Getting started with Semantic MediaWiki



Approval not enabled

Semantic MediaWiki Basics
Define properties
Define properties
Create a template
Create a form
Integrate form edit mode
Template styling
Query the data
Naming conventions

Table of contents

1 Semantic MediaWiki Basics 4
2 Example: Customer data 13
2.1 Define properties 13
2.2 Create a template 17
2.3 Create a form 22
2.4 Integrate form edit mode 26
3 Template styling 28
4 Query the data 33
5 Naming conventions 37

1 Semantic MediaWiki Basics

Contents

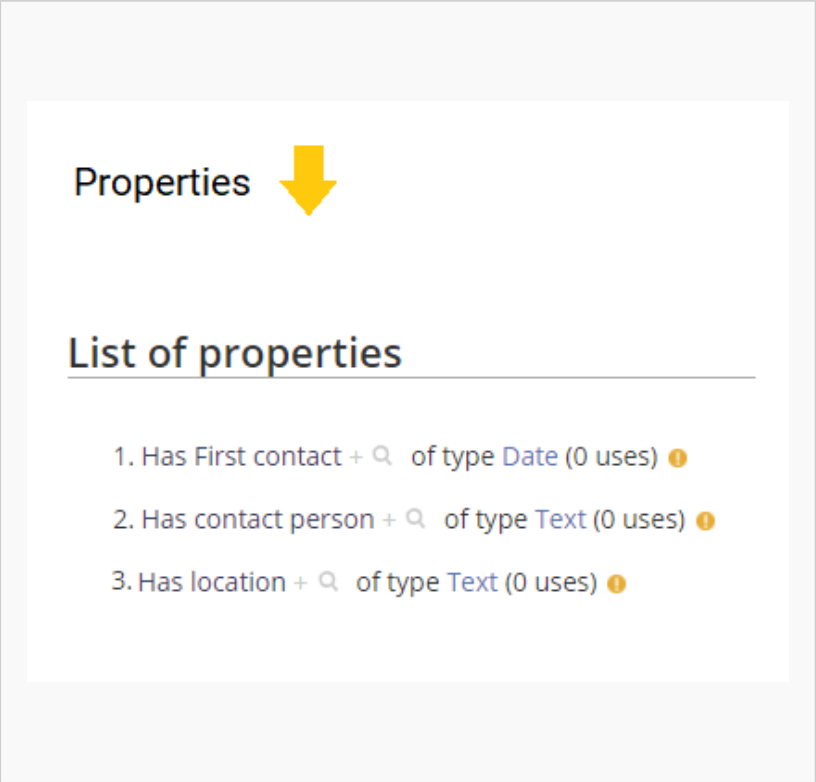
1 Scope and terminology 5
2 Define properties 8
3 Create a template 8
4 Create a form 8
5 Assign a category 9
6 Example 9

Scope and terminology

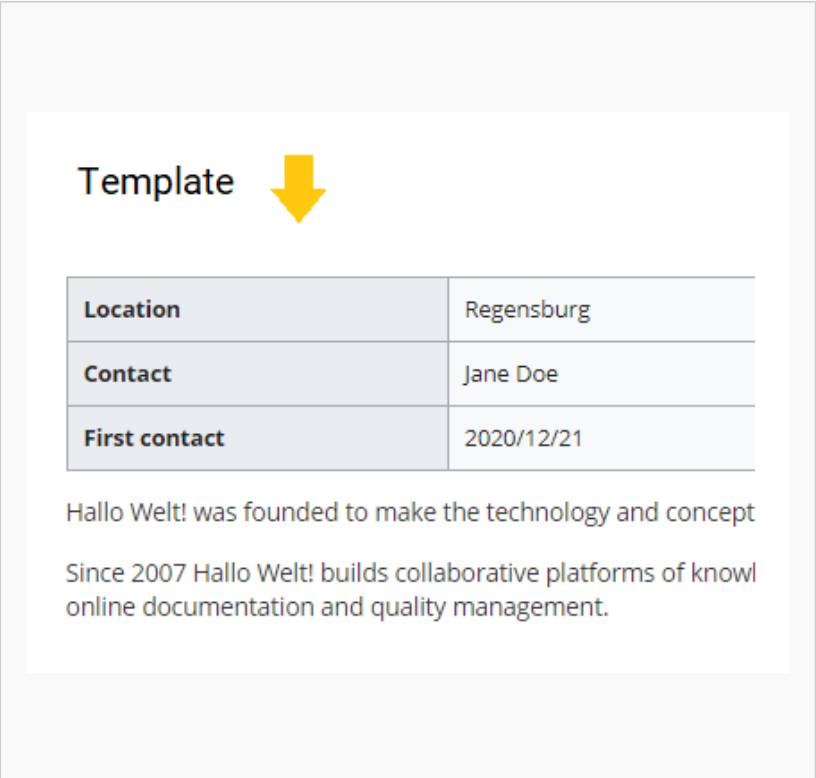
Semantic MediaWiki (SMW) is a MediaWiki extension that enables you to define and query data on a wiki page.

The following steps are usually taken to work with data-based content in your wiki :

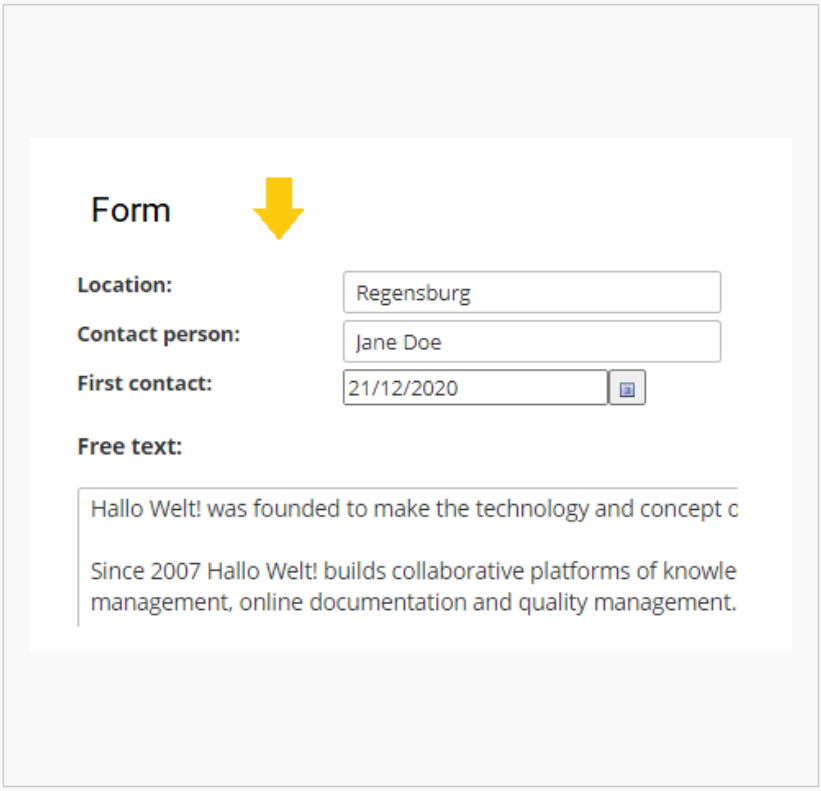
1. **Define properties:** Properties determine which values can be queried in the wiki.
2. **Create a template:** This creates the output format of the data set on the wiki pages.
3. **Create a form:** Using this form, Wiki users can comfortably enter their data.
4. **Assign a category:** Each page that contains a semantic query is usually tagged with a category that was created for this purpose. This, in addition to the attributes, provides an important means of querying and filtering the pages. It also creates a form editing mode for the wiki pages that contain this category.



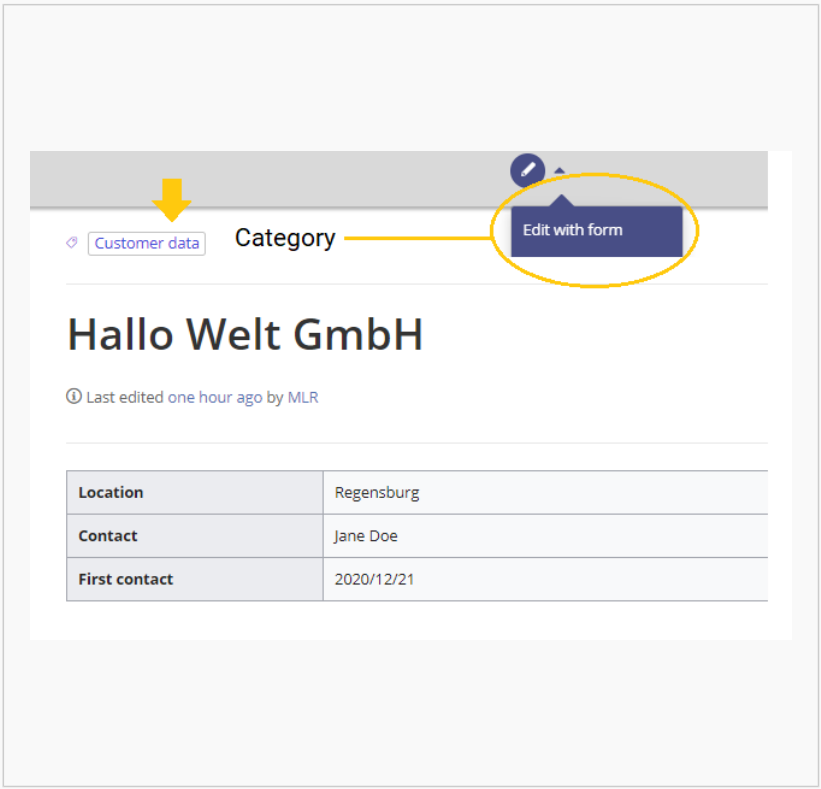
Semantic properties



Semantic template



Data entry form



Category to organize data

Define properties

To make information available as data on a wiki page, we create so-called **properties**. For example, if we want to semantically record customer data, we create a wiki page for each customer. We collect customer data such as location on every customer page. To do this, we create a property that records the location of the customer.

```
[[Has location::Regensburg]]
```

Afterwards I can list all customers at a certain location or all customers with their associated locations in the wiki.

