BlueSpiceSocialBlog provides the SocialEntity type blog.

**Usage / features**

The extension BlueSpiceSocialBlog is the basis for the `Special:Blog` page. In every new wiki installation, this page is linked directly from the main navigation. Customized blog views can be added to a wiki page using the `<bs: timeline />` tag.
Technical information

Configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Default</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>bsgSocialBlogPreloadTitle</td>
<td>article that is used as preload for SocialEntities of type blog</td>
<td>str</td>
<td>Template: Blog</td>
<td>LocalSettings.php</td>
</tr>
<tr>
<td>bsgBSSocialUseBlogTeaser</td>
<td>usa a teaser-text system for SocialEntities of type blog</td>
<td>bool</td>
<td>true</td>
<td>ConfigManager</td>
</tr>
</tbody>
</table>

Rights (permissions)

This extension does not provide any additional rights.

Requirements

Requires BlueSpiceFoundation and BlueSpiceSocial.

More info

For a detailed description and usage info visit our help page.

Demo

Take a look at the page Special:Blog in our BlueSpice Demo.

GPL v3 only

Redirect to:

- GPL v3

Blog

The extension BlueSpiceSocialBlog is part of BlueSpiceSocial, the communication module of BlueSpice. In addition to the type blog, there are other types of so-called social entities, such as comment, discussion topic, attachment, or microblog.
About the blog

Basically, a blog is a list of comments, questions, or feedback from different users. The blog is often sorted chronologically downwards, in the sense of a public journal. A lively communication can arise when the contributions are further commented, recommended, linked with related articles or placed on a personal watchlist.

The blog function is ready for use on the special page Special:Blog. This page is linked directly from the main navigation in every new wiki installation. So you can start using the blog right away.

It is also possible to include the blog functionality on any wiki page. Social contributions of many types can be consolidated and displayed chronologically in a timeline and filtered and sorted as needed.

Adding the blog to a page

If you want to include a view of your blog on another page of your wiki, for example your homepage, you need to insert the necessary codeblock in the source code of your page.

To include a blog view:

1. Go to the page where you want to insert the blog or create a new page.
2. Open the page in Source editing mode.
3. **Copy the following code block** to the section in your page where you want to show the blog.

```xml
<bs:timeline>
  {
    "showentitylistmenu": true,
    "preloadtitles": {
      "blog": "Vorlage:Blog"
    },
    "preloadedentities": [{
      "type": "blog"
    }],
    "headlinemessagekey": "Community-Blog",
    "showheadline": true,
    "usemorescroll": false,
    "morelink": "Special:Blog",
    "limit": 5,
    "sort": [
      {
        "property": "timestampcreated",
        "direction": "DESC"
      }
    ],
    "lockedfilternames": [
      "type"
    ],
    "filter": [
      {
        "type": "list",
        "property": "type",
        "value": [
          "blog"
        ],
        "comparison": "ct"
      }
    ]
  }
</bs:timeline>
```

After saving the page, you will see a list of existing blog entries based on the filter and sorting criteria you provided.
The following table shows the blog elements and the necessary parameters to create the output:

<table>
<thead>
<tr>
<th>Element</th>
<th>Parameters</th>
<th>Parameter and element description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 -</td>
<td>headlinemessagekey</td>
<td>Header text</td>
</tr>
<tr>
<td>header</td>
<td>showheadline</td>
<td>Show the header text</td>
</tr>
<tr>
<td></td>
<td>showentitylistmenu</td>
<td>Show or hide the blog menu</td>
</tr>
<tr>
<td></td>
<td>showentityspawner</td>
<td>Show the menu item for creating a blog entry</td>
</tr>
<tr>
<td></td>
<td>availablesorterfields</td>
<td>Determines which data fields are included as sorting options in the sort menu by date created or by the user who created a blog entry:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;availablesorterfields&quot;: [&quot;timestampcreated&quot;, &quot;ownerid&quot;]</td>
</tr>
<tr>
<td>Element</td>
<td>Parameters</td>
<td>Parameter and element description</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>availablesorterfields</td>
<td><img src="Image" alt="List of typical sort fields" /></td>
<td>Determines which settings for sorting the user cannot change.</td>
</tr>
<tr>
<td>lockedoptionnames</td>
<td>Determines which data fields are included as filter options in the filter menu based on Creation date and Linked pages:</td>
<td></td>
</tr>
<tr>
<td>availablefilterfields</td>
<td><img src="Image" alt="Available filter fields" /></td>
<td>&quot;availablefilterfields&quot;: [&quot;timestampcreated&quot;, &quot;tags&quot;]</td>
</tr>
<tr>
<td>lockedfilternames</td>
<td>Determines which settings for filtering the user cannot change. Example: a user cannot change the filter values for <em>Linked pages</em>. The text field is greyed out:</td>
<td></td>
</tr>
<tr>
<td>persistsettings</td>
<td>Initial settings and custom sort and filter settings are included in the page url. The url can then be used to create a link to the page with the same filter and sort settings.</td>
<td></td>
</tr>
<tr>
<td>persistsettings</td>
<td><img src="Image" alt="" /></td>
<td>&quot;persistsettings&quot;: true</td>
</tr>
</tbody>
</table>

