Send Mail with PHP

If you're a PHP application developer, you might need to send email notifications through your custom PHP applications. With XAMPP, there are two ways to do this: you can use the included *sendmail.exe* binary, or you can use the third-party PHPMailer class. This guide walks you through both options.

NOTE

This guide assumes that you have an existing Gmail account, and that you will use Gmail's SMTP server to transmit email. You can also use your organization or ISP's mail server, although you'll need to replace key details, such as the SMTP server name and port, with appropriate values for your environment. Your network administrator or ISP support desk will typically be able to provide this information. The example code provided in this tutorial is illustrative only and should not be used in production environments.

To configure XAMPP to use *sendmail.exe* for email notifications, follow these steps:

1. Edit the *php.ini* file in the *php*\ subdirectory of your XAMPP installation directory (usually, *C:\xampp*). Within this file, find the [mail function] section and replace it with the following directives:

   ```
   sendmail_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"
   ```

2. Edit the *sendmail.ini* file in the *sendmail*\ subdirectory of your XAMPP installation directory. Within this file, find the [sendmail] section and replace it with the following directives:

   ```
   smtp_server=smtp.gmail.com
   smtp_port=465
   smtp_ssl=auto
   error_logfile=error.log
   auth_username=your-gmail-username@gmail.com
   auth_password=your-gmail-password
   ```

   Remember to replace the dummy values shown with your actual Gmail address and account password.

3. Restart the Apache server using the XAMPP control panel.

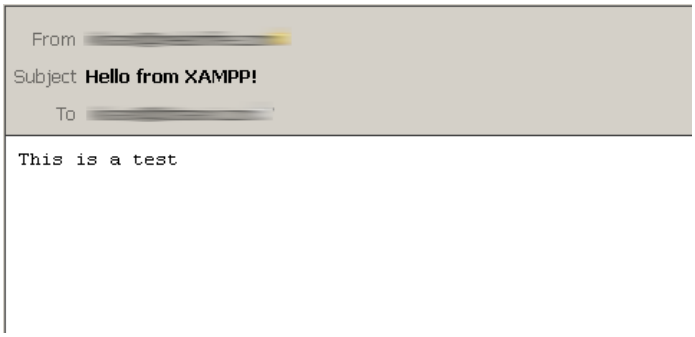
You can now use PHP's *mail()* function to send email from your application. To illustrate how it can be used with your Gmail account, use your text editor to create an example script named *sendmail.php* in the *htdocs*\ subdirectory and fill it with the following code. Once done, save your changes.

```
<?php
$to = 'recipients@email-address.com';
$subject = 'Hello from XAMPP!';
$message = 'This is a test';
$headers = "From: your@email-address.com\r\n";
if (mail($to, $subject, $message, $headers)) {
   echo "SUCCESS";
} else {
   echo "ERROR";
}
```

Remember to replace the dummy values shown with valid email addresses. For this simple test, use your own

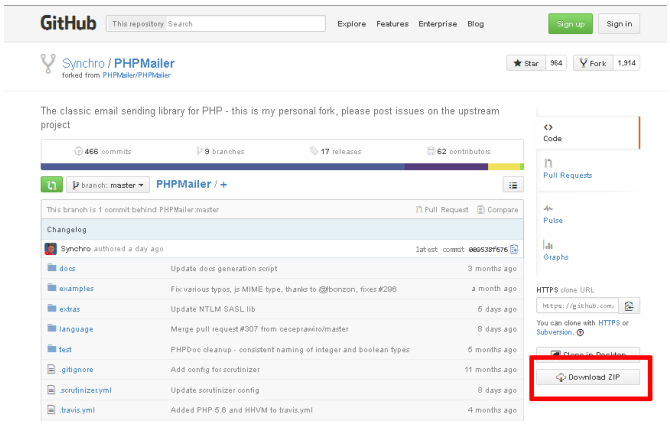
email address as the recipient address.

Now, browse to the URL http://localhost/sendmail.php to execute the script and send the email message. If all goes well, you should see a success notification in your browser. If you used your own email address for the recipient address, you should also receive the email message.



To configure XAMPP to use PHPMailer for email notifications, follow these steps:

1. Download PHPMailer from its Github repository using the "Download Zip" button.



2. Create a directory for your new application within the *htdocs*\ subdirectory of your XAMPP installation directory. In this tutorial, the application directory is named *example*\.

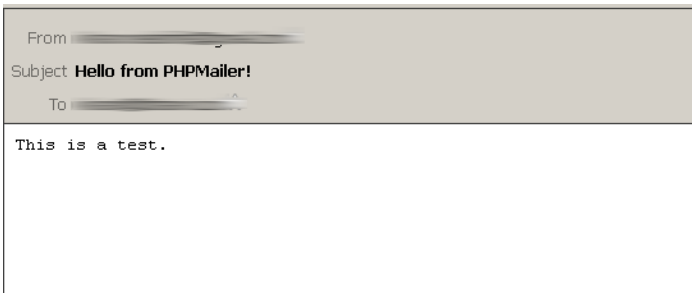
3. Extract the contents of the PHPMailer ZIP archive to the application directory.

You can now use PHPMailer to send email from your application. To illustrate how it can be used with your Gmail account, use your text editor to create an example script named *phpmailer.php* in the application directory, and fill it with the following code. Once done, save your changes.

```
<?php
require 'PHPMailer-master/PHPMailerAutoload.php';
$mail = new PHPMailer;
$mail->isSMTP();
$mail->SMTPSecure = 'ssl';
$mail->SMTPAuth = true;
$mail->Host = 'smtp.gmail.com';
$mail->Port = 465;
$mail->Username = 'your-gmail-username@gmail.com';
$mail->Password = 'your-gmail-password';
$mail->setFrom('your@email-address.com');
$mail->addAddress('recipients@email-address.com');
$mail->Subject = 'Hello from PHPMailer!';
$mail->Body = 'This is a test.';
//send the message, check for errors
if (!$mail->send()) {
    echo "ERROR: " . $mail->ErrorInfo;
} else {
    echo "SUCCESS";
}
```

Remember to replace the dummy values shown with your actual Gmail address and account password. You should also use a valid sender and recipient address. For this simple test, use your own email address as the recipient address.

Now, browse to the URL http://localhost/example/phpmailer.php. This should execute the script and send the email message. If all goes well, you should see a success notification in your browser. If you used your own email address for the recipient address, you should also receive the email message.



NOTE

As a security precaution, Gmail will automatically rewrite the From: and Reply-to: headers in your email message to reflect your Gmail address. If you want to avoid this, you must add and validate your custom email address in your Gmail account as a sender. Refer to Gmail's documentation for more information and important restrictions.