

# Gửi Email sử dụng PHP

PHP phải được định cấu hình một cách chính xác trong **php.ini** file với các chi tiết về cách hệ thống của bạn gửi email. Mở php.ini có sẵn trong thư mục **/etc/** và tìm phần có bắt đầu với **[mail function]**.

Người dùng Windows nên chắc chắn rằng có 2 chỉ thị được cung cấp. Cái đầu tiên được gọi là SMTP, nó định nghĩa địa chỉ email Server của bạn. Thứ 2 được gọi là `sendmail_from`, nó định nghĩa địa chỉ email của riêng bạn.

Cấu hình cho Windows trông giống như sau:

```
[mail function]
; For Win32 only.
SMTP = smtp.secureserver.net

; For win32 only
sendmail_from = webmaster@tutorialspoint.com
```

Người dùng Linux cần cho PHP biết vị trí của ứng dụng *sendmail*. Đường dẫn path phải được xác định cho chỉ thị `sendmail_path`.

Cấu hình cho Linux trông giống như sau:

```
[mail function]
; For Win32 only.
SMTP =

; For win32 only
sendmail_from =

; For Unix only
sendmail_path = /usr/sbin/sendmail -t -i
```

Bây giờ chúng ta cùng thực hiện việc gửi Email bởi sử dụng PHP.

## Gửi email văn bản đơn giản trong PHP

PHP sử dụng hàm **mail()** để gửi một email. Hàm này yêu cầu 3 tham số bắt buộc mà xác định địa chỉ email người nhận, chủ đề của thông điệp (message) và nội dung thông điệp, ngoài ra có thêm 2 tham số tùy ý nữa.

```
mail( to, subject, message, headers, parameters );
```

Bảng dưới miêu tả các tham số này:

Tham số	Miêu tả
to	Bắt buộc. Chỉ ra địa chỉ email của người nhận
subject	Bắt buộc. Chỉ ra chủ đề của email. Tham số này không thể chứa kí tự newline
message	Bắt buộc. Chỉ ra nội dung thông điệp. Các dòng phân cách nhau bởi một LF (\n). Mỗi dòng không vượt quá 70 kí tự
headers	Tùy ý. Xác định đầu đề bổ sung, như From, Cc và Bcc. Các đầu đề bổ sung nên được tách biệt với một CRLF (\r\n)
parameters	Tùy ý. Xác định một tham số bổ sung cho chương trình gửi mail

Ngay khi hàm mail được gọi, PHP sẽ gửi email này, sau đó nó sẽ trả về true nếu thành công và false nếu thất bại.

Để gửi mail cho nhiều người nhận, tham số đầu tiên trong hàm mail() là danh sách người nhận được phân biệt nhau bởi dấu phẩy.

## Gửi HTML email trong PHP

Khi bạn gửi một thông điệp văn bản sử dụng PHP, khi đó tất cả nội dung sẽ được xử lý như văn bản đơn giản. Thậm chí nếu bạn thêm các thẻ HTML trong một thông điệp văn bản, nó sẽ hiển thị như văn bản thông thường và các thẻ HTML sẽ không được định dạng theo cú pháp HTML. Tuy nhiên, PHP cung cấp tùy chọn gửi thông điệp HTML như một thông điệp HTML thực sự.

Khi gửi một thông điệp email, bạn có thể xác định một Mime version, content-type và tập kí tự để gửi một email HTML.

## Ví dụ

Ví dụ sau sẽ gửi một thông điệp email HTML tới xyz@somedomain.com sao chép nó sang afgh@somedomain.com. Bạn có thể code chương trình này trong theo một cách mà nó sẽ nhận được tất cả các nội dung từ người sử dụng và sau đó nó sẽ gửi một email.

```
<html>

  <head>
    <title>Sending HTML email using PHP</title>
  </head>

  <body>

    <?php
      $to = "xyz@somedomain.com";
      $subject = "This is subject";

      $message = "<b>This is HTML message.</b>";
      $message .= "<h1>This is headline.</h1>";

      $header = "From:abc@somedomain.com \r\n";
      $header = "Cc:afgh@somedomain.com \r\n";
      $header .= "MIME-Version: 1.0\r\n";
      $header .= "Content-type: text/html\r\n";

      $retval = mail ($to,$subject,$message,$header);

      if( $retval == true )
      {
        echo "Message sent successfully...";
      }
    }
  </body>
</html>
```

```
else
{
    echo "Message could not be sent...";
}
?>

</body>
</html>
```

## Gửi đính kèm cùng Email trong PHP

Để gửi một email với nội dung hỗn hợp thì điều cần thiết là thiết lập **Content-type** header thành **multipart/mixed**. Sau đó văn bản và phần đính kèm có thể được xác định bên trong **boundaries**.