**Shows the text field for creating a new blog entry. If it is not set, the text field is not available.**

To show the text field:
<table>
<thead>
<tr>
<th>Element Description</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 - new blog entry</td>
<td>preloadentities</td>
<td>&quot;preloadedentities&quot;: [{&quot;type&quot;:&quot;blog&quot;}]</td>
</tr>
<tr>
<td>4 - existing blog entry</td>
<td>outputtypes</td>
<td>It defines the display format of the blog entry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Standard types are: List, Short, Default, Page. (see examples)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Each blog entry is a &quot;social entity&quot;.</td>
</tr>
<tr>
<td>4a - avatar</td>
<td>-</td>
<td>Shows the avatar of the user who created the blog entry or comment.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It is displayed in all output types except List.</td>
</tr>
<tr>
<td>4b - title bar</td>
<td>-</td>
<td>In the output type List, the username is not displayed in the title bar.</td>
</tr>
<tr>
<td>4c - blog entry content</td>
<td>-</td>
<td>Shows the text body of a blog entry. The output type Default collapses the first paragraph with a &quot;More&quot; link.</td>
</tr>
<tr>
<td>4d - social tags</td>
<td>-</td>
<td>Blog entries always include the social tags Comments, Recommend, Linked Page</td>
</tr>
<tr>
<td>4e - user menu</td>
<td>-</td>
<td>Users can edit or delete their own blog entries or comments at any time.</td>
</tr>
<tr>
<td>5 - new comment</td>
<td>-</td>
<td>Text field for creating a new comment.</td>
</tr>
<tr>
<td>6 - existing comment</td>
<td>-</td>
<td>Comments include the social tag Recommend.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Each comment is a &quot;social entity&quot;.</td>
</tr>
<tr>
<td>7 - &quot;more&quot; button</td>
<td>morelink</td>
<td>Link target of the &quot;more&quot; button.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For blogs, it makes sense to link to the page</td>
</tr>
<tr>
<td></td>
<td>useendlessscroll</td>
<td>More blog entries are automatically loaded after the user scrolls to the end</td>
</tr>
<tr>
<td></td>
<td>showentitylistmore</td>
<td>Shows a &quot;more&quot; button below the blog which loads more blog entries on click</td>
</tr>
</tbody>
</table>

Note: "useendlessscroll" is set to false.
**Sorting**

Most commonly, a blog shows the latest blog entries first. However, there are many different options for sorting the list of blog entries.

**Syntax**

```json
{
    "property": "timestamptouched",
    "direction": "DESC"
}
```

**Properties**

There are some common properties of a blog entry that are useful for sorting:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestampcreated</td>
<td>creation date</td>
</tr>
<tr>
<td>timestamptouched</td>
<td>last edit date</td>
</tr>
<tr>
<td>commentcount</td>
<td>number of comments</td>
</tr>
<tr>
<td>ratingcount</td>
<td>number of ratings</td>
</tr>
<tr>
<td>ownerid</td>
<td>username of the creator of a blog entry</td>
</tr>
<tr>
<td>ownerrealname</td>
<td>real name of the user</td>
</tr>
</tbody>
</table>

**Direction**

Blog entries can be sorted in descending and ascending order. Descending is the default order.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESC</td>
<td>sorts the blog entries in descending order</td>
</tr>
<tr>
<td>ASC</td>
<td>sorts the blog entries in ascending order</td>
</tr>
</tbody>
</table>

**Filtering**

Without setting the filter parameter, there will be no output on the page.

To show the blog entries of your wiki, include the following codeblock like shown in the code example above:
This filter includes all social entities which match the type blog. Since we only want to show social entities of type blog, we only include blog entries in our filter. If you want to create a social timeline rather than a blog view, you can also add other types of social entities. The following example shows a timeline view that includes blog entries, discussion entries from wiki pages, and user profile information:

```
"filter": [ {
  "type": "list",
  "property": "type",
  "value": [ "blog" ],
  "comparison": "ct"
}
]
```

### Output type examples

The output type for a blog is added using the `outputtypes` parameter. Don't forget to include a comma before and after this parameter (unless this is the last parameter in your code):

```
"outputtypes": {
  "blog": "Default"
}
```

### Default

The output type Default shows the content of a blog entry truncated after the first paragraph. To view the full text, users can click on the More link.
Short

The output type Short does not show the content of the blog entry. To view the content, a user has to click on the title of a blog entry. This loads the actual page for the selected blog entity.

Page

The output type Page shows all blog elements, including the comments, in an expanded view.
List

The output type **List** only shows the title and the timestamp.

- **Suggestion for improving search results** Created one hour and 56 minutes ago
- **Greg Rundlett über BlueSpice** Created 2 months ago

Configuration

In the **Config manager**, you can change the following settings:

- **Use teaser text for blog entries in the timeline**: When you create a blog entry, you can create a teaser text for it by clicking the cogwheel symbol in the blog entry form. If this setting is active, the teaser text will be shown on the page *Special:Timeline* instead of the first part of the content.
Blog teaser text

Related info

- Micro-Blog
- Discussions and Attachments
- Discussion topics
- Social comments
- User Profile

*Linked pages

- Reference:BlueSpiceSocial

Reference:BlueSpiceFoundation
The extension **BlueSpiceFoundation** is the interface between BlueSpice and Mediawiki. It provides basic functions for the individual BlueSpice extensions.

If you install any BlueSpice extension in a MediaWiki environment - that is outside of BlueSpice free or BlueSpice pro - the reference information for the extension will indicate if it requires BlueSpiceFoundation.

### Technical information

#### Requirements

This extension requires **ExtJSBase**.

For download info, visit [mediawiki.org](http://mediawiki.org).

### Reference: BlueSpiceSocial

**BlueSpiceSocial** is a group of extensions that add social media features to the wiki. This allows users to quickly communicate on the wiki and to express their opinions on pages more easily.
Usage / features

BlueSpiceSocial provides the architecture for all Social extensions. It contains base classes, SourceTitle namespace, SocialEntity ContentModel, notifications, etc.

This extension is required by all Social extensions. Without another Social extension (e.g., BlueSpiceSocialBlog), this extension has no effect.

The following Social extensions are available:

- SocialArticleActions
- SocialBlog
- SocialComments
- SocialMicroBlog
- SocialProfile
- SocialRating
- SocialResolve
- SocialTags
- SocialTimelineUpdate
- SocialTopics
- SocialWatch
- SocialWikiPage

Technical information

Rights (permissions)

<table>
<thead>
<tr>
<th>Permission</th>
<th>Description</th>
<th>Default role</th>
</tr>
</thead>
<tbody>
<tr>
<td>social-deleteothers</td>
<td>delete SocialEntities that have been created by other users</td>
<td>admin, maintenanceadmin</td>
</tr>
<tr>
<td>social-editothers</td>
<td>edit SocialEntities that have been created by other users</td>
<td>admin, maintenanceadmin</td>
</tr>
<tr>
<td>social-source</td>
<td>administrative access to SourceTitles in the SocialEntity namespace</td>
<td>maintenanceadmin</td>
</tr>
</tbody>
</table>

Requirements

This extension requires BlueSpiceFoundation.
More info

For detailed description and usage, visit our [help page](#).

Demo

You can test different social functions in our [BlueSpice Demo](#).