

Tutorial: Editor customizing

Author: David Haasler

Table of contents

1	Editor customizing.....	2
1.1	Localizing identifiers.....	2
1.2	Setting the display name and icon.....	4
1.3	Object structure window.....	5

Requirements

No other prerequisites are needed for this tutorial than for the preceding tutorials.

- > [Define Information Model](#)
- > [Developing transformations](#)

Description

Part three shows how maintaining in the onion.net Editor can be improved with minimum effort through self-explanatory and located identifiers, additional sortable list views and helpful pictograms. This step is not absolutely necessary to the functioning of our quotation administration, but it is worth doing.

Note: You will make the content creator's work easier through the assignment of pictograms alone. It is no longer necessary to read each structure point.

Signs and symbols



Boxes marked with an arrow symbol and a green border contains instruction of what to do next.

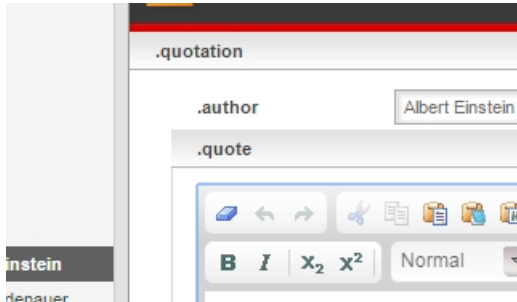


This kind of boxes contains tips and tricks.

Source code is shown in blue boxes.

1 Editor customizing

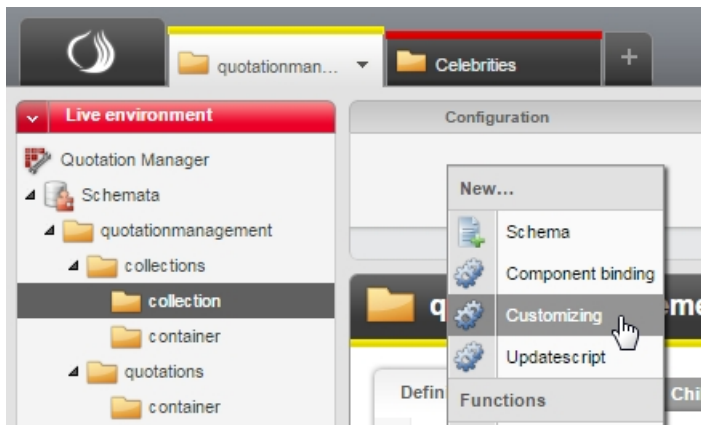
After the schema has been created, all fields in the maintenance mask of the editorial system have the element names allocated in the schema. In the case of a quotation, these are **quotation**, **author**, **quote** and **image**.



The field labels can be adapted to the needs of the editor by way of localisations. The localizations can be created on the respective schemas directly.



In the module, select the schema **quotation** and right-click in the top right-hand section (object structure window) before then selecting *Customizing*. In the dialogue, select *runtime* as the type and *English* as the language. Then select *Create*.



1.1 Localizing identifiers

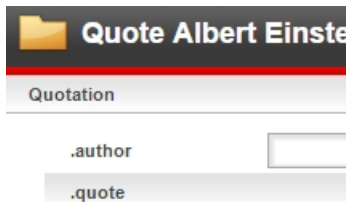
Language-specific settings of the data types are recorded in the objects "runtime.LANGUAGE", in our case "runtime.en". The newly created configuration file is filled with the element "o:settings" by default.

The data types of our schema are now localised within this element. To do this, we create a section with the appropriate configuration for each component. We shall use the following XML as our first example.

```
<o:component aspect="Label" match=".quotation">
  <label>Zitat</label>
</o:component>
```



Copy the XML fragment into the element **o:settings** and save your change. Then switch to the editorial system and open a quotation. You should now be displayed **quotation** as the object heading.



Depending on the kind of type to be localised, different notations are necessary in the attribute *match*. The localization of elements begins with a dot followed by the element name. The actual localization is then made with the tag **Label**. Further settings can then be made here depending on the type.



