A brief introduction into the Custom Question-Type Editor

What is this editor for?

For creating, prototyping and administering custom question types
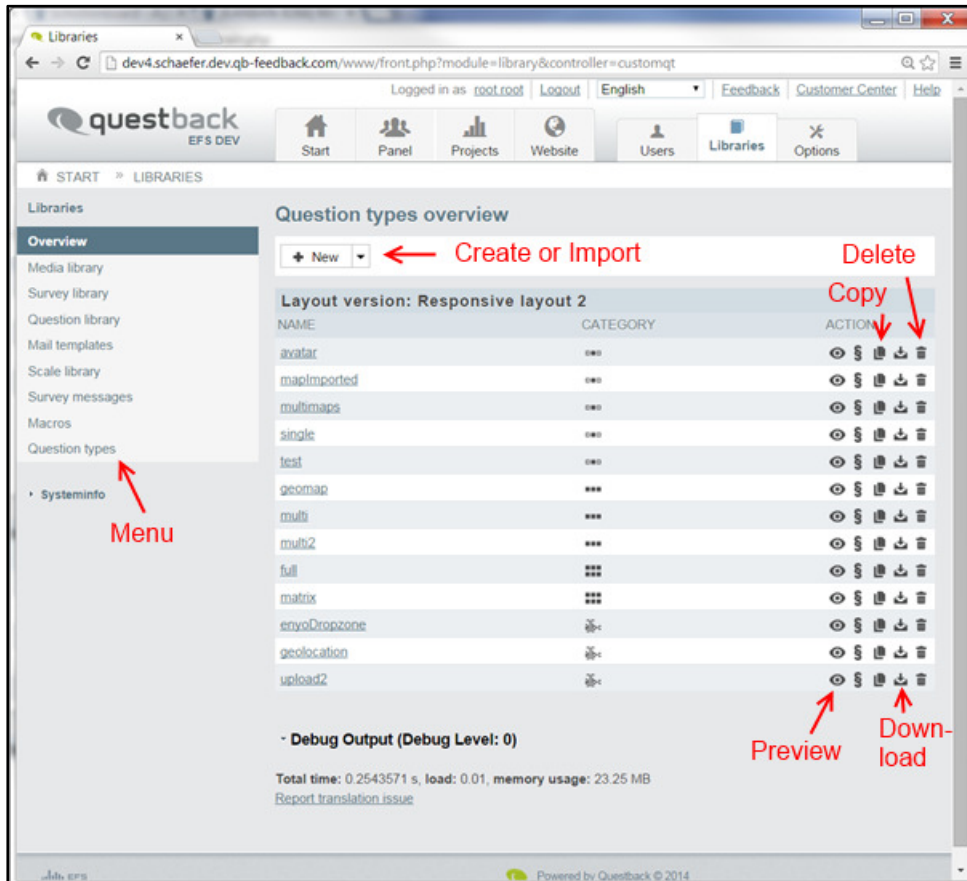
Pre-Requisites and Requirements

Minimum skills and knowledge in:

EFS Survey, CSS, HTML5, SMARTY2, JQUERY, JAVASCRIPT

When developing advanced applications:

PHP, MySQL, EFS Module Knowledge and whatever you think of

Overview

The overview page offers the following actions:

- Create a new custom question type from an empty template skeleton
- Import an archived custom questiontype (zip file) into the questiontype editor
- Copy an existing custom questiontype
- Preview an existing custom questiontype
- Delete an existing custom questiontype