Create a template

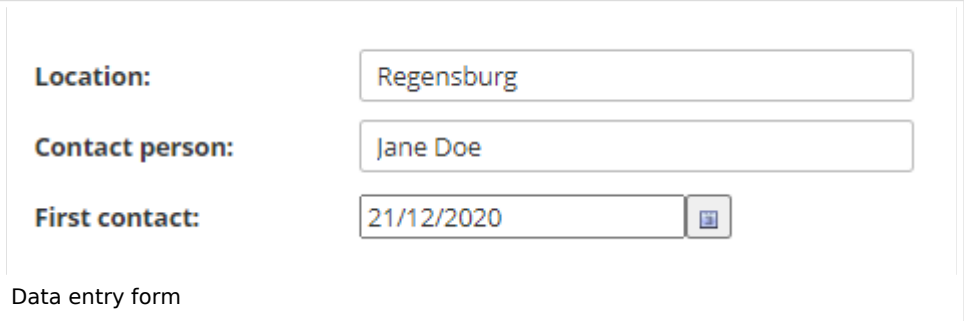
Since we often want to display several such properties and their values on one page at the same time, we format a special area that clearly shows the collected page attributes. This requires a (data) template, which can then be used on any wiki page.

Location	Regensburg
Contact	Jane Doe
First contact	2020/12/21

Data template

Create a form

To support our users with data entry tasks, we create an input form. The input form is shown when users switch to the edit mode of a wiki page that has the semantic template included.



Data entry form

Assign a category

Wiki pages that use semantic templates are tagged with a category. This category enables the assignment of the input form to the wiki page. The category is also used to query the data collected via the template and display it anywhere in the wiki.

Example

On the following pages, we will create an example that collects and displays customer data it in the wiki.

Example: Collect customer data

2 Example: Customer data

In this exampla for SMW, we create a query for customer data. The collected information is shown at the beginning of each customer page.

Location	Regensburg
Contact	Jane Doe
First contact	2020/12/21

Example: Customer data

This requires 4 steps:

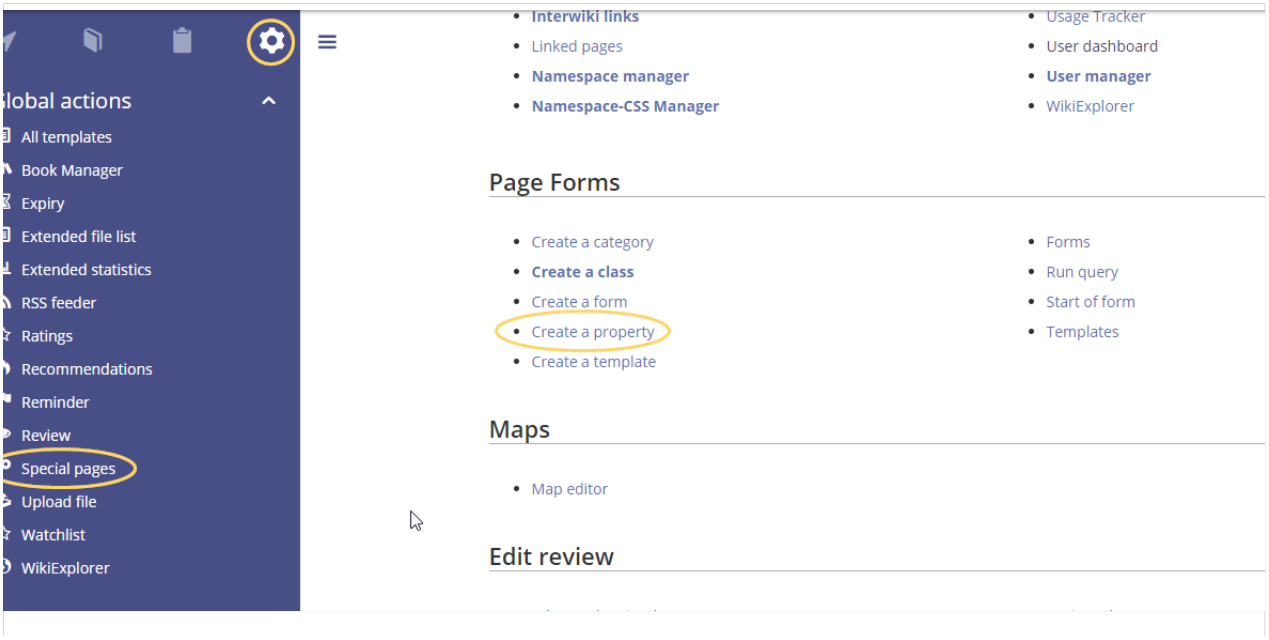
1. Defining properties
2. Creating a template
3. Creating a form
4. Creating a category

Defining properties

For our customer query, we define 3 properties:

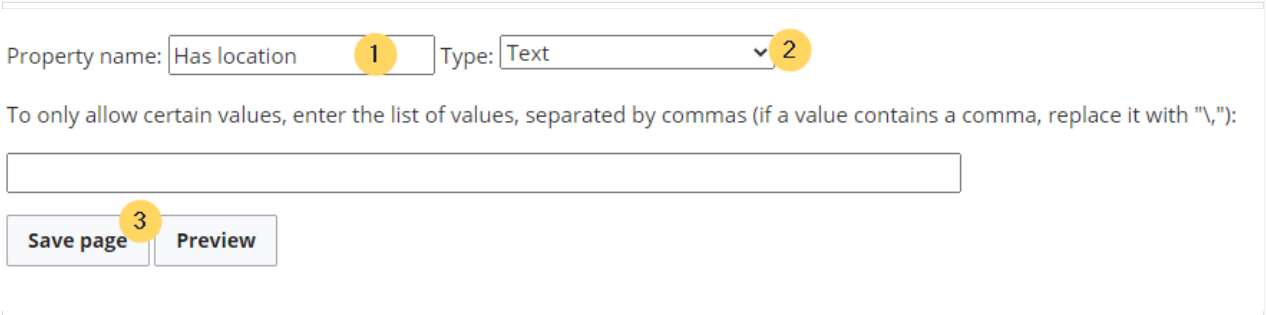
- Location
- Contact person
- Date of first contact

1. Switch to *Special pages* in the navigation menu *Global Actions*. Scroll down to section **Page Forms**.



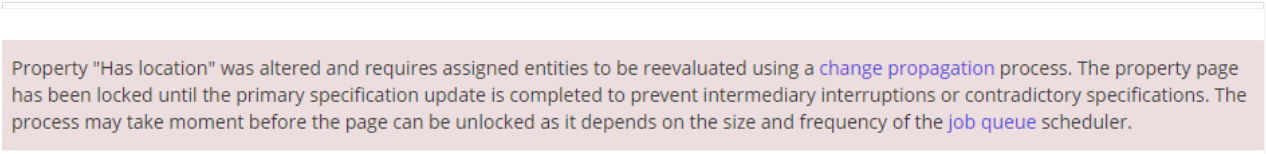
Create a property

2. Click *Create a property*. In the form, enter the name and data type for the first property Has location:: - Property name (1): Hat Location - Data type (2): Text



Create the first property

Click *Save page* (3). A message box appears that shows you that the property is now being processed in the wiki. You can ignore this message.



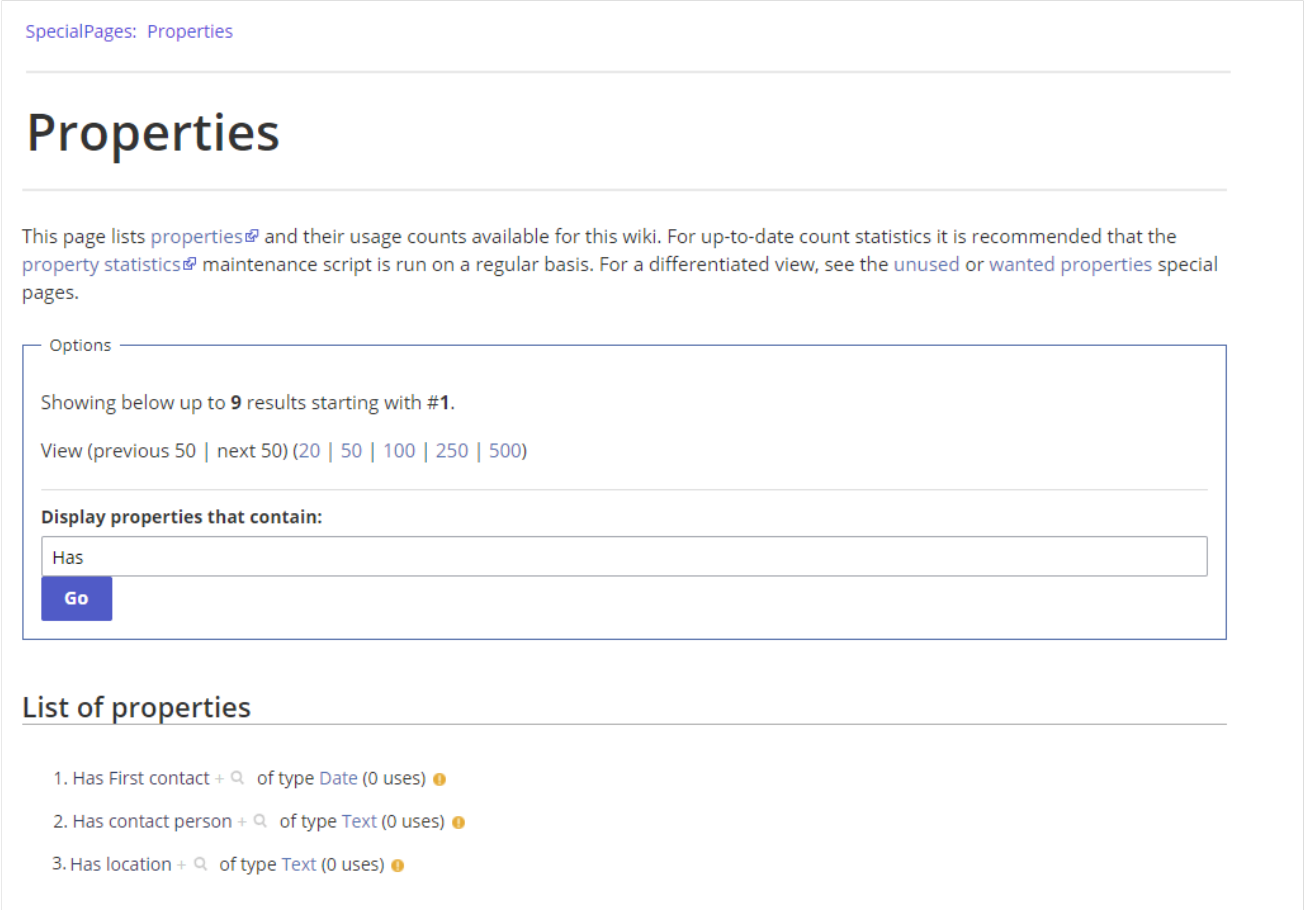
Property gets processed

For each property, a page is created in the namespace Property in the wiki. This page contains the information about the data type.