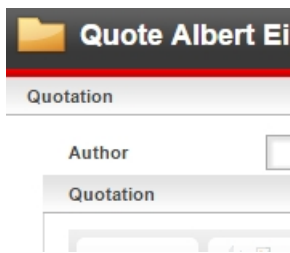
Tip: If elements with different names are to be given the same designation, the indication of all element names, separated by commas, is also possible in the match attribute. (Example: *match=".quotation, .quote"*)

With the same principle, we now locate the outstanding fields.

```
<o:component aspect="Label" match=".quotation, .quote">
  <label>Zitat</label>
</o:component>
<o:component aspect="Label" match=".author">
  <label>Autor</label>
</o:component>
<o:component aspect="Label" match=".image">
  <label>Bild</label>
</o:component>
```



Copy the XML block and replace the previous content of the settings element. Save the changes and check the result in the editorial system.



1.2 Setting the display name and icon

If we create a new object in the editorial view, the entire schema name and folder icon will always be shown. This can soon become unclear for a content creator if there are many different objects.

onion.net therefore offers the possibility of giving the created schemas a display name and an appropriate icon.

In the quotation administration module, switch to the schema **quotation**. Select the *Settings* tab and then the *Localisation* tab underneath it. Select *Quotation* as the display name and choose a suitable icon for the quotation.



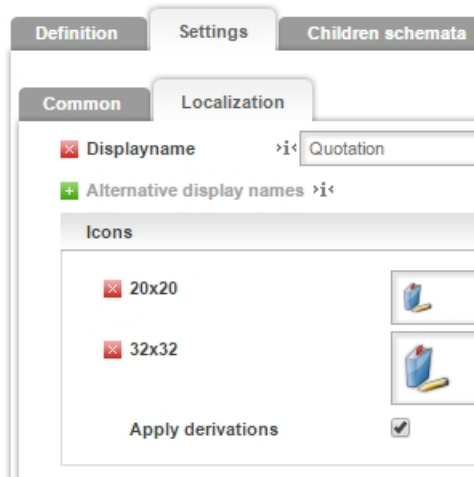
Tip: The small icon should be 20x20 pixels in size; the large icon 32x32. Many web pages offering free downloadable icons or iconsets for certain topics can be found via relevant search engines.

Follow the exact same procedure for all root elements created:

- > quotations/container
- > Name: quotations
- > quotations/quotation
- > Name: quotation
- > collections/container
- > Name: quotation collections
- > collections/collection
- > Name: quotation collection



In order to localise the display name for other languages also, a display name per language can be defined under *Alternative display names*.



You can check the changes directly after saving.



In the content management, right-click on the root element and the names **quotation collections** and **quotations** just allocated will already appear under *New...* Under **quotations**, you can create a **quotation** via *New...* under the **quotation collections**, a **quotation collection**.

1.3 Object structure window

In the next step, we will define an object structure window. This marks the content of objects in a list form and serves for clarity. The result will be a list view similar to the one displayed when the editor module **quotation management** is selected.



For this purpose, create an object of the type *Detail Views* below the module **quotation management** by right-click. Then assume the following default content as the content.

```

<childrenViews xmlns="http://onion.net/modulesystem"
  xmlns:i18n="http://onion.net/common/i18n"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:onion="http://onionworks.net/2004/data"
  xmlns:dw="http://onion.net/editor/detailview"
  onion:schema="Onion:http://onion.net/2010/editor/modulesystem/schemata/childrenviews"
  i18n:language="de"
>
  <detailView schemaLocations="" applyToDerivations="false">
    <columns>
      <column
        id="name"
        label="name"
        i18n:key="name"
      />
    </columns>
    <childType
      schemaLocations=""
      structureInvisible="false"
      applyToDerivations="true"
    >
      <column ref="name">
        <xsl:value-of select="onion:meta/@onion:name" />
      </column>
    </childType>
  </detailView>
</childrenViews>

```

In order to define which schema the list view is to be used for, we enter the appropriate schema into the *schemaLocation* attribute of the element *<detailView>*. In this step it is to be a list view for our quotation container (**quotations/container**).



