файла конфигурации, он будет содержать настройки к которым я, возможно, буду обращаться много раз. И так, я создаю файл с название config.php в**/home/mywebsite/public\_html/config.php**. Создаем внем массив с названием **$site** с моими ключами и значениями, которые я буду использовать в дальнейшем. В этом руководстве я опишу переменные, которые мы будем использовать в классе PHPMailer. Вот пример моего файла **config.php**:

|  |
| --- |
| // Настройки для MY site  // Настройки Email  $site['from\_name'] = 'мое имя'; // from (от) имя  $site['from\_email'] = 'email@mywebsite.com'; // from (от) email адрес  // На всякий случай указываем настройки  // для дополнительного (внешнего) SMTP сервера.  $site['smtp\_mode'] = 'disabled'; // enabled or disabled (включен или выключен)  $site['smtp\_host'] = null;  $site['smtp\_port'] = null;  $site['smtp\_username'] = null; |

Этот пример очень прост, чтобы объяснять его. Далее мы продолжим свою работу и будем брать эти настройки по мере надобности.

**Создаем класс, расширяющий PHPMailer**

Сразу подчеркну, что вы должны создать класс расширения, это позволит упростить вам жизнь, далее я покажу как.

Класс расширения вызовет класс PHPMailer() и установит начальные значения, такие, как Email адрес от имени которого вы будете посылать письма, настройки почтового сервера, и так далее.

Каждый из этих параметров берется по умолчанию из файла **config.php**, но вы также можете переписать их при вызове нашего класса расширения. Таким образом, если вы не переопределите какой-либо из параметров в классе расширения, то он будет взят поумолчанию из файла **config.php**, и вам не надо каждый раз переопределять эти параметры. В этом и красота данного подхода!

Взгляните на наш класс расширения:

|  |
| --- |
| require\_once($\_SERVER['DOCUMENT\_ROOT'].'/lib/phpmailer/class.phpmailer.php');  class FreakMailer extends PHPMailer  {  var $priority = 3;  var $to\_name;  var $to\_email;  var $From = null;  var $FromName = null;  var $Sender = null;    function FreakMailer()  {  global $site;    // Берем из файла config.php массив $site    if($site['smtp\_mode'] == 'enabled')  {  $this->Host = $site['smtp\_host'];  $this->Port = $site['smtp\_port'];  if($site['smtp\_username'] != '')  {  $this->SMTPAuth = true;  $this->Username = $site['smtp\_username'];  $this->Password = $site['smtp\_password'];  }  $this->Mailer = "smtp";  }  if(!$this->From)  {  $this->From = $site['from\_email'];  }  if(!$this->FromName)  {  $this-> FromName = $site['from\_name'];  }  if(!$this->Sender)  {  $this->Sender = $site['from\_email'];  }  $this->Priority = $this->priority;  }  } |

Далее мы рассмотрим этот код.

**Разберем класс FreakMailer**

Наш класс, показанный выше, очень прост. И вам необходимо обладать только базовыми навыками ООП, чтобы использовать его. Давайте разберемся.

Сначала нам необходимо подключить класс PHPMailer (файл **class.phpmailer.php** в директории **lib**). Это позволяет нам расширить класс PHPMailer, потому что делает объект доступным. Вы могли бы подключить его и в другом месте, но здесь это наиболее эффективно.

|  |
| --- |
| require\_once($\_SERVER['DOCUMENT\_ROOT'].'/lib/phpmailer/class.phpmailer.php'); |

**Структура управления классом**

Далее, мы определяем структуру управления классом и даем ему имя при расширении класса PHPMailer.

|  |
| --- |
| class FreakMailer extends PHPMailer  { |

**Переменные класса**

Идем дальше. Теперь мы определяем внутренние переменные. Большинство из них установлены поумолчанию в *NULL*, чтобы позже мы могли их переопределить, если есть необходимость заменить значения, установленные в **config.php**.

|  |
| --- |
| var $priority = 3;  var $to\_name;  var $to\_email;  var $From = null;  var $FromName = null;  var $Sender = null; |

Давайте рассмотрим эти переменные:

* **$priority** – это устанавливает приоритет почты поумолчанию: 1 – высоко, 3 – нормально, 5 – низко.
* **$to\_name** – Имя человека, которому вы посылаете сообщение.
* **$to\_email** – адрес этого человека.
* **$From** – адрес, с которого вы хотите послать письмо.
* **$FromName** – имя отправителя.

После определения переменных, мы с вами можем обсудить функцию **FreakMailer()**.

