Upload Fortschritt mit PHP + AJAX

== Voraussetzung ==
Voraussetzung zum Betrieb des Upload Scripts ist ein aktiviertes APC. Wie das funktioniert erfahrt ihr hier: [wiki]APC unter Linux installieren[/wiki].

== Normaler PHP Upload ==
Zuerst benötigen wir ein normales Upload Script. Wir erstellen dazu 2 Dateien. Die Datei index.html dient als Sender. Sie nimmt die Formulareingaben entgegen und bietet den Upload Button.
Sie sendet die Eingabe an die Datei upliad.php, die den eigentlich Uploadvorgang durchführt.

Das komplette Beispiel sieht folgendermaßen aus:

Quellcode

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" xml:lang="de">
3. <head>
4. <title>APC Upload</title>
5. </head>
6. <body>
7. <form action="upload.php" method="post" enctype="multipart/form-data">
8. <div>
9. <input type="file" name="upload" />
10. <input type="submit" />
11. </div>
12. </form>
13. </body>
14. </html>
```

Alles anzeigen

Quellcode

```
1. <?php
2. if(isset($_FILES['upload'])) {
3. echo "File upload successfull";
4. move_uploaded_file($_FILES['upload']['tmp_name'], 'cache/'.$_FILES['upload']['name']);
5. }
6. ?>
```

Die Statusmeldung, dass der Upload komplett war, wird erst nach dem gesamten Datentransfer ausgegeben.
Dies verdeutlicht auch den Nutzen einer Fortschrittsanzeige für den Benutzer, der womöglich mehrere Minuten oder gar Stunden warten muss, ohne dass er ein visuelles Feedback erhält.

== PHP Upload ohne Seite blockieren ==
Sind noch andere Elemente auf der Webseite vorhanden, sind diese während des Uploadprozesses blockiert. Der Sender wurde verlassen oder der Empfänger ist noch nicht fertig, Man befindet sich in einem Status, der nichts ganzes und nichts halbes ausdrückt.

Um ein Blockieren der Seite zu verhindern, führen wir den Uploadprozess in einem separaten Fenster durch. Da mehrere Fenster irritieren nutzen wir ein Fenster im Fenster - einen Iframe.
Diesen Adressieren wir über einen Namen und das Form-Attribut "target". Zusätzlich können wir das Iframe Fenster durch die Style Eigenschaft display:none unsichtbar machen.

Quellcode

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" xml:lang="de">
<head>
<title>APC Upload</title>
</head>
<body>
<iframe src="upload.php" name="hidden_upload"></iframe>
<form action="upload.php" target="hidden_upload" method="post" enctype="multipart/form-data">
<div>
<input type="file" name="upload" />
<input type="submit" />
</div>
</form>
</body>
</html>
```

Alles anzeigen

== Uploadfortschritt mit APC ==
APC Uploads selbst lassen sich über IDs adressieren. Mit Hilfe dieser ID kann man parallel zum Upload in jedem beliebigem Fenster und von jedem beliebigem Browser den Status abfragen.
Wir erzeugen die ID selbst mit Hilfe der Funktion uniqid() - hier kann aber jede beliebige andere Funktion genutzt werden.
Damit der Upload die ID verwendet müssen wir das Formularelement APC_UPLOAD_PROGRESS zusammen mit der Datei übertragen.
Um den Status parallel in einem zweiten Prozess abzufragen müssen wir ihm diese ID mitteilen. Dazu nutzen wir JavaScript.
Um den Status regelmäßig abzufragen starten wir beim Drücken auf den Upload Button einen JavaScript Timer und zeigen in einem regelmäßigen Intervall den Status des Upload an.
Die Methode "ajax" nimmt den Wert von APC_UPLOAD_PROGRESS als Parameter und fragt mit diesem Wert alle 750 Millisekunden den aktuellen Upload Status ab. Gezeigt wird die Rückgabe per AJAX in einem DIV mit der ID status.

Für den Status der Datei benötigen wir eine weitere Datei, die wir status.php nennen.
Sie hat folgenden Inhalt:

Quellcode

```
<?php
$arr = apc_fetch("upload_{$_GET['uid']}");
printf("<pre>%s</pre>", print_r($arr,true));
?>
```

Die Funktion apc_fetch liefert uns ein assoziatives Array aus denen sich das Verhältnis zwischen "bereits hochgeladen" und "Gesamtgröße" berechnen lässt. Die Funktion kann weiter genutzt werden um die Uploadgeschwindigkeit oder viele weitere Statistiken anzuzeigen.

Quellcode

```
Array (
[total] => 1142543
[current] => 1142543
[rate] => 1828068.8
[filename] => test
[name] => file
[temp_filename] => /tmp/php8F
[cancel_upload] => 0
```

```
9. [done] => 1
10. )
```

Die Aktualisierte index.php die nun php benötigt sieht wie folgt aus:

Quellcode

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" xml:lang="de">
3. <head>
4. <title>APC Upload</title>
5. <script type="text/javascript">
6. //<![CDATA[
7. function ajax(uid) {
8. var req;
9. try {
10. req = window.XMLHttpRequest?new XMLHttpRequest():
11. new ActiveXObject("Microsoft.XMLHTTP");
12. } catch (e) {
13. //Kein AJAX Support
14. }
15. req.onreadystatechange = function() {
16. if ((req.readyState == 4) && (req.status == 200)) {
17. document.getElementById("status").innerHTML = req.responseText;
18. }
19. }
20. req.open('GET', 'status.php?uid='+uid);
21. req.send(null);
22. }
23. //]]>
24. </script>
25. </head>
26. <body>
27. <iframe src="upload.php" name="hidden_upload" style="display:none"></iframe>
28. <div id="status"></div>
29. <form action="upload.php" target="hidden_upload" method="post" enctype="multipart/form-data">
30. <div>
31. <input type="hidden" name="APC_UPLOAD_PROGRESS" value="<?php echo uniqid();?>"/>
32. <input type="file" name="upload" />
33. <input type="submit" onclick="this.disabled=true;
    setInterval('ajax(\''+this.form.APC_UPLOAD_PROGRESS.value+'\')', 750); "/>
34. </div>
35. </form>
36. </body>
37. </html>
```

Alles anzeigen

== Upload großer Dateien ==
Der Upload großer Dateien funktioniert nur, wenn der Server dies auch erlaubt. Prüft dazu die folgenden beiden PHP Einstellungen: **upload_max_filesize** und **post_max_size**.
post_max_size integer setzt die maximal erlaubte Größe von POST-Daten. Diese Option betrifft auch den Datei-Upload. Um größere Dateien hochzuladen, muss der Wert größer sein als upload_max_filesize. Wenn eine maximale Speichergrenze während des Kompilierens aktiviert wurde, dann betrifft auch memory_limit den Datei-Upload.

Ihr setzt die Einstellung in der php.ini Datei.

== Demo ==
Den kompletten Download gibt es unter: demo.easy-coding.de/php/ajax-upload-progress/download.zip