

PHP Email Form

PHP Email Form is simple and easy to use PHP script for sending the data submitted by web HTML forms (like contact forms) to your email inbox. The library is created by the BootstrapMade team and available in the paid versions of templates published on BootstrapMade.com. It works out of the box with the included contact forms in our templates.

Setting up the PHP Email Form as contact form

We do provide contact forms in all our templates. The PHP Email Form works out of the box with our contact forms. You just need to configure your receiving email address if you have PHP support on your hosting.

1. If you already started working with the free version of the template: Download the pro version from your member area, unzip it and copy the **/assets/vendor/php-email-form/php-email-form.php** to the **/assets/vendor/php-email-form/** folder of your working project.
2. Edit **/forms/contact.php** and replace `contact@example.com` with your email address and you're done!

PHP Email Form uses the [PHP mail\(\) function](#) for mailing by default. The **/forms/contact.php** also comes with a configuration for SMTP, in case your hosting doesn't allow sending emails with the PHP mail() function. The SMTP configurations are commented. You need to uncomment the SMTP configuration lines and add your credentials.

Adding multiple receiving email addresses

You can also add multiple receiving email addresses for the submitted forms via **cc** and **bcc** methods. Edit and add the below code to your `forms/contact.php` just before the line `echo $contact->send();`

```
$contact->cc = array('ccreceiver1@example.com', 'ccreceiver2@example.com');  
$contact->bcc = array('bccreceiver1@example.com', 'bccreceiver2@example.com');
```

We use cookies to provide our services and enhance your experience. By using this website, you agree with our Privacy & Cookie Policy

OK



form, which is hidden for your users and not filled while submitting it. Spam bots miss it thinking that it's a standard required field and alert us it's activity. If the honeypot field is filled in, we can confidently reject the form as spam. To use the honeypot method:

First, edit your form and add an extra field, e.g:

```
<div class="form-group d-none">
<input type="text" class="form-control" name="first_name">
<div class="validate"></div>
</div>
```

Then, edit your **/forms/contact.php** and add the below code just before the line `echo $contact->send();`

```
$contact->honeypot = $_POST['first_name'];
```

You can change the field name *first_name* as you wish.

2. Spam Protection with Google reCaptcha

Google reCaptcha is the most popular form submission protection in the world. PHP Email Form comes with built in support for the latest version 3.0 of Google reCaptcha. Follow the below steps to enable Google reCaptcha protection for your form.

1. Go to [Google reCaptcha administration](#) website, set-up your website and obtain your keys (site key and secret key)
2. Add the below line to your web page in the footer part along with the other script inclusions.

```
<script src="https://www.google.com/recaptcha/api.js?render=Your_reCAPTCHA_site_key"></script>
```

Replace *Your_reCAPTCHA_site_key* with the site key you obtained in the step 1

3. Add `data-recaptcha-site-key` to your form tag, e.g:

```
<form action="forms/contact.php" method="post" role="form" class="php-email-form" data-recaptcha-site-key="Your_reCAPTCHA_site_key">
```

Replace *Your_reCAPTCHA_site_key* with the site key you obtained in the step 1

We use cookies to provide our services and enhance your experience. By using this website, you agree with our [Privacy & Cookie Policy](#)

OK



Adding "Accept terms/privacy policy" checkbox to your form

Here is an example code in case you need to add a checkbox to your form and require your users to accept your terms or privacy policy before submitting the form.

First, add the below code to your form, just before the submit button:

```
<div class="form-check form-group ps-0">
  <input id="privacy-policy" type="checkbox" name="privacy" value="accept" required>
  <label class="form-check-label ps-1" for="privacy-policy">
    Accept our <a href="terms.html">terms of service</a> and <a href="privacy.html">privacy
  </label>
</div>
```

Then, edit your `/forms/contact.php` and add the below code just before the line `echo $contact->send();`

```
if($_POST['privacy'] != 'accept') {
  die('Please, accept our terms of service and privacy policy');
}
```

Advanced Usage and Customization of PHP Email Form

The **PHP Email Form** can also be used for emailing any forms with unlimited custom inputs. It also comes with integrated SMTP support that allows you to send emails without a local mail server.

You can just duplicate the **forms/contact.php** to a new file and use it as a starting point, e.g: myform.php

You also need to set the action property to your new form to the newly created myform.php, e.g:

```
<form action="forms/myform.php" method="post" class="php-email-form">
```

Below is a full documentation of each line of the **forms/contact.php** and how you can customize them for your needs. First, below is the full content of the contact.php. Scroll down for detailed explanation.

