

QFQ - Support #10345

Templates - Patterns QFQ Style

02.04.2020 23:10 - Benjamin Baer

Status:	New	Start date:	02.04.2020
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Discuss:			

Description

Ein "Pattern" im Design ist eine Vorgabe wie etwas aussehen soll.

Nehmen wir mal als Beispiel das teachingtool:

```
<div class="item">
  <div class="item-background"></div>
  <a class="item-image" style="background-image: url('typo3conf/ext/qfq/Classes/Api/download.php?s=5e8651ff802ce')" href="?id=tool&toolId=118"></a>
  <div class="item-description">Hinweise für die Übertragung einer geplanten Multiple-Choice-Prüfung in ein alternatives digitales Format</div>
  <span class="item-meta">1.4.2020 17:02 Uhr</span><a class="item-title" href="?id=tool&toolId=118">MC-Prüfungen (Remote)</a>
  <div class="item-tags">
    Tags:
    <a class="label label-primary" href="index.php?id=alltools&tagId=104">Leistungsnachweise</a>
  >
  <a class="label label-primary" href="index.php?id=alltools&tagId=126">Digitale Lehre</a>
</div>
```

Nun brauchen wir diesen Satz an mehreren Orten vom Teachingtool - und wenn man etwas am Grunddesign ändert, muss man an jeden Ort denken in dem es vorkommt.

Mit dingen wie `_link` machen wir es uns relativ einfach diese Dinge zu erstellen - ist in diesem Sinne bereits ein Pattern.

Nun, wandeln wir obigen Satz mal in ein Pattern um:

```
<div class="item">
  <div class="item-background"></div>
  <a class="item-image" style="background-image: url('{{backgroundImage:R}}')" href="{{toolLink:R}}"></a>
  <div class="item-description">{{itemDescription:R}}</div>
  <span class="item-meta">{{dateTime:R0}}</span><a class="item-title" href="{{toolLink:R}}">{{itemTitle:RE}}</a>
  <div class="item-tags">
    Tags:
    {{@tagLoopStart}}<a class="label label-primary" href="{{itemTagLink:R}}">{{itemTagName:R0}}</a>
  {{@tagLoopEnd}}
  </div>
</div>
```

Und dazu dann der SQL Aufruf:

```
10 {
```

```
sql = SELECT "blabla" AS _itemTitle, ....  
pattern = item
```

```
20 {  
    sql = SELECT CONCAT('index.php?id=tag&tagId=', tag.id) AS _itemTagLink, tag.name AS _itemT  
agName ...  
    loop = tagLoop  
}  
}
```

Die Designs selbst koennten dabei aehnlich wie System Variabeln definiert werden und seitenweit abgelegt sein. Oder besser wie Form direkt ueber eine Art editor im frontend erstellt (bzw. gleich als git datei - um die eigenen HTML Editoren verwenden zu koennen)

Pattern wird natuerlich hier immer quasi erst "renr" ausgefuehrt, da einzelne Teile aus subqueries kommen koennten (und mit loops sogar muessen). Neben Store :R koennte man auch die anderen Stores abfragen, vorallem Y und U.

Related issues:

Related to QFQ - Feature #10713: QFQ functions (reusable reports)

Priorize

04.06.2020

History

#1 - 02.04.2020 23:12 - Benjamin Baer

- Description updated

#2 - 02.04.2020 23:13 - Benjamin Baer

- Description updated

#3 - 02.04.2020 23:14 - Benjamin Baer

- Description updated

#4 - 02.04.2020 23:15 - Benjamin Baer

- Description updated

#5 - 02.04.2020 23:18 - Benjamin Baer

- Description updated

#6 - 04.06.2020 19:22 - Marc Egger

- Related to Feature #10713: QFQ functions (reusable reports) added

#7 - 04.06.2020 20:43 - Nicola Chiapolini

Mindestens den Darstellungsteil kann man mit Twig-Template-Dateien loesen.