Additionals Documentation

Release 3.0.0

Alexander Meindl

Nov 24, 2020

Contents

Additionals is a Redmine plugin for customizing Redmine, providing wiki macros and act as a library/function provider for other Redmine plugins.

- Documentation: https://additionals.readthedocs.io
- Redmine.org plugin page: https://www.redmine.org/plugins/additionals
- Github: https://github.com/alphanodes/additionals

CHAPTER 1

Requirements

Redmine version	>= 4.1.0
Ruby version	>= 2.4.0
Database version	MySQL >= 5.7 or PostgreSQL >= 9.6

Note: If you use MySQL, make sure all database tables using the same storage engine (InnoDB is recommended) and character set (utf8mb4 is recommended).

Note: For more information use the official Redmine install documentation

CHAPTER 2

Installation

Install `additionals` plugin for Redmine.

```
$ cd $REDMINE_ROOT
$ git clone -b v3-stable https://github.com/AlphaNodes/additionals.git plugins/
↪additionals
$ bundle install --without development test
$ bundle exec rake redmine:plugins:migrate RAILS_ENV=production
$
$ # if you want to use smiley/emoji legacy support, you have to put emoji icons to
$ # $REDMINE_ROOT/public/images/emoji
$ # To obtain image files, run the gemoji extract command on macOS Sierra or later:
$ bundle exec gemoji extract public/images/emoji
$
$ # if you to not have macOS, you can put these files manually to $REDMINE_ROOT/
↪public/images/emoji
$ # see https://github.com/github/gemoji for more infos
```

Restart your application server (apache with passenger, nginx with passenger, unicorn, puma, etc.) and `Additionals` is ready to use.

More information about installation of Redmine plugins, you can find in the official Redmine plugin documentation.

CHAPTER 3

Update

Update `additionals` plugin for Redmine.

```
$ cd $REDMINE_ROOT/plugins/additionals
$ git pull
$ cd ../..
$ bundle install --without development test
$ bundle exec rake redmine:plugins:migrate RAILS_ENV=production
```

Restart your application server (apache with passenger, nginx with passenger, unicorn, puma, etc.) and `Additionals` is ready to use.

CHAPTER 4

Uninstall

Uninstall `additionals` plugin for Redmine.

```
$ cd $REDMINE_ROOT
$ bundle exec rake redmine:plugins:migrate NAME=additionals VERSION=0 RAILS_
↪ENV=production
$ rm -rf plugins/additionals public/plugin_assets/additionals
```

CHAPTER 5

Features

- Dashboard (Drag&Drop) Support
- Text for login page
- Global footer for all projects
- Welcome text for login page
- Global sidebar content support
- Note for new issues above issue content (e.g. for guidelines)
- PDF for wiki pages
- Wiki macros for:
  - asciinema
  - cryptocompare
  - date
  - fa
  - gihub gist
  - google_docs
  - gmap
  - group_users
  - iframe
  - issues
  - members
  - meteoblue
  - new_issue
  - projects

  - reddit
  - redmine.org issue and wiki page (redmine_issue and reminde_wiki)
  - slideshare
  - tradingview
  - twitter
  - user
  - vimeo
  - youtube
- macro list with all available macros at /help/macros
- macro button for wiki toolbar with available macros with project and permission context support
- option to remove "My page" and/or "Help" from top menu
- customize top menu items
- disable (hide) modules for projects
- open external urls in new window
- smiley/emoji legacy support
- anonymize referrer for external urls
- hide role in project memberbox
- change issue author
- spam protection on registration form
- add involved issue users as watcher automatically
- create issue on user profile
- "assign to me" link on issue
- change issue state on sidebar
- configurable issue rules
  - freeze closed issue
  - change assigned_to automatically, if issue status changes
  - assigned_to has changed, but status change is required, too

CHAPTER 6

Developer features

As Redmine does not support asset pipeline, we need to install Javascript plugins as Redmine plugins to load them globally.

If no common files are used as library plugin, every plugin has to deliver the same files. And if more than one version of a library is delivered with each Redmine plugin, there is a problem.

Therefore if developer uses this plugin for the provided libraries, everything works smoothly. Don't worry, if you only need a subset of the provided libraries. If you do not load a library, it is not used.

It provides :

- Chart.js 2.9.3
- Chart.js Plugin colorschemes 0.4.0
- Chart.js Plugin datalabels 0.7.0
- clipboardJS 2.0.6
- d3 6.2.0
- d3plus v2.0.0-alpha.29
- FontAwesome 5.15.1
- mermaid 8.8.2
- Select2 4.0.13

And a set of various Rails helper methods (see below).

It provides the following Rails helper methods :

CHAPTER 7

Libraries assets loader

```
additionals_library_load(module_name)
```

This method loads all JS and CSS files needed by the required module.

The following modules are available :

- chartjs
- chartjs_colorschemes
- chartjs_datalabels
- clipboardjs
- d3
- d3plus
- font_awesome
- mermaid
- select2

CHAPTER 8

Redmine Plugins, which are using `additionals`

- redmine_automation
- redmine_codimd
- redmine_db
- redmine_git_hosting
- redmine_hrm
- redmine_omniauth_saml
- redmine_passwords
- redmine_postgresql_search
- redmine_privacy_terms
- redmine_reporting
- redmine_sudo

If you know other plugins, which are using `additionals`, please let us know or create a PR.

CHAPTER 9

Contact and Support

I am glad about your feedback on the plugin, pull requests, issues, whatever. Feel free to contact me for any questions.

9.1 Manual

The plugin configuration takes place in the administration area by a user with administration permission. Go to *Plugins / Additionals* to open it.

The additionals plugin configuration is divided into several sections, which are described here.

9.1.1 General section

The general section tab allows you to define some special behaviours for your Redmine installation. It is divided into two areas called `Contents` and `Settings`.

Contents

The following plugin options are available to be edited by users with administration rights in the area **Contents**:

Text for login page This section is for a short information on the login page below the login mask. For example who to contact in order to get Redmine access.

Global sidebar Place your global information here, if needed. Use wiki links or available macros that can be seen by every one.

Project wide footer In case you want to put some information about your company or for your imprint. Here you can also use wiki syntax for structuring your text.

Settings

And the following options can be edited by users with administratios rights in the area **Settings**:

- Activate the option `Open external URLs` in order to open those URLs in a new window or browser tab if someone wants to visit them.
- Add "Go to top" link. If you have a lot of long pages, it is enabling users to easily return to the top of a page.
- Activate the `Legacy smileys support` if you want to use the manual smiley code in your text (e.g. `:)`). If you already use a plugin that supports Smileys this option should stay deactivated. For more info on Emoji-Browser support read http://caniemoji.com/. Have a look at the Emoji cheat sheet at https://www.webpagefx.com/tools/emoji-cheat-sheet for available Emoji-Codes.
- `Disable modules`, this feature will hide the selected modules in the project settings. Even if the module is enabled for use in the admin area it can not be selected by the project manager within the projects. Please note, if these modules already activated in existing projects, you will have to change and re-save the respective project settings first.

Note: Please restart the application server, if you make changes to the external urls settings as well as if you activate the Smileys support.

Note: You can use the following manual smiley codes: :), =), :D, =D, :'(, :(, ;), :P, :O, :/, :S, :|, :X, :*, O:), >:), B), (!), (?), (v), (x), (/) and @}->-

9.1.2 Wiki section

If you click on this tab you get to the area, where users with administration rights can customize contents and settings for your Wiki pages in Redmine. It is divided into two areas called **Contents** and **PDF Wiki settings**.

