

HTML.form.guide

All Articles

Home » Email Forms

How to create PHP based email form with file attachment

 php email form attachment

email form

This article shows you how to create a PHP based email form that supports file attachment. The article will also show you how to validate the type and size of the uploaded file.

Note: It is easy to create complete file upload forms using **Simfatic Forms**. Visually design the form, get form submissions by email, save submissions to database and more. No coding required. **Read more about Simfatic Forms here.**

The HTML form with file upload box

The code for an HTML form with a file upload box is given below. User can click on the 'Browse' button to select the file from his/her local machine.

```
<form method="POST"
name="email_form_with_php"
action="php-form-action.php"
enctype="multipart/form-data">

<label for='name'>Name: </label>
<input type="text" name="name" >

<label for='email'>Email: </label>
<input type="text" name="email" >

<label for='message'>Message:
</label>
<textarea name="message">
</textarea>

<label for='uploaded_file'>Select
A File To Upload:</label>
<input type="file"
name="uploaded_file">

<input type="submit"
value="Submit" name='submit'>
</form>
```

The form will look like this:



Please note that we have added:

```
"enctype="multipart/form-data"
```

while defining the <form> tag. This is to tell the browser that this form will be used to upload files. Then we have added the "name" and "email" fields to collect the user info. The third form field is the file upload box.

```
<input type="file"
name="uploaded_file">
```

On hitting the "Submit" button, the form data along with the file data is posted to the script pointed to by the 'action' attribute of the form.

Getting the uploaded file in the PHP script

In the PHP script, we will first validate the submission and if the validation succeeds, we will send the submission by email.

We can access the uploaded file and its different attributes by using the **\$_FILES** array. This array will contain the name, size, path and other attributes of the uploaded file. The code below gets the name, type, and size of the uploaded file:

```
//Get the uploaded file
information
$name_of_uploaded_file =

basename($_FILES['uploaded_file']
['name']);

//get the file extension of the
file
$type_of_uploaded_file =
substr($name_of_uploaded_file,

strpos($name_of_uploaded_file,
'.') + 1);

$size_of_uploaded_file =
$_FILES["uploaded_file"]
["size"]/1024;//size in KBs
```

The code above is getting the different attributes of the uploaded file from the **\$_FILES[]** array.

Validating the size and extension of the uploaded file

Suppose we don't want to allow files greater than the size of 100KB and we only want to allow image files to be uploaded. The validation code goes like this:

```
//Settings
$max_allowed_file_size = 100; //
size in KB
$allowed_extensions = array("jpg",
"jpeg", "gif", "bmp");

//Validations
if($size_of_uploaded_file >
$max_allowed_file_size )
{
$errors .= "\n Size of file
should be less than
$max_allowed_file_size";
}

//----- Validate the file
extension -----
$allowed_ext = false;
for($i=0;
$i<sizeof($allowed_extensions);
$i++)
{
if(strcasecmp($allowed_extensions[$i]
== 0)
{
$allowed_ext = true;
}
}

if(!$allowed_ext)
{
$errors .= "\n The uploaded file
is not supported file type. "
" Only the following file types
are supported:
implode(',', $allowed_extensions);
}


```

In the code above, we are validating the file size and type. We have the maximum allowed file (**\$max_allowed_file_size**) size set to 100KB. The **\$allowed_extensions** array contains the file extensions of all allowed file types. The validation code checks to see whether the file extension matches any of the extensions in the **\$allowed_extensions** array.

If there are errors found in the validation, the error is displayed. Else we proceed with sending the email.

Copy the uploaded file

Now, its time to send the uploaded file with the user message to the recipient's email address.

First of all, we shall copy the file to a folder on the server.

```
//copy the temp. uploaded file to
uploads folder
$path_of_uploaded_file =
$upload_folder .
$name_of_uploaded_file;
$tmp_path =
$_FILES["uploaded_file"]
["tmp_name"];

if(is_uploaded_file($tmp_path))
{

if(!copy($tmp_path,$path_of_uploaded
{
$errors .= '\n error while
copying the uploaded file';
}
}


```

This code copies the uploaded file to the 'uploads' folder. You can change the uploads folder by updating **\$upload_folder**. Please make sure that "uploads" folder has "777" permissions.

Sending the Email

The next step is to compose and send the email. We will use the **Pear library** for composing and sending the email. (see the Pear installation instructions **below**) The pear classes **PEAR::Mail** and **PEAR::Mail_Mime** are used for sending the email with the attachment.

First, we need to include the pear library files for these classes.

```
include_once('Mail.php');
include_once('Mail_Mime/mime.php');
```

The code below composes and sends the email

```
$message = new Mail_mime();

$message->setTXTBody($text);

$message-
>addAttachment($path_of_uploaded_fil

$body = $message->get();

$extraheaders =
array("From"=>$from,
"Subject"=>$subject,"Reply-
To"=>$visitor_email);

$headers = $message-
>headers($extraheaders);

$mail = Mail::factory("mail");

$mail->send($to, $headers, $body);


```

Mail_mime() class helps in composing a **MIME message**. In the code above, a Mail_mime object is created, the text body is updated (**\$message->setTXTBody(\$text);**) and the attachment is added (**\$message->addAttachment(file)**)

The MIME-encoded message is then sent using the Mail class.

The sample PHP upload form

[Click here to download php-email-form-attachment.zip](#)

The download contains a complete PHP upload form that sends the uploaded by email.

How to Install the PEAR Library

In this article, we used the **PEAR::Mail** and **PEAR::Mail_Mime** classes to send the email with attachment. Before using these classes, you need to install the PEAR package on your server. It is beyond the scope of this tutorial to discuss the installation of PEAR. But, I want to give you a quick tip. Get the PEAR installer script from <http://pear.php.net/go-pear>

Save the file as "pear-installer.php". Upload this file to your server in any directory. Then run this file from your browser, like this:

<http://www.yourdomain.com/pear-installer.php>

This display the web interface to install the PEAR on your website. The interface shows detailed instructions. After Pear is installed, search and install packages "mail" and "mail_mime".

Also see: more free [file upload forms from ReusableForms.com](#)



Build Forms with Calculations in 5 minutes!

Try Simfatic Forms





Build Unlimited Online Forms

Simfatic Forms Form Builder

Download

Categories

Contact Forms

Code Snippets

HTML Forms

Best Practices

HTML5 Forms

Form Widgets

PHP Form Handling

jQuery Form Handling

Email Forms

Form Mail

Web Forms

Calculation Forms

Checkboxes

File Upload

Form Action

Google Forms

Fix typos in your Laravel app

Automated spelling and grammar

Automated grammar checks and fixes for your Laravel app - no more typos in the user comms

grammarchi.com

OPEN

About

© 2020 form.guide

Sitemap