

## QFQ - Feature #11998

### Custom QFQ-Funktion

11.02.2021 17:25 - Elias Villiger

<b>Status:</b>	Closed	<b>Start date:</b>	11.02.2021
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Carsten Rose	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	21.3.1	<b>Spent time:</b>	0.00 hour
<b>Discuss:</b>			

#### Description

Einige Ids werden auf verschiedensten Seiten jedesmal neu berechnet. Das bedeutet 1) Code duplication und 2) ev. Performance-Einbusse

Beispiel: die person.id des eingeloggten Users:

```
SELECT pId FROM account WHERE accountType='uzh shortname' AND username='{{feUser:UTE}}'
```

oder die geoleanAccess property ids für die aktuelle feUserGroups:

```
SELECT GROUP_CONCAT(id) FROM gGroup WHERE name='geoleanAccess' AND gType='property' AND t1 IN('{{feUserGroup:UT0}}')
```

Wichtig: Zugriff auf den Typo3-Store.

Die Variablen für meine Anwendungen müssten nur selten aktualisiert werden. Einmal pro Page Load oder allenfalls pro Session würde reichen.

#### Mögliche Lösungen:

- Bereits jetzt möglich: Die gewünschten Variablen über QFQ im User-Store definieren, z.B. auf der Seite nach dem Login, oder über Einbinden eines QFQ-Records auf allen Seiten mittels template:

```
// Einbinden des Records mit uid:96 auf allen Seiten
page.200 = RECORDS
page.200 {
  tables = tt_content
  source = 96
}
```

- Neue Option, Custom QFQ-Code zu definieren. In Konfig, dynamisch gerendert? In tt\_content Record?

#### Related issues:

Related to QFQ - Feature #11569: Custom QFQ-Code pro Session	<b>Closed</b>	<b>21.11.2020</b>
Has duplicate QFQ - Feature #7883: Link to QFQ record: '... AS _qfq'	<b>Rejected</b>	<b>14.02.2019</b>
Has duplicate QFQ - Feature #10713: QFQ functions (reusable reports)	<b>Rejected</b>	<b>04.06.2020</b>

#### Associated revisions

##### Revision 50c06e12 - 19.03.2021 00:42 - Carsten Rose

WIP Refs #11998 - first implementation - throws an exception

##### Revision 1e49ae1a - 19.03.2021 10:58 - Carsten Rose

Refs #11998 - first running implementation.

##### Revision c1f6172f - 19.03.2021 13:37 - Carsten Rose

Refs #11998 - function() is now fired even if there is no 'sql = ...'. NEW: '10.sql=' is allowed and 'thead' and 'althead' are processed even if 10.sql is

an empty string.

#### Revision 3cb504de - 20.03.2021 18:22 - Carsten Rose

Refs #11998 - Change funtion notation from ':' to '=>'. Make special column definition available via '{{&...:R}}'

#### Revision cc7e6397 - 20.03.2021 18:23 - Carsten Rose

Refs #11998 - Add documentation.

#### Revision da4212db - 20.03.2021 18:55 - Carsten Rose

Refs #11998 - Add qfq function cache

#### Revision ceb35066 - 20.03.2021 19:10 - Carsten Rose

Refs #11998 - Add explanation to qfq function example code.

#### Revision 1c09b4e2 - 20.03.2021 21:02 - Carsten Rose

Refs #11998 - minor doc add

#### Revision dc4271bf - 20.03.2021 22:25 - Carsten Rose

Refs #11998 - By theory, getBodyText() should now be MultiDB aware.

