Sending Emails in PHP with PHPMailer

Manage your PHP libraries with Composer. Watch our screencast PHP Dependencies Made Easy with Composer to learn how.

PHPMailer is one of the most popular open source PHP libraries to send emails with. It was first released way back in 2001 and since then it has become a PHP developer's favorite way of sending emails programmatically, aside from a few other fan favorites like Swiftmailer.

In this article we'll talk about why you should use PHPMailer instead of PHP's `mail()` function and we'll show some code samples on how to use this library.

Is it an alternative to PHP's `mail()` function?

In most cases, it's an alternative to PHP's `mail()` function, but there are many other cases where the `mail()` function is simply not flexible enough to achieve what you need.

First of all, PHPMailer provides an object oriented interface, whereas `mail()` is not object oriented. PHP developers generally hate to create `$headers` strings while sending emails using the `mail()` function because they require a lot of escaping – PHPMailer makes this a breeze. Developers also need to write dirty code (escaping characters, encoding and formatting) to send attachments and HTML based emails when using the `mail()` function whereas PHPMailer makes this painless.

Also, the `mail()` function requires a local mail server to send out emails. PHPMailer can use a non-local mail server (SMTP) if you have authentication.

Further advantages include:

- It can print various kinds of errors messages in more than 40 languages when it fails to send an email
- Integrated SMTP protocol support and authentication over SSL and TLS
- Can send alternative plaintext version of email for non-HTML email clients
- Very active developer community which keeps it secure and up to date

PHPMailer is also used by popular PHP content management systems like WordPress, Drupal, Joomla etc.

Installing PHPMailer

You can install PHPMailer using Composer:

```
composer require phpmailer/phpmailer
```

Sending Email from Local Web Server using PHPMailer

Here is the simplest example of sending an email from a local web server using PHPMailer

```
<?php

require_once "vendor/autoload.php";

//PHPMailer Object
$mail = new PHPMailer;
```

```
//From email address and name
$mail->From = "from@yourdomain.com";
$mail->FromName = "Full Name";

//To address and name
$mail->addAddress("recepient1@example.com", "Recepient Name");
$mail->addAddress("recepient1@example.com"); //Recipient name is optional

//Address to which recipient will reply
$mail->addReplyTo("reply@yourdomain.com", "Reply");

//CC and BCC
$mail->addCC("cc@example.com");
$mail->addBCC("bcc@example.com");

//Send HTML or Plain Text email
$mail->isHTML(true);

$mail->Subject = "Subject Text";
$mail->Body = "<i>Mail body in HTML</i>";
$mail->AltBody = "This is the plain text version of the email content";

if(!$mail->send())
{
    echo "Mailer Error: " . $mail->ErrorInfo;
}
else
{
    echo "Message has been sent successfully";
}
```

The code and comments should be sufficiently clear to explain everything that’s going on.

Sending an E-Mail with Attachments

Let’s see an example on how to send an email with attachments using PHPMailer.

```
 <?php

require_once "vendor/autoload.php";

$mail = new PHPMailer;

$mail->From = "from@yourdomain.com";
$mail->FromName = "Full Name";

$mail->addAddress("recipient1@example.com", "Recipient Name");

//Provide file path and name of the attachments
$mail->addAttachment("file.txt", "File.txt");
$mail->addAttachment("images/profile.png"); //Filename is optional

$mail->isHTML(true);

$mail->Subject = "Subject Text";
$mail->Body = "<i>Mail body in HTML</i>";
$mail->AltBody = "This is the plain text version of the email content";

if(!$mail->send())
{
    echo "Mailer Error: " . $mail->ErrorInfo;
}
else
{
    echo "Message has been sent successfully";
}
```

Here we are attaching two files i.e., `file.txt` which resides in the same directory as the script and `images/profile.png` `images` which resides in directory of the script directory.

To add attachments to the email we just need to call the function `addAttachment` of the PHPMailer object by passing the file path as argument. For attaching multiple files we need to call it multiple times.

Using SMTP

You can use the mail server of an another host to send email, but for this you first need to have authentication. For example: to send an email from Gmail's mail server you need to have a Gmail account.

SMTP is a protocol used by mail clients to send an email send request to a mail server. Once the mail server verifies the email it sends it to the destination mail server.

