https://wiki.micylou.com/index.php/Mediawiki_Extension_PageForms

Mediawiki Extension PageForms

Mediawiki Extension PageForms

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-04-14
Document type:	Documentation
Status:	Active
Access:	free

□

Contents

- 1 Description
- 2 Installation
  - 2.1 Download
  - 2.2 Activation
  - 2.3 Configuration
- 3 Quick start guide
  - 3.1 The easy way - Special:CreateClass
  - 3.2 Example
- 4 Example
- 5 Using Page Schemas

Description

Page Forms (known before 2016 as Semantic Forms) is an extension to MediaWiki that allows users to add, edit and query data using forms. It was originally created as an offshoot of the Semantic MediaWiki extension, to be able to edit templates that store their parameters via SMW, which is why it was originally called "Semantic Forms". However, it can now also work with the alternative Cargo extension, or with neither extension installed.

Very simply, Page Forms allows you to have forms for creating and editing pages on your wiki, as well as for querying data, all without any programming. Forms can be created and edited not just by administrators, but by users themselves.

The main components of Page Forms functionality are form definition pages, which exist in a separate namespace, 'Form:'. These are pages consisting of markup code which gets parsed when a user goes to a form. Since forms are defined strictly through these definition pages, users can themselves create and modify forms, without the need for any actual programming.

The Page Forms extension is mostly used to add and edit calls to infobox-style templates within pages. A form allows a user to populate a pre-defined set of templates, as well as page sections, for

any page; if Cargo or Semantic MediaWiki are used, the data within the templates can then be stored and queried.

Forms can also be used to edit the data in an existing page, and you can enable an "edit with form" tab to show up on any page.

Page Forms also supports autocompletion of fields, so users can easily see what the previously-entered values were for a given field. This greatly helps to avoid issues of naming ambiguity, spelling, etc.

Data in a page that doesn't fit into the form, like a free-form text description of the page's subject, isn't ignored when the page is edited with a form; rather, it is placed into a separate input box called "free text".

Page Forms also includes a wide variety of other functionality related to forms.

Below are links to the entire documentation.

Installation

Download

You can download] the Page Forms code from github, in .zip format or clone it usint the command

```
  git clone https://gerrit.wikimedia.org/r/mediawiki/extensions/PageForms.git
```

Activation

Once you have downloaded the code as a directory called PageForms in your extensions folder, you just need to add the following line to your LocalSettings.php file:

```
wfLoadExtension( 'PageForms' );
```

Configuration

There are four User rights defined for Page Forms:

1. 'createclass' is the permission to create new classes (see Extension:Page Forms/Special pages). By default it is given to all members of the 'user' group.
2. 'editrestrictedfields' is the "permission" to edit restricted fields. By default it is given to all members of the 'sysop' group.
3. 'multipageedit' is the "permission" to use the MultiPageEdit special page, which allows to edit multiple pages through a single spreadsheet-like interface. By default it is given to all members of the 'user' group.
4. 'viewedittab' is the "permission" to see the "edit with form" tab. By default it is given to everyone (i.e, '*').

You can modify the set of users who have any of these permissions. For example, to have the "edit with form" tab show up only for administrators, you could add the following, below the inclusion of Page Forms:

```
$wgGroupPermissions['*']['viewedittab'] = false;
$wgGroupPermissions['sysop']['viewedittab'] = true;
```

Quick start guide

So, you've set up all the software on your site. Now what? Now it's time to create the structures to hold, display and enable adding and editing of your data; thankfully all of this can be done simply by creating various wiki pages.

The first step should be to figure out your data structure. What types of pages will the site have? What data will be contained on each one? You can change all of this around later, but you should have a starting plan. Once you have that (ideally, written down), you can move on to creating the necessary wiki pages. You will need to create or modify pages for templates and forms, probably categories as well, and (if you're using Semantic MediaWiki) properties.

You can create all of these from scratch, although helper forms also exist to help with creating them. The right approach depends on which other extensions you're using; we will limit this to Semantic Mediawiki.

The easy way - Special:CreateClass

The page Special:CreateClass, defined by Page Forms, provides a single helper form to define everything about a data "class" in one place. If you fill out the fields and hit "submit", the page will create all the necessary property, template, form and category pages automatically. Then you can go to the form page that was created, and start entering data.

If not all the pages seem to be getting created, it could be that your job queue lacks enough memory. See the job-queue-related item here for how to add more memory.