3. Switch back to the page Wechseln Sie erneut zur Seite *Spezial:CreateProperty*. Create the property ***Has contact person*** with data type ***Text*** as shown before.
4. Switch back to *Special:CreateProperty*. Create the property ***Has First contact*** with data type ***Date***.

Tip: In your personal navigation, you can create a quick link to the Page Forms section of the special pages main page. Just enter the following link: **[[Special:SpecialPages#Page_Forms]]

You can review the properties in your wiki by going to the page *Special:Properties*.



Special page: Properties

The necessary properties for data collection are now available. In the next step, we create the template that will be included in our customer pages to display the customer data.

Create the template

Related info

- https://www.semantic-mediawiki.org/wiki/Help:Properties_and_types

2.1 Define properties

In this exampla for SMW, we create a query for customer data. The collected information is shown at the beginning of each customer page.

Location	Regensburg
Contact	Jane Doe
First contact	2020/12/21

Example: Customer data

This requires 4 steps:

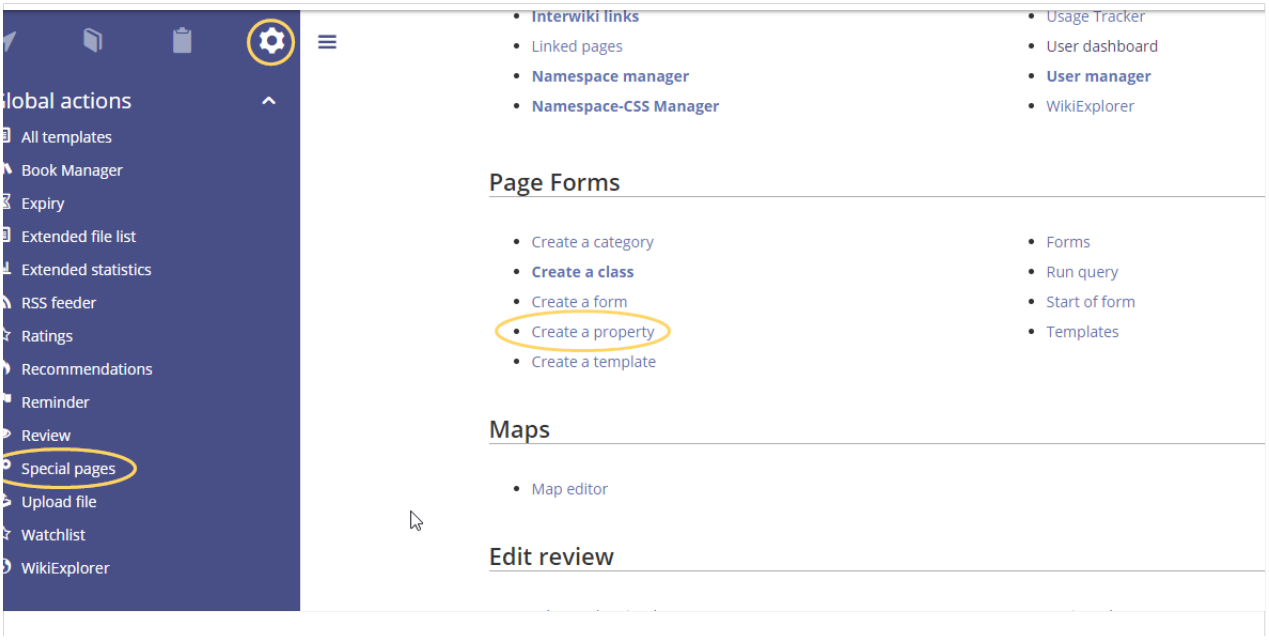
1. Defining properties
2. Creating a template
3. Creating a form
4. Creating a category

Defining properties

For our customer query, we define 3 properties:

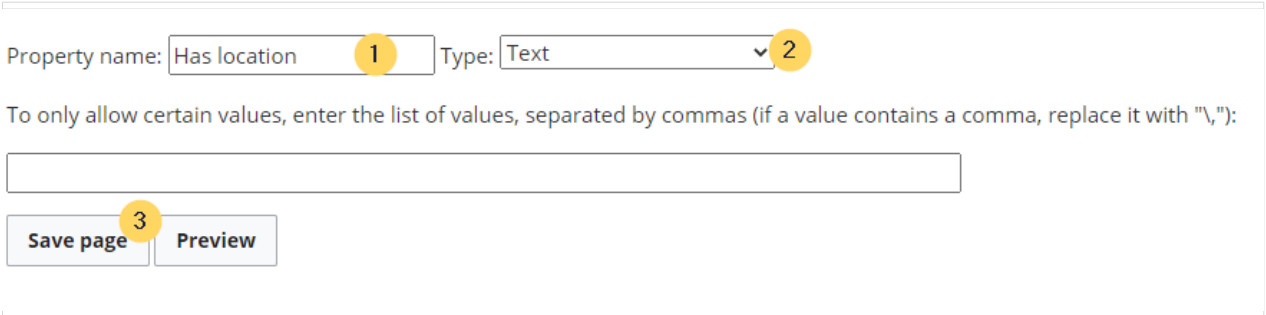
- Location
- Contact person
- Date of first contact

1. Switch to *Special pages* in the navigation menu *Global Actions*. Scroll down to section **Page Forms**.



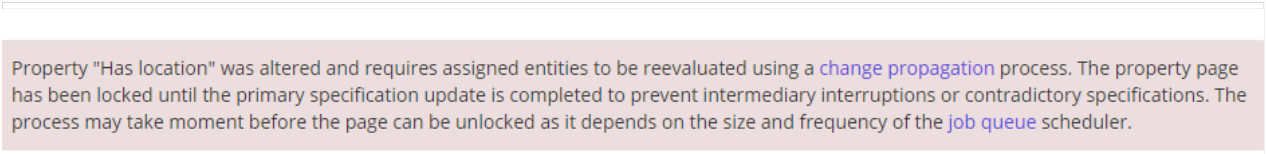
Create a property

2. Click *Create a property*. In the form, enter the name and data type for the first property Has location:: - Property name (1): Hat Location - Data type (2): Text



Create the first property

Click *Save page* (3). A message box appears that shows you that the property is now being processed in the wiki. You can ignore this message.



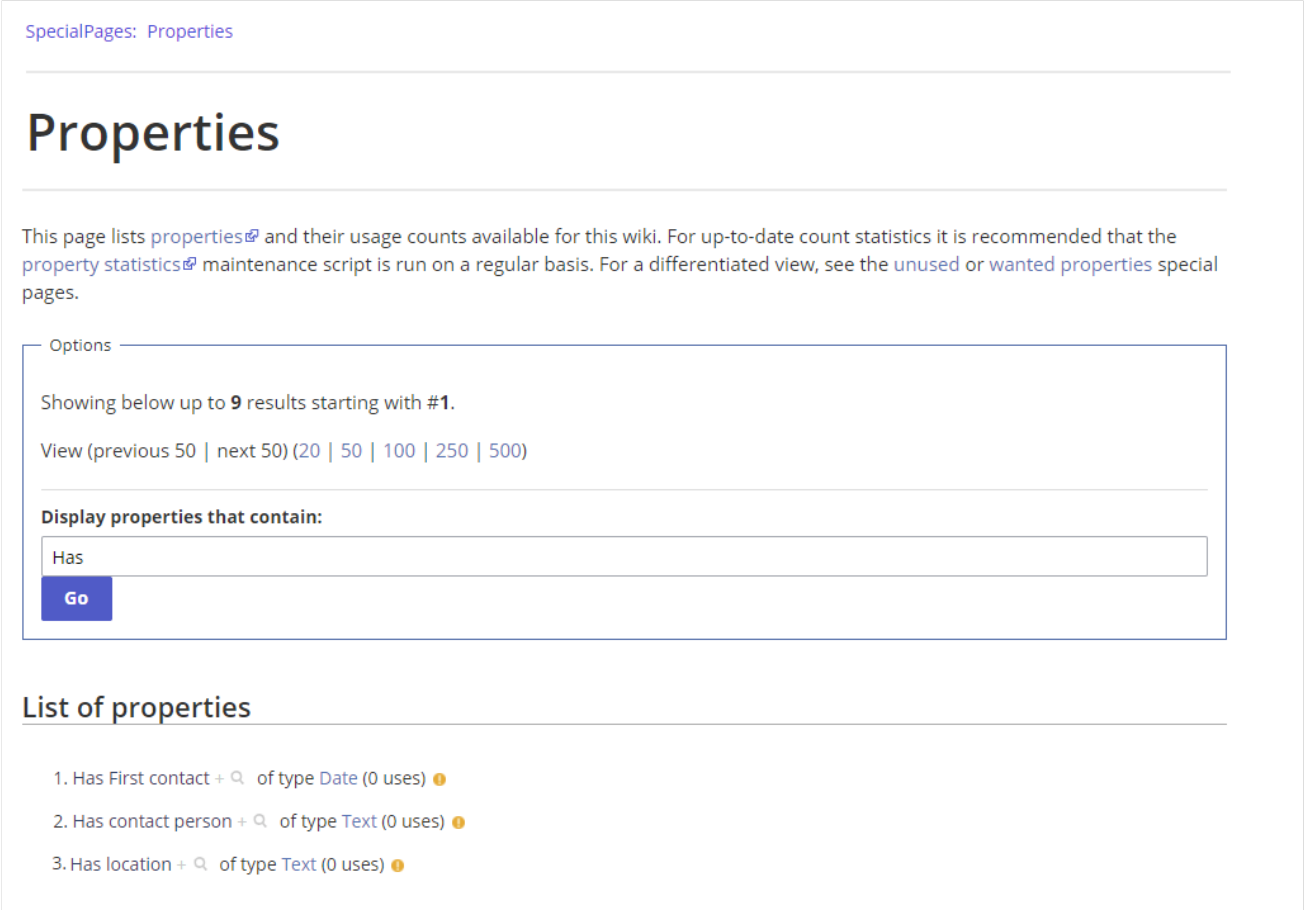
Property gets processed

For each property, a page is created in the namespace Property in the wiki. This page contains the information about the data type.

