

Envío de formularios autenticados con PHPMailer



Es habitual encontrarnos en las **páginas web** un formulario de contacto con el que permitir a los usuarios que la visitan ponerse en contacto con el administrador del portal, ya sea para recibir cualquier tipo de información o para comentar algo referente a lo que se pueda ver en ese sitio. El uso de estos formularios es algo muy necesario para poder interactuar con los usuarios, pero en muchas ocasiones se convierte en un gran problema porque suelen ser objeto de ataques por parte de los hackers en busca de alguna brecha no segura, desde donde poder llevar a cabo sus acciones de envío de spam.

Ante esta situación, muchos proveedores de **hosting** han tomado medidas extra de seguridad para evitar estos problemas, las cuales pueden hacer que los envíos desde el formulario no se lleguen a realizar si no se hace de forma autenticada.

A lo largo de este White Paper os explicaremos cómo realizar este tipo de envío utilizando para ello la librería PHPMailer.

¿Qué es PHPMailer?

PHPMailer es una clase desarrollada en PHP cuyo principal objetivo es facilitar al desarrollador la tarea del envío de correos electrónicos desde la propia página web. Esta clase para enviar mails está basada en el componente active server ASPMail. Gracias a ella, los desarrolladores de aplicaciones web pueden hacer de forma sencilla tareas complejas como por ejemplo:

- Enviar correos electrónicos con archivos adjuntos.
- Enviar mensajes en formato HTML.
- Realizar envíos de mails utilizando el protocolo SMTP autenticado.

Además de poder utilizar el envío mediante SMTP, la librería también ofrece la posibilidad de realizar ese envío de otras dos formas distintas: mediante la función “mail()” de PHP y mediante el uso de “sendMail”. A pesar de disponer de estas tres opciones, lo más recomendado es hacer uso del envío vía SMTP debido a varias razones:

- PHPMailer permite hacer uso de varios servidores SMTP para hacer el envío. Al repartirlo entre varias máquinas se reducirá la carga de éstas.
- Se implementan todas las funcionalidades necesarias para añadir a varios destinatarios, las direcciones privadas y las opciones de respuesta.

Conceptos a tener en cuenta para empezar a utilizar PHPMailer

Para hacer uso de esta librería, lo primero que debemos hacer es **descargarla**. La librería estará formada por dos archivos:

- class.phpmailer.php: que es la librería propiamente dicha.
- class.smtp.php: que es el que nos permite hacer el envío de emails a través del SMTP.

Estos dos archivos son los que tendremos que añadir a nuestro proyecto cuando queramos hacer el envío de emails mediante PHPMailer.

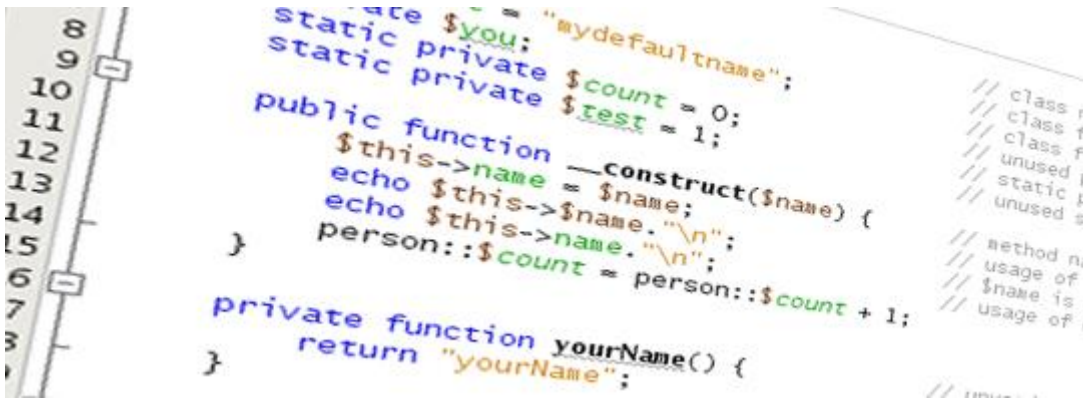
La librería nos ofrece una serie de propiedades que podemos utilizar a la hora de programar nuestro envío. Veamos las más importantes.

- **AltBody:** Permite establecer el cuerpo del mensaje en formato texto plano. Útil cuando estamos enviando mensaje en formato HTML y queremos enviar texto plano para aquellos casos en los que el servidor no interprete HTML.
- **Body:** Establece el cuerpo del mensaje. Puede ser texto simple o con formato HTML. Si es con formato HTML hay que ejecutar el método `IsHTML(True)`.
- **ConfirmReadingTo:** Establece la dirección de correo a la que se enviará una confirmación de lectura.
- **ErrorInfo:** Informa del error más reciente al intentar enviar un email.
- **From:** Establece la dirección de correo origen del mensaje.
- **FromName:** Sirve para establecer el nombre de quien realiza el envío.
- **Host:** Establece el servidor SMTP. Pueden ser varios separados por “;”.
- **Mailer:** Propiedad que nos permite indicar el método que queremos utilizar a la hora de hacer el envío. Puede ser mail, sendmail o smtp.
- **Password:** Establece la contraseña del servidor SMTP.
- **Port:** Para indicar el puerto SMTP. Por defecto es el 25 aunque hay servidores que utilizan uno alternativo como el 587, cosa que ocurre con los servidores de Hostalia.
- **ReplyTo:** Sirve para indicar todas las direcciones Reply-To.
- **SMTPAuth:** Sirve para indicar la autenticación SMTP. Por defecto tiene el valor “false”, pero si vamos a utilizar la autenticación debemos indicarlo como “true”.
- **Subject:** Establece el asunto del mensaje.
- **Username:** Aquí indicaremos el nombre de usuario SMTP que utilizaremos en la autenticación.