We use cookies to provide our services and enhance your experience. By using this website, you agree with our Privacy & Cookie Policy

OK



```
$receiving_email_address = 'contact@example.com';

if( file_exists($php_email_form = '../assets/vendor/php-email-form/php-email-form.php' ))
include( $php_email_form );
} else {
die( 'Unable to load the "PHP Email Form" Library!' );
}

$contact = new PHP_Email_Form;
$contact->ajax = true;

$contact->to = $receiving_email_address;
$contact->from_name = $_POST['name'];
$contact->from_email = $_POST['email'];
$contact->subject = $_POST['subject'];

// Uncomment below code if you want to use SMTP to send emails. You need to enter your cor
/*
$contact->smtp = array(
'host' => 'example.com',
'username' => 'example',
'password' => 'pass',
'port' => '587'
);
*/

$contact->add_message( $_POST['name'], 'From');
$contact->add_message( $_POST['email'], 'Email');
$contact->add_message( $_POST['message'], 'Message', 10);

echo $contact->send();
?>
```

```
// Replace contact@example.com with your real receiving email address
$receiving_email_address = 'contact@example.com';
```

Here we set the receiving email address variable, which we will later use when setting up the PHP Email Form class properties. It's not required to do it this way, we just moved this configuration at the top in the contact.php since it's the only configuration setting that requires editing for the default

We use cookies to provide our services and enhance your experience. By using this website, you agree with our Privacy & Cookie Policy

OK



Here we check if the PHP Email Form library exists and then include it. The script will end execution if php-email-form.php library file doesn't exist.

```
$contact = new PHP_Email_Form;
```

Here we initiate the PHP Email Form and assign it to \$contact variable. You can change the \$contact to your own.

```
$contact->ajax = true;
```

We use the AJAX method of posting the form in our contact form. This setting checks if the post method is really coming from an AJAX call and outputs error if isn't. The `ajax` property is false by default

```
$contact->to = $receiving_email_address;  
$contact->from_name = $_POST['name'];  
$contact->from_email = $_POST['email'];  
$contact->subject = $_POST['subject'];
```

All of these four properties `to` `from_name` `from_email` `subject` are required and need to be set up.

The `to` property is the receiving email address of the form. You can set it directly from here as `$contact->to = 'contact@example.com';` or use the `$receiving_email_address` variable as we do.

The `from_name` property is the email sender name. In our contact form we capture the input value with name "name". You can set it directly from here as `$contact->from_name = 'Custom Name';` or other input value with a different name, e.g: `$contact->from_name = $_POST['sender_name'];`

The `from_email` property is the sender email address. In our contact form we capture the input value with name "email". You can set it directly from here as `$contact->from_email = 'custom@email.com';` or other input value with a different name, e.g: `$contact->from_email = $_POST['email_address'];`

The `subject` property is the email subject. In our contact form we capture the input value with name "subject". You can set it directly from here as `$contact->subject = 'My Subject';` or other input value with a different name, e.g: `$contact->subject = $_POST['form_subject'];`

We use cookies to provide our services and enhance your experience. By using this website, you agree with our Privacy & Cookie Policy

OK



PHP Email Form uses the [PHP mail\(\) function](#) for mailing by default. You can use SMTP if your hosting doesn't support the PHP mail() function or you can prefer the SMTP. To use SMTP, just uncomment the above code and add your credentials.

```
$contact->add_message( $_POST['name'], 'From');  
$contact->add_message( $_POST['email'], 'Email');  
$contact->add_message( $_POST['message'], 'Message', 10);
```

Here we prepare and compose the message content that will be included in the email body with using the `add_message()` method, which is a part of PHP Mail Form. It can be used unlimited times based on your needs. As you can see, we use it 3 times in our contact.php. The formatted message that you'll receive in your email will be as an example below:

Email: Sender Name

From: sender@email.com

Message: The message text

`add_message()` method accepts 3 parameters. The first one is the message text, the second one is a title/label and the third one is length check number (it will output error if the provided message text characters count is under the set length).

```
echo $contact->send();
```

It will output a text message "OK" if the email is sent successfully or an error message if the email is not sent for some reason.

Customizing the error messages

You can also customize the returned error messages by the PHP Email Form. Here are the available error message properties and their default values.

```
$contact->invalid_to_email = 'Email to (receiving email address) is empty or invalid!';  
$contact->invalid_from_name = 'From Name is empty!';  
$contact->invalid_from_email = 'Email from: is empty or invalid!';  
$contact->invalid_subject = 'Subject is too short or empty!';  
$contact->short = 'is too short or empty!'; // If the length check number is set and the  
$contact->ajax_error = 'Sorry, the request should be an Ajax POST'; // If ajax property is
```

We use cookies to provide our services and enhance your experience. By using this website, you agree with our [Privacy & Cookie Policy](#)

OK



function is not supported.

Join Our Newsletter

Subscribe to our newsletter to receive emails about new template releases and updates



High quality free Bootstrap website templates. Build your next website with our modern and fully responsive web templates!



Company

- > [About](#)
- > [License](#)
- > [PHP Email Form](#)
- > [Terms of service](#)
- > [Privacy policy](#)

Popular Templates

We use cookies to provide our services and enhance your experience. By using this website, you agree with our Privacy & Cookie Policy

OK



My Resume

Popular Categories

- > [Bootstrap 5 Templates](#)
- > [Website Templates](#)
- > [One Page Template](#)
- > [Portfolio Templates](#)
- > [Free Website Templates](#)

Copyright © 2022 [BootstrapMade](#)

We use cookies to provide our services and enhance your experience. By using this website, you agree with our Privacy & Cookie Policy

OK