3. Switch back to the page Wechseln Sie erneut zur Seite *Spezial:CreateProperty*. Create the property ***Has contact person*** with data type ***Text*** as shown before.
4. Switch back to *Special:CreateProperty*. Create the property ***Has First contact*** with data type ***Date***.

> **Tip:** In your personal navigation, you can create a quick link to the Page Forms section of the special pages main page. Just enter the following link: **[[Special:SpecialPages#Page_Forms]]

You can review the properties in your wiki by going to the page *Special:Properties*.



Special page: Properties

The necessary properties for data collection are now available. In the next step, we create the template that will be included in our customer pages to display the customer data.

Create the template

Related info

- https://www.semantic-mediawiki.org/wiki/Help:Properties_and_types

2.2 Create a template

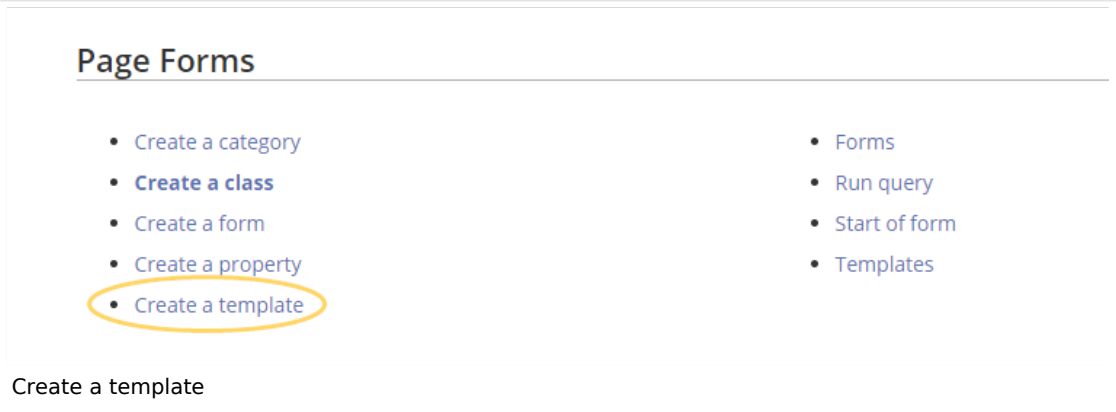
Contents

1 Creating the display template 18
2 Including the template on a page 19
3 Next step 21
4 Related info 21

Creating the display template

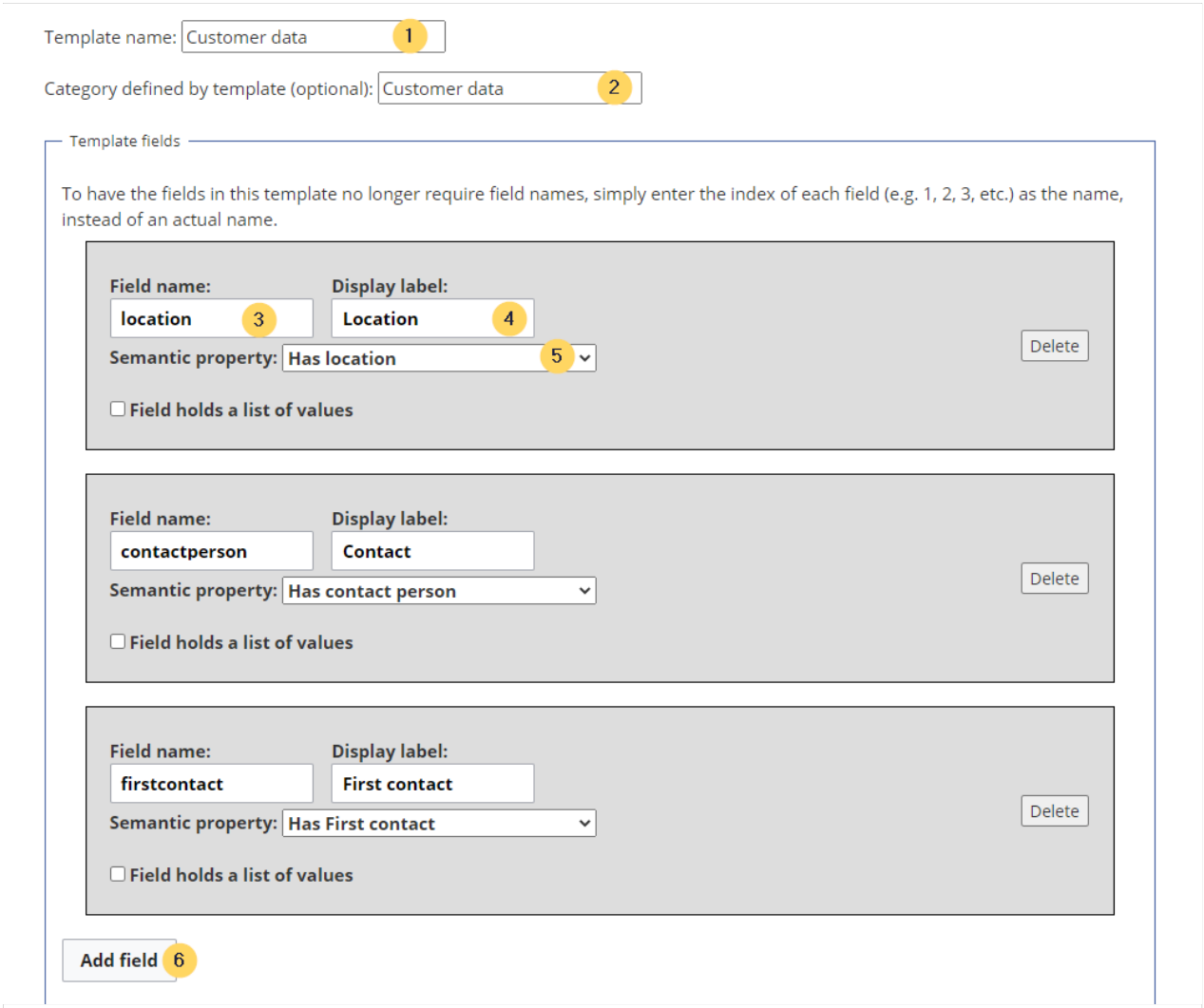
To create the template, we go to the page *Special:Create template*. On this page, we have access to the properties that we have already created.

1. Like in the previous step (creating properties), switch to the Page Forms section on the page *Special:SpecialPages*.



Create a template

2. Define the data fields that you want to use in your template. For each property that you created, you need to have a field in the template.



Create the template

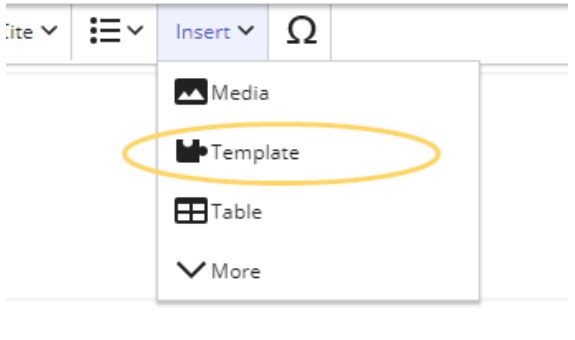
- **Template name (1):** Enter *Customer data* as name for the template.
- **Category (2):** Define a category which will be assigned to all customer pages that use this template. It makes sense to also call this category Customer data.
- **Field name (3):** Create a field name for the property Has location. This field name will later be used to connect the template with the data entry form. The name should not contain spaces.
- **Display label (4):** This is the field label tha tis shown on the customer pages. It can contain spaces.
- **Semantic property (5):** Choose the property *Has location* that you created in the previous step.
- **Add field (6):** Click Add field to create the fields for the properties *Has First contact* and *Has contact person*.

3. Click *Save page* at the end of the page *Seite speichern*.

The template is now available and can be viewed in the *Template* namespace. You do not have to edit the template any further at this point.

