

QFQ - Feature #9600

Report als File

20.11.2019 10:14 - Carsten Rose

Status:	Closed	Start date:	20.11.2019
Priority:	Normal	Due date:	
Assignee:	Marc Egger	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Discuss:			
Description			
Nach Diskussion mit BB, ME soll QFQ umgebaut werden, so das die T3 QFQ tt-content Records den Inhalt aus dem Filesystem laden koennen.			
<ul style="list-style-type: none">• Damit sollte (fuer bestehende tt-content records) ein GIT commit/pull (mergen) durch mehrere Entwickler moeglich werden.• Im QFQ tt-content Record wird ein Keyword 'filebased' angegeben. Damit weiss QFQ das es filebased arbeiten soll und dann den Inhalt im Filesystem.			
Es sind noch etliche Detailfragen zu loesen - insbesondere beim mergen. Evtl. koennen extended attributes helfen um den Dateinamen sprechend zu machen.			
<ul style="list-style-type: none">• Moegliche Directory Struktur<ul style="list-style-type: none">◦ Page als Verzeichnis: id-<title>◦ Content als Datei: id-[column]-[language id]-<title> (column: Inrb)• Ist der Inhalt im tt-content record neuer als die Datei, wird die Datei aktualisiert. Timestamps werden aktualisiert.• Ist der Inhalt in der Datei neuer als der tt-content record, wird der Record aktualisiert. Timestamps werden aktualisiert.• Somit kann auf Filesystemebene und im Browser gearbeitet werden.			
Related issues:			
Related to QFQ - Feature #9602: Form definition as JSON		New	20.11.2019

Associated revisions

Revision 39fa2e75 - 27.07.2020 12:12 - Marc Egger

Refs #9600 add todos, parse file keyword

Revision 11fa9e42 - 27.07.2020 18:42 - Marc Egger

Refs #9600 report export and file read works.

Revision 29ac8bc8 - 29.07.2020 09:30 - Marc Egger

Refs #9600 todos

Revision 00d02c9d - 29.07.2020 13:09 - Marc Egger

Refs #9600 Edit report file using frontend editor.

Revision 1eac8e30 - 18.08.2020 10:26 - Marc Egger

Refs #9600 Edit report file using frontend editor.

Revision b4537ff1 - 18.08.2020 11:22 - Marc Egger

Refs #9600 Todos

History

#1 - 20.11.2019 10:15 - Carsten Rose

- Related to Feature #3402: Syntax Highlighting via CodeMirror added

#2 - 20.11.2019 10:22 - Carsten Rose

- Related to Feature #9602: Form definition as JSON added

#3 - 20.11.2019 10:22 - Carsten Rose

- Related to deleted (Feature #3402: Syntax Highlighting via CodeMirror)

#4 - 11.12.2019 16:01 - Carsten Rose

- Status changed from New to Some day maybe

#5 - 11.12.2019 16:17 - Carsten Rose

- Status changed from Some day maybe to New

#6 - 01.02.2020 15:55 - Carsten Rose

- Tracker changed from Support to Feature

#7 - 01.02.2020 23:20 - Carsten Rose

- Target version changed from 146 to next

#8 - 01.02.2020 23:22 - Carsten Rose

- Target version changed from next to next2

#9 - 04.06.2020 19:06 - Marc Egger

- Status changed from New to Priorize

- Assignee changed from Carsten Rose to Marc Egger

#10 - 04.06.2020 19:07 - Marc Egger

- Subject changed from T3 tt-content record (Report) fuer QFQ: File based to Report als File

#11 - 23.07.2020 12:32 - Marc Egger

Ich bin mir nicht ganz sicher ob die oben genannte Lösung die beste ist.

Nach der Implementierung von Form As File, habe ich erlebt, wie kompliziert es sein kann, eine "simple" Synchronisation zwischen Files und Datenbank zu implementieren. Befolgen wir die obige Lösung, müssen wir nicht nur Files sondern auch die Ordnerstruktur synchronisieren. Natürlich wird es dadurch einfacher, dass wir die Ids mit synchronisieren, dies bindet die Files aber wiederum sehr stark an die Datenbank. Wie interpretieren wir drag-and-drop und copy-paste der files, wenn diese von der Datenbank id abhängig sind?