Here is an example of sending an email from Gmail's mail server from your domain. You don't need a local mail server to run the code. We will be using the SMTP protocol:

```
 <?php

require_once "vendor/autoload.php";

$mail = new PHPMailer;

//Enable SMTP debugging.
$mail->SMTPDebug = 3;
//Set PHPMailer to use SMTP.
$mail->isSMTP();
//Set SMTP host name
$mail->Host = "smtp.gmail.com";
//Set this to true if SMTP host requires authentication to send email
$mail->SMTPAuth = true;
//Provide username and password
$mail->Username = "name@gmail.com";
$mail->Password = "super_secret_password";
//If SMTP requires TLS encryption then set it
$mail->SMTPSecure = "tls";
//Set TCP port to connect to
```

```
$mail->Port = 587;

$mail->From = "name@gmail.com";
$mail->FromName = "Full Name";

$mail->addAddress("name@example.com", "Recepient Name");

$mail->isHTML(true);

$mail->Subject = "Subject Text";
$mail->Body = "<i>Mail body in HTML</i>";
$mail->AltBody = "This is the plain text version of the email content";

if(!$mail->send())
{
    echo "Mailer Error: " . $mail->ErrorInfo;
}
else
{
    echo "Message has been sent successfully";
}
```

Gmail requires TLS encryption over SMTP so we set it accordingly. Before you send via SMTP, you need to find out the host name, port number, encryption type if required and if authentication is required you also need the username and password. Note that having a two-factor authentication enabled on Gmail won’t let you use their SMTP with username/password – instead, additional configuration will be required.

One big advantage in using remote SMTP over local mail is that if you use PHP’s `mail()` function to send email with the `from` address domain set to anything other than the local domain name (name of the server), then the recipient’s email server’s attack filters will mark it as spam. For example, if you send an email from a server with actual host name `example.com` with the `from` address `name@gmail.com` to `name@yahoo.com`, then Yahoo’s servers will mark it as spam or display a message to the user not to trust the email because the mail’s origin is `example.com` and yet it presents itself as if coming from `gmail.com`. Although you own `name@gmail.com`, there is no way for Yahoo to find that out.

Retrieving E-Mails using POP3

PHPMailer also allows POP-before-SMTP verification to send emails. In other words, you can authenticate using POP and send email using SMTP. Sadly, PHPMailer doesn’t support retrieving emails from mail servers using the POP3 protocol. It’s limited to only sending emails.

Displaying Localized Error Messages

`$mail->ErrorInfo` can return error messages in 43 different languages.

To display error messages in a different language, copy the `language` directory from PHPMailer’s source code to the project directory.

To return error messages in, for example, Russian, set the PHPMailer object to the Russian language using the below method call:

```
$mail->setLanguage("ru");
```

You can also add your own language files to the `language` directory.

Conclusion

If you are a PHP developer, there is little chance of avoiding having to send emails programmatically. While you may opt for third party services like Mandrill or Sendgrid, sometimes that just isn’t an option, and rolling your own email sending library even less so. That’s where PHPMailer and its alternatives (Zend Mail, Swiftmailer, etc..) come in.

You can learn about this library’s APIs in the official documentation. Do you use PHPMailer? Or do you rather rely on fully remote API based solutions? Let us know in the comments!

Are you getting bogged down with PHP library dependencies? Watch our screencast and learn about how Composer can help you manage this for you.

Meet the author

Narayan Prusty □ □

Narayan is a web astronaut. He is the founder of QNimate. He loves teaching. He loves to share ideas. When not coding he enjoys playing football. You will often find him at QScutter classes.

All PHP Books

Replies

PetitPaul

▪ April 24, 2015

I've been using phpmailer for years and it works real fine for me.

If you need to send email in languages that have accented characters, don't forget to set the proper character encoding. I believe phpmailer assumes ISO-latin 1, but if your text is in UTF-8 all the non standard characters will be screwed-up:

```
$mail->CharSet = 'UTF-8';
```

Also, it seems recommended to end your script by:

```
$mail->ClearAllRecipients();
$mail->ClearAttachments();
```

Don't forget to set the "sender" option properly as most anti-spam soft will reject mail without a "sender".

```
$mail->Sender = 'somebody@domain.com';
```

Too bad, phpmailer can't intelligently translate html to text by itself. I use a class html2text available here:

https://github.com/tcz/html2text

but it does not seem to be updated. Any better option?

Synchro

▪ July 7, 2015

You don't need to call `clearAllRecipients` or `clearAllAttachments` unless you're in a loop where you're reusing the same instance of PHPMailer, such as in the mailing list example provided with PHPMailer.

`Sender` is used to provide a path back to the origin - it gets converted into the return-path by the receiver. It's not just used in a sender header within the message, it's also used as the SMTP envelope sender, and so is commonly used for VERP addressing. If you don't set it yourself, your mail server will do it for you, but that may mean it's not set to what you expect.

html to text conversion is really a bit outside PHPMailer's scope - there used to be a bundled converter, but that had to be removed for license reasons. Now there is only a basic converter, but you can inject your own through the html2text function that is also used internally when you call `msgHTML`.

Reply To This Discussion

1 more reply

Stuff We Do

- Premium
- Forums
- Corporate Memberships

- Jobs
- Deals

About

- Our Story
- Press Room

Contact

- Contact Us
- FAQ
- Write for Us
- Advertise

Legals

- Terms of Use
- Privacy Policy

Connect

□ □ □ □