Một boundary được bắt đầu với 2 dấu gạch ngang theo sau bởi một số duy nhất mà không thể xuất hiện trong thông điệp của email. Hàm **md5()** của PHP được sử dụng để tạo một số hexa 32 kí số và là số duy nhất. Một boundary cuối cùng biểu thị phần cuối cùng của email cũng phải kết thúc với hai dấu gạch nối.

```
<?php
// request variables // important
$from=$_REQUEST["from"];
$email=$_REQUEST["email"];
$file=$_REQUEST["file"];

if ($file)
{
    function mail_attachment ($from , $to, $subject, $message, $attachment){
        $fileatt = $attachment; // Path to the file
        $fileatt_type = "application/octet-stream"; // File Type

        $start = strrpos($attachment, '/') == -1 ? strrpos($attachment, '//') :
        strrpos($attachment, '/')+1;

        $fileatt_name = substr($attachment, $start, strlen($attachment)); // Filename that
        will be used for the file as the attachment
```

```
$email_from = $from; // Who the email is from
$subject = "New Attachment Message";

$email_subject = $subject; // The Subject of the email
$email_txt = $message; // Message that the email has in it
$email_to = $to; // Who the email is to

$headers = "From: ".$email_from;
$file = fopen($fileatt,'rb');
$data = fread($file,filesize($fileatt));
fclose($file);

$msg_txt="\n\n You have recieved a new attachment message from $from";
$semi_rand = md5(time());
$mime_boundary = "==" . Multipart_Boundary_x{$semi_rand} . "x";

$headers .= "\nMIME-Version: 1.0\n" . "Content-Type: multipart/mixed;\n" . "
boundary=\"{$mime_boundary}\"";

$email_txt .= $msg_txt;

$email_message .= "This is a multi-part message in MIME format.\n\n" . "--
{$mime_boundary}\n" . "Content-Type:text/html; charset=\"iso-8859-1\"\n" . "Content-Transfer-
Encoding: 7bit\n\n" . $email_txt . "\n\n";

$data = chunk_split(base64_encode($data));

$email_message .= "--{$mime_boundary}\n" . "Content-Type: {$fileatt_type};\n" . "
name=\"{$fileatt_name}\" \n\n" . //"Content-Disposition: attachment;\n" . //"
filename=\"{$fileatt_name}\" \n\n" . "Content-Transfer-Encoding: base64\n\n" . $data . "\n\n" .
"--{$mime_boundary}--\n";

$ok = mail($email_to, $email_subject, $email_message, $headers);

if($ok)
{
    echo "File Sent Successfully.";
    unlink($attachment); // delete a file after attachment sent.
```

```
    }

    else
    {
        die("Sorry but the email could not be sent. Please go back and try again!");
    }
}

move_uploaded_file($_FILES["filea"]["tmp_name"], 'temp/'.basename($_FILES['filea']['name']));

    mail_attachment("$from", "youremailaddress@gmail.com", "subject", "message",
("temp/".$_FILES["filea"]["name"]));
}
?>

<html>
    <head>

        <script language="javascript" type="text/javascript">
            function CheckData45()
            {
                with(document.filepost)
                {
                    if(filea.value != "")
                    {
                        document.getElementById('one').innerText = "Attaching File ... Please Wait";
                    }
                }
            }
        </script>

    </head>
    <body>

        <table width="100%" height="100%" border="0" cellpadding="0" cellspacing="0">
```

```
<tr>
  <td align="center">
    <form name="filepost" method="post" action="file.php" enctype="multipart/form-
data" id="file">

    <table width="300" border="0" cellspacing="0" cellpadding="0">
      <tr valign="bottom">
        <td height="20">Your Name:</td>
      </tr>

      <tr>
        <td><input name="from" type="text" id="from" size="30"></td>
      </tr>

      <tr valign="bottom">
        <td height="20">Your Email Address:</td>
      </tr>

      <tr>
        <td class="frmtxt2"><input name="emaila" type="text" id="emaila"
size="30"> </td>
      </tr>

      <tr>
        <td height="20" valign="bottom">Attach File:</td>
      </tr>

      <tr valign="bottom">
        <td valign="bottom"><input name="filea" type="file" id="filea"
size="16"></td>
      </tr>

      <tr>
```

```
        <td height="40" valign="middle"><input name="Reset2" type="reset"
id="Reset2" value="Reset">
        <input name="Submit2" type="submit" value="Submit" onClick="return
CheckData45()"></td>
    </tr>
</table>

</form>

<center>
    <table width="400">
        <tr>
            <td id="one">
            </td>
        </tr>
    </table>
</center>

</td>
</tr>
</table>

</body>
</html>
```