



Konzept

Werner Hahn

Kivitendo - Dateiuploader
Version 2





Inhaltsverzeichnis

a. Versionen	2
b. Allgemeines	3
b.1 Allgemeine Beschreibung	3
c. Spezifikation	4
c.1 Abschnitt A01: Datenbank - Modell	4
c.2 Abschnitt A02: Backend - Controller	4
c.3 Abschnitt A03: Frontend - View	5
c.4 Abschnitt A04: Tests	5
c.5 Abschnitt A05: TODO	5

a. Versionen

Version	Datum	Beschreibung
1	19.01.2016 15:02	Initiale Version.
2	19.01.2016 15:06	Konzept statt Pflichtenheft

Konzept: "Kivitendo - Dateiuploader"
Version 2 vom 19.01.2016 15:06

Seite 2



b. Allgemeines

b.1 Allgemeine Beschreibung

Der Dateiaploader soll hochzuladene Dateien in Webdavordnern abspeichern.

Ideensammlung wofür:

- Kunden/Lieferanten: Korrespondenz, Verträge usw.
- Artikel: Beschreibungen, Datenblätter, Bilder, Shopbilder, Gebrauchsanweisungen, Etiketten usw.
- Belege: Einkaufrechnungen und Lieferscheine usw.
Mitzusendende Broschüren, Handbücher usw.
- Pflichtenheft: individuelles Titelbild, Anhänge
- Briefe: Anhänge

Optional soll eingegeben werden können, ob zu der Datei ein Datenbankeintrag geschehen soll. Das ist z.B. notwendig für die Shopimages und deren Sortierung.



c. Spezifikation

c.1 Abschnitt A01: Datenbank - Modell

>>> Funktionsblock FB001
Beschreibung Tabelle **files**

- **id** #int PK autointeger bzw postgresfunktion, die die id vergibt
- **modul** #text Tabellennamen des Moduls z.B. parts, customer ...
- **trans_id** #int FK Fremdschlüssel auf die id der Tabelle aus der Spalte modul
- **filename** #text Dateiname
- **location** #text Speicherort der Datei im Dateisystem Webdav
- **description** #text Zusätzliche Bildbeschreibung
- **sort_order** #int Sortierung bei mehreren Bildern unique zusammen mit trans_id
- **itime** #timestamp without timezone now() Erfassungszeit
- **mtime** #timestamp without timezone now() ??

TODO:

1. DB-Upgradescript schreiben.
2. DB-Upgradescript ausführen lassen, damit die Tabelle existiert.
3. In SL::DB::Helper::Mappings deine Tabelle in das große Mapping-Hash eintragen.
4. rose_auto... mit Tabellennamen ausführen.
5. In SL::DB::Helper::ALL deine neue Objekt-Klasse einfügen.

Gewichtung: nicht bewertet

c.2 Abschnitt A02: Backend - Controller

>>> Funktionsblock FB004
Beschreibung **SL::CONTROLLER::HELPER::Fileuploader** schreiben der auf die Funktionen aus SL::Webdav zugreift

Gewichtung: nicht bewertet

>>> Funktionsblock FB005
Beschreibung Actions in Controllern z.B. bei Parts den Bildupload für die Shopbilder inkl. Beschreibung und Sortierung für den Shopexport

Gewichtung: nicht bewertet

Konzept: "Kivitendo - Dateiploader"
Version 2 vom 19.01.2016 15:06

Seite 4



c.3 Abschnitt A03: Frontend - View

>>>	Funktionsblock	FB003
	Beschreibung	Sind die Bildinformationen in der Datenbank kann die Sort_order mit L.sortable_element umsortiert werden
	Gewichtung:	nicht bewertet

>>>	Funktionsblock	FB002
	Beschreibung	in ?? SL/Template/Plugin/L.pm Funktion fileupload_tag(DATEIPFAD,ACTION) schreiben, die die notwendigen Felder zurückliefert. ungefähr so: <pre><div id="fileupload"> <input type="hidden" value="DATEIPFAD" name="fupload_dateipfad"> <input type="hidden" value="ACTION" name="fupload_action"> <input type="file" name="fupload_filename"> <input type="textarea" name="fupload_description"> <input type="submit" name="fupload_submit"> </div></pre> <p>Sind (DATEIPFAD,ACTION) nicht definiert wird der Standardpfad "webdav/MANDANTEN_ID/MODUL /MODUL_NUMBER/fupload.filename und ACTION SL::Controller::Helper::Fileuploader verwendet. Das Textareafeld ist ausgeblendet, da nichts in die Datenbank geschrieben werden soll.</p> <p>Ist nur (DATEIPFAD) definiert wird der Standardpfad überschrieben und ACTION SL::Controller::Helper::Fileuploader verwendet. Das Textareafeld ist ausgeblendet, da nichts in die Datenbank geschrieben werden soll.</p> <p>Sind (DATEIPFAD,ACTION) definiert kann die Action alles Mögliche machen (Datenbankeintrag). In dieser Action kann dann auch auf SL::Controller::Helper::Fileuploader zugegriffen werden</p> <p>Die Funktion kann dann in den Templates aufgerufen werden. L.fileupload_tag(Dateipfad, Action).</p> <p>??? Ob ich mit dem Submitbutton direkt auf SL::Controller::Helper::Fileuploader bzw. die definierte Aktion weiss ich noch nicht. ???</p>
	Gewichtung:	nicht bewertet

c.4 Abschnitt A04: Tests

Mit Tests hab ich absolut noch keine Erfahrungen gesammelt

c.5 Abschnitt A05: TODO

- Falls Datei in der Datenbank steht, diese aber direkt gelöscht wird, bleibt der Eintrag bestehen.