Contents

Changes you can make in the area **Contents** are:

Field	Description
Global wiki sidebar	Here you can implement useful macros like a display of your page hierarchy. But remember - only people with the correspondent rights will get a display of the content. You can also implement useful macros in this section. For example to implement some date or author macros (e.g. last_updated_at, last_updated_by)

Note: Use simple text, macros and wiki syntax for your content.

PDF Wiki settings

Changes you can make in the area **PDF Wiki settings** are:

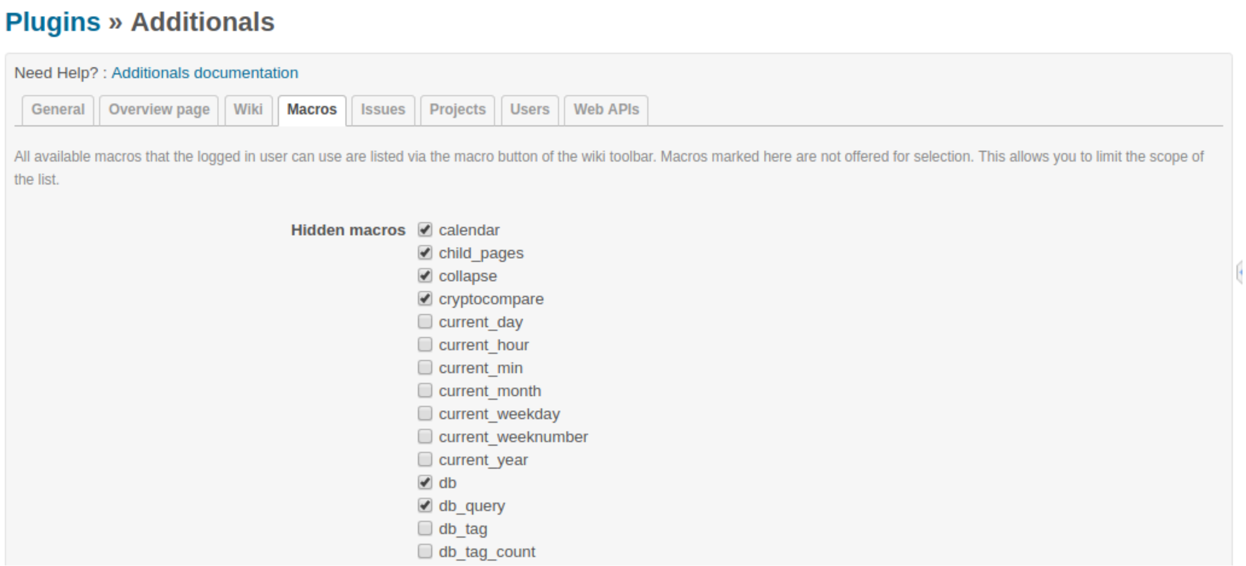
Field	Description
Remove Wiki title from PDF view	When activated the general Wiki title info in the page header of the PDF view will not be displayed.
Remove attachments from PDF view	When activated the attachments will not be displayed in the PDF view of a Wiki page.

9.1.3 Macros section

Redmine itself and Redmine plugins are sometimes equipped with additional macros for usage. Macros can be used in the Wiki of a project or in the text area of an issue. For more information on how to add macros use the Redmine help documentation.

The *Macros section* of the additionals plugin lists all available macros that the logged in user can use with the *macro button* of the wiki toolbar. If you leave them deactivated they are all available to your users for selection.

Macros marked here are not offered for selection. This allows you to limit the scope of the list for a better usability.



All available macros that the logged in user can use are listed via the macro button of the wiki toolbar. Macros marked here are not offered for selection. This allows you to limit the scope of the list.

Hidden macros

☑ calendar
☑ child_pages
☑ collapse
☑ cryptocompare
□ current_day
□ current_hour
□ current_min
□ current_month
□ current_weekday
□ current_weeknumber
□ current_year
☑ db
☑ db_query
□ db_tag
□ db_tag_count

If all macros are deactivated the *Macro button* of the Wiki toolbar will disappear.

Note: If you deactivate some macros here this does not mean the user may not implement them. All available macros of installed plugins will work even if they are not part of the macro button. The macro button is just a little helper for Redmine users with no macro experience to make it easier for them to use macros or to remember them.

9.1.4 Macro button for Wiki toolbar

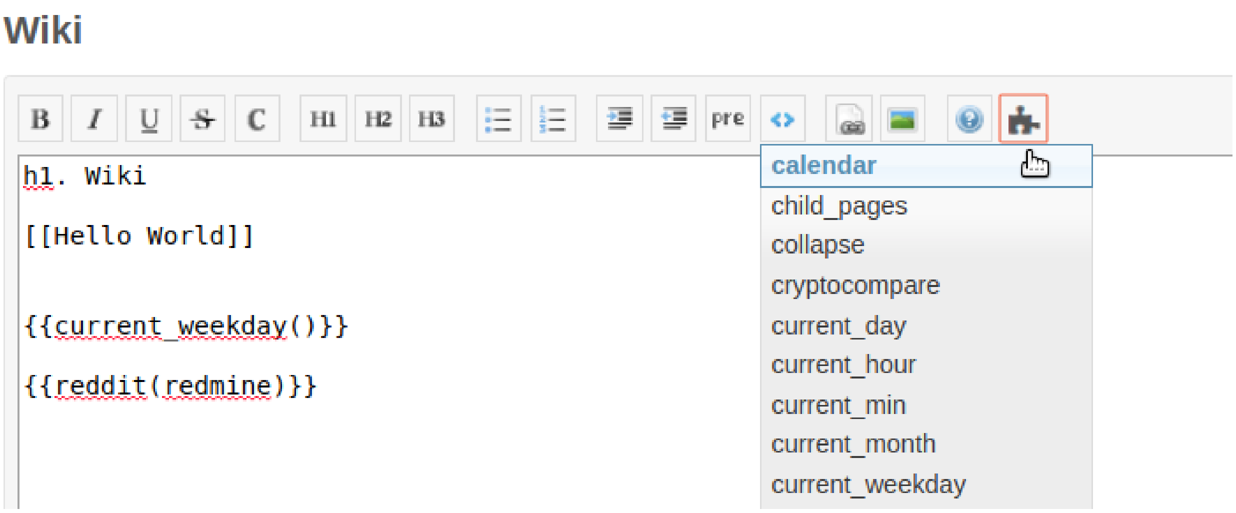
Many plugins are equipped with a number of useful macros. Unfortunately it is difficult for the normal user to find out which macros are usable without a look at the plugin documentation.

With the macro button for the Wiki toolbar we want to simplify the implementation of macros for users somehow and above all also promote. Because the use of macros belongs to the daily tools of the trade when dealing with the Wiki.

The macro button for the Wiki toolbar is acessible for every user of a project. For reasons of clarity, the list of available macros is restricted according to the following criteria.

A user can see in the macro list:

- the macros that can be used for the respective area. Macros that only work in the wiki are not available in the issue area and vice versa.
- The macros, which he / she can use due to his / her role and the associated rights in the respective project.
- only the macros of modules activated in the project. Macros for deactivated functions are hidden in the list.



Wiki

```
h1. Wiki

[[Hello World]]


{{current_weekday()}}

{{reddit(redmine)}}
```

Fig. 1: Figure 2: The Wiki toolbar macro button is a useful helper in order to select available project macros for your content.