Create a custom question type:
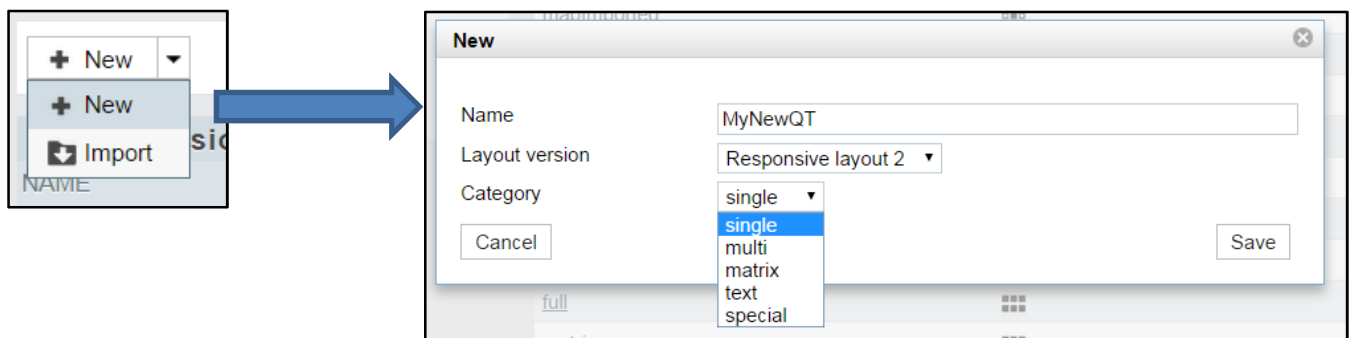
By clicking the "New"-split-button above the overview list, you open the "New"-dialog. In the opened dialog you have to name your new CQT. In the second step you can choose the layout version. Finally, you have to select a category and click the save button. For the moment there are five categories to choose:

Category	Question-Type's Base Data Model
Single	111
Multi	121
Matrix	311
Text	143
Special	911

Let's start with a single CQT to explain the next steps.

myNewQT

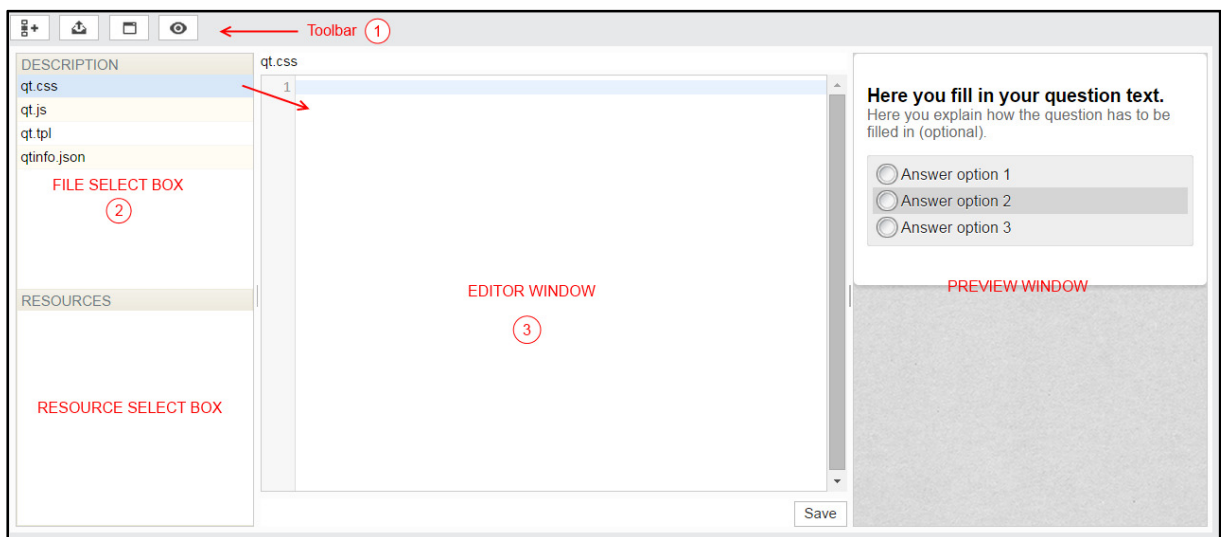After saving the dialog a new entry will occur in the overview. Now, click on the link as shown above. At the beginning the question type contains some empty presets.

DESCRIPTION
qt.css
qt.js
qt.tpl
qtinfo.json

RESOURCES

qt.css
1

Here you fill in your question text.
Here you explain how the question has to be filled in (optional).

Answer option 1
Answer option 2
Answer option 3

Save

The editor

DESCRIPTION
qt.css
qt.js
qt.tpl
qtinfo.json

FILE SELECT BOX
②

RESOURCES

RESOURCE SELECT BOX

Toolbar ①

qt.css
1

EDITOR WINDOW

③

Here you fill in your question text.
Here you explain how the question has to be filled in (optional).