Including the template on a page

1. To include the template on a customer page, you simply add it from the Visual Editor *Insert* menu.



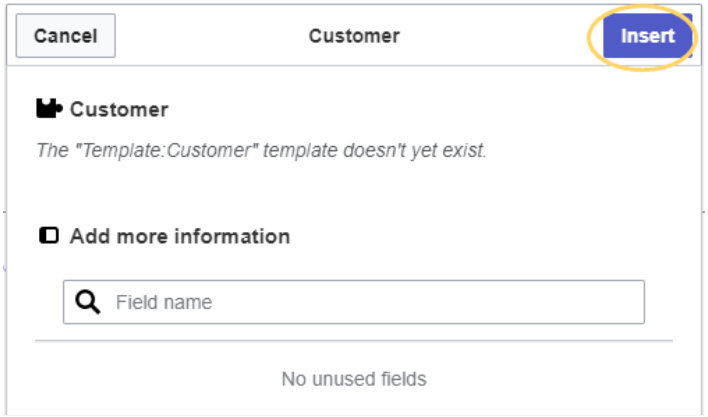
Insert the template on a page

2. Start typing the name *Customer data* and select the template (1). Then, click *Add template* (2).



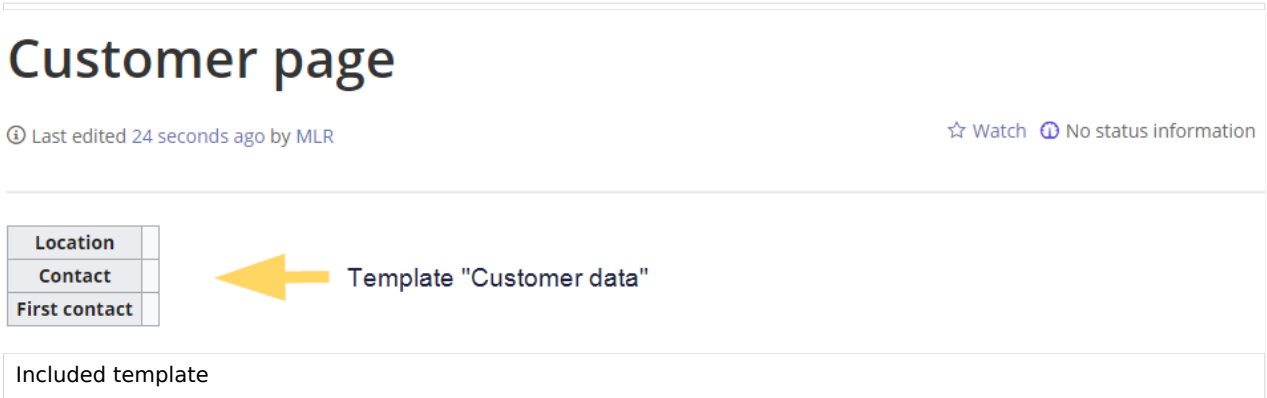
Select template

3. Click *Insert.*



Insert template

4. Save the page. The template is now included on the page.



Included template

Next step

Next, we create a form that makes it easy for users to enter the data.

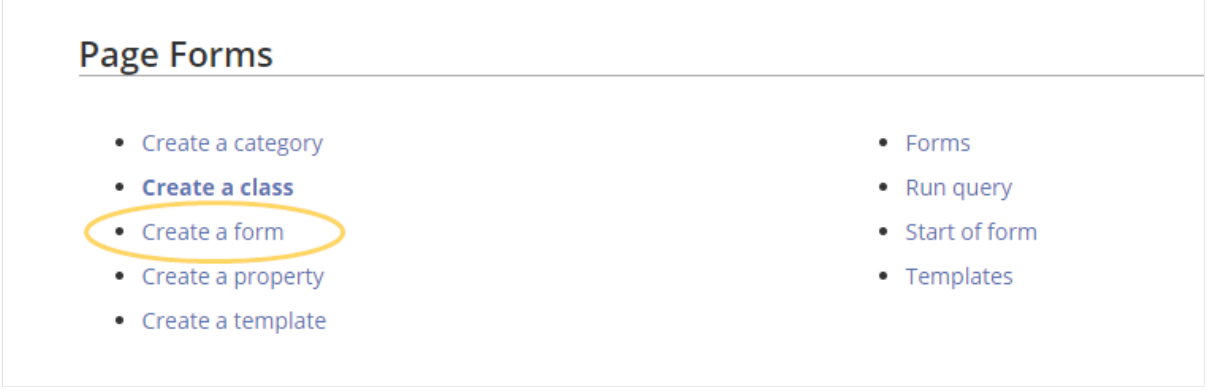
Create a form

Related info

- https://www.semantic-mediawiki.org/wiki/Help:Semantic_templates Help page for semantic templates

2.3 Create a form

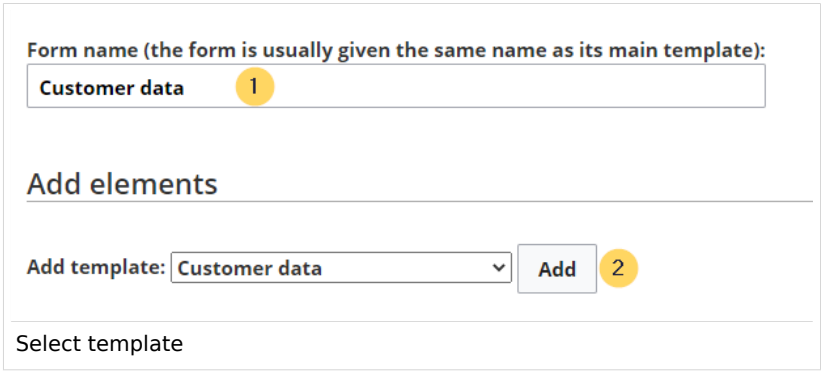
With the page *Special:CreateForm*, we now create the data entry form.



Create a form

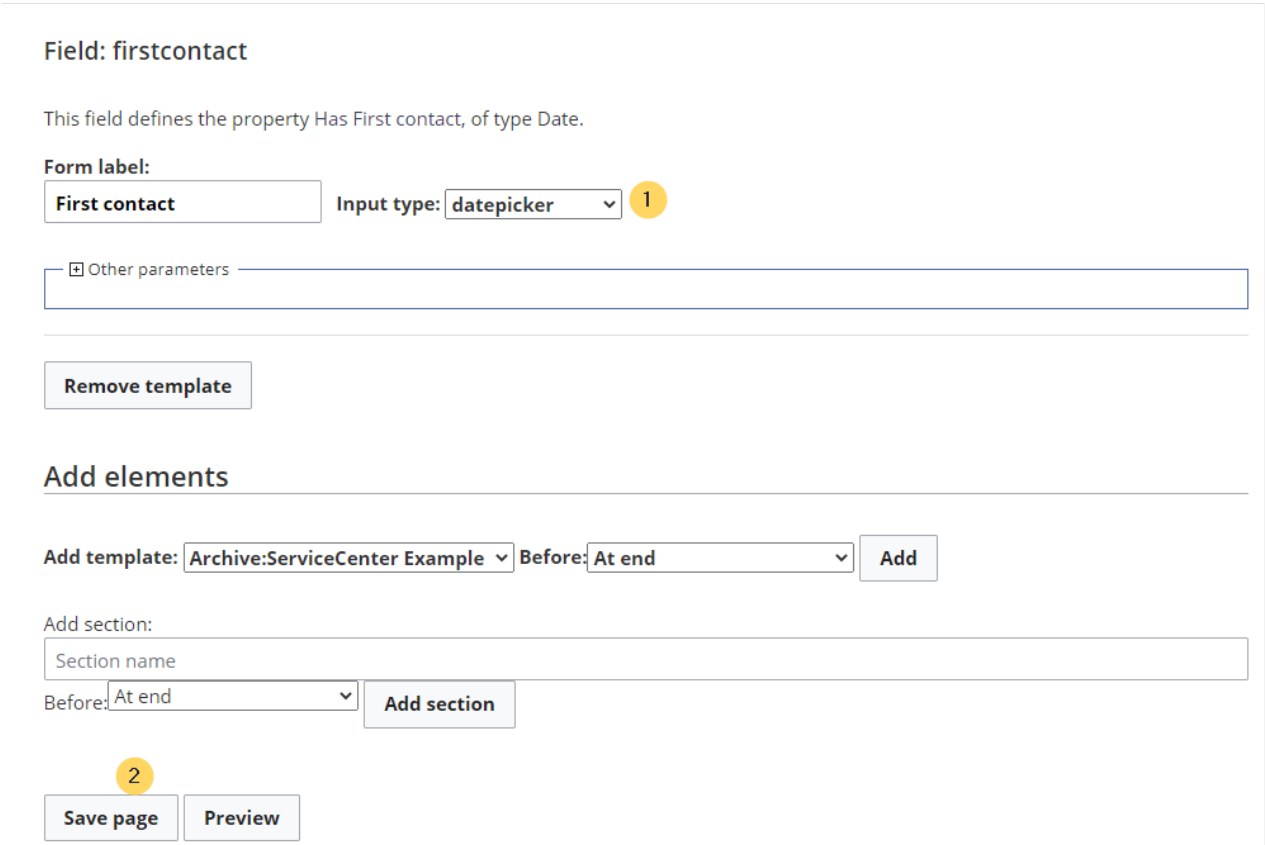
Creating the form

1. On the "Create a form" page, enter a name for your new form. Since we create it for template *Customer data*, we give the form the same name (1). Select the template Customer data from the choice menu and click *Add* (2).



Select template

2. The customer data fields from the template Customer data are automatically integrated in the form. To facilitate the data entry for the date field, change the input type of the field *firstcontact* to *datepicker.* (1) Then, click *Save page* (2).



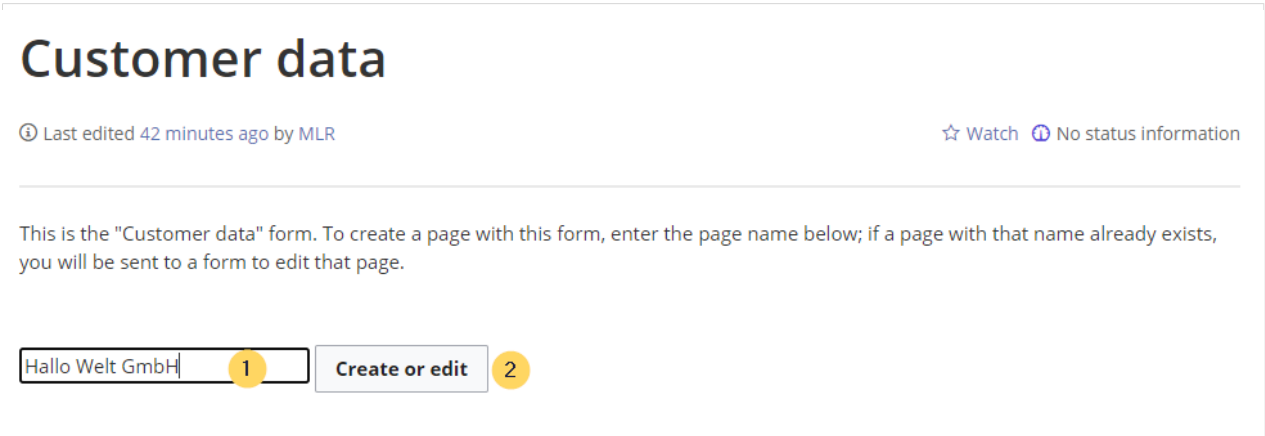
Selecting the input type

Now the form is available for data entry. It can be found in the namspace *Form.*