**Функция FreakMailer()**

Эта функция, в основном, настраивает начальные значения для класса PHPMailer, чтобы посылать почту.

Сначала мы вызываем массив $site из файла **config.php**, чтобы иметь возможность использовать его в пределах данной функции и класса. Чтобы сделать это есть два способа: передать массив при вызове класса или сделать массив глобальным. Последний способ самый легкий и он работает, поэтому его и используем!

|  |
| --- |
| function FreakMailer()  {  global $site; // Берем из файла config.php массив $site |

Все дальнейшие действия – это, в основном, определение значений переменным класса PHPMailer. Если внутренняя переменная (**$this->setting**) не определена, мы берем значения из файла **config.php**. Я уже упоминал о том, что вы можете переопределить значения из**config.php**, здесь мы проверяем это.

|  |
| --- |
| if($site['smtp\_mode'] == 'enabled')  {  $this->Host = $site['smtp\_host'];  $this->Port = $site['smtp\_port'];  if($site['smtp\_username'])  {  $this->SMTPAuth = true;  $this->Username = $site['smtp\_username'];  $this->Password = $site['smtp\_password'];  }  $this->Mailer = "smtp";  }    if(!$this->From)  {  $this->From = $site['from\_email'];  }  if(!$this->FromName)  {  $this->FromName = $site['from\_name'];  }  if(!$this->Sender)  {  $this->Sender = $site['from\_email'];  }  $this->Priority = $this->priority;  } |

Самое главное, что вы должны понять, это то, что все возможности PHPMailer все еще присутствуют и могут использоваться, несмотря на то, что мы его расширили. Единственное, что мы сделали здесь – это создали расширение, которое заботится о том, чтобы нам не надо было каждый раз вводить одни и теже данные, когда мы хотим послать сообщение.

Теперь, когда мы достаточно имеем представление о классе расширения, давайте попробуем отправить почту!

**Отправка электронной почты средствами PHPMailer**

Предварительная работа сделана и теперь у нас есть все, чтобы посылать письма средствами PHPMailer.

**Первоначальное испытание**

Это испытание очень важно, потому что в рамках данной статьи мы будем постоянно возвращаться к приведенному здесь коду. Если данный код у вас не заработает, перечитайте этот материал сначала и снова попробуйте. Наш первый пример мы сохраним в файле**mailtest.php**, который положим в корень нашего проекта. Вот его содержимое:

|  |
| --- |
| // Читаем настройки config  require\_once($\_SERVER['DOCUMENT\_ROOT'].'/config.php');  // Подключаем класс FreakMailer  require\_once($\_SERVER['DOCUMENT\_ROOT'].'/lib/MailClass.inc');  // инициализируем класс  $mailer = new FreakMailer();  // Устанавливаем тему письма  $mailer->Subject = 'Это тест';  // Задаем тело письма  $mailer->Body = 'Это тест моей почтовой системы!';  // Добавляем адрес в список получателей  $mailer->AddAddress('foo@host.com', 'Eric Rosebrock');  if(!$mailer->Send())  {  echo 'Не могу отослать письмо!';  }  else  {  echo 'Письмо отослано!';  }  $mailer->ClearAddresses();  $mailer->ClearAttachments(); |

Давайте разберем этот код для лучшего его понимания.

Сначала мы подключаем наш конфигурационный файл, чтобы иметь доступ к массиву **$site**.

Далее, подключаем класс FreakMailer.

Теперь необходимо инициализировать наш класс расширения, а также класс PHPMailer.

|  |
| --- |
| // Читаем настройки config  require\_once($\_SERVER['DOCUMENT\_ROOT'].'/config.php');  // Подключаем класс FreakMailer  require\_once($\_SERVER['DOCUMENT\_ROOT'].'/lib/MailClass.inc');  // инициализируем класс  $mailer = new FreakMailer(); |

Теперь PHPMailer готов и мы идем дальше.

Задаем тему письма и его содержание.

|  |
| --- |
| // Устанавливаем тему письма  $mailer->Subject = 'Это тест';  // Задаем тело письма  $mailer->Body = 'Это тест моей почтовой системы!'; |

Если вы используете в качестве письма обычный текст, то необходимо разделять строки или . И вы должны использовать двойные кавычки при определении переменной **$mailer->Body**. Если вы используете одинарные кавычки, то перенос текста с одной строки на другую возможен с помощью клавиши .

