**Doku „file\_manager()“**

2014-05-31 DM

Unten stehend findest Du eine Kurz-Doku zur File-Manager-Klasse. Diese kannst Du verwenden, wenn Du mit Files arbeitest.  
Wichtig: Da Files im normalen Betrieb nur in einem vom direkten Apache-Zugriff versteckten Verzeichnis gespeichert werden sollten (in "/usr/local/www/apache22/data-hidden/..."), ist diese Klasse auch nur dafür ausgelegt. Wenn Du ausnahmsweise direkt z.B. auf ein Pluginverzeichnis zugreifen musst, kannst Du diese Klasse nicht verwenden.  
  
Grundsätzlich unterscheiden wir zwischen drei verschiedenen Spaces:

* globalSpace (globale Daten; darauf hat theoretisch jeder Mandant Zugriff [z.Bsp. global gültige Dokumentvorlagen])
* customerSpace (Daten, die nur den Mandant des aktuell eingeloggten User betreffen; andere Mandanten haben keinen Zugriff auf diese Daten)
* userSpace (Daten, die nur den aktuell eingeloggten User betreffen)

Die Klasse kannst Du überall im Programmcode mittels "$fm = new file\_manager();" instanzieren und gleich verwenden (ein Include ist nicht nötig).  
  
Um in einem der oben erwähnten Spaces zu arbeiten, musst Du diesen jeweils spezifizieren:

* $fm->globalSpace() //heisst: Du möchtest im globalen Verzeichnis arbeiten
* $fm->customerSpace() //heisst: Du möchtest im Verzeichnis des aktuellen Kunden (Mandanten) arbeiten
* $fm->userSpace() //heisst: Du möchtest im Verzeichnis des aktuellen Users arbeiten

Entweder Du gehst Zeile für Zeile vor:  
$fm = new file\_manager(); //neues Objekt  
$fm->customerSpace(); //Du möchtest im Verzeichnis des aktuellen Kunden (Mandanten) arbeiten. Mit "customerSpace()" wird quasi das Root-Verzeichnis auf den aktuellen Mandanten eingestellt.  
$fm->setPath("TEST/dest/"); //hier gibst Du an, in welchem Unterverzeichnis vom aktuellen customerSpace Du arbeiten möchtest (immer relativ zum Root-Verzeichnis des aktuellen Mandanten)  
$fm->setFile("blaTest.txt"); //Datei spezifizieren, mit der Du arbeiten möchtest  
$fm->putContents("blabla"); //Der Text "blabla" wird in die Datei ".../AKTUELLER\_KUNDE/TEST/dest/blaTest.txt" geschrieben.  
  
Der obige Code kann auch mit nur zwei Zeilen geschrieben werden:  
$fm = new file\_manager();  
$fm->customerSpace()->setPath("TEST/dest/")->setFile("blaTest.txt")->putContents("blabla");  
  
Wenn Du mal mit setPath und setFile angegeben hast mit welchem Unterverzeichnis und welcher Datei Du arbeiten möchtest, musst Du setPath und setFile nicht immer wiederholen. D.h. zum Auslesen des soeben geschriebenen "blabla"-Text reicht folgende Zeile:  
echo $fm->getContents();  
  
Falls Du das Verzeichnis beibehalten, aber eine andere Datei auslesen möchtest, musst Du nur den Dateinamen mit setFile neu definieren:  
echo $fm->setFile("andereDatei.txt")->getContents();  
  
  
  
Hier ein paar Beispiele:  
  
$fm = new file\_manager();  
$fm->customerSpace()->setPath("TEST/dest/")->makeDir(); //Verzeichnis "TEST/dest/" erstellen  
$fm->setFile("blaTest.txt")->putContents("blabla"); //Datei "blaTest.txt" mit Text "blabla" schreiben  
echo $fm->getContents()."<br>"; //Inhalt auslesen  
echo $fm->size()."<br>"; //Dateigrösse ermitteln  
  