Using the form

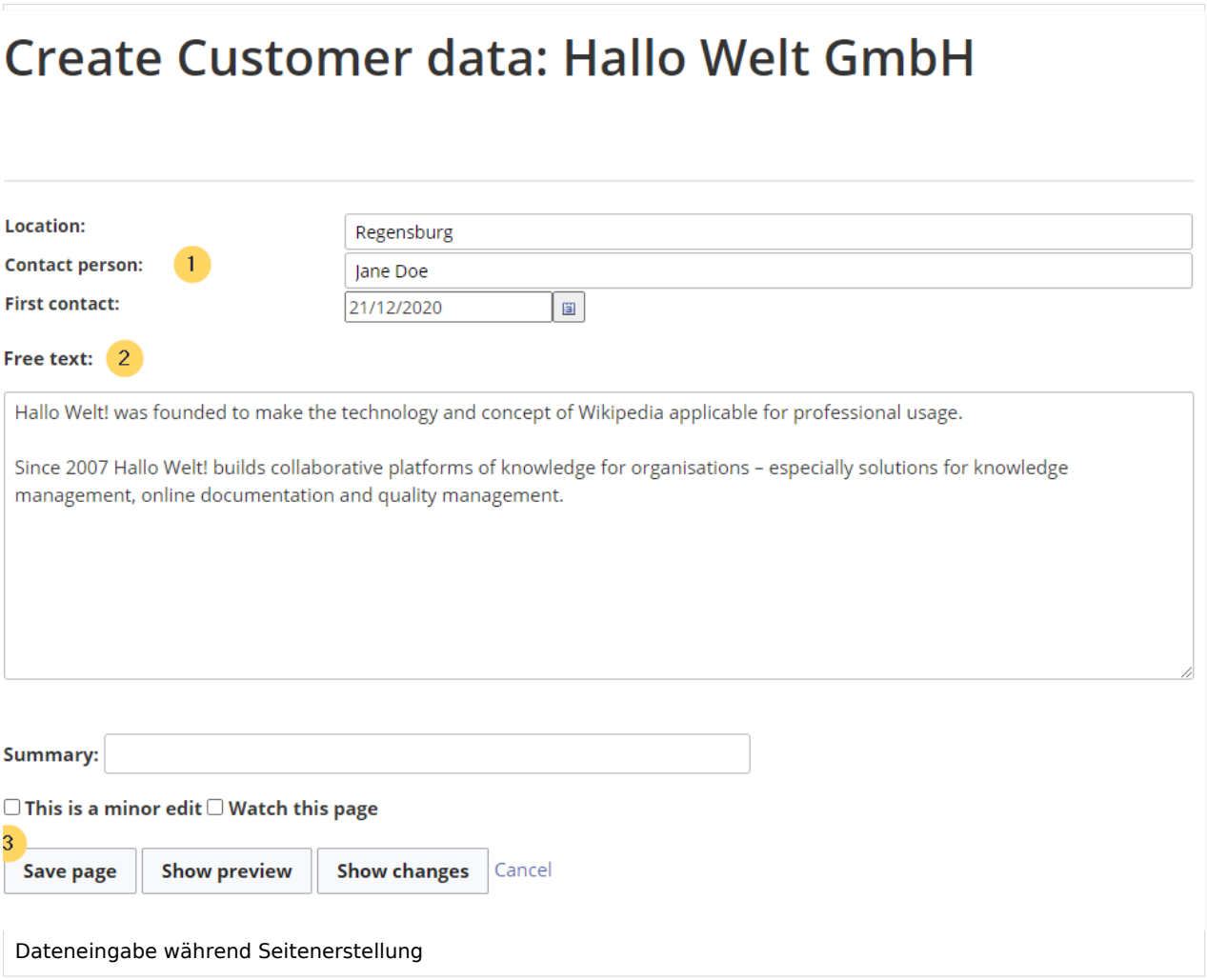
When we create a new customer page, we can now use the form page to create the page and to automatically include the customer data template in the page. The form page has the name *Form: Customer data*.

1. To create a customer page, enter a page name in the form field (1) and click *Create or edit* (2).

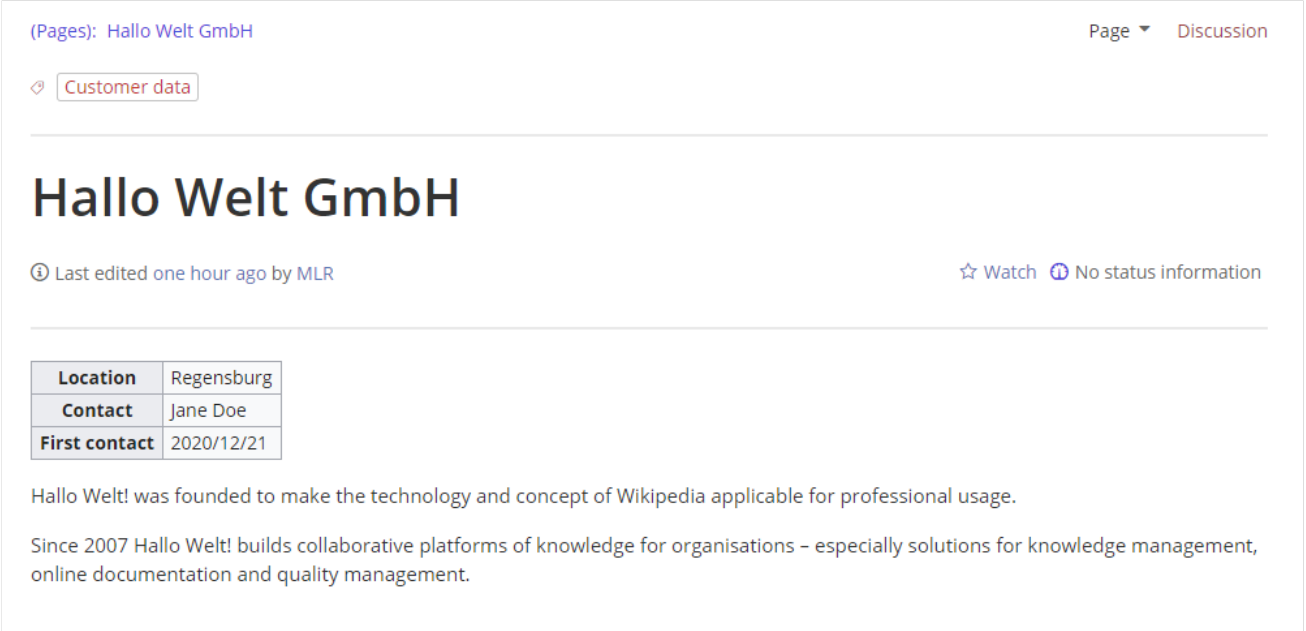


Create a customer page

2. Enter the customer data (1). You can additionally create some free text (2). Click *Save page* after editing the page (3).



Now you have created a customer page that collects customer data and shows the entered data on the page.



Customer page with semantic template

Note: You can customize the template style later. First, you should integrate the form with the customer pages.

Next step

As the next step, we add the form edit mode option to pages that include the *Customer data* template.

Edit pages in form edit mode

Related info

- https://www.mediawiki.org/wiki/Extension:Page_Forms/Quick_start_guide Help page for the extension *PageForms*

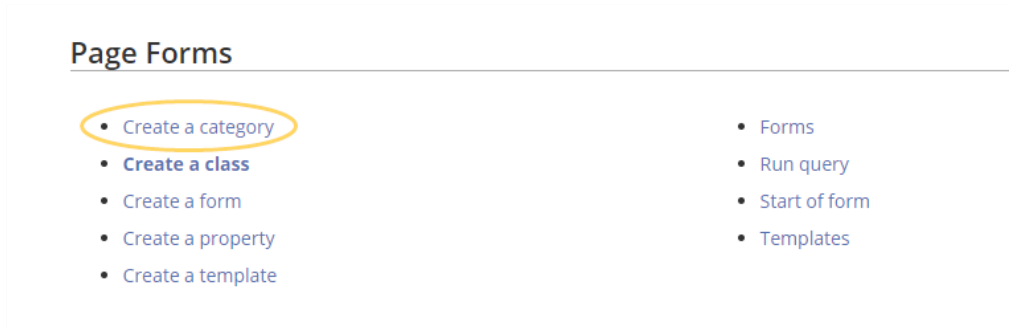
2.4 Integrate form edit mode

With the page *Special:CreateCategory,* we now connect the category *Customer data* with the form *Customer data*.

This adds a **form edit link** to all pages that use the **template** *Customer data*. Additionally, it makes it possible to list pages that include customer data throughout the wiki.

Activating the form edit mode

1. Switch again to the page *Special:Special pages*. In the section *Page Forms,* click the link *Create a category.*



Create a category

2. Enter the category name you chose in step "Create template". In our case, this is the category *Customer data* (1). Select the form *Customer data* (2). Click *Save page* (3).



Associate category and form

The category page *Customer data* is now associated with the form *Customer data*.

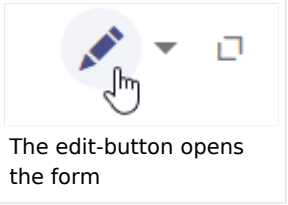
Location	Regensburg
Contact	Jane Doe
First contact	2020/12/21

Hallo Welt! was founded to make the technology and concept of Wikipedia applicable for professional usage.

Since 2007 Hallo Welt! builds collaborative platforms of knowledge for organisations – especially solutions for knowledge management, online documentation and quality management.

Category that is associated with a form

The *Customer data* category page is now linked to the *Customer data* form. As a result of this link, the edit button opens the form on the customer pages.



