

PHP File Upload

In this tutorial you'll learn how to upload a file to the remote web server with PHP.

Uploading Files with PHP

In this tutorial we will learn how to upload files on remote server using a Simple HTML form and PHP. You can upload any kind of file like images, videos, ZIP files, Microsoft Office documents, PDFs, as well as executables files and a wide range of other file types.

Step 1: Creating an HTML form to upload the file


The following example will create a simple HTML form that can be used to upload files.

Example	Download
<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <title>File Upload Form</title> </head> <body> <form action="upload-manager.php" method="post" enctype="multipart/form-data"> <h2>Upload File</h2> <label for="fileSelect">Filename:</label> <input type="file" name="photo" id="fileSelect"> <input type="submit" name="submit" value="Upload"> <p>Note: Only .jpg, .jpeg, .gif, .png formats allowed to a max size of 5 MB.</p> </form> </body> </html></pre>	

Note: In addition to a [file-select](#) field the upload form must use the [HTTP post](#) method and must contain an `enctype="multipart/form-data"` attribute. This attribute ensures that the form data is encoded as multipart MIME data — which is required for uploading the large quantities of binary data such as image, audio, video, etc.

Step 2: Processing the uploaded file

Here's the complete code of our "upload-manager.php" file. It will store the uploaded file in a "upload" folder on permanent basis as well as implement some basic security check like file type and file size to ensure that users upload the correct file type and within the allowed limit.

Example	 Download
<pre><?php // Check if the form was submitted if(\$_SERVER["REQUEST_METHOD"] == "POST"){ // Check if file was uploaded without errors if(isset(\$_FILES["photo"]) && \$_FILES["photo"]["error"] == 0){ \$allowed = array("jpg" => "image/jpg", "jpeg" => "image/jpeg", "gif" => "image/gif", "png" => "image/png"); \$filename = \$_FILES["photo"]["name"]; \$filetype = \$_FILES["photo"]["type"]; \$filesize = \$_FILES["photo"]["size"]; // Verify file extension \$ext = pathinfo(\$filename, PATHINFO_EXTENSION); if(!array_key_exists(\$ext, \$allowed)) die("Error: Please</pre>	

Note: The above script prevents uploading a file with the same name as an existing file in the same folder. However, if you want to allow this just prepend the file name with a random string or timestamp, like `$filename = time() . '_' . $_FILES["photo"]["name"];`

You might be wondering what this code was all about. Well, let's go through each part of this example code one by one for a better understanding of this process.

Explanation of Code

Once the form is submitted information about the uploaded file can be accessed via PHP superglobal array called `$_FILES`. For example, our upload form contains a file select field called photo (i.e. `name="photo"`), if any user uploaded a file using this field, we can obtain its details like the name, type, size, temporary name or any error occurred while attempting the upload via the `$_FILES["photo"]` associative array, like this:

- `$_FILES["photo"]["name"]` — This array value specifies the original name of the file, including the file extension. It doesn't include the file path.
- `$_FILES["photo"]["type"]` — This array value specifies the MIME type of the file.
- `$_FILES["photo"]["size"]` — This array value specifies the file size, in bytes.
- `$_FILES["photo"]["tmp_name"]` — This array value specifies the temporary name including full path that is assigned to the file once it has been uploaded to the server.

- `$_FILES["photo"]["error"]` — This array value specifies error or status code associated with the file upload, e.g. it will be 0, if there is no error.

The PHP code in the following example will simply display the details of the uploaded file and stores it in a temporary directory on the web server.

Example	Download
<pre data-bbox="207 380 1349 865"><?php if(\$_FILES["photo"]["error"] > 0){ echo "Error: " . \$_FILES["photo"]["error"] . "
"; } else{ echo "File Name: " . \$_FILES["photo"]["name"] . "
"; echo "File Type: " . \$_FILES["photo"]["type"] . "
"; echo "File Size: " . (\$_FILES["photo"]["size"] / 1024) . " KB
"; echo "Stored in: " . \$_FILES["photo"]["tmp_name"]; } ?></pre>	

Tip: Once a file has been successfully uploaded, it is automatically stored in a temporary directory on the server. To store this file on a permanent basis, you need to move it from the temporary directory to a permanent location using the PHP's `move_uploaded_file()` function.