$fm->customerSpace()->setPath("TEST/bla/")->setFile("blaTest.txt")->renameFile("einsundeins.txt"); //Datei "TEST/bla/blaTest.txt" nach "TEST/bla/einsundeins.txt" umbenennen  
  
$fm->customerSpace()->setPath("TEST/bla/")->setFile("blaTest.txt")->deleteFile(); //Datei "TEST/bla/blaTest.txt" löschen  
  
$fm->customerSpace()->setPath("TEST/bla/")->deleteDir(); //Verzeichnis "TEST/bla/" löschen  
  
$fm->customerSpace()->setPath("TEST/bla/")->listDir(); //liefert einen Array mit dem Verzeichnisinhalt (momentan nur Dateinamen; das könnte vielleicht mal mit Dateityp, Grösse, Datum, etc. erweitert werden)  
  
$fm->customerSpace()->setPath("BLA/")->moveTo("TEST/BLA/"); //ganzes Verzeichnis "BLA/" rekursiv(!) nach "TEST/BLA/" verschieben  
$fm->customerSpace()->setPath("TEST/BLA/bla")->setFile("blaTest.txt")->moveTo("TEST/BLA/"); //Datei "blaTest.txt" von "TEST/BLA/bla/" nach "TEST/BLA/" verschieben  
  
$fm->customerSpace("bestehenderKunde")->cloneCustomer("neuerKunde"); //Macht eine Kopie der Daten des Kunden "bestehenderKunde" und erstellt daraus die ganze Verzeichnisstruktur für "neuerKunde"  
  
$fm->userSpace("bestehenderKunde","3")->deleteUser(); //löscht beim Mandant "bestehenderKunde" alle Dateien des Users mit der core\_user\_ID "3". Macht Sinn, wenn der User mit core\_user\_ID=3 auch in der DB gelöscht wird...  
  
$fm->customerSpace()->setPath("TEST/BLA/")->renameDir("GNNN"); //aus Verzeichnis "TEST/BLA/" wird neu "TEST/GNNN/"  
  
$fm->customerSpace()->setPath("TEST/bla/")->setFile("blaTest.txt")->renameFile("brrr.txt"); //Datei "blaTest.txt" wird nach "brrr.txt" umbenennt  
  
$fm->getTemplatePath("payroll","meineDokumentvorlage.xlst"); //Diese Funktion holt das für den aktuell eingeloggten User gültige Template (1.Param=Pluginname/2.Param=Templatedatei). Dabei wird zuerst im userSpace geschaut. Wenn das Template dort nicht existiert, dann wird im customerSpace und zuletzt noch im globalSpace geschaut. Damit kann man bei Bedarf alle im globalSpace hinterlegten Templates übersteuern.  
  
$fm->customerSpace()->setPath("TEST/bla/")->setFile("blaTest.txt")->getFullPath(); //liefert Dir den kompletten, absoluten Pfad zur jeweiligen Datei  
  
$token = $fm->createTmpDir(); //erstellt beim aktuell eingeloggten User ein temporäres Verzeichnis und liefert einen Token zurück. Wenn dieser Token an setTmpDir() übergeben wird, kann diese Funktion damit das TMP-Verzeichnis wieder ausfindig machen und setzt Objekt-intern den korrekten Pfad [setPath ist hier also nicht nötig].  
if( $fm->setTmpDir($token) ) {  
    $path = $fm->getFullPath();  
    // ....  
    $fm->deleteDir();  
}  
  
  
Wenn obige Funktionen nicht reichen und Du auf einem "lower level" mit den Dateien arbeiten musst, dann kannst Du die fopen und fclose Befehle verwenden.  
  
$handle = $fm->customerSpace()->setPath("TEST/bla/")->setFile("blaTest.txt")->fopen("w"); //"w" ist der "write"-Modus. Hier kannst Du alle Standard-Modes vom PHP-fopen verwenden  
fwrite($handle, '123456'); //normale fwrite-Funktion von PHP  
$fm->fclose(); //zum Schliessen des Files sollte wieder die fclose-Funktion vom "file\_manager()" verwendet werden.  
  
Gruss  
Daniel