

Tutorial to do upload and sessions

Key components:

\$_SESSIONS

\$_FILES

Lets start with uploading.

When you upload in PHP, you use a file input. Usually you also specify the type of data you are sending. So in most cases it'll look like this.

```
<html>
<body>

<form action="upload.php" method="post"
enctype="multipart/form-data">
<input type="file" name="file" id="file" /> |
<input type="submit" name="submit" value="Submit" />
</form>

</body>
</html>
```

Action = the php file that will handle the upload

Type file will give you an open browser

This is the html non AJAX style.

If you'd like to do AJAX style. I suggest using this <http://www.phpletter.com/Our-Projects/AjaxFileUpload/> plugin. I'll go over how you can use this as well.

After you download it, include it in your header and then do this

```
$(document).ready(function(e) {  
    $.ajaxFileUpload  
    (  
        {  
            url:'upload.php',  
            secureuri:false,  
            fileElementId:'file',  
            dataType: 'json',  
            success: function (data, status)  
            {  
                if(typeof(data.error) != 'undefined'  
                {  
                    if(data.error != '')  
                    {  
                        alert(data.error);  
                    }else  
                    {  
                        alert(data.msg);  
                    }  
                }  
            },  
            error: function (data, status, e)  
            {  
                alert(e);  
            }  
        }  
    )  
});
```

url = the link to your php file

secureuri = you don't need to know this, just keep it at false

fileElementId = the input file's id

dataType= returned data's format. If you have it at json then it'll return an object instead of a string

So basically you can wrap this around a button click like so `$("#button").click(function(e){
$.ajaxFileUpload(...
}`

The client side is all done. That's everything you need to make the upload work. We are not going to go on to the server side.

In your upload.php you'll be using `$_FILES` to get information of the uploaded file. As soon as you submit/ajaxsubmit the file, your php will have stored data in the `$_FILES` just like it'll have stored data in the `$_GET` and `$_POST`

```
<?php
//check for type of file
if ((($_FILES["file"]["type"] == "image/jpeg"))
//check for how big the file is in bits
&& ($_FILES["file"]["size"] < 10000))
{
    //check file error like screwed up uploads
    if ($_FILES["file"]["error"] > 0)
    {
        echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
    }
    else
    {
        //name of the file
        echo "Upload: " . $_FILES["file"]["name"] . "<br />";
        //type of the file
        echo "Type: " . $_FILES["file"]["type"] . "<br />";
        //size of the file in bits, divided by 1024 will make it into kb. And if divided AGAIN by 1024 with get it
to MB
        echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
        //temporary storage name
        echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

        if (file_exists("upload/" . $_FILES["file"]["name"]))
        {
            echo $_FILES["file"]["name"] . " already exists. ";
        }
        else
        {
            //function to move the file to a location, in this case to upload/the name of the file
            move_uploaded_file($_FILES["file"]["tmp_name"], "upload/" . $_FILES["file"]["name"]);
            echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
        }
    }
}
else
{
    echo "Invalid file";
}
?>
```

`$_FILES` will return multiple arrays as commented in the photo

`$_FILES['html_input_name']['type']` tells you what type of file is uploaded

`$_FILES['html_input_name']['size']` tells you how big the files are in bits. To make it in kb you'll have to convert it with the bit to kb ratio `$_FILES['html_input_name']['type']/1024`.

Similarly to convert to MB you divide by the same ration again to go from kb to MB

`$_FILES['html_input_name']['type']/1024/1024`

AND so on...

`$_FILES['html_input_name']['name']` is the name of the file you uploaded

`$_FILES['html_input_name']['tmp_name']` is a temporary hash name that is stored in the server until you move it

`move_uploaded_file($_FILES['html_input_name']['tmp_name'], "link/of/new/location/name.ext")` moves your server stored file (tmp_name) to the location you desire for it to go. You can also change the name of the file if you wish. So for all I care, my uploaded file could be named "popo.jpg". But I can do this

`move_uploaded_file($_FILES['html_input_name']['tmp_name'], "folder/mojojojo.jpg")`. Then my newly uploaded file is automatically renamed to mojojojo.jpg

And that's it! That's all there is for uploading. Now after the upload you know exactly where the link is, so you can call on to your DB file and store the new link! For example I could make a call to upload.php using a get variable. "upload.php?port_id=1". You can dynamically set it to the portfolio you're on. Then do the following below.

```
//function to move the file to a location, in this case to upload, the name of the file
move_uploaded_file($_FILES["file"]["tmp_name"], "upload/newfile.jpg");
db = new port_db();
db->set_portfolio_id($_GET['port_id']);
db->set_img_link("upload/newfile.jpg");
db->update_portfolio();
```

Will update with sessions shortly... STAY TUNED