#### Revision b83706b1 - 20.03.2021 22:35 - Carsten Rose

Refs #11998 - minor doc add

#### Revision dfdd9a07 - 21.03.2021 10:47 - Carsten Rose

Refs #11998 - more specific error message if function not found

#### Revision ebae6839 - 21.03.2021 10:56 - Carsten Rose

Refs #11998 - remove urlParam from getBodyText ... not necessary here

#### Revision 92963fb7 - 21.03.2021 15:20 - Carsten Rose

Refs #11998 - implement central download logic

## History

---

### #1 - 11.02.2021 17:25 - Elias Villiger

- Related to Feature #11569: Custom QFQ-Code pro Session added

### #2 - 11.02.2021 17:46 - Carsten Rose

Vorschlag

- Idee: QFQ Records koennen an beliebiger Stelle in REPORT eingebunden werden. Bsp.

```
10 {  
    sql = SELECT ...  
    20.sql = SELECT ...  
    30.uid = 123|accountId,feUserId  
    40.uid = 124|*  
}
```

```
# Alternative um die `uid` lesbarer zu machen: Die Spalte tt_content.sub_header wird im T3 Backend  
# bei QFQ Records editierbar gemacht und es kann ein 'Funktionsname' gesetzt werden:  
# Sollten mehrere tt_content records den gleichen Funktionsnamen haben, zeigt QFQ einen Fehler an.
```

```
50.uid = userLogin|feUserId,accountId
```

```
60.uid = getPeriod|*|skip cache
```

- Die Variablennamen hinter dem uid/Funktionsname gibt an, was durch die Funktion via STORE\_RECORD exportiert wird. Damit ist klar welche Parameter zurueck kommen.
- Diskussion: sollten auch die aufrufenden Parameter deklariert werden?
- Der \*\*: alles was in der Funktion in den STORE\_RECORD kopiert.
- STORE\_USER ist immer schreib/lesbar.
- cache = 30 - Der STORE\_RECORD der Funktion wird gecacht und der tt\_content Record maximal alle 30 Sekunden gefeuert. Ein erneuter

Aufruf innerhalb der Cache time liefert den Cache.

- cacheUser = 5 - wie Cache, aber pro User Session, 5 Sekunden.
- 'skip cache' - Die Query wird immer ausgeführt, unabhängig ob eine Cache Time definiert ist.
  
- Beispiel: tt\_content.uid = 123

```
cache = 30
10.sql = SELECT 1 AS accountId, 2 AS feUserId
```

### #3 - 13.02.2021 14:02 - Elias Villiger

Ist insgesamt viel mehr als ich eigentlich brauche, aber warum nicht?

Einige Kommentare/Fragen:

- Ich würde auch aufrufende Parameter ermöglichen. In meinen Beispielen bräuchte ich das zwar nicht, kann ich mir aber gut vorstellen, dass das gebraucht würde. Z.B. mit ähnlicher Schreibweise wie bei den Excel Exports: 10.uid = 123&isProf=1|feUserId,accountId
- Falls QFQ-Record-Funktionsnamen umgesetzt werden - gerne auch für die Excel-Exporte verwenden
- Cache: falls auch für andere QFQ-Records verwendet, müsste man natürlich auch die Ausgabe des Records cachen, nicht nur STORE\_RECORD.
  - Den Cache braucht's aus meiner Sicht nicht zwingend. Wenn eine Seite geladen wird, möchte ich eigentlich in den meisten Fällen sicher sein, dass die Daten aktuell sind, nicht 30s alt. Die Variablen würden so oder so nur einmal pro Page Load berechnet und stünden dann beliebig oft zur Verfügung im nachfolgenden Code.
- uid: mögliche andere Namen: 10.load oder 10.import

### #5 - 19.02.2021 15:09 - Philipp Gröbelbauer

Check ob man #8450 damit lösen könnte?!