Теперь необходимо добавить адрес получателя.

|  |
| --- |
| // Добавляем адрес в список получателей  $mailer->AddAddress('foo@host.com', 'Eric Rosebrock'); |

Далее, посылаем сообщение и обрабатываем ошибки, если возникли.

В случае ошибки – вы увидите сообщение «Не могу отослать письмо!», в противном случае - «Письмо отослано!».

|  |
| --- |
| if(!$mailer->Send())  {  echo 'Не могу отослать письмо!';  }  else  {  echo 'Письмо отослано!';  } |

Далее, мы очищаем список адресатов и список вложений.

|  |
| --- |
| $mailer->ClearAddresses();  $mailer->ClearAttachments(); |

Если вам только что пришло сообщение, которое вы отослали с помощью PHPMailer, – поздравляю!

**Основные проблемы**

Вот список некоторых (типичных) проблем, которые могут возникнуть у вас при отправлении почты при помощи PHPMailer (вероятнее всего, эти проблемы у вас возникли бы и при обычном отправлении писем):

* на вашей машине не установлено или не запущено ни одного SMTP сервера;
* неправильные настройки внутри PHP скрипта, перечитайте обучающий курс еще раз;
* Сервер Apache не разрешает релей через SMTP сервер на локальной машине (типичный случай);
* вы не правильно определили получателя письма.

Но, если у вас все заработало, давайте перейдем к отправке писем с определенными особенностями.

**Использование дополнительных возможностей PHPMailer**

Если вы уже здесь, но не читали при этом предыдущего материала – вернитесь и прочитайте, потому что здесь мы будем заниматься расширением базового примера.

PHPMailer имеет множество особенностей, например, добавление вложений, рассылка по списку адресатов и др. В этой части нашего материала мы с вами посмотрим, как это выглядит на практике.

**Обработка адресов электронной почты**

PHPMailer поддерживает много особенностей Адреса электронной почты, типа «Для» TO, «от» FROM и списков получателей, точная копия CC и Слепая точная копия BCC, Reply-TO и др. Давайте посмотрим, как использовать эти особенности.

Помните, что мы опираемся на основной пример, приведенный выше.

**Добавление адреса отправителя (FROM)**

В нашем основном примере адрес отправителя уже задан посредствам config.php и класса расширения, но вы в любой момент можете его переопределить. Например:

|  |
| --- |
| $mailer->FromName = 'Ваше Имя';  $mailer->From = 'You@yourdomain.com'; |

Если не определить FromName, то большинство почтовых клиентов в качестве имени подставят емаил.

**Добавление адреса для ответа (Reply-To)**

Поумолчанию, адрес для ответа совпадает с адресом, с которого вы отослали письмо (FROM). Однако вы можете определить разные адреса для ответа на ваше письмо:

|  |
| --- |
| $mailer->AddReplyTo('billing@yourdomain.com', 'Департаменту оплаты'); |

**ЗАМЕЧАНИЕ:**

Вы можете задать несколько адресов для ответа. Для этого необходимо продублировать приведенный выше код столько раз, сколько вы имеете адресов (не забудьте в примере менять адреса на нужные вам).

**Добавление нескольких получателей**

Этот метод позволяет вам добавить нескольких получателей для письма. Я не рекомендую этот способ для анонимных листов рассылки. Смотрите позже пример для списка рассылки.

Для добавления нескольких получателей вы должны повторить вызов функции **AddAddress** столько раз, скольким людям вы хотите отослать ваше сообщение. Далее, пример для трех адресатов. В данном примере адрес является обязательным, а имя адресата опциональным и может быть опущено.

|  |
| --- |
| $mailer->AddAddress('recipient1@domain.com', 'Первый человек');  $mailer->AddAddress('recipient2@domain.com', 'Второй человек');  $mailer->AddAddress('recipient3@domain.com', 'Третий человек'); |

**ЗАМЕЧАНИЕ:**

Не рекомендуем пользоваться этим методом для рассылки по списку адресатов! Каждый получатель вашего письма будет видеть ВЕСЬ список получателей, а, затем, сможет использовать его по своему усмотрению. Позже мы расскажем как правильно отсылать письма по списку адресатов.

