

QFQ - Feature #10979

Ajax Calls an API - dataReport

07.08.2020 12:25 - Philipp Gröbelbauer

Status: Closed	Start date: 07.08.2020
Priority: Normal	Due date:
Assignee: Marc Egger	% Done: 0%
Category:	Estimated time: 0.00 hour
Target version: 20.11.0	Spent time: 0.00 hour
Discuss:	
Description - QFQ Page Content per AJAX ausführen - Über API, sodass kein Performance Verlust durch Typo3 - return Resultat des QFQ Page Contents (HTML Snippet zum Testen hängt an diesem Ticket)	

Associated revisions

Revision 7d60d5d1 - 06.09.2020 11:11 - Carsten Rose

Refs #10979 Not finally implemented, but should be documented not to forget it and it's already used in LK.

History

#1 - 09.08.2020 15:30 - Carsten Rose

- Subject changed from Ajax Calls an API to Ajax Calls an API - dataReport

- Status changed from New to In Progress

#2 - 09.08.2020 16:23 - Carsten Rose

- In einer ersten Version wird die reguläre Link Klasse genutzt.
- Es wird kein HTML/ Javascript generiert, das muss explizit gemacht werden

```
SELECT '<a href="typo3conf/ext/qfq/Classes/Api/dataReport.php?s='  
, 'U:uid=159&arg1=Hello&arg2=World|s|r:8' AS _link  
, '">Call 159 (this record)</a>'
```

- Das '... AS _link' liefert `badcaffee1234`.

Als Return von dem Aufruf kommt 1:1 der Code den der Report ausgibt (kein JSON).

#3 - 09.08.2020 17:20 - Carsten Rose

- File dataReport2.png added

- File dataReport.png added



Quick Form Query

[Call 159 \(this record\)](#)

[Call 160 \(api record\)](#)

Home

Report

Download

Action

Output

Chart

altsql, shead, stail

Thumbnail

Spaces

FSkipWrap

Escape Class

Variable not found

dataReport

QFQ JQW

- First Page
- Home
- Report
 - Download
 - Session
 - Letter
 - export
 - header
 - body
 - Excel
 - ExcelData
 - Action
 - Output
 - Chart
 - altsql, shead, stail
 - Thumbnail
 - Spaces
 - FSkipWrap
 - Escape Class
 - Variable not found
 - dataReport
- Notation
 - StoreRecord
 - StoreUser
 - Column: hide/reuse
 - Column: mimetype

Edit Page Content "[QFQ] dataReport" on page "dataReport"

General Access Appearance Categories

Header: [QFQ] dataReport **Type:** QFQ Elem

Code:

```

1 #
2 # uid:159
3 #
4
5 10 {
6   sql = SELECT '<a href="typo3conf/ext/qfq/Classes/Api/dataReport.php?s='
7   , 'U:uid=159&arg1=Hello&arg2=World|s|r:8' AS _link
8   , '>Call 159 (this record)</a>', '<br><br>'
9
10  , '<a href="typo3conf/ext/qfq/Classes/Api/dataReport.php?s='
11  , 'U:uid=160&arg1=John&arg2=Doe|s|r:8' AS _link
12  , '>Call 160 (api record)</a>'
13
14   FROM (SELECT '' AS fake
15   WHERE {{uid:S0}}=0
16  }
17 }
18
19 20 {
20  sql = SELECT '{{arg1:S}} {{arg2:S}} ', NOW()
21   FROM (SELECT '' AS fake
22   WHERE {{uid:S0}}!=0
23  } |

```

#4 - 10.08.2020 22:12 - Carsten Rose

Diskussion PG, CR:

- Vorschlag von CR, mit 'JS Templates' zu arbeiten, scheint keine nennenswerte Vorteile zu bringen.
- Vorschlag von PG, ein '... AS _link' der in den link Parametern Callback fuer 'success', 'error' vorgibt, scheint keine nennenswerte Vorteile gegenueber der nativen JS Implementation zu bringen.
- Auf folgendes geeinigt:

```
20 {
  sql = SELECT 'J:uid=100&xId=3|s' AS '_link|coll' # Output: 'typo3conf/ext/qfq/Classes/Api/dataReport.php?s=000000000'
}

30.head = <script>
function writeYourOwnAjax(){
$.ajax({
url: '{{coll:RE}}',
data: {a:30, b:430, c:0},
method: 'POST',
dataType: 'JSON',
success: function(response) {ajaxSuccess(response);},
error: function(jqXHR, textStatus, errorThrown) {ajaxError(jqXHR, textStatus, errorThrown);}
});
}
</script>
```

- Neuer Hauptqualifier 'J' fuer Ajax.
- `typo3conf/ext/qfq/Classes/Api/dataReport.php` wird hart reinkodiert.
- Parameter koennen 'plain' oder 'SIP kodiert' uebergeben werden.
- `r:8` ist nicht mehr noetig.
- Der Output der QFQ-Records wird ausgegeben.
- Es gibt zwei neue Variablen (ohne Zeilennummer, werden ersetzt nachdem Report durchgelaufen ist):

```
statusCode = {{statusCode:R}} //200 404
statusMessage = missing value
```

- Enthaelt der statusCode einen Wert !=200, wird JS den 'error' Callback aufrufen.
- statusMessage kann mehr Details enthalten.
- Sollte es OS / PHP Probleme geben, sind diese Werte sowieso gestzt.
- Neu kann zusaetzlich der User Code die Werte beeinflussen. Ein StatusCode !=200 koennte (wenn implementiert) eine Standard QFQ Error Box anzeigen.

#5 - 18.08.2020 10:50 - Carsten Rose

- Assignee changed from Carsten Rose to Marc Egger

#6 - 18.08.2020 10:57 - Marc Egger

- Status changed from In Progress to Priorize

#7 - 06.09.2020 11:16 - Carsten Rose

Unfinished implementation (already used by Teaching Tools) and small documentation under <http://docs.qfq.io/en/develop/Report.html#api-call-qfq-report>.

#8 - 23.09.2020 10:09 - Marc Egger

- Status changed from Priorize to Closed

Es ist nun moeglich, auch die header der antwort zu setzen.
Ich habe die Doku etwas ausgebessert.
Bis ein neuer Request kommt, wuerde ich die Implementation so lassen.

#9 - 12.11.2020 22:46 - Carsten Rose

- Target version changed from 21.8.0 to 20.11.0

Files

html_snippet.html	2.76 KB	07.08.2020	Philipp Gröbelbauer
dataReport2.png	41.9 KB	09.08.2020	Carsten Rose