The more comprehensive way[edit] You should take the following steps:

- Create properties. The basic building blocks of any semantic site are the connections between data, which in Semantic MediaWiki are known as properties. A property is used to specify a single piece of information about the topic of this page. Every property should be defined on your wiki, with a page in the "Property:" namespace. Create each property by going to the Special:CreateProperty page.
- Create templates. A template sets the display of the data on a page, holds the markup to turn the data into actual semantic information, and (often) defines the page as being of a certain category, and thus of a certain page type. There will generally be one template per page type, although sometimes a single page type will contain more than one template. A template can also be used for aggregation, by displaying a list of all other pages that have a certain relationship to this page (see the inline queries documentation for much more information on this). Create the template(s) by going to the Special:CreateTemplate page.
- Create forms. Now you can create forms to allow users to easily add and edit pages of various types. There should be one form per page type; the form will populate the template or templates that this page type contains. As before, there's a special page to create new forms: Special:CreateForm.
- Create categories. One template in each form should tag each article that the form creates as being part of a certain category. You should create a page for each such category, and specify a default form for it so that every article is automatically editable

with the same form that created it. The easiest way to do that is by using the Special:CreateCategory page.
- Enable links to forms. Besides creating categories that have a default form, there are other steps you should take to enable users to access the forms that you have created. These include adding links to add data in the sidebar and elsewhere, and using #formredlink in order to have red-links to nonexistent pages point to a form for creating them. These actions are all explained in the section "Linking to forms".
- Add links on sidebar and elsewhere. The sidebar (which, for English-language wikis, can be edited at the page "MediaWiki:Sidebar") should hold links to add each of the data types, and to the categories for each of these data types as well. You can also include such links on the main page and elsewhere.
- Add data. Now you can start entering data! Just go to the form page(s), and start creating pages.
- Customize. Once the structure is in place, you can customize all you want - changing the look-and-feel of the entire wiki, of the various templates, and of the forms, adding and changing fields and adding and changing inline queries.

Example

Example

Let's go through a specific example of setting up a Page Forms-based wiki from scratch, in the "more comprehensive way".</translate> <translate> Let's say that you want to create a wiki that shows all the books you have at home.</translate> <translate> We'll then go through the previous steps specified, with some parts different for Cargo vs. SMW.

1. Figure out the data structure.</translate> <translate> You want two kinds of pages on your wiki: one for books, and one for authors.</translate> <translate> Each book page should hold a title, an author name (or names), a genre (or genres), a year of publication, and a number of pages.</translate> <translate> Each author page should hold the author name, their country of origin, and a list of all the books they have written that you own.

2. Create properties.</translate>

<translate> With Cargo</translate>	**<translate> With SMW</translate>**

	<translate> For books, there is no need to create a property for the title, since the title of each page representing a book will be the title of that book.</translate> <translate> That leaves four fields:</translate> <translate> • Author(s) - using the "Create a property" page (at `Special:CreateProperty`), you create a property called **"Was written by"**, of type "<tvar|page>Page</>" (since each author will have their own page).</translate> <translate> • Genre - you create a property called **"Has genre"**, of type "<tvar|txt>Text</>", and give it allowed values such as: "Art, Cookbook, Fiction, History, Poetry, Science".</translate> <translate> • Year of publication - you create a property called **"Was published in year"**, of type "<tvar|nb>Number</>".</translate> <translate> • Number of pages - you create a property called **"Has number of pages"**, of type "<tvar|nb>Number</>".</translate> <translate> For author pages, you only need to create one more property:</translate> <translate> • Country of origin - you create a property called **"Is from country"**, of type "<tvar|txt>Text</>".</translate>
<translate> There is no equivalent step for Cargo.</translate>	

<translate> **3. Create templates.**</translate> <translate> You need to create two templates, one for books and one for authors, both at the "Create a template" page at `Special:CreateTemplate`.

First, books:</translate>

<translate> With Cargo</translate>	**<translate> With SMW</translate>**

create>Special:CreateTemplate</>, set the template name to be "Book", and set the category name to be "Books".</translate> <translate> You add to it four fields, and for each field you specify at least three values: its field name in the template source code, its display label on each page, and its type.</translate> <translate> There are a few other values you can set, but they will not be necessary for most of these fields.</translate>
<translate>
1. For the first field, set the field name to be "Authors", the label to be "Author(s)" and the type to be "<tvar|page>Page</>".</translate> <translate> You also need to check the box indicating that this field can hold a list of values (since a book can have more than one author).</translate>
<translate>
1. Click **Add field** and then, for the second field, the three values to be "Genres", "Genre(s)" and "<tvar|str>String</>".</translate> <translate> You again need to check the box indicating that this field can hold a list of values.</translate>
<translate> You should then specify the set of allowed values; this can be done as either a simple list, or a hierarchy.</translate> <translate> You could set a simple list like "Art, Cookbook, Fiction, History, Poetry, Science", and so on; or you could set a complete hierarchy, which could look like:</translate>

```
* Fiction
** Science fiction
*** Cyberpunk
** Historical fiction
```

<translate> ...and so on.</translate>
3. <translate> Click **Add field** and then, for the third field, set the values to be "Year", "Year of publication" and "Date" (a better choice than "<tvar|int>Integer</>").</translate>
4. <translate> Click **Add field** and then, for the fourth field, set them to be "Number of pages", "Number of pages" and "Integer".</translate>
5. <translate> Click the **Save page** button.</translate>

create>Special:CreateTemplate</>, set the template name to be "Book", and set the category name to be "Books".</translate> <translate> You add to it four fields, and for each field you specify three values: its field name in the template source code, its display label on each page, and the semantic property it populates.</translate>
<translate>
1. For the first field, set the field name to be "Authors", the label to be "Author(s)" and the semantic property to be "Was written by", and check the box indicating that this field can hold a list of values (since a book can have more than one author);</translate>
<translate>
1. Click **Add field** and then, for the second field, set the three values to be "Genres", "Genre(s)" and "Has genre", and also check the box for a list of values;</translate>
<translate>
1. Click **Add field** and then, for the third field, set the values to be "Year", "Year of publication" and "Was published in year";</translate>
<translate>
1. Click **Add field** and then, for the fourth field, set the values to be "Number of pages", "Number of pages" and "Has number of pages".</translate>
<translate>
1. Click the **Create** button.</translate>