The function is easy to use. Just click the button with the left mouse. The dropdown list shows all your available macros. Select the one you want to use. The selected macro will be pasted to the cursor position. All you have to do is adapt missing parameters (if needed). That's it.

A list of all available macros provided by the Additionals plugin is also available in this manual.

9.1.5 Issues section

Here you can define issue rules, which are used in issues of all projects.

Note for new issues Use this section if you want to place important issue notes above every new issue (edit mode). Keep it short and use a link to a common wiki page with further information.

Note: You can use wiki syntax for your text, but use it wisely. And make sure the wiki page you link to is accessible for every user.

New issue on user profile Activate this option in case you want to display the symbol link ''New issue" on a user's profile page in the top right corner in order to add a new issue for this user directly from it's user profile page.

Show ''Assign to me" on issue Activate this option if you want to display the symbol link ''Assign to me" in the issue overview page of an issue that you are able to assign it directly to yourself without *editing* the issue. The link is only shown to users who are also members in the correspondent project.

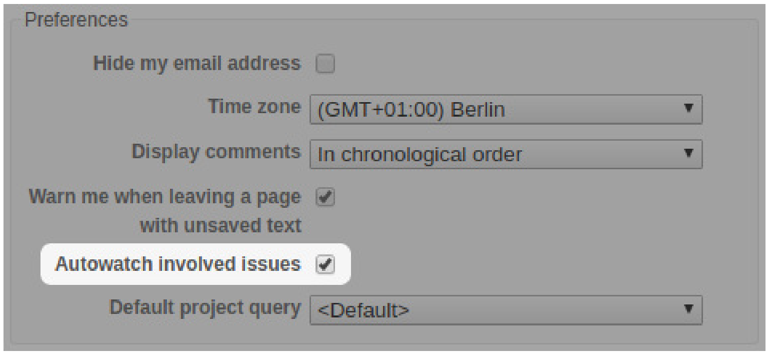
Issue status on sidebar Activate this option in case you want to display a list of available issue status options in the right sidebar of the issue view page. One click on the option changes the issue status directly without opening the *edit* mode.

Note: All options above only work for users with appropriate issue tracking rights in the administration area "Roles and permissions" (view, create, edit).

Add involved users as watcher automatically This means, everyone who is or has been involved in the issue (Assignee, Editor, Author etc.) will automatically be notified about further changes. At the same time the user finds an additional option in his user account edit mode called `Autowatch involved issues` (see figure 1). Deactivate this option if you don't want to be notified.

Disallow editing of closed issues (Freeze). This option should be activated if already closed issues should no longer be edited and commented.

If "Assignee" is unchanged and the issue status changed from x to y, than the author is assigned to the issue. Issues should be automatically assigned to the author, if the status changes to "Approval".



Preferences

Hide my email address □

Time zone (GMT+01:00) Berlin

Display comments In chronological order

Warn me when leaving a page with unsaved text ☑

Autowatch involved issues ☑

Default project query <Default>

Fig. 2: Figure 1: Deactivate this option in your account in case you don't want to be notified even if the admin activated it.

Note: Use Case for this option is that issues should be automatically assigned to author, if the status changes to *Approval*.

Current issue status x is only allowed if *Assignee* is the current user.

Note: Is this function used in combination with the setting option *Issue status on sidebar* then the current user will automatically be assigned to the issue while changing the issue status. Use Case here: Users are only allowed to change the status to *In Progress* if they are the person who is actually working on the issue right now.

If ''Assigned to" is not assigned to a user and the new issue status is x then the issue is auto assigned to the first group with users of the pre-defined role.

Note: Use Case: The issue author does not know whom the issue should be assigned to. Or he is unsure who will be responsible for solving the task. In that case the issue for example with the status "To Do" is automatically assigned to the first group, which does contain a user of the pre-selected project manager role. Imagine you have a group called "Support", and there are users assigend to the "Manager" role, this support group will be automatically adressed to solve the issue when the issue author saves it.

Time log for issues required.

Note: For each issue of the selected trackers, a time logging is necessary if the issue is to receive one of the defined status properties. The time logging is not required if there does no rights exist for the creation of time logging or if the user has the authorization *Time logging not required.*

Please note, that a user with administration rights will always be able to do all those things a normal user is not allowed to.

9.1.6 Menu section

First of all: This section is only visible in case the *Redmine HRM Plugin* is not installed. If you are also using the *Redmine HRM Plugin* this section disappears because the functionality is also an important part of ‘’HRM”. Otherwise, you can define here new top menu items and change some standard settings on the menu behaviour.

Note: Please restart the application server, if you remove a menu item or change permissions as well as changing the custom help url.

This area offers you the possibility to add up to 5 additional menu items to your Redmine main menu. The following input fields are available for each entry:

Field	Description
Name	Enter the name of the menu item.
URL	Enter an URL starting with `http://`
Title	Enter an title (optional)
Permissions	Select one ore more existing roles to which the menu item will be displayed. Only members of selected roles will be displayed in this list.

Settings

In the **Settings** area of the menu tab there are the following functions available.

- Enter a `Custom help URL` instead of linking to the help on Redmine.org. Make sure you restart your application server after your changes.
- Remove `My Page` from top menu in order you don’t want your users to use this page.

9.1.7 Users section

In case you want to use a SPAM protection for the user registration page you can activate a spam protection function in this area which is based on the honeypot strategy to provide a better user experience. We use the simple and flexible SPAM protection solution for Rails applications called `invisible_captcha` for it.

In order to activate the checkbox `SPAM protection for registration` you must have enabled the registration process in your Redmine settings (Authentication - Self-registration). Otherwise it is not possible to use this function.

Note: CAPTCHA is short for `Completely Automated Public Turing test to tell Computers and Humans Apart`. It is a type of test to determine whether or not the user is human. The honeypot strategy is an alternative and fairly simple. It puts a field onto your form that humans won’t fill out because they don’t see it. Most spam bots search for forms, fill out every field and submit it. If the honeypot captcha field is filled out you know for sure it is a spam submission.

9.1.8 Web APIs section

In case you want to use the Gmap Macro you have to implement your Google API Key into the field `Google Maps Embed API Key` first. After this you can use the Macro everywhere inside Redmine (Wiki, Issues - everywhere you use wiki syntax).

9.1.9 Help menu

We have implemented a help symbol in the global top menu of the header section that opens the new help menu.

There you find useful manual links to various Redmine related topics which will automatically be implemented, when the plugin has been installed. The menu is divided into two parts.

- There is a menu only for Redmine users without administration rights.
- There is a menu extension for Redmine users with administration rights.

The following menu items are currently implemented if a plugin is installed, that supports this additionals-function:

- FontAwesome icons
- Redmine macros (for more information see: Macros)
- Redmine guide
- Redmine changelog
- Redmine Security Advisories
- Redmine Upgrade

Known external plugins that have a user manual and support this feature are currently:

- additionals
- redmine_automation
- redmine_privacy_terms
- redmine_db
- redmine_passwords
- redmine_hrm
- redmine_reporting

9.1.10 Additionals permissions

The following role permissions are provided by the plugin and must be configured in the administration area `Roles and permissions` for the plugin functions to make sure it's working properly.

According to the selected Role you can activate / deactivate the following option:

Hide This option hides `members` of the selected role in the member box of each project overview page. Project members of other roles will still be listed. For example: You don't want others to catch a glimpse of your development team (Role: Development). So activate the role "Developer" to hide the list of your team members assigned to this role in this box. This function is also useful if your member box is too crowded.