**Добавить получателей точной копии (CC)**

Вы можете добавить получателей точной копии точно также, как и предыдущих примерах, используя функцию **AddCC**.

|  |
| --- |
| $mailer->AddCC('recipient1@domain.com', 'Первый человек');  // Если вам надо добавить более чем одного – просто повторяйте эту команду!  $mailer->AddCC('recipient2@domain.com', 'Второй человек');  $mailer->AddCC('recipient3@domain.com', 'Третий человек'); |

**Добавление получателей слепой точной копии (BCC)**

Невидимых получателей или BCC можно добавить используя следущую функцию:

|  |
| --- |
| $mailer->AddBCC('recipient1@domain.com', 'Первый человек');  // Если вам надо добавить более чем одного BCC, просто продолжайте!  $mailer->AddBCC('recipient2@domain.com', 'Второй человек');  $mailer->AddBCC('recipient3@domain.com', 'Третий человек'); |

**Требование подтверждения прочтения**

Если вы хотите запросить у человека, читающего ваше письмо, подтверждение о прочтении, подайте следующую команду:

|  |
| --- |
| $mailer->ConfirmReadingTo = 'you@youdomain.com'; |

Теперь, когда мы рассмотрели все адреса для отправки почты, давайте посмотрим, как посылать письма ввиде HTML!

**Отправление писем ввиде HTML при помощи PHP и PHPMailer**

Формирование HTML письма – это одна из самых сложных задач при отправке почты средствами PHP. Настройка MIME заголовков и построение HTML – нелегкая задача и требует дополнительных знаний и исследований. Однако, PHPMailer сделает вашу жизнь проще, далее, мы покажем как это реализуется.

**Важное замечание относительно почты ввиде HTML**

Прежде чем мы приступим к отправке таких писем, я хотел бы, чтобы вы поняли, что важно знать о том, каким образом должны подключаться изображения и стили CSS, и т.д. Простое правило состоит в том, чтобы хранить эти файлы на сервере, а в письме давать полные ссылки на них. Если вы сходите сума и добавляете все эти файлы к письму, то готовьтесь к головной боли. Например, тело моего письма выглядит примерно так:

|  |
| --- |
| <html>  <head>  <title>My HTML Email</title>  </head>  <body>  <img src="http://www.phpfreaks.com/images/phpfreaks\_logo.jpg" alt="PHP Freaks" />  <h2>PHP Freaks Rules!</h2>  <p>We invite you to visit <a href="http://www.phpfreaks.com" title="PHP Freaks">PHP Freaks.com</a>  for a loving community of PHP Developers who enjoy helping each other learn the language!</p>  <p>Sincerely,<br />  PHP Freaks Staff</p> |

Как вы видете, я сделал каждый URL полным, а не сокращенным относительно моего документа. Если этого не сделать, то ваши изображения не будут грузиться и пользователь их не увидит!

Идем дальше, теперь нам необходимо отослать наше письмо. Для этого нам необходимо определить тело нашего письма и задать **isHTML** настройки. Следующий пример расширяет первоначальный базовый пример.

|  |
| --- |
| $htmlBody = '  <head>  <title>My HTML Email</title>  </head>  <body>  <img src="http://www.phpfreaks.com/images/phpfreaks\_logo.jpg" alt="PHP Freaks" />  <h2>PHP Freaks Rules!</h2>  <p>We invite you to visit <a href="http://www.phpfreaks.com" title="PHP Freaks">PHP Freaks.com</a>  for a loving community of PHP Developers who enjoy helping each other learn the language!</p>  <p>Sincerely,<br />  PHP Freaks Staff</p>';  $mailer->Body = $htmlBody;  $mailer->isHTML(true);  // Отправляем сообщение |

**Дополнительные «тела» письма**

Никогда не полагайтесь только на HTML при отправке писем, если ваше сообщение очень важное. Вы должны позаботиться о получателе и помимо HTML письма прикрепить и только text-only (только текст) версию сообщения, поскольку некоторые почтовые клиенты не умеют показывать HTML письма. Мы можем достигнуть такого эффекта добавляя дополнительные тела письма функцией **AltBody** класса PHPMailer. Давайте добавим текстовую версию вашего сообщения. Как только мы добавляем несколько тел письма – PHPMailer автоматически сформирует многослойное письмо.

|  |
| --- |
| // настраиваем класс почты  $htmlBody = 'My HTML Body....';  $textBody = 'My text-only body....';  $mailer->Body = $htmlBody;  $mailer->isHTML(true);  $mailer->AltBody = $textBody;  // Отправляем письмо |

**Прикрепление файлов к письму**

Прикрепить файл к письму – это чень легкая задача, если вы используете PHPMailer. Просто добавьте их письму точно также, как вы добавляли адреса. Вот пример:

|  |
| --- |
| // настройка класса для отправки почты  $mailer->AddAttachment('/home/mywebsite/public\_html/file.zip', 'file.zip'); |

Функция AddAttachment имеет 4 аргумента:

* путь до файла;
* имя файла;
* кодирование;
* тип заголовка.

