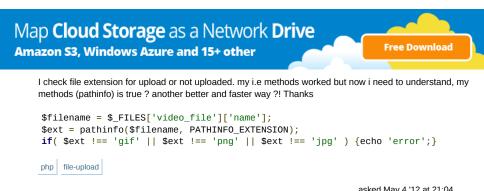
sign up log in tour help careers 2.0

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Take the 2-minute tour

php check file extension in upload form



asked May 4 '12 at 21:04

BBKing

296 4 15 33

4 why is the filed called 'video_file' yet you're only allowing images.. - cream May 6 '13 at 7:53

add a comment

5 Answers

```
Using if( $ext !== 'gif') might not be efficient what if you allow like 20 different extensions

Try

$allowed = array('gif', 'png', 'jpg');
$filename = $_FILES['video_file']['name'];
$ext = pathinfo($filename, PATHINFO_EXTENSION);

if(!in_array($ext,$allowed)) {
    echo 'error';
}

answered May 4'12 at 21:09

Baba
49.4k 12 55 103
```

add a comment



You can check the extension, but that's not the most reliable method. For better way of checking file type you should check it's mime type. Mime Types

From PHP Manual:

```
<?php
$finfo = finfo_open(FILEINFO_MIME_TYPE); // return mime type ala mimetype extens
foreach (glob("*") as $filename) {
    echo finfo_file($finfo, $filename) . "\n";</pre>
```

1 of 3 16/08/2014 11:00

```
finfo_close($finfo);
The above example will output something similar to:
 text/html
image/gif
application/vnd.ms-excel
                                                                         answered May 4 '12 at 21:08
                                                                               Reza Sanaie
                                                                               2,940 13 30
1 I almost doubt there is a way to get the REAL file extension. Everywhere there are answers "to analyze
    filename" which is the worst way... Thanks a lot for this answer! - instead Sep 24 '13 at 17:23
add a comment
Personally,I prefer to use preg_match() function:
if(preg_match("/\.(gif|png|jpg)$/", $filename))
or in_array()
$exts = array('gif', 'png', 'jpg');
if(in_array(end(explode('.', $filename)), $exts)
With in_array() can be useful if you have a lot of extensions to validate and perfomance question.
Another way to validade file images: you can use @imagecreatefrom*(), if the function fails, this mean
the image is not valid.
For example:
 function testimage($path)
     if(!preg_match("/\.(png|jpg|gif)$/",$path,$ext)) return 0;
    $ret = null;
     switch($ext)
         case 'png': $ret = @imagecreatefrompng($path); break;
         case 'jpeg': $ret = @imagecreatefromjpeg($path); break;
         default: $ret = 0;
     return $ret;
}
then:
$valid = testimage('foo.png');
 Assuming \ that \quad foo.png \ \ is \ a \ PHP-script \ file \ with \quad .png \quad extension, \ the \ above \ function \ fails. \ It \ can \ avoid 
attacks like shell update and LFI.
                                            edited May 4 '12 at 21:29
                                                                         answered May 4 '12 at 21:08
                                                                              3,705 4 41 95
add a comment
```

pathinfo is cool but your code can be improved:

```
$filename = $_FILES['video_file']['name'];
$ext = pathinfo($filename, PATHINFO_EXTENSION);
$allowed = array('jpg','png','gif');
if( ! in_array( $ext, $allowed ) ) {echo 'error';}
```

Of course simply checking the extension of the filename would not guarantee the file type as a valid image. You may consider using a function like getimagesize to validate uploaded image files.

answered May 4 '12 at 21:09

2 of 3 16/08/2014 11:00



add a comment

```
Not sure if this would have a faster computational time, but another option...

$acceptedFormats = array('gif', 'png', 'jpg');

if(!in_array(pathinfo($filename, PATHINFO_EXTENSION), $acceptedFormats))) {
    echo 'error';
}

answered May 4'12 at 21:10
```

Not the answer you're looking for? Browse other questions tagged php file-upload or ask your own question.

3 of 3 16/08/2014 11:00