The most simple way to do this is again by dragging & dropping: drag the schema **quotations/container** from the schema administration into the empty attribute *schemaLocation*.

```

<detailView schemaLocation="quotationmanagement/quotations/container" />

```

The quotations are to be displayed as a child type (*<childType>*) for the list view. Drag the schema **quotations/quotation** into the *schemaLocation* attribute of the *<childType>* element.

```

<childType
  schemaLocation="quotationmanagement/quotations/quotation"
  structureInvisible="false"
  handleDerivations="true"
/>

```


Our list view is to be given three columns. We want to represent the author, the quotation and the picture in a table.



To do this, we adapt the content of the *Columns* elements so that it contains the following elements.

```
<column
  id="author"
  label="Author"
  i18n:key="author"
/>
<column
  id="quote"
  label="Quotation"
  i18n:key="quote"
/>
<column
  id="image"
  label="Image"
  i18n:key="image"
/>
```

By the ID of the respective column, indicate the name of the element from the schema to be displayed.



Now customize the elements that have just been inserted:

- > column id="author"
- > Add a width of 15% (width= "15%")
- > column id="image"
- > Add a width of 5% (width= "5%")
- > Center the picture (align="center")
- > Define the schema node as a picture (type="image")

We will now describe how the quotation is represented in the list view.

For this purpose, we also need three <column> elements under <childType>.



In the attribute *refs* we enter the ID of the column allocated above under <columns>. The desired output is now generated using XSLT. For the special field **image**, we need an auxiliary method since a graphic is involved:

```
dw:binaryUrl(c.id(), onion:content/quotation/image)
```

This call generates a uniform representation of 48x48 pixels for any picture. The complete configuration now looks as follows:

```







<childrenViews xmlns="http://onion.net/modulesystem"
  xmlns:i18n="http://onion.net/common/i18n"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:onion="http://onionworks.net/2004/data"
  xmlns:dw="http://onion.net/editor/detailview"
  onion:schema="Onion:http://onion.net/2010/editor/modulesystem/schemata/childrenviews"
  i18n:language="de"
>
  <detailView schemaLocations="quotationmanagement/quotations/container"
  applyToDerivations="false">
    <columns>
      <column
        id="author"
        label="Author"
        i18n:key="author"
        width="15%"
      />
      <column
        id="quote"
        label="Quotation"
        i18n:key="quote"
      />
      <column
        id="image"
        label="Image"
        i18n:key="image"
        width="5%"
        align="center"
        type="image"
      />
    </columns>
    <childType
      schemaLocations="quotationmanagement/quotations/quotation"
      structureInvisible="false"
      applyToDerivations="true"
    >
      <column ref="author">
        <xsl:value-of select="onion:content/quotation/author" />
      </column>
      <column ref="quote">
        <xsl:value-of select="onion:content/quotation/quote" />
      </column>
      <column ref="image">
        <xsl:if test="string-length(onion:content/quotation/image) > 0">
          <xsl:value-of select="dw:binaryUrl(c.id(),
onion:content/quotation/image)" />
        </xsl:if>
      </column>
    </childType>
  </detailView>
</childrenViews>

```

```
</detailView>  
</childrenViews>
```



Now update the editor and click on quotations on the left. You will now see the list view that has just been created on the right above the familiar detail view.

Author	Quotation	Image
 Albert Einstein	Imagination is more important than knowledge. For knowledge is limited.	
 Konrad Adenauer	All of us live under the same sky, but we don't all have the same horizon.	
 Thomas Carlyle	The greatest of all faults, I should say, is to be conscious of none.	



Tip: Each list view can be custom-sorted. This can be done by simply clicking once on the heading of the respective column.

You've done it!

With just a few configuration steps, we have created a customized maintenance environment for the editor.