Ich würde folgende Alternative vorschlagen:

- Zusätzlich zu dem bisherigen qfq tt-content record, führen wir einen qfq-file tt-content record ein.
- Mittels Dropdown oder Textfeld kann man ein report file diesem tt-content record zuweisen.
- Wenn das qfq-file tt-content record gerendert wird, wird das referenzierte File geladen und gerendert.
- Ein file kann in mehreren tt-content records referenziert werden.

Vorteile:

- Keine Synchronisation !!!
- Files sind unabhängig von tt-content ids und page ids.
- Files koennen problemlos zwischen verschiedenen Typo3 Instanzen hin und her kopiert werden.
- File Struktur darf, muss aber nicht die Page Struktur widerspiegeln.
- Ich schätze den Aufwand kleiner ein.
- Die Lösung ist unabhängig von Typo3, Einsatz von QFQ in anderen CMS/Frameworks wird ermöglicht.
- Diese Lösung erscheint mir stabiler gegenüber Typo3 updates.
- Update auf die neue QFQ Version (mit Report as File) ändert nichts an bestehender QFQ Applikation. Kann also nichts kaputt machen.

Nachteile:

- Beim erstellen des tt-content records muss ein File ausgewählt (oder erstellt) werden.
- Die Filestruktur spiegelt nicht automatisch die page Struktur.
- Der Report kann nicht innerhalb von Typo3 bearbeitet werden.
 - Wir könnten allerdings einen eigenen web-editor in qfq bauen und diesen im Typo3 Interface verlinken.
- Es wird keine automatische History der Reports mehr erstellt. Versioning muss komplett über Git stattfinden.
- Die T3 Datenbank muss nachwievor synchronisiert werden. (Dies ist aber auch in der vorgesehenen Lösung der Fall und ist kaum zu verhindern)

#12 - 23.07.2020 12:43 - Benjamin Baer

~~ich wuerde die id ignorieren und automatisch eine speaking url aus dem ordnernamen generieren fuer die page und beim tt content record einfach title = dateiname.~~

#13 - 23.07.2020 16:14 - Marc Egger

Falls wir meine Variante machen, haette Benj gerne ein Feature, mit dem man einmalig alle bestehenden tt-content records in report files exportieren und umwandeln kann.

hmm aber ja, kannst es auch so machen wie du gesagt hast
vermutlich besser, dann haben wir nicht viel mit der typo3 datenbank am hut
solange du ein ~sync~ export script macht der beim ersten mal alles schoen in ordner ablegt die nach pages benannt sind <3

- Benj

#14 - 27.07.2020 10:13 - Marc Egger

CR und ME entschieden:

- Es gibt **kein neues tt-content element**.
- Stattessen führen wir ein **neues keyword** ein: File=<path>
 - Ist das keyword gesetzt, wird der report aus angegebenem file geladen.
 - Ist es nicht gesetzt, wird der **tt-content pfad im filesystem einmalig abgebildet** und der inhalt tt-content bodytext in ein file in diesem pfad kopiert. Der **bodytext des tt-content element wird ersetzt mit File=<path>**.
- Den inhalt der report files kann man nicht im T3 backend direkt bearbeiten.
- Im **Frontend wird weiterhin ein edit symbol angezeigt**, wenn man im T3 backend eingeloggt ist. Damit kann man den inhalt des report files bearbeiten.

#15 - 29.07.2020 13:15 - Marc Egger

- Status changed from Priorize to In Progress

#16 - 29.07.2020 13:16 - Marc Egger

Implemented first working version with frontend report editor:

Branch: F9600ReportAlsFile

Unittests fail. Have to be corrected.

#17 - 18.08.2020 11:27 - Marc Egger

Fertig implementiert.

TODO:

- Mit Carsten die Notes in FormAsFile.php besprechen
- Selenium tests schreiben

#18 - 18.08.2020 11:27 - Marc Egger

- Status changed from In Progress to Priorize

#19 - 16.03.2021 17:58 - Marc Egger

- Status changed from Priorize to Closed

Report As File wird in den Selenium Tests jetzt abgedeckt durch den mitgelieferten Form Report.

#20 - 17.05.2021 22:33 - Carsten Rose

- Target version deleted (next2)