Путь до файла, естественно, полный путь до файла на диске, имя файла – название файла, который вы хотите прикрепить, кодирование – поумолчанию base64, тип заголовка – это тип заголовка, который вы хотите послать – поумолчанию – **Application/octet-stream**.

Как видите, все очень легко! Давайте перейдем с вами к использованию SMTP.

**Использование внешнего(их) SMTP для работы с почтой средствами PHP**

Сначала давайте обсудим использование внешнего SMTP вместо локального. В базовом примере мы с вами сделали **config.php**, где задали несколько вариантов для SMTP сервера. Если вы хотите указать больше SMTP серверов – сделать это можно также расширив**config.php**, установив **$site['smtp\_enabled'] = 'enabled';** (включено).

В **$site ['smtp\_host']** вы можете расширить список SMTP серверов, разделяя их точкой с запятой. Например:

|  |
| --- |
| $site['smtp\_host'] = 'server1.host.com;server2.host.com;server3.host.com'; |

Как я понимаю, PHPMailer попытается послать почту через первый сервер, если это не удается – через второй и так далее.

**ЗАМЕЧАНИЕ:**

Помните, что вы всегда можете включить авторизацию через SMTP или поменять порт в файле **config.php**.

Важное замечание, обращаю ваше внимание, что для отправки писем через SMTP необходимо использовать не стандартную функцию **Send()**, а функцию **SMTPSend()**. Например:

|  |
| --- |
| // настройки класса  // смотри базовый пример  // Не SMTP режим:  // $mailer->Send();  // SMTP режим:  $mailer->SmtpSend(); |

**Проблемы с отправкой через SMTP**

Есть много вариантов возникновения ошибок при работе с SMTP, в основном, все они касаются прав доступа:

* имеет ли ваш хостинг разрешение на отправку почты через указанный SMTP?
* Возможно, перед подключением по SMTP необходимо подключиться по протоколу POP (сначала скачать – потом отправлять);
* вах хостинг требует авторизации соединения;
* вы уверены, что вы ввели правильные параметры пользователя и пароля?

К сожалению, я не уверен, что PHPMailer поддерживает соединение по протоколу POP перед соединением по SMTP. Если это создает проблему – свяжитесь со своим администратором и попросите хозяина релея прописать ваш IP адрес для сервера.

**Использование qmail и Sendmail**

Если вы хотите обойти функцию **[mail()](http://php.net/mail" \t "_blank)** PHP и использовать бинарные программы для работы с почтой, то вам необходимо определить тип почтового сервера, а затем PHPMailer запустит указанный бинарный файл, что, возможно, немного ускорит операции. Пример:

|  |
| --- |
| //qmail пример:  // настройте класс  $mailer->IsQmail();  $mailer->Send();  // Sendmail пример:  // настройте класс  $mailer->IsSendmail();  $mailer->Send(); |

Как видите, это довольно легко! Давайте перейдем к построению простого списка адресатов.

**Пример списка рассылки**

Теперь, когда вы уже имеете достаточное представление о работе PHPMailer, давайте рассмотрим некоторые особенности списков рассылки. Выскажу несколько соображений о том, что вам может понадобиться при рассылке: настройка темы письма, тела письма. И, что наиболее важно, необходимо скрыть адреса получателей друг от друга.

Следующий пример предполагает, что у вас уже есть некоторая база данных со списком адресов и вы знаете как соединиться с базой MySQL. Пока мы предположим, что вы имеете что-то следующей структуры:

* Имя
* Фамилия
* Электронный адрес
* Тип письма (text/html)

Предположим, что вы имеете 50 пользователей и хотите сделать настраиваемую рассылку по этим адресам. Следующий пример поможет вам в этом:

|  |
| --- |
| // подключаем файл конфигурации  require\_once($\_SERVER['DOCUMENT\_ROOT'].'/config.php');  // подключаем класс FreakMailer  require\_once($\_SERVER['DOCUMENT\_ROOT'].'/lib/MailClass.inc');  // Настраиваем тело письма  $textBody = "Dear {MEMBER\_NAME},  Check out PHP Freaks: http://www.phpfreaks.com  Sincerely,  Admin";  $htmlBody = "Dear {MEMBER\_NAME},<br /><br />  Check out PHP Freaks: http://www.phpfreaks.com<br /><br />Sincerely,<br />Admin";  // инициализируем класс  $mailer = new FreakMailer();  // получаем список адресов пользователей  $sql = mysql\_query("SELECT FirstName,LastName,EmailAddress,MailType FROM users WHERE 1");  while($row = mysql\_fetch\_object($sql))  {  // Отправляем письма в цикле  $member\_name = $row->FirstName;  if(!empty($row->LastName))  {  $member\_name .= ' '.$row->LastName;  }    if($row->MailType == 'html')  {  $mailer->Body = str\_replace('{MEMBER\_NAME}', $member\_name, $htmlBody);  $mailer->IsHTML(true);  $mailer->AltBody = str\_replace('{MEMBER\_NAME}', $member\_name, $textBody);  }  else  {  $mailer->Body = str\_replace('{MEMBER\_NAME}', $member\_name, $textBody);  $mailer->isHTML(false);  }    $mailer->AddAddress($row->EmailAddress, $member\_name);    $mailer->Send();  $mailer->ClearAddresses();  $mailer->ClearAttachments();  $mailer->IsHTML(false);  echo "Письмо отправлено. Пользователь: $member\_name ";  } |

**Разбор кода списка рассылки**

Первая часть такая же, как и в нашем первом примере – мы подключаем необходимые файлы.

|  |
| --- |
| require\_once($\_SERVER['DOCUMENT\_ROOT'].'/config.php');  require\_once($\_SERVER['DOCUMENT\_ROOT'].'/lib/MailClass.inc'); |

Далее, мы определяем тело нашего письма. Вы можете также читать его из файла, но мы, для простоты, держим его внутри этого скрипта.

|  |
| --- |
| $textBody = "Dear {MEMBER\_NAME},  Check out PHP Freaks: http://www.phpfreaks.com  Sincerely,  Admin";  $htmlBody = "Dear {MEMBER\_NAME},<br /><br />  Check out PHP Freaks: http://www.phpfreaks.com<br /><br />Sincerely,<br />Admin"; |

Обратите внимание на условный тег **{MEMBER\_NAME}**. Позже, когда мы будем выполнять действия в цикле, командой **[str\_replace()](http://php.net/str_replace" \t "_blank)** мы заменим этот тег на имя пользователя из базы данных.

Далее, мы инициализируем класс для работы с почтой.

|  |
| --- |
| $mailer = new FreakMailer(); |

Далее, мы получаем данные об адресатах из базы данных. И организуем цикл, в котором получаем последовательно каждый ряд в объект.

|  |
| --- |
| $sql = mysql\_query("SELECT FirstName,LastName,EmailAddress,MailType FROM users WHERE 1");  while($row = mysql\_fetch\_object($sql))  { |

Теперь мы дошли до места, где происходит непосредственная отсылка почты. Черезвычайно важно разобраться в том, что мы здесь делам. Первая часть кода в основном занимается формированием переменной **$member\_name**, которая будет содержать все данные текущего пользователя.

|  |
| --- |
| $member\_name = $row->FirstName;  if(!empty($row->LastName))  {  $member\_name .= ' '.$row->LastName;  } |

Следующая часть кода определяет предпочтения пользователя – HTML или plain-text. Если пользователь предпочитает HTML, то мы пошлем ему письмо ввиде HTML с включенным телом текстового письма. В противном случае – пошлем только текстовый вариант.

|  |
| --- |
| if($row->MailType == 'html')  {  $mailer->Body = str\_replace('{MEMBER\_NAME}', $member\_name, $htmlBody);  $mailer->IsHTML(true);  $mailer->AltBody = str\_replace('{MEMBER\_NAME}', $member\_name, $textBody);  }  else  {  $mailer->Body = str\_replace('{MEMBER\_NAME}', $member\_name, $textBody);  $mailer->isHTML(false);  } |

Внимательно проанализируйте текст выше, пока не достигните полного понимания. Вся необходимая информация для этого была в предыдущих частях нашего материала.

Наконец, мы отсылаем письмо и очищаем адреса в объекте $mailer от любых вложений.

|  |
| --- |
| $mailer->AddAddress($row->EmailAddress, $member\_name);  $mailer->Send();  $mailer->ClearAddresses();  $mailer->ClearAttahements();  $mailer->IsHTML(false);  echo "Mail sent to: $member\_name "; |

**ЗАМЕЧАНИЕ:**

Удостоверьтесь, что вы используете функции **$mailer->ClearAddresses()** и **$mailer->ClearAttahements()**, иначе адреса пользователей будут добавляться в конец списка.