**Doku BLOB**

2014-05-28

Technisch ist bereits alles für eine Speicherung von Dateien nach Mandant und Benutzer vorbereitet. Im Prinzip werden sämtliche PDFs, die Sie heute schon im System abrufen können, auf der Server-Festplatte pro Benutzer gespeichert. Allerdings nur temporär, bis der Browser die Daten empfangen hat. Danach werden die PDFs auf dem Server gelöscht. Es wäre aber kein Problem, den bestehenden Mechanismus so zu beeinflussen, dass die PDFs permanent (anstatt nur temporär) gespeichert werden.  
  
Ich habe eigens dafür eine Programm-Bibliothek mit Funktionen für das Erstellen, Löschen und Kopieren/Verschieben von Dateien entwickelt. Auch Verzeichnisse können damit ausgelesen werden. Eine Beschreibung dieser Programm-Bibliothek für die Verwaltung von Dateien und Verzeichnissen werde ich in den nächsten Tagen an Harald Müller senden.  
  
Es gäbe in MySQL tatsächlich die Möglichkeit ganze Dateien (Bilder, Dokumente, etc.) in der Datenbank abzulegen. Wie überall gibt es auch hier Vor- und Nachteile. Wobei aus meiner Sicht die Nachteile überwiegen. Das Speicherformat von Dateien in Datenbanken nennt sich übrigens "BLOB" (Abk. f. "Binary Large Object"). Hier zwei Links zum Thema:  
<http://www.patrick-gotthard.de/bilder-in-mysql-datenbank-speichern>  
[http://amor.cms.hu-berlin.de/~kunert/blog/2012/09/03/du-sollst-nicht-blobben/](http://amor.cms.hu-berlin.de/%7Ekunert/blog/2012/09/03/du-sollst-nicht-blobben/)  
  
Vorteile:

* Sämtliche Daten und Dateien eines Kunden sind in einem einzigen "Datengefäss" untergebracht
* In einem einzigen Backup sind sowohl die DB-Daten als auch alle Dateien enthalten

Nachteile:

* Datenbanken sind eigentlich nicht fürs Speichern von Dateien optimiert
* Die sonst schon grossen Datenbank-Dumps (DB-Backups) werden noch viel grösser
* Eine Datenbank kennt intern keine Verzeichnisstruktur. Diese müsste mittels eigener Programmlogik imitiert werden.
* Trotz DB-interner Speicherung von Dateien müssten einige Dateien im File-System abgelegt werden (temporäre Dateien, globale/mandantenübergreifende Dateien), was zu einer unschönen Vermischung von Dateien in der DB und Dateien im File-System führt
* Der Dateizugriff (lesen) via Datenbank ist langsamer als "normal" via File-System
* Das Speichern von Dateien ist im Falle von DB-Speicherung ebenfalls langsamer, da die Datei ohnehin zuerst im File-System generiert werden muss, um dann noch in einem zusätzlichen Schritt in die DB importiert zu werden, wo sie im Prinzip DB-intern ein zweites Mal als Datei gespeichert wird.

Ich hoffe, dieser Input ist hilfreich für die Entscheidungsfindung, ob künftig "BLOBs" eingesetzt werden oder ob Dateien weiterhin im File-System verwaltet werden.  
  
Beste Grüsse  
Daniel Müller