### #6 - 18.03.2021 21:51 - Carsten Rose

- Subject changed from Custom QFQ-Code zentral definiert to Custom QFQ-Funktion

### #7 - 18.03.2021 22:11 - Carsten Rose

Naming:

```
10.function = getFeUser(pId, pName) => accountId, feUserId
```

- function ist klarer als uid
- Beispiel: getFeUser ist der subheader Name des aufzurufenden tt-content records. Wird eine reine Zahl angegeben, ist die uid gemeint. Bsp 10.function = 123.
- Sind neben .function noch weitere Token wie sql, head, ... angegeben: function wird als erstes ausgeführt, danach wie gewohnt alles andere.
- Ist nur function definiert, aber kein sql werden keine Sublevel gefeuert.
- Die runden Klammern mit (pId, pName) entsprechen der bekannten Notation eines Funktionsaufrufes.
- Die Returnwerte werden durch '=>' angezeigt.
- Die Funktion sollte so implementiert werden das aus dem STORE\_RECORD nur die definierten Variablen verwendet werden können.
- Die Aufrufparameter und die Returnparameter werden im STORE\_RECORD uebergeben.
- Der STORE\_RECORD wird nicht veraendert (ausser den definierten Returnwerten).
- Der Output der in der Funktion erzeugt wird (optional) kann abgerufen werden via {{\_output:R}}
- Ein Cache wird erstmal nicht implementiert.
- Optional könnte man auch ueber die Verwendung des STORE\_PARENT nachdenken.

Frage:

- Können in einer Funktion weitere Functions aufgerufen werden?

### #8 - 19.03.2021 08:57 - Elias Villiger

Das tönt echt gut.

Hier noch ein anderer Notationsvorschlag:

```
10.function = getFeUser(pId, pName) => accountId, feUserId
```

Wichtig wäre mir, dass es auch Funktionen ohne Return-Wert geben kann. Gebrauch z.B.:

- Funktion speichert etwas in der DB, ohne Return value
- Funktion speichert etwas oder mehrere Werte auf einmal im U-Store (Bsp. feUserId, acctId, isDev, isProf, isSecretary, ...), diese sind dann sozusagen der globale Return value, aber ohne eigentlichen Return value der Funktion im R-Store

Koennen in einer Function weitere Functions aufgerufen werden?

- Wäre schon brauchbar.
- Wenn kompliziert, dann "vorerst" darauf verzichten?
- Das Aufrufen von einzelnen Funktionen ist an sich schon ein riesiger Sprung vorwärts.

#### **#9 - 20.03.2021 20:32 - Carsten Rose**

Aktuell haben wir bei diversen Tools, bei denen PDFs & Attachments zu einem PDF zusammengefuehrt werden und diese Full-PDF Links auf diversen Seiten benoetigt werden, eine Zwischenseite eingebaut. Damit war die PDF Definition nur an einer Stelle, und man konnte im gesamten Tool Links auf diese Zwischenseite bauen.

Der Nachteil: die User mussten immer zweimal klicken um an ein PDF zu gelangen.

Neu soll als weitere Download Source (neben 'F,p,u,uid') auch source:<qfq function>&arg1=value1\$arg2=... moeglich sein.

In <qfq function> werden dann die ueblichen Sourcen ausgegeben in der Form 'F:file1.pdf|p:myBaseData&appld=123|F:file2.pdf'. Dabei ist es egal aus wieviel Spalten/Rows, es wird alles implodiert und muss ein (multiple) Sources Argument ergeben.

#### **#10 - 21.03.2021 18:05 - Carsten Rose**

- Status changed from New to Closed

#### **#11 - 21.03.2021 18:09 - Carsten Rose**

- Related to Feature #7883: Link to QFQ record: '... AS\_qfq' added

#### **#13 - 21.03.2021 18:11 - Carsten Rose**

- Related to deleted (Feature #7883: Link to QFQ record: '... AS\_qfq')

#### **#14 - 21.03.2021 18:11 - Carsten Rose**

- Has duplicate Feature #7883: Link to QFQ record: '... AS\_qfq' added

#### **#15 - 21.03.2021 20:30 - Carsten Rose**

- Target version changed from 21.8.0 to 21.3.1

#### **#16 - 07.09.2021 13:45 - Marc Egger**

- Has duplicate Feature #10713: QFQ functions (reusable reports) added