<translate> Next, create the template for Authors - using the "Create a template" page, you create a template called **"Author"**, and set it to define the category "Authors".</translate>

<translate> With Cargo</translate> **<translate> With SMW</translate>**

1. <translate> Just add to this template one field, for which you set the three values to be "Country", "Country of origin" and "<tvar|page>Page</>".</translate>
2. <translate> Click the **Save page** button.</translate>

1. <translate> Just add to this template one field, for which you set the three values to be "Country", "Country of origin" and "Is from country".</translate>
2. <translate> For this template, you also fill out the "Aggregation" fields in the form, since you want each author page to list all the books the author has written.</translate> <translate> Under "Semantic property", you select "Was written by", and for "Title for list" you put "Books by this author".</translate>
3. <translate> Click the **Save page** button.</translate>

<translate> **4. Create forms.**</translate> <translate> You need to create two forms, one for books and one for authors:</translate>

- <translate> Books - using the "Create a form" page (at <tvar|create>`Special:CreateForm`</>), you make a form called **"Book"**, select the template "Book" and click the **Add** button.</translate> <translate> You could also optionally add a section for the plot of the book - to do that, add a section (not a template) called "Plot".</translate> <translate> You do not do any further customizations on this form, but just save the form once you have created it.</translate>
- <translate> Authors - using the "Create a form" page, you make a form called **"Author"**, and add to it the template "Author".</translate> <translate>

As before, you do not do any further customizations - you just save the page.

5. Create categories.</translate> <translate> You need to create two categories, one for books and one for authors; you have already specified these categories' names before:</translate>

- <translate> Books - using the "Create a category" page (at <tvar|1>`Special:CreateCategory`</>), you make a category called **"Books"**, and give it the default form "Book".</translate>
- <translate>

Authors - using the "Create a category" page, you make a category called **"Authors"**, and give it the default form "Author".

6. Enable links to forms.</translate>

<translate> With Cargo</translate>

tpl>`Template:Book`</>) find the portion of the display containing the tag:</translate>
```
{{{Authors|}}}
```
<translate> Replace this with something like:</translate>
```
{{#arraymap:{{{Authors|}}}|,|x|{{#formredlink:target=x|form=Author}}}}
```

<translate> With SMW</translate>

tpl>`Template:Book`</>) most likely has a portion that looks like</translate>
```
{{#arraymap:{{{Authors|}}}|,|x|[[Was written by::x]]}}
```
<translate> Replace this with something like</translate>
```
{{#arraymap:{{{Authors|}}}|,|x|{{#formredlink:target=x|form=Author}}{{#set:Was written by=x}}}}
```

<translate> This will make links to author pages that don't exist yet point to the form for creating those pages.

7. Add data.</translate> <translate> Using the new "Book" form (at <tvar|1>`Form:Book`</>), you create pages for some books; you check to make sure that both the form and the resulting pages look alright.</translate> <translate> Each book page will now have a red link to its author, that should point to the "Author" form for adding that author (you may need to refresh the page for the right link to show up).</translate> <translate> You can then add a few authors, either through

these links or from the form page directly, to make sure that this setup works correctly too.</translate> <translate> Each created author's page should show a list of the books you've saved that have that person as the author.

8. Add links on sidebar and elsewhere.</translate> <translate> Your structures are now all set up, and you can add links to the sidebar for easy adding and viewing, by modifying the page <tvar|bar>MediaWiki:Sidebar</> (or its equivalent if you're using a language other than English).</translate> <translate> You can add links to the "Book" and "Author" forms to make it easy to add new ones.</translate> <translate> You could also link to <tvar|drilldown>Special:Drilldown</> (if you have Cargo installed), or <tvar|BrowseData>Special:BrowseData</> (if you have the [[<tvar|1>Extension:Semantic Drilldown</>|Semantic Drilldown]] extension installed), to let users view all the data at a glance.

Using Page Schemas

The Page Schemas extension is specifically intended for the automated creation of data structures, including forms. Its big advantage over the standard PF helper forms is that it can be used to create and modify these data structure pages many times, not just create them one time. It can be used with either Cargo or Semantic MediaWiki installed, or with neither one installed. If Cargo and/or SMW are installed, the Page Schemas form will include many additional form fields in order to create the additional tags needed. To create a page schema, you need to go to the category page (whether or not it has been created yet), go to the tab action for creating that page's schema, fill out the form, then go the tab action for generating pages based on that schema. See the Page Schemas documentation for more information.

Back to top of page - Back to Welcome Page