Answer option 1
Answer option 2
Answer option 3

PREVIEW WINDOW

Save

The Toolbar

The editor toolbar contains four icons:

[icon] Back to overview

[icon] Upload file

[icon] Open pop-up preview window

[icon] Toggle preview window to maximize or minimize editor window

[icon] Upload dialog

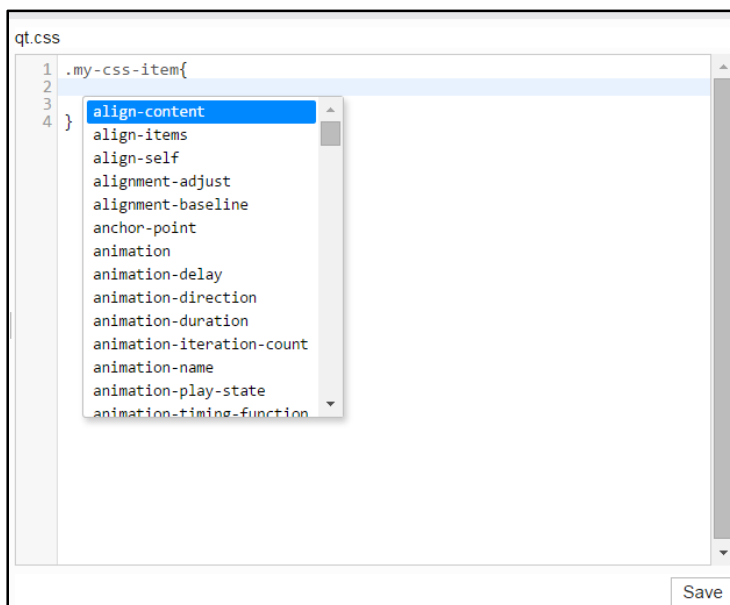When clicking the upload file icon a new dialog pops up.
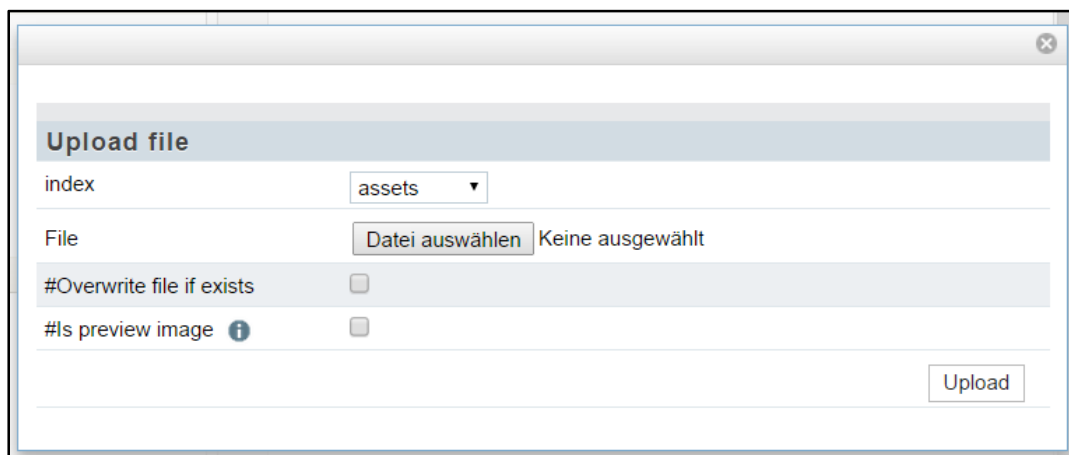
Here you can manage all files your CQT uses. BTW, when you have access to the server, you can put the files directly to the CQT folder. You can find your currently created custom question type folder by regarding the following rule:

/{rootdir-of-installation}/uc/custom/tpl_layout_responsive2/questiontypes/single/myNewQT/
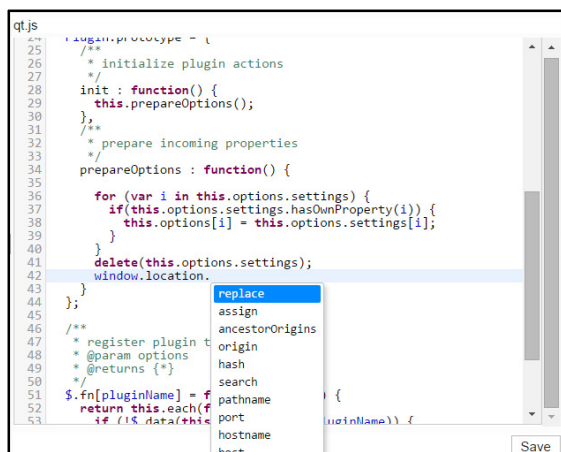
Behind the scene

There are different editors managing your code in the editor window:

CSS files are edited with CodeMirror which supports syntax highlighting and gives you hints when dealing with css elements and attributes. The example below gives you an image what you can do with these autocompleting hints in a css environment. But, you need to know what CSS is for. When you do not know what CSS is for, then skip the whole tutorial.

```
qt.css
1  .my-css-item{
2
3          align-content
4  }       align-items
           align-self
           alignment-adjust
           alignment-baseline
           anchor-point
           animation
           animation-delay
           animation-direction
           animation-duration
           animation-iteration-count
           animation-name
           animation-play-state
           animation-timing-function
                                                    Save
```

CodeMirror supports JS (Javascript) files by giving you some basic hints for common Javascript browser objects and syntax highlighting the source code.
By pressing Ctrl-Space right after the dot an inline autosuggest pop-up appears. It contains some basic Javascript commands.

```
qt.js
    Plugin.prototype = {
25      /**
26       * initialize plugin actions
27       */
28      init : function() {
29          this.prepareOptions();
30      },
31      /**
32       * prepare incoming properties
33       */
34      prepareOptions : function() {
35
36          for (var i in this.options.settings) {
37              if(this.options.settings.hasOwnProperty(i)) {
38                  this.options[i] = this.options.settings[i];
39              }
40          }
41          delete(this.options.settings);
42          window.location.
43      }           replace
44  };              assign
45                  ancestorOrigins
46      /**         origin
47       * register plugin t  hash
48       * @param options    search
49       * @returns {*}       pathname
50       */                  port
51      $.fn[pluginName] = f           {
52          return this.each(f          pluginName)) {
53              if (!$.data(this          hostname
                                          host              Save
```

6

If you open a Smarty template file the editor will highlight your smarty code. You can use EMMET autocompletion to write html snippets quickly.

What is EMMET? Go to the emmet homepage here…

A brief description of Emmet:

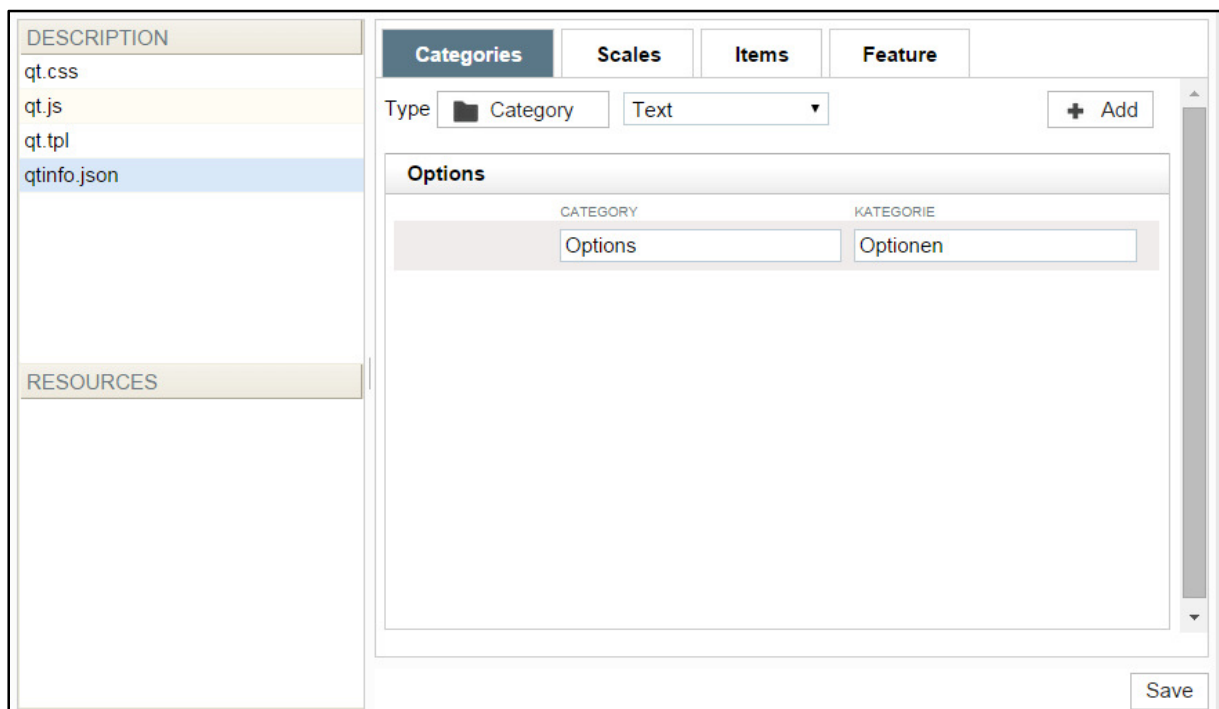Emmet is a plugin for many popular text editors which greatly improves HTML & CSS workflow:

HTML from CSS: You've already known how to use Emmet abbreviations: its syntax is inspired by CSS selectors.

Dynamic snippets: Each abbreviation is transformed in runtime: just slightly change its name to get a different result.

Ultra-fast coding: With Emmet you can quickly write a bunch of code, wrap code with new tags, quickly traverse and select important code parts and more!

When clicking on the qtinfo.json file the configuration dialog will open in the editor window.

7

Each of the different CQT comes along with a specific setup of this configuration dialog. Behind the scene it is an overlay to the qtinfo.json file. The php description file (not shown in the picture) is being generated from the settings in the json file.

The Configuration Dialog
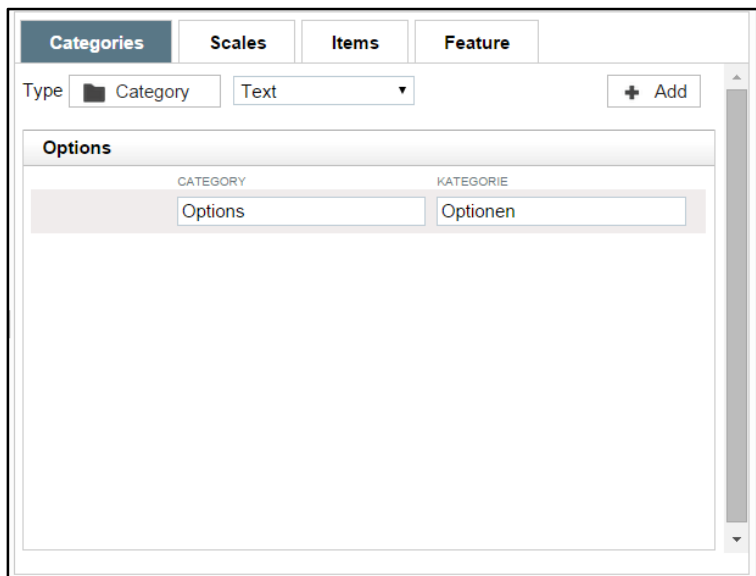
The screen contains a tab section where you switch between the different configuration features:

The categories tab holds the form for defining option headers and its fields. An option header is a collapsible section in the survey's question editor page.
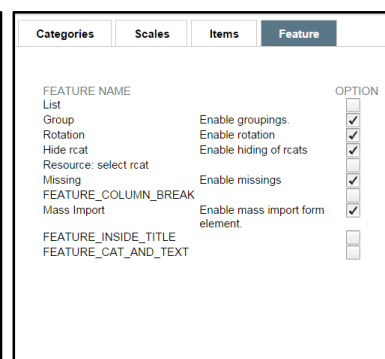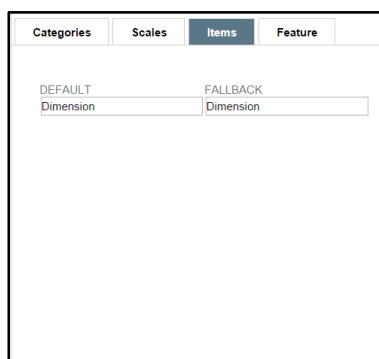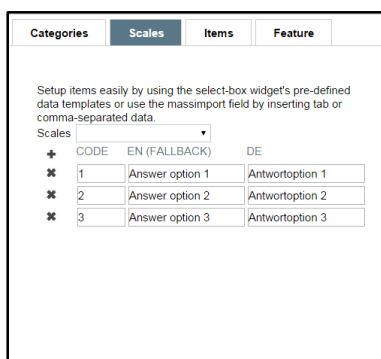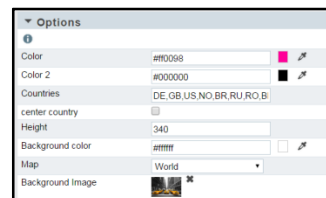

The scales tab is for the definition of scale items, in the items tab you change the settings for item lists. And, the feature tab holds the checkboxes for activating features per custom question type.
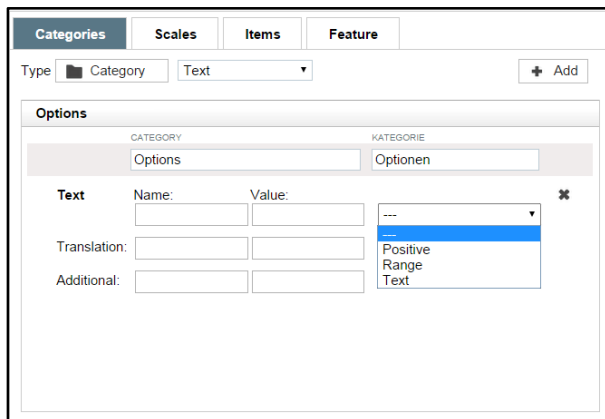
Adding controls
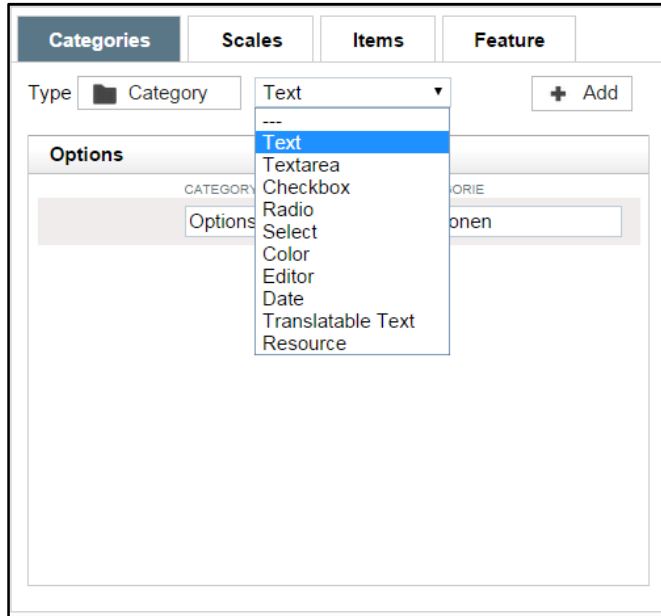
The categories tab contains two areas: The toolbar and the controls section.

By clicking the Category button you can add a new option header to the categories tab.
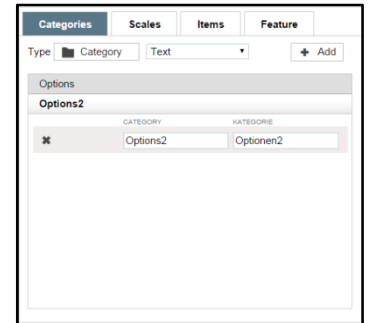
Each option header becomes a collapsible area in the question editor.
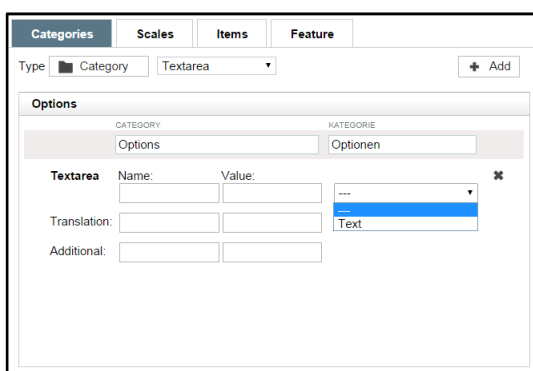
The text field control

This is control is deletable by clicking the x-Button.