The edit-button opens the form

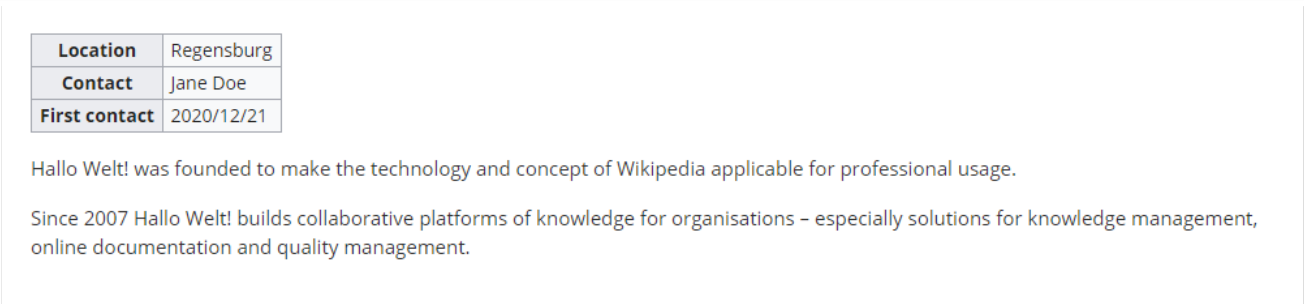
Note: If you do not want to use a category to make the "form edit" option available on a page, you can also associate the form with a namespace or simply add the form reference directly to the template. For more information see the instructions on mediawiki.org

Related info

- Styling the template
- Query the collected data
- Naming conventions for semantic elements
- Video tutorials for Semantic MediaWiki

3 Template styling

By default, a semantic template that was created with the page *Special:CreateTemplate,* is displayed in a basic table format. Each property/value pair is shown in a new table row.The table width adjusts to the table content.



Location	Regensburg
Contact	Jane Doe
First contact	2020/12/21

Hallo Welt! was founded to make the technology and concept of Wikipedia applicable for professional usage.

Since 2007 Hallo Welt! builds collaborative platforms of knowledge for organisations – especially solutions for knowledge management, online documentation and quality management.

Default table of a semantic template

To change the table layout, we can add some styling information to the template.

Adjusting the template styling

To format the table, we create a subpage to the template. It contains the custom template styles. In the following example, we create a variation of the original table with a width of 100%, left-aligned labels and more padding in the table cells.



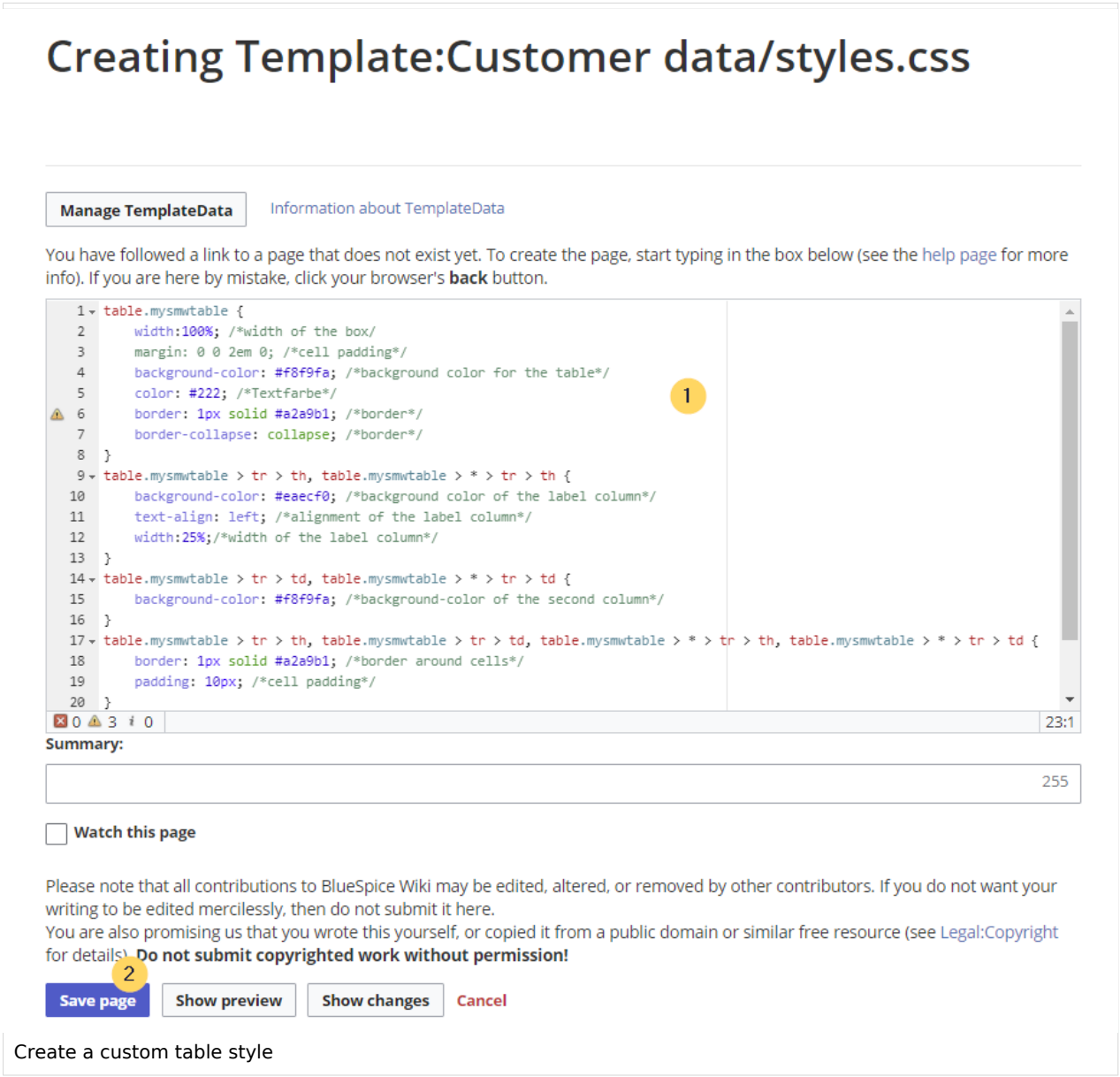
Location	Regensburg
Contact	Jane Doe
First contact	2020/12/21

Customized table styling

1. Create the subpage styles.css. For the template Customer data, you need to create the page *Template: Customer data/styles.css*.
2. To achieve the shown table styling, add the following code to the new *styles.css*.

```
table.mysmwtable {
        width:100%; /*width of the box/
        margin: 0 0 2em 0; /*cell padding*/
        background-color: #f8f9fa; /*background color for the table*/
        color: #222; /*Textfarbe*/
        border: 1px solid #a2a9b1; /*border*/
        border-collapse: collapse; /*border*/
}
table.mysmwtable > tr > th, table.mysmwtable > * > tr > th {
        background-color: #eaecf0; /*background color of the label column*/
        text-align: left; /*alignment of the label column*/
        width:25%;/*width of the label column*/
}
```

```
table.mysmwtable > tr > td, table.mysmwtable > * > tr > td {
        background-color: #f8f9fa; /*background-color of the second column*/
}
table.mysmwtable > tr > th, table.mysmwtable > tr > td, table.mysmwtable > * > tr >
th, table.mysmwtable > * > tr > td {
        border: 1px solid #a2a9b1; /*border around cells*/
        padding: 10px; /*cell padding*/
}
form.createbox table.formtable th {
        padding:10px 0; /*distance between form fields*/
}
```



Create a custom table style

Save the page.

3. Now edit the page *Template:Customer data*.
   (1) Add the reference to the page *styles.css* at the beginning of the template. You can copy the following tag:
   <templatestyles src="Customer_data/styles.css" />
   (2) Change the name of the css-class used in the table from *wikitable mysmwtable*.



Add style tag to the template

Save the changes.

Now you can see the new table styling..

As a variation, you can also define the background color of the table cells. For example, you can create white cell backgrounds. Change the following lines in your *styles.css*:

```
table.mysmwtable > tr > th, table.mysmwtable > * > tr > th {
        background-color: white; /*background color of the label column*/
}
```

sowie

```
table.smwtable > tr > td, table.smwtable > * > tr > td {
        background-color: white; /*Hintergrundfarbe der Wertespalte*/
}
```

Im CSS ändern sich also folgende Zeilen:

```
table.mysmwtable {
    width:100%; /*width of the box/
    margin: 0 0 2em 0; /*cell padding*/
    background-color: #f8f9fa; /*background color for the table*/
    color: #222; /*Textfarbe*/
    border: 1px solid #a2a9b1; /*border*/
    border-collapse: collapse; /*border*/
}
table.mysmwtable > tr > th, table.mysmwtable > * > tr > th {
    background-color: white; /*background color of the label column*/
    text-align: left; /*alignment of the label column*/
    width:25%;/*width of the label column*/
}
table.mysmwtable > tr > td, table.mysmwtable > * > tr > td {
    background-color: white; /*background-color of the second column*/
}
table.mysmwtable > tr > th, table.mysmwtable > tr > td, table.mysmwtable > * > tr > th, table.mysmwtable > * > tr > td {
    border: 1px solid #a2a9b1; /*border around cells*/
    padding: 10px; /*cell padding*/
```

White cell backgrounds

Die Zellhintergründe werden entsprechend angepasst.

Location	Regensburg
Contact	Jane Doe
First contact	2020/12/21

Hallo Welt! was founded to make the technology and concept of Wikipedia applicable for professional usage.

Since 2007 Hallo Welt! builds collaborative platforms of knowledge for organisations – especially solutions for knowledge management, online documentation and quality management.

Display of white table backgrounds

Infobox

You can also format the table as an infobox. An infobox is floating to the left or right of the text flow.

Hallo Welt! was founded to make the technology and concept of Wikipedia applicable for professional usage.

Since 2007 Hallo Welt! builds collaborative platforms of knowledge for organisations – especially solutions for knowledge management, online documentation and quality management.

Location	Regensburg
Contact	Jane Doe
First contact	2020/12/21

Infobox

1. Add the following css-code to the page Template:*Customer_data/styles.css* and save the template. Delete any previous styles for the class .mysmwtable to avoid any inheritance problems for the styles:

```
table.mysmwtable {
        float:right; /*right-aligned infobox*/
        width:300px; /*width of the box*/
        margin: 0 0 2em 2em; /*cell padding*/
        background-color: #f8f9fa; /*background color for the table*/
        color: #222; /*Textfarbe*/
        border: 1px solid #a2a9b1; /*border*/
        border-collapse: collapse; /*border*/
}
table.mysmwtable > tr > th, table.mysmwtable > * > tr > th {
        background-color: #eaecf0; /*background color of the label column*/
        text-align: left; /*alignment of the label column*/
}

table.mysmwtable > tr > th, table.mysmwtable > tr > td, table.mysmwtable > * > tr >
th, table.mysmwtable > * > tr > td {
        border: 1px solid #a2a9b1; /*border around cells*/
        padding: 0.2em 0.4em; /*cell padding*/
}
form.createbox table.formtable th {
        padding:10px 0; /*distance between form fields*/
}
```

Tip: If you need different output formats, create separate templates to accomodate the styling. The templates can then use different subpages (styles.css) for each styling.