Note: This setting is also taken into account in all related query lists (e.g. issues, spent time etc.). So that the members with this role are also hidden there. **Important:** If the user has in one project a role, that allows him / her to view a specific "hidden role" this rule is not taken into account anymore.

According to the selected Role you can activate / deactivate the following permissions:

Show hidden roles Section Project. In case you have hidden roles in a project that should not be displayed in the member box for example you can give to some special user roles the permission to display the members however. *Important:* If someone in a project has a role that has this right, then this user sees all users everywhere even if he or she is user of another project with different roles.

Save dashboards Section Project. Allow this user role to add, edit or delete dashboards of their own. So this users can add dashboards for every purpose they want and switch between them.

Set system dashboards Section Project. This permission requires *Save dashboards* permission. If you activate it, you allow users with this permission to change the system wide dashboard. Those user may create dashboards and make them system default. They get additional options: *Visible*, *System default* and *Always expose*.

Share dashboards Section Project. Allow user roles with this permission to share dashboards with other users. This means if someone else assigns you a dashboard via *Author* selection, you are allowed to edit those dashboards as long as you are the assigned *Author* and also have the permission to *Save dashboards*.

Edit issue author Section Issue tracking. This permission will always record any changes made to the issue author. You can change the author only in the issue edit mode. This function replaces the use of external plugins (e.g. change_author)

Edit closed issues Section Issue tracking. Set this option to those roles you do not want to edit closed issues. Normally a closed issue should not be edited anymore.

Set author of new issues Section Issue tracking. This permission should be set carefully, because in case you allow this, there is no history entry set for this. You will never know if the author has been originally someone else. Normally you don’t want this.

Log time to closed issues Section Time tracking. Our plugin does not allow time logs to closed issues. In case you still want to allow your members to log time to closed issues, you need to change the permission here.

9.2 Dashboards

The additionals plugin version 3.0.0 comes with new dashboard support (Drag & Drop). The new *Dashboard* feature of the additionals plugin provides customizable areas to display and organize content from Redmine that support the functionality.

Specifically, you can customize the traditional Redmine main page, as well as the project overview pages according to your needs using the available dashboard blocks. The goal is to give the viewer an immediate overview of the issues or to display the metrics that are important to you.

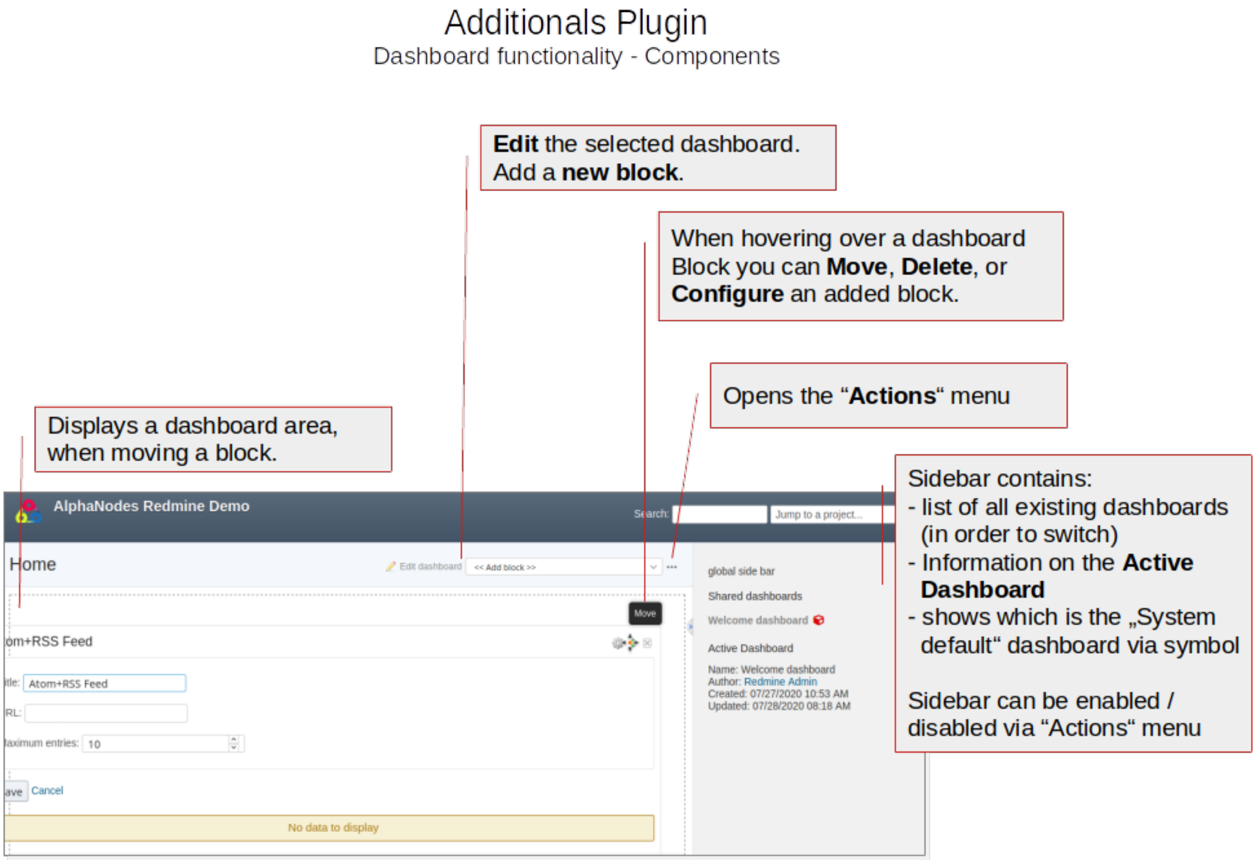
The dashboard configuration takes place directly in Redmine:

- On the “Home” page
- On the Project overview page
- If other plugins are installed and support the dashboard functionality there might be also other areas. For example:
  - Redmine HRM Plugin: HRM overview page

Dashboard examples

With the *Dashboard* of the *additionals* plugin for Redmine you can for example:

- design the Redmine main page or project overview page so that the information relevant to you is displayed there at a glance.
- create your own (private) dashboards independent of the predefined default dashboard and switch between them as needed.
- create dashboards for other users or specific roles. As well as share your own dashboards publicly and make them available to your team.



Additionals Plugin

Dashboard functionality - Components

Edit the selected dashboard.
Add a **new block**.

When hovering over a dashboard Block you can **Move**, **Delete**, or **Configure** an added block.

Opens the “**Actions**“ menu

Displays a dashboard area, when moving a block.

Sidebar contains:
- list of all existing dashboards (in order to switch)
- Information on the **Active Dashboard**
- shows which is the „System default“ dashboard via symbol

Sidebar can be enabled / disabled via “Actions“ menu

Fig. 3: Figure shows the main areas of the dashboard function

Note: When using the dashboard function make sure you have already configured the *additionals permissions* the way you need them.

Dashboard permissions

If you want to use the dashboard functionality you need to configure the permissions for the respective user roles, which are:

- Save dashboards
- Set system dashboards
- Share dashboards

9.2.1 Dashboard: Home

When accessing Redmine you probably get redirected to the "Home" page of the system. Users with appropriate permissions can modify the existing dashboard or add new dashboards by using the respective action links in the head section of the content area.