The text field control is defined by 7 single fields:

- Name of control
- Value of control
- GUI Translation default
- GUI Translation fallback
- Additional default
- Addition fallback

The textarea / editor control

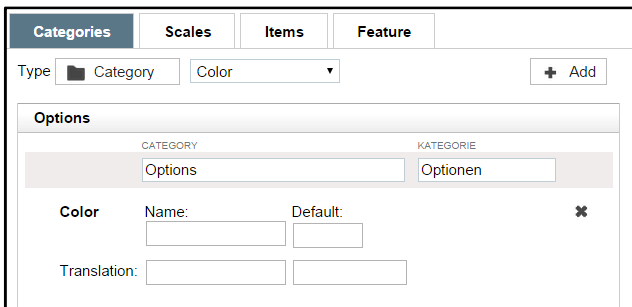The textarea/ editor control is defined by these fields:

- Name of control
- Value of control
- GUI Translation default
- GUI Translation fallback

9

- Additional default
- Addition fallback

This control is deletable by clicking the x-Button. The difference to the text field control is the missing validator type for the editor field. In the future the editor field will be taken for complex editors being used in the question editor.
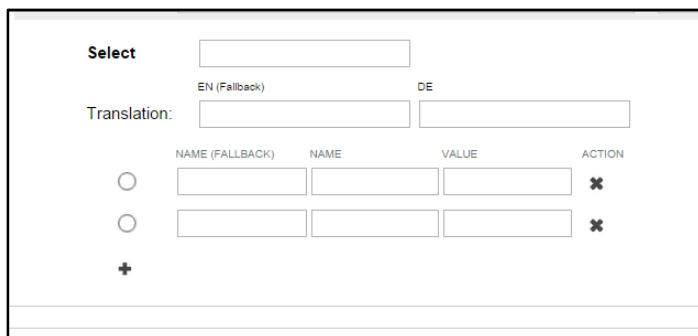
The color field control

The color field control is defined by 4 fields:

- Name of control
- Value of control
- GUI Translation default
- GUI Translation fallback

This is control is deletable by clicking the x-Button.

The select / radio control

The select /radio control defines a selectbox/radio group widget in the question editor.

You have to name the field and its translation properties. This time there also is a plus button of the control, which can be used to add options to the control widget. Each option has two input fields for text translation, a radio field to define the default option and a delete button for removing the option from the list.

Further fieldtypes:

- Date control
- Translatable text control
- Resource control

The scales tab

Categories	Scales	Items	Feature	

Setup items easily by using the select-box widget's pre-defined data templates or use the massim inserting tab or comma-separated data.

Scales [▼]

	CODE	EN (FALLBACK)	DE
✖	1	Germany	Deutschland
✖	2	United Kingdom	Vereingtes Königreich
✖	3	USA	USA
✖	4	Norway	Norwegen
✖	5	Brazil	Brasilien
✖	6	Russia	Rußland
✖	7	Romania	Rumänien
✖	8	Belgium	Belgien
✖	9	Netherlands	Niederlande

The scales tab is self-explanatory. Here you can add items to a scale used by the CQT. To delete an item simply click on the left-hand x-Button. Each scale item consists of three input fields.

Atop there is a select box with some pre-defined scales. Once selected the scale tab will be filled with the scale set behind the selected option.

The items tab

In the items tab you can create entries for your item list. Each category has its special setup for this section. A CQT based on the single category has only a single changeable item.

Category: Single

Category: Matrix

The items form works the same way as the scale form works.

The feature tab

The feature tab offers a variable amount of checkboxes which are used to enable question type features in the question editor. Each category has its specific feature set according to the definition of a question model.