4 Query the data

You can query and display the data that you collect in the wiki at at any time on any wiki page.

Contents

1 "Ask" queries 34
2 Adjusting the output format 34
2.1 Table formatting 34
2.2 Simple page list 35
3 Adjusting the query 35
3.1 Examples 35
4 Related info 36

"Ask" queries

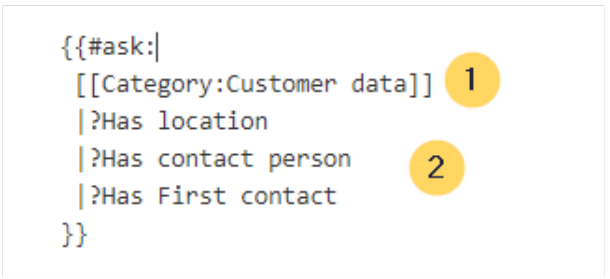
With the parser function "#ask", we can now use the semantic properties and the category association to show the collected data on our wiki pages.

As an example, we list the pages that collect customer data. You can include this "ask" query on any wiki page:

```
{{#ask:
 [[Category:Customer data]]
 |?Has location
 |?Has contact person
 |?Has First contact
}}
```

This query first defines via the category *Customer data* that only pages within this category should be included in the data query (1).

Then, using the pipe-syntax `|?` we define which property values we want to display (2).



Simple "ask" query

The default results output is in a table format. The page from which the customer data originates is automatically added as a first column (this can be customized as needed).

Adjusting the output format

The output of the data is very flexible and offers many customizing options. Let's look at some examples.

Table formatting

Output with adjusted table width of 100% (format=broadtable) and with adjusted column labels. The first column also gets a custom label (mainlabel=Customer):

Query:

```
{{#ask:
 [[Category:Customer data]]
 |?Has location = Location
 |?Has contact person = Contact
 |?Has First contact= Date of first contact
 |mainlabel = Customer
 |format= broadtable
}}
```

Simple page list

Output of pages that use the template "Customer data" in a list format:

Query:

```
{{#ask:
 [[Category:Customer data]]
 |format= ul
}}
```

Adjusting the query

In semantic queries, results can be filtered by namespaces, categories and semantic attributes.

Examples

The following query shows all results for the namespace *Manual* and the category *Customer data*. The output shows the location of the customers:

```
{{#ask:[[Manual:+]] [[Category:Customer data]]
|?Has location = Location
}}
```

Result:

The following query shows all results for the namespace *Manual* **and *for the namespace* (Main)** *and*" for the category *customer data*. All three selection criteria have to be valid for a page to be included in the results. (Note: The namespace (Main) is referenced as empty prefix (`:+`).

```
{{#ask:[[Manual:+||:+]] [[Category:Customer data]]
|?Has location = Location
|mainlabel = Customer
}}
```

Result:

The following query shows the results from namespace *Manual* and from the category *Customer data*.

```
{{#ask:[[Category:Customer data]] [[Has location::Regensburg]]
|?Has location = Location
|mainlabel = Customer
}}
```

Result:

Related info

- https://www.semantic-mediawiki.org/wiki/Help:Inline_queries Help page for semantic inline queries

5 Naming conventions

{{Messagebox|

Contents

1 Consistent naming of pages 38
2 Classification of information 38
2.1 Categories 38
2.2 Properties 38
3 Subpages 39
4 Related info 40

Consistent naming of pages

Before you create a semantic template, you first need to define which data to collect. Usually the data can be seen as a data set.

> **Example:** We want to collect customer data. All customers get their own wiki page. On each page we enter information about the company *location*, the primary *contact person* and the *date of first contact* with the company.

It makes sense to name the required pages consistently. In our example, we would create the pages *Template:Customer data*, *Form:Customer data* and *Category:Customer data.* You can even consider putting the pages in their own namespace namespace *Customer*. Manual:The concept of namespaces For the properties, you have to consdier that the property *location* could also be used outside the context of customer pages. There could be locations for partners or vendors, for example. Each context could get their own property (e.g. *Customer location*) or share the same property location and then use a different category (Customer data, Partner data, ...) on the wiki pages to distinguish them.

Classification of information

Categories

Generell unterscheiden wir bei der Klassifizierung von Seiten zwischen Kategorien und Attributen. Mit Kategorien wird die Seite an sich beschrieben. Am Beispiel Kunden kategorisieren wir jede Kundenseite mit dem Schlagwort *Kundendaten.* Die Kategorie sammelt also alle Seiten, auf der sich Kundendaten befinden.

When classifying pages, we generally differentiate between categories and properties. The page itself is described with categories. Using the example of customers, we categorize each customer page with the keyword *Customer data*. The category therefore collects all pages on which customer data is located.

Properties

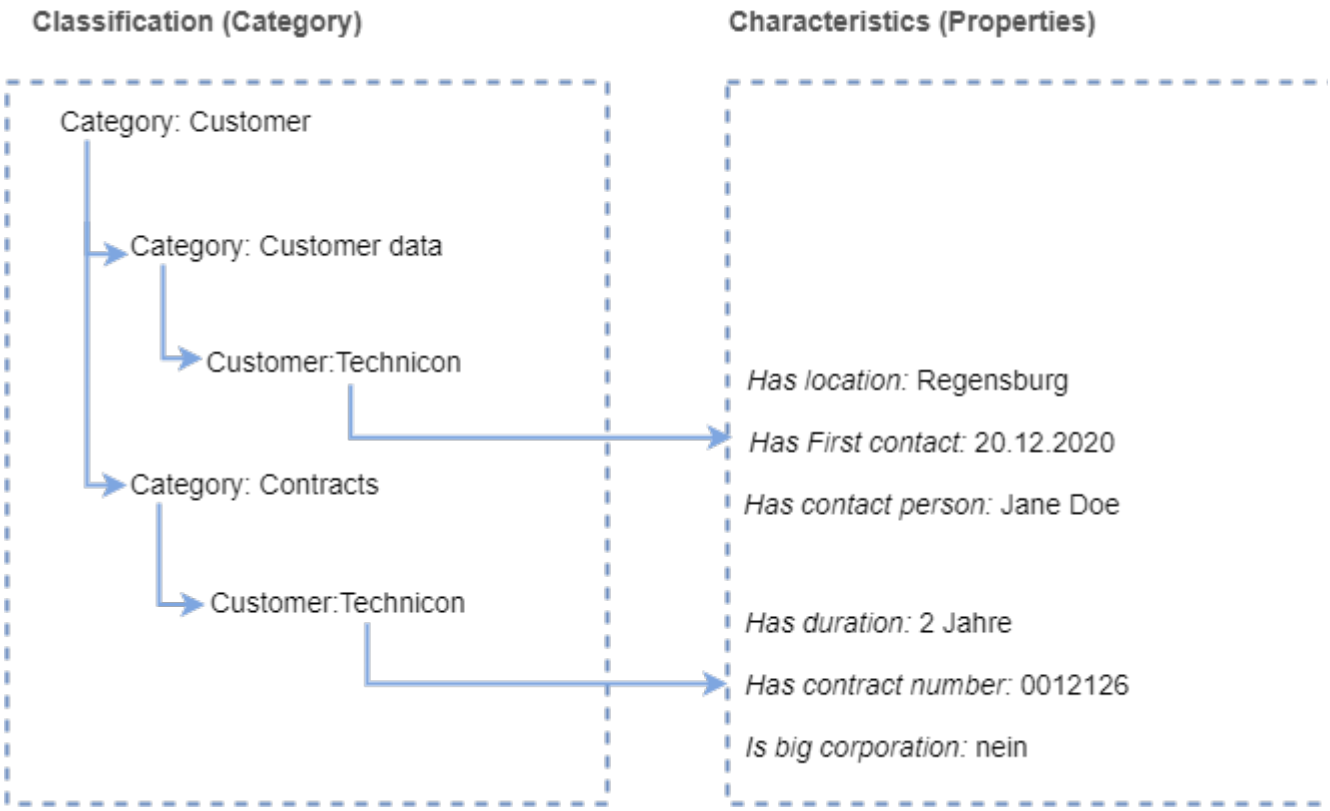
In den Kundendaten werden nun bestimmte Eigenschaften gesammelt, die jeden Kunden genauer beschreiben. Hierzu werden Attribute erstellt. Im Normalfall stehen diese Attribute in einer direkten Beziehung zur Seite selbst. Daher kann es hilfreich sein, die semantische Beziehung über das Attribut auszudrücken:

Certain properties that describe each customer more precisely are now collected as customer data. Normally, these properties are directly related to the page itself. It can therefore be helpful to express the semantic relationship directly in the property:

```
Customer Technicon has location Regensburg.
         (page)   (property)    (value)
```

Therefore, we capture the relationship to the page in the property name: *Has location*.

> **Note:** It is not required to express this relationship function of properties (as predicates). The property can also simply be names "location" if its intended use is clear.
>
> There is, however a difference between "Has location" and "Is location of". For example, the customer Technicon has the location Regensburg. The city of Regensburg, on the other hand, is the location of the customer Technicon.



Subpages

We can also work with a subpage system and work with properties like Property:Customer /Has_location, Property:Customer/ Has_First_contact, and so on. If the property "Location" is also to be used elsewhere, it is advisable to define Property: Has_location instead of Property: Customer/Has_location.

When properties can be clearly assigned to a use case or have several use cases, it makes sense to name them accordingly. For example, Property:Customer/Contract_number can be a sequential, whole number, but Property:Partner/Contract_number can contain entries such as "1.1.5" or "4.3.7".

Related info

- https://www.semantic-mediawiki.org/wiki/Help:Classification