

## 21. PHP File Upload

With PHP, it is possible to upload files to the server.

### 21.1. Create an Upload-File Form

To allow users to upload files from a form can be very useful.

Look at the following HTML form for uploading files:

```
<html>
<body>

<form action="upload_file.php" method="post"
enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="file" id="file"><br>
<input type="submit" name="submit" value="Submit">
</form>

</body>
</html>
```

Notice the following about the HTML form above:

- The enctype attribute of the <form> tag specifies which content-type to use when submitting the form. "multipart/form-data" is used when a form requires binary data, like the contents of a file, to be uploaded
- The type="file" attribute of the <input> tag specifies that the input should be processed as a file. For example, when viewed in a browser, there will be a browse-button next to the input field

**Note:** Allowing users to upload files is a big security risk. Only permit trusted users to perform file uploads.

### 21.2. Create The Upload Script

The "upload\_file.php" file contains the code for uploading a file:

```
<?php
if ($_FILES["file"]["error"] > 0)
{
```

```
        echo "Error: " . $_FILES["file"]["error"] . "<br>";
    }
    else
    {
        echo "Upload: " . $_FILES["file"]["name"] . "<br>";
        echo "Type: " . $_FILES["file"]["type"] . "<br>";
        echo "Size: " . ($_FILES["file"]["size"] / 1024) . "
kB<br>";
        echo "Stored in: " . $_FILES["file"]["tmp_name"];
    }
    ?>
```

By using the global PHP \$\_FILES array you can upload files from a client computer to the remote server.

The first parameter is the form's input name and the second index can be either "name", "type", "size", "tmp\_name" or "error". Like this:

- \$\_FILES["file"]["name"] - the name of the uploaded file
- \$\_FILES["file"]["type"] - the type of the uploaded file
- \$\_FILES["file"]["size"] - the size in kilobytes of the uploaded file
- \$\_FILES["file"]["tmp\_name"] - the name of the temporary copy of the file stored on the server
- \$\_FILES["file"]["error"] - the error code resulting from the file upload

This is a very simple way of uploading files. For security reasons, you should add restrictions on what the user is allowed to upload.

## 21.3. Restrictions on Upload

In this script we add some restrictions to the file upload. The user may upload .gif, .jpeg, and .png files; and the file size must be under 20 kB:

```
<?php
$allowedExts = array("gif", "jpeg", "jpg", "png");
$extension    = end(explode(".", $_FILES["file"]
["name"]));
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/jpg")
|| ($_FILES["file"]["type"] == "image/pjpeg")
|| ($_FILES["file"]["type"] == "image/x-png")
|| ($_FILES["file"]["type"] == "image/png"))
&& ($_FILES["file"]["size"] < 20000))
```

```
&& in_array($extension, $allowedExts))
{
    if ($_FILES["file"]["error"] > 0)
    {
        echo "Error: " . $_FILES["file"]["error"] . "<br>";
    }
    else
    {
        echo "Upload: " . $_FILES["file"]["name"] . "<br>";
        echo "Type: " . $_FILES["file"]["type"] . "<br>";
        echo "Size: " . ($_FILES["file"]["size"] / 1024) .
        " kB<br>";
        echo "Stored in: " . $_FILES["file"]["tmp_name"];
    }
}
else
{
    echo "Invalid file";
}
?>
```

## 21.4. Saving the Uploaded File

The examples above create a temporary copy of the uploaded files in the PHP temp folder on the server.

The temporary copied files disappears when the script ends. To store the uploaded file we need to copy it to a different location:

```
<?php
$allowedExts = array("gif", "jpeg", "jpg", "png");
$extension = end(explode(".", $_FILES["file"]["name"]));
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/jpg")
|| ($_FILES["file"]["type"] == "image/pjpeg")
|| ($_FILES["file"]["type"] == "image/x-png")
|| ($_FILES["file"]["type"] == "image/png"))
&& ($_FILES["file"]["size"] < 20000)
&& in_array($extension, $allowedExts))
{
    if ($_FILES["file"]["error"] > 0)
    {
        echo "Return Code: " . $_FILES["file"]["error"] .
```

```
"<br>";
    }
    else
    {
        echo "Upload: " . $_FILES["file"]["name"] . "<br>";
        echo "Type: " . $_FILES["file"]["type"] . "<br>";
        echo "Size: " . ($_FILES["file"]["size"] / 1024) . "
kB<br>";
        echo "Temp file: " . $_FILES["file"]["tmp_name"] .
"<br>";

        if (file_exists("upload/" . $_FILES["file"]["name"]))
        {
            echo $_FILES["file"]["name"] . " already exists. ";
        }
        else
        {
            move_uploaded_file($_FILES["file"]["tmp_name"],
"upload/" . $_FILES["file"]["name"]);
            echo "Stored in: " . "upload/" . $_FILES["file"]
["name"];
        }
    }
}
else
{
    echo "Invalid file";
}
?>
```

The script above checks if the file already exists, if it does not, it copies the file to a folder called "upload".