A la hora de utilizar la librería, no es obligatorio utilizar todos estos parámetros, sino únicamente aquellos que necesitamos. Además de las propiedades, la librería también nos ofrece una serie de métodos que podemos utilizar para hacer el **envío** de los correos. Veamos los más importantes.

- **AddAddress(\$address, \$name):** Método que nos permite añadir una dirección de destino. El parámetro \$name es opcional.
- **AddAttachment(\$path, \$name, [\$encoding = "base64"], [\$type = "application/octet-stream"]):** Método que nos permite adjuntar archivos a nuestros correos.
- **AddBCC(\$address, \$name):** Añade una dirección BCC.
- **AddCC(\$address, \$name):** Añade una dirección CC.
- **AddReplyTo(\$address, \$name):** Añade una dirección Reply-to.
- **IsHTML():** Establece si el cuerpo del mensaje contiene HTML.
- **IsSMTP():** Sirve para indicar qué se enviará mediante SMTP.
- **Send():** Método encargado de realizar el envío del mensaje. Devuelve ‘false’ en caso de producirse algún error.

Ejemplo de uso de la librería PHPMailer



Veamos a continuación un ejemplo completo de cómo debemos utilizar esta librería para realizar un envío utilizando para ello SMTP autenticado.

Lo primero que necesitaremos será conocer los datos de configuración del correo que queremos utilizar para hacer el envío. Los datos necesarios son:

- Nombre de usuario
- Contraseña del correo
- Servidor SMTP

Una vez que conocemos estos datos, empezaremos a crear el código que realizará esta acción. Lo primero de todo será añadir a nuestro proyecto las librerías necesarias.

```
require('PHPMailer/class.phpmailer.php');
```

```
require('PHPMailer/class.smtp.php');
```

Lo siguiente que hacemos es instanciar un objeto de la clase.

```
$mail = new PHPMailer();
```

A ese objeto le indicamos que vamos a utilizar el envío mediante SMTP.

```
$mail->isSMTP();
```

Y que vamos a utilizar el envío autenticado.

```
$mail->SMTPAuth = true;
```

Por último le indicamos que en el cuerpo del mensaje, enviaremos información con formato HTML.

```
$mail->isHTML(true);
```

A continuación le indicaremos los datos que tiene que utilizar para hacer el envío del correo mediante el protocolo SMTP Autenticado.

```
$mail->Host = "smtp.hostalia.com";
```

```
$mail->Username = "micuentaremitente@hostalia.com";
```

```
$mail->Password = "contraseña_de_la_cuenta_remitente";
```

Hecho este paso, indicamos el nombre de quien hace el envío y la dirección de mail que lo envía. Hay que tener cuidado de no confundir con el nombre y la dirección de mail del destinatario.

```
$mail->From = "micuentaremitente@miservidordecorreo.com";
```

```
$mail->FromName = "El nombre de remitente que queremos que aparezca";
```

A continuación indicamos el o los correos a los que se le envía el mail. En nuestro caso sólo pondremos uno, pero se puede añadir tanto como queramos.

```
$mail->AddAddress("destinatario@eldominioquesea.com");
```

Indicamos el asunto que llevará el mail.

```
$mail->Subject = "El asunto del mensaje";
```

El cuerpo del mensaje, que en nuestro caso contendrá código HTML.

```
$mail->Body = "<b>Mensaje con código HTML</b>";
```

Para terminar, le añadimos el texto que queremos que vean aquellos usuarios cuyos clientes de mail no muestran HTML. Se trata de un campo optativo pero muy recomendable de utilizar.

```
$mail->AltBody = "Este es el texto para clientes de correo que no muestren HTML";
```

Hechos todos estos pasos, sólo faltaría enviar el correo.

```
$mail->Send();
```

Si juntamos todo el código, nos quedaría algo similar a lo que os dejamos a continuación.

```
require('PHPMailer-master/class.phpmailer.php');
```

```
require('PHPMailer-master/class.smtp.php');
```

```
$mail = new PHPMailer();
```

```
$mail->IsSMTP();
```

```
$mail->SMTPAuth = true;
```

```
$mail->isHTML(true);
```

```
$mail->Host = "smtp.miservidordecorreo.com";  
  
$mail->Username = "micuentaremitente@miservidordecorreo.com";  
  
$mail->Password = "contraseñaadelacuentaremitente";  
  
$mail->From = "micuentaremitente@miservidordecorreo.com";  
  
$mail->FromName = "El nombre de remitente que queremos que aparezca";  
  
$mail->AddAddress("destinatario@eldominioquesea.com");  
  
$mail->Subject = "El asunto del mensaje";  
  
$mail->Body = "Este es el HTML que compondrá el cuerpo del mensaje, con sus formatos,  
etiquetas y demás"  
  
$mail->AltBody = "Este es el texto para clientes de correo que no muestren HTML";  
  
$mail->Send();
```

Con el código visto anteriormente ya sería suficiente para realizar el envío, pero si además quisiéramos hacerlo más completo podríamos haber añadido destinatarios ocultos mediante la siguiente línea de código.

```
$mail->AddBCC("destinatariooculto@hostalia.com");
```

E incluso añadir documentos adjuntos al correo.

```
$mail->addAttachment('ruta_al_documento_adjunto');
```

A lo largo de este WhitePaper hemos visto lo fácil que puede resultar hacer envíos desde PHP mediante el protocolo SMTP autenticado, garantizando de esta forma que el **correos electrónico** será enviado al destinatario.