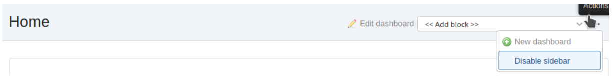


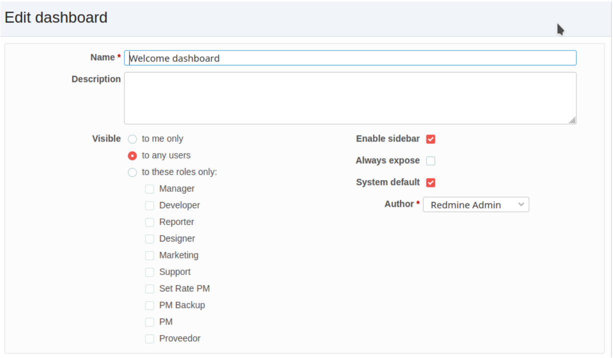
Fig. 4: Figure shows the Redmine *Home* page and where to find the dashboard actions

What you can do here is:

- use the "Edit dashboard" link for modifications
- add a "New dashboard"
- Enable sidebar or Disable sidebar
- Switch to other existing dashboards
- Add a dashboard block
- Move, Delete or configure added dashboard blocks

Edit Dashboard

Users with appropriate permission can edit an existing dashboard by clicking the "Edit dashboard" link to open the edit window.



Edit dashboard

Name * Welcome dashboard

Description

Visible
○ to me only
● to any users
○ to these roles only:
□ Manager
□ Developer
□ Reporter
□ Designer
□ Marketing
□ Support
□ Set Rate PM
□ PM Backup
□ PM
□ Proveedor

Enable sidebar ☑

Always expose □

System default ☑

Author * Redmine Admin

There you can make changes to the following fields:

Name The standard dashboard after the plugin installation is called "Welcome Dashboard". Modify it according to your needs.

Description The dashboard description will be displayed in the sidebar next to the dashboard as soon as a dashboard has been been selected to be active.

Visible There are various visibility settings for a dashboard like "to me only", "to any users", "to these roles only". If you want to mak a dashboard publicly available to all other users you have to choose the option "to any users". For a specific role choose the respective role instead.

Enable sidebar The dashboard sidebar contains some useful information for the user. Since it is sometimes disturbing, it is hidden. To prevent this you can select this option.

System default If you want to make your project dashboard system default, activate this option. So every user will have to work with it for the selected project.

Always expose If you want to make the system default dashboard name visible to the users in the head section of the dashboard page, you can activate this option. Then also the dashboard information in the sidebar is displayed.

Author You can change the dashboard author in case it is necessary. This is sometimes necessary, if you create a dasboard for someone else but want this person to be able to edit it afterwards. Or if a user will no longer be active in your Redmine system. *Important:* Only the author of a dashboard is able to edit or remove it. If the author is no longer an active user in your system, only a user with administration rights is allowed to modify the dashboard.

Hint: In general, dashboards can only be edited by a single author. Not by a group. Users with administrator rights can change the author in an emergency.

Note: The selection of the Author depends on the set permission for a role in the administration area. Only those user roles with the minimum permission **Save dashboards** or users with administration rights are selectable.

New Dashboard

Users with appropriate permission can add a new dashboard by clicking the "New dashboard" link in the "actions menu" to open the "New dashboard" window to fill out the following fields.

Name Assign a meaningful and unique name. The dashboard name will be displayed in the sidebar for later selection. If the sidebar has been disabled, you can select the dashboard from the "Actions" menu. The name must be unique.

Description The dashboard description will be displayed in the sidebar next to the dashboard as soon as a dashboard has been been selected to be active.

Visible There are various visibility settings for a dashboard like "to me only", "to any users", "to these roles only". If you want to mak a dashboard publicly available to all other users you have to choose the option "to any users". For a specific role choose the respective role instead.

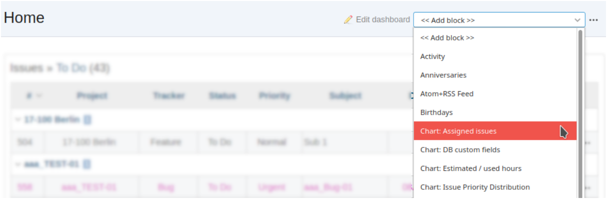
Enable sidebar The dashboard sidebar contains some useful information for the user. Since it is sometimes disturbing, it is hidden. To prevent this you can choose this option.

System default If you want to make your project dashboard system default, activate this option. So every user will have to work with it for the selected project.

Always expose If you want to make the system default dashboard name visible to the users in the head section of the dashboard page, you can activate this option. Then also the dashboard information in the sidebar is displayed.

Add Dashboard blocks

You can fill existing dashboards with content by using the select box “Add block”.

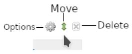


The blocks that are available here depend on the functions you are using and the plugins you have on your system. They need to support the dashboard functionality of the additionals plugin. Which is not hard to do so, because it’s easy to implement for plugin developers.

1. In order to add a new dashboard block, just select the respective option from the selection box. The block will be added right away.

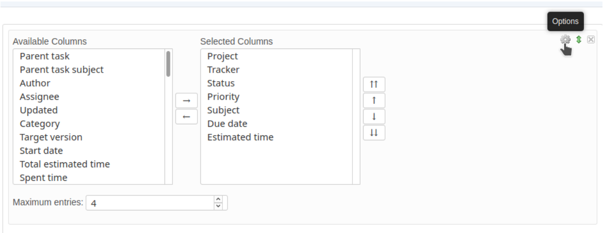
Note: The selection displays only blocks, that are allowed to be displayed on the Redmine Home page. Others are not available for selection. Currently the following plugins have additionals dashboard support implemented: DB, Passwords, Reporting, HRM, Automation, additionals

2. You probably need to position the block. In that case hover your mouse over the right block corner and grap the “Move” icon. Now position it somewhere else. If you need to remove it again use the “Delete” icon.



Note: Be careful with the deletion option. If you delete a block it will be gone right away.

3. Some blocks can be configured. In that case hover your mouse over the right block corner and click the “Options” icon. The configuration option appears and you can make your changes. Done.



Available Columns
Parent task
Parent task subject
Author
Assignee
Updated
Category
Target version
Start date
Total estimated time
Spent time

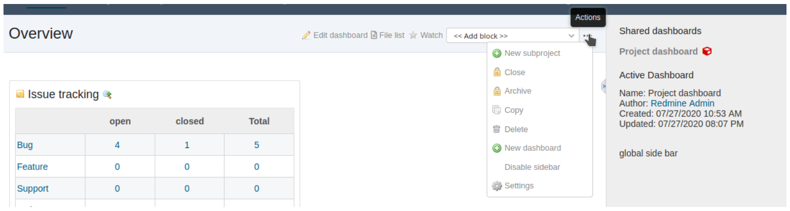
Selected Columns
Project
Tracker
Status
Priority
Subject
Due date
Estimated time

Maximum entries: 4

Note: Not every block is configurable and the block configuration may differ depending on the selection you have made. Some blocks may allow to change the column settings and other only the maximum entries, for example.

9.2.2 Dashboard: Project overview

The project overview page is also supporting the new Dashboard function. Users with appropriate permissions can modify the existing dashboard or add new dashboards by using the respective action links in the head section of the content area.



Issue tracking

	open	closed	Total
Bug	4	1	5
Feature	0	0	0
Support	0	0	0

Shared dashboards

Project dashboard

Active Dashboard

Name: Project dashboard
Author: Redmine Admin
Created: 07/27/2020 10:53 AM
Updated: 07/27/2020 08:07 PM

global side bar

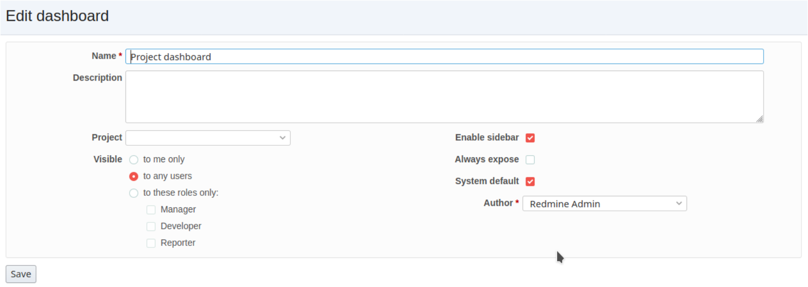
Fig. 5: Figure shows the project overview page and where to find the dashboard actions

What you can do here is:

- use the “Edit dashboard” link for modifications
- add a “New dashboard”
- Enable sidebar or Disable sidebar
- Switch to other existing dashboards
- Add a dashboard block
- Move, Delete or configure added dashboard blocks

Edit project dashboard

Users with appropriate permission can edit an existing dashboard by clicking the “Edit dashboard” link to open the edit window. What fields are displayed also depends on the permission the role your user has.



Edit dashboard

Name * Project dashboard

Description

Project

Visible ○ to me only
● to any users
○ to these roles only:
□ Manager
□ Developer
□ Reporter

Enable sidebar ☑

Always expose □

System default ☑

Author * Redmine Admin

There you can make changes to the following fields:

Name The standard dashboard after the plugin installation is called “Project Dashboard”. Modify it according to your needs. The name must be unique.

Description The dashboard description will be displayed in the sidebar next to the dashboard as soon as a dashboard has been been selected to be active.

Project Here you can select the project this dashboard should be assigned to in case you decide to make it available to special project, only. It is for all projects, if you do not select a project.

Visible There are various visibility settings for a dashboard like “to me only”, “to any users”, “to these roles only”. If you want to mak a dashboard publicly available to all other users you have to choose the option “to any users”. For a specific role choose the respective role instead.

Enable sidebar The dashboard sidebar contains some useful information for the user. Since it is sometimes disturbing, it is hidden. To prevent this you can select this option.

System default If you want to make your project dashboard system default, activate this option. So every user will have to work with it for the selected project.

Always expose If you want to make the system default dashboard name visible to the users in the head section of the dashboard page, you can activate this option. Then also the dashboard information in the sidebar is displayed.

Author You can change the dashboard author in case it is necessary. This is sometimes necessary, if you create a dasboard for someone else but want this person to be able to edit it afterwards. Or if a user will no longer be active in your Redmine system. *Important:* Only the author of a dashboard is able to edit or remove it (required permissions assumed). If the author is no longer an active user in your system, only a user with administration rights is allowed to modify the dashboard.

Hint: In general, dashboards can only be edited by a single author. Not by a group. Users with administrator rights can change the author in an emergency.

Note: The selection of the Author depends on the set permission for a role in the administration area. Only those user roles with the minimum permission **Save dashboards** or users with administration rights are selectable.

Note: Usually the system default "Project dashboard" looks like the standard project page of Redmine. It contains the same areas as "blocks". These can be repositioned or partially hidden. It is up to you if you want to change it. If not, just leave it the way they are.

New project dashboard

Users with appropriate permission can add a new project dashboard by clicking the "New dashboard" link in the "actions menu" to open the "New dashboard" window in order to fill out the following fields. What fields are displayed also depends on the permission the role your user has.

Name Assign a meaningful and unique name. The dashboard name will be displayed in the sidebar for later selection. If the sidebar has been disabled, you can still select the dashboard from the "Actions" menu. It will then be listed there.

Description The dashboard description will be displayed in the sidebar next to the dashboard as soon as a dashboard has been been selected to be active.

Project Select a project for which the new dashboard is planned. If no project is selected here, it will be available to all projects.

Visible There are various visibility settings for a dashboard like "to me only", "to any users", "to these roles only". If you want to mak a dashboard publicly available to all other users you have to choose the option "to any users". For a specific role choose the respective role instead.

Enable sidebar The dashboard sidebar contains some useful information for the user. Since it is sometimes disturbing, it is hidden. To prevent this you can choose this option.

System default If you want to make your project dashboard system default, activate this option. So every user will have to work with it for the selected project.

Always expose If you want to make the system default dashboard name visible to the users in the head section of the dashboard page, you can activate this option. Then also the dashboard information in the sidebar is displayed.

Add project dashboard blocks

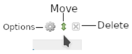
You can fill existing project dashboards with content by using the select box "Add block".

The blocks that are available here depend on the functions you are using and the projects. They need to support the dashboard functionality of the additionals plugin. Which is not hard to do so, because it’s easy to implement for plugin developers.

1. In order to add a new project dashboard block, just select the respective option from the selection box. The block will be added right away.

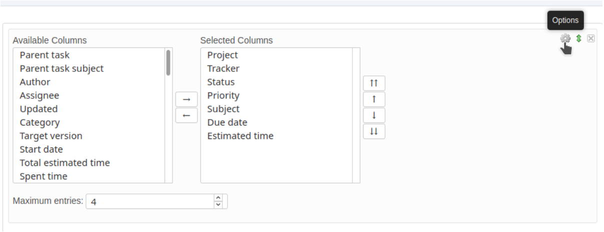
Note: The selection displays only blocks, that are allowed to be displayed on the Redmine project area. Others are not available for selection. Currently the following plugins have additionals dashboard support implemented: DB, Passwords, Reporting, HRM, Automation, additionals

2. You probably need to position the block. In that case hover your mouse over the right block corner and grap the “Move” icon. Now position it somewhere else. If you need to remove it again use the “Delete” icon.



Note: Be careful with the deletion option. If you delete a block it will be gone right away.

3. Some blocks can be configured. In that case hover your mouse over the right block corner and click the “Options” icon. The configuration option appears and you can make your changes. Done.



Available Columns
Parent task
Parent task subject
Author
Assignee
Updated
Category
Target version
Start date
Total estimated time
Spent time

Selected Columns
Project
Tracker
Status
Priority
Subject
Due date
Estimated time

Maximum entries: 4

Note: Not every block is configurable and the block configuration may differ depending on the selection you have made. Some blocks may allow to change the column settings and other only the maximum entries, for example.

9.2.3 General dashboard actions

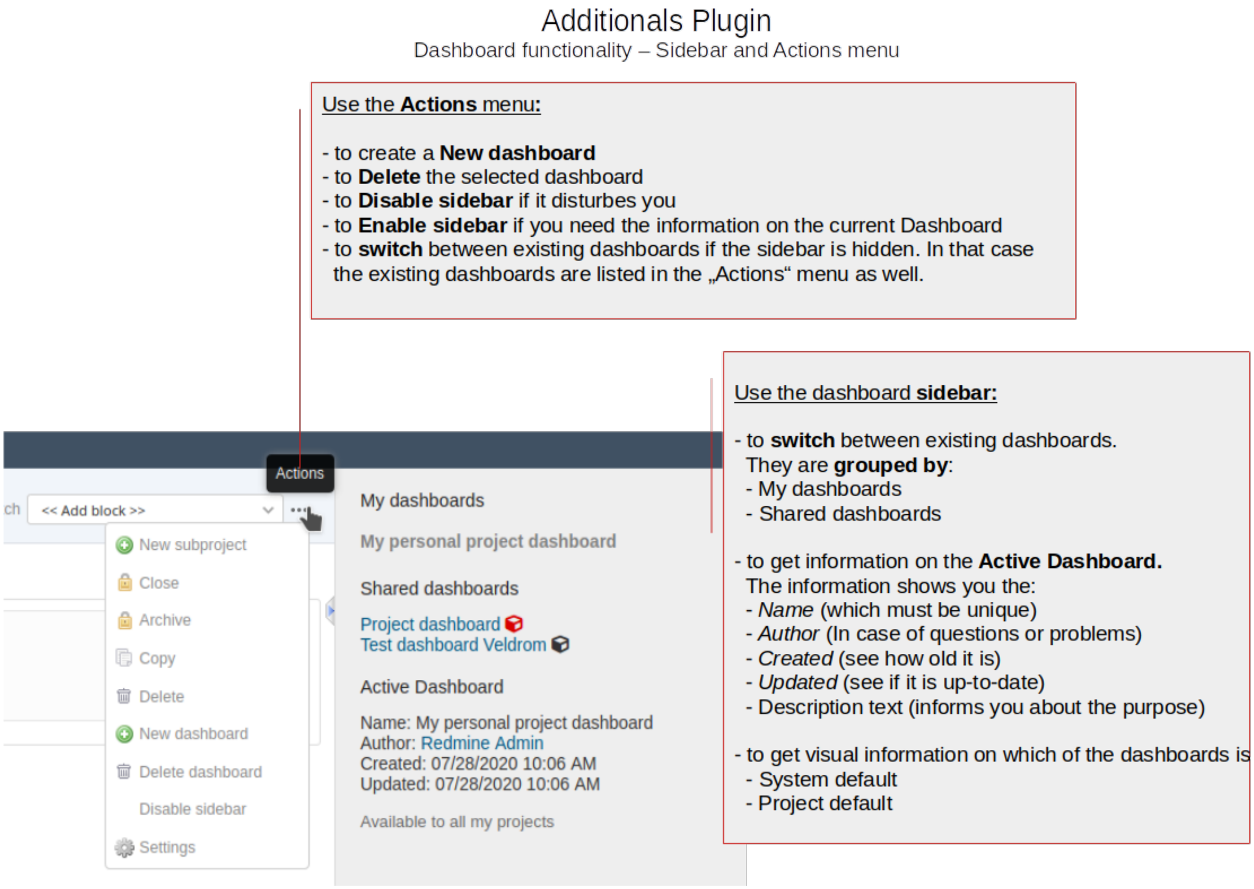
Some actions on the dashboards are the same whether you are on a dashboard of the home page or the project overview page.

Switching dashboards You have two options to switch between existing dashboards. If the sidebar is activated for the dashboard you can select the respective dashboard from the sidebar section. They are grouped by *My dashboards* or *Shared dashboards*.

Enable or disable sidebar If the sidebar is hidden by default you can activate it via the *Actions* menu. Just select the menu option *Enable sidebar*. If the sidebar is displayed but you want to hide it select the menu option *Disable sidebar*.

View sidebar information The sidebar contains always the same information for the user like general globa infos, that are stored there or dashboard specific information regarding the *Active Dashboard* or other dashboards, which are available to you as user.

Some other general dashboard actions are the selection box in order to



Additionals Plugin

Dashboard functionality – Sidebar and Actions menu

Use the **Actions** menu:

- to create a **New dashboard**
- to **Delete** the selected dashboard
- to **Disable sidebar** if it disturbes you
- to **Enable sidebar** if you need the information on the current Dashboard
- to **switch** between existing dashboards if the sidebar is hidden. In that case the existing dashboards are listed in the „Actions“ menu as well.

Use the dashboard **sidebar:**

- to **switch** between existing dashboards.
  They are **grouped by**:
  - My dashboards
  - Shared dashboards
- to get information on the **Active Dashboard.**
  The information shows you the:
  - *Name* (which must be unique)
  - *Author* (In case of questions or problems)
  - *Created* (see how old it is)
  - *Updated* (see if it is up-to-date)
  - Description text (informs you about the purpose)
- to get visual information on which of the dashboards is
  - System default
  - Project default

My dashboards

My personal project dashboard

Shared dashboards

Project dashboard

Test dashboard Veldrom

Active Dashboard

Name: My personal project dashboard
Author: Redmine Admin
Created: 07/28/2020 10:06 AM
Updated: 07/28/2020 10:06 AM

Available to all my projects

Add blocks The content which can be added to the active dashboard depends on the functions you are using on your Redmine and the developer of the function. Those need to support the dashboard *Drag & Drop* functionality with their plugin. Regardless of this, the dashboard is always filled with content via the "Add block" selection.

9.2.4 Dashboard blocks

The following blocks are delivered as standard when using the additionals dashboard function.

General blocks for all areas (e.g. *Home* and *Project overview*) are:

Query: Issues You can configure the block content by selecting the *Issue Query* you want to display and choosing a number for *Maximum entries* to be displayed. Maximum number of such blocks per dashboard: **8**

Text You can add individual text passages to your dashboard and use the wiki editor for structuring this text. If you want to work with macros, that include large amount of information (e.g. issue lists), use the *Text (asynchronous)* block instead, because it is cached asyncronous for 30 seconds due to performance reasons. Maximum number of such blocks per dashboard: **8**

Text (asynchronous) This block is perfect if you want to implement macros into your text area. You can add individual text passages to your dashboard and use the wiki editor for structuring this text. Please note, that some macros are only usable in Wiki pages of a project, but not in issues or dashboard blocks. In that case no content will be displayed, but only the macro code. Maximum number of such blocks per dashboard: **8**

Latest news Add the latest news block to your dashboard to show your user's what is going on in your system. The block configuration allows you to edit the number of *Maximum entries*. Maximum number of such blocks per dashboard: **1**

Documents Add the *Documents* block to your dashboard to show your user's the latest documents, uploaded in Redmine. The block configuration allows you to edit the number of *Maximum entries*. Maximum number of such blocks per dashboard: **1**

Static spent time Add the *Spent time* block to your dashboard to show the spent time activities. The block configuration allows you to edit the number of *Days* displayed in the block. Maximum number of such blocks per dashboard: **1**

Atom+RSS Feed You can configure the block content by changing the *Title*, adding an *URL* and choosing a number for *Maximum entries* to be displayed. Maximum number of such blocks per dashboard: **8**

Special blocks for the welcome page *Home* are:

Standard content left / Standard content right There are plugin developers who do not support our dashboard function and may have implement own content on the Redmine start page (legacy_left and legacy_right). These are usually not displayed due to the lack of dashboard support. If you still want to see this, select one of the two blocks or both for displaying such content.

Activity You can configure the activity block content by choosing a number for *Maximum entries* to be displayed. And in case you are only interested in your own activities by activating the option *Only my activity*. Maximum number of such blocks per dashboard: **1**

Welcome If you have specified a welcome text in the Redmine system settings you can display this text in the dashboard as well, by choosing the *Welcome* block. Maximum number of such blocks per dashboard: **1**

Special blocks for *Project overview* are:

Standard content left / Standard content right There are plugin developers who do not support our dashboard function and may have implement own content on the Redmine project overview page (legacy_left and legacy_right). These are usually not displayed due to the lack of dashboard support. If you still want to see this, select one of the two blocks or both for displaying such content as well.

Project information Displays the standard project information of Redmine in a block. Containing the project information from the project settings like custom fields for projects, description, URL etc. The block is not configurable, but you can remove it if not needed or reposition it.

Subprojects Displays the standard subproject information of Redmine in a block, if you have configured them in the project settings. The block is not configurable, but you can remove it if not needed or reposition it.

Issue tracking Displays the standard issue tracking information of Redmine in a block. The block is not configurable, but you can remove it if not needed or reposition it.

Time tracking Displays the standard issue tracking information of Redmine in a block including the additionals adaption to the information displayed there. The block is not configurable, but you can remove it if not needed or reposition it.

Members Displays the standard member box of Redmine in a block including the additionals permission adaption to the information displayed there. The block is not configurable, but you can remove it if not needed or reposition it.

Note: These are the basic blocks that come with the additionals plugin dashboard functionality. Other plugins, that also support the *Drag&Drop* dashboard function may deliver additional blocks for selection.

Note: Notes about caching: All blocks are cached asynchron (every 30 seconds). The RSS Feed blocks are cached asynchron (every 10 minutes). The time tracking block is chached asynchron (every 1 hour).

9.2.5 Default Dashboards

When installing the Additionals plugin it comes with default dashboards for the Redmine *Home* page as well as the Redmine *Project overview* page. The typical default dashboard has the following key options:

- Visible: *to any users*
- Marked as *System default*

In case you want to restore the default dashboard you can simply add a new one or arrange an existing one by using the following blocks and position them in the left or right area.

Default Dashboard blocks: Home Welcome (left), Standard content left (left), Standard content right (right)

Default Dashboard blocks: Project overview Project information (left), Issue tracking (left), Time tracking (left), Members (right), Sub projects (right)

9.2.6 Dashboard FAQ

Is there a restore function? No, there is no restore function for your default dashboard. If you add dashboard blocks you don't like, just remove them and add other ones.

I messed up my default Dashboard There is nothing bad about it. Just remove the dashboard blocks you don't like and start over. The best way to start a new default dashboard anyway is to create a new dashboard, first and add the blocks you like. If everything is the way you want it, make it "Default". Keep the old "Default" as backup.

I accidently deleted a Dashboard block This will always happen, if you work as user with the appropriate permission to do so (e.g. administration permissions). We recommend not do work with those permission in your regular work day. Create your dashboards one time and switch user permissions afterwards. So this will not happen again.

How many default Dashboards can be created? There is just one system default dashboard possible for every area. This means one default dashboard for the Redmine "Home" page. And one for the general project overview page. But you can create also one default for a specific project overview page, which will than be "Project default".

Does every user sees the content of every Dashboard block? You do not control the content a user in your project sees via the dashboard block you add, but still by setting up the correct user permissions in the administration area "Roles and permissions". Those permissions for a user role are relevant for the content a user can view in your project and must be set correctly.

9.3 Macros

These wiki macros are available with Additionals.

If a parameter is in brackets, this parameter is optional.

9.3.1 Asciinema

Graphical application developers often use screencasts to demonstrate functions of their programs. With Asciinema you have the possibility to record such videos, but are restricted completely to the terminal. So you can record and publish actions in the shell. With the additionals plugin macro for Asciinema you can implement such videos into Redmine.

asciinema.org wiki macro for Redmine.

`{{asciinema(cast_id)}}`
show asciinema.org cast

Parameters **`cast_id`** (*`string`*) – asciinema.org asciicast id

Scope

This macro works in all text fields with formatting support.

Examples

show asciinema.org cast_id `113463`

```
{{asciinema(113463)}}
```

9.3.2 CryptoCompare

CryptoCompare is an Internet plattform where you can interactively inform yourself about the latest trends in crypto currencies and follow the development of the crypto market in real-time.

CryptoCompare wiki macro for Redmine.

See https://www.cryptocompare.com/dev/widget/wizard/ for more information.

`{{cryptocompare(options)}}`
show CryptoCompare information

Parameters

- **fsym** (*string*) – default BTC
- **tsym** (*string*) – default EUR
- **fsyms** (*string*) – default BTC,ETH,LTC (if supported by widget type)
- **tsyms** (*string*) – default EUR,USD (if supported by widget type)
- **period** (*string*) – (if supported by widget type)
  - 1D = 1 day (default)
  - 1W = 1 week
  - 2W = 2 weeks
  - 1M = 1 month
  - 3M = 3 months
  - 6M = 6 months
  - 1Y = 1 year
- **type** (*string*) – widget type has to be one of
  - advanced
  - chart (default)
  - converter
  - header
  - header_v2
  - header_v3
  - historical
  - list
  - news
  - summary
  - tabbed
  - titles

Scope

This macro works in all text fields with formatting support.

Examples

Show `header_v3` widget type for crypto currencies `BTC` and `ETH`

```
{{cryptocompare(fsyms=BTC;ETH, type=header_v3)}}
```

9.3.3 Date

`{{date([TYPE])}}`

Display current date

Parameters **`TYPE`** (*string*) – current_date (default) current date current_date_with_time current date with time current_year current year current_month current month current_day current day current_hour current hour current_minute current minute current_weekday current weekday current_weeknumber current week number (1 - 52) The week starts with Monday YYYY-MM-DD e.g. 2018-12-24, which will formated with Redmine date format

Scope

This macro works in all text fields with formatting support.

Examples

Show current date.

```
{{date}}
```

Show current date with time

```
{{date(current_date_with_time)}}
```

Show current year

```
{{date(current_year)}}
```

Show current week number

```
{{date(current_weeknumber)}}
```

Show date 2018-12-24 in right format, which is specified in Redmine Settings, which shows e.g. 24.12.2018

```
{{date(2018-12-24)}}
```

9.3.4 Font Awesome

Font Awesome wiki macro for Redmine.

`{{fa(icon [, class=CLASS, title=TITLE, text=TEXT, link=LINK, color=COLOR])}}`

Displays a Font Awesome icon

Parameters

- **`icon`** (*string*) – font awesome icon name, e.g. adjust or fas_adjust
- **`class`** (*string*) – additional css classes
- **`title`** (*string*) – mouseover title
- **`text`** (*string*) – text, which is displayed after font awesome icon
- **`link`** (*string*) – link to this url
- **`color`** (*string*) – css color code

Scope

This macro works in all text fields with formatting support.

Examples

Show font awesome icon "fas fa-adjust"

```
{{fa(adjust)}}
```

Show font awesome icon "fas fa-stack" and inverse

```
{{fa(adjust, class=fa-inverse)}}
```

Show font awesome icon "fas fa-adjust" with size 4x

```
{{fa(adjust, size=4x)}}
```

Show font awesome icon "fas fa-adjust" with title "Show icon"

```
{{fa(fas_adjust, title=Show icon)}}
```

Show font awesome icon "fab fa-angellist"

```
{{fa(fab_angellist)}}
```

Show font awesome icon "fas fa-adjust" and link it to https://www.redmine.org

```
{{fa(adjust, link=https=//www.redmine.org))}}
```

Show font awesome icon "fas fa-adjust" with name "Go to Redmine.org" and link it to https://www.redmine.org

```
{{fa(adjust, link=https=//www.redmine.de, name=Go to Redmine.org))}}
```

Icons

There are currently more than 1300 free Font Awesome Icons available for implementation. The full list can be found at: https://fontawesome.com/icons?d=gallery&m=free

Note: All you have to do is use the icon name and implement it into your macro as mentioned above.

9.3.5 Gist

GitHub, a service for hosting larger projects, also operates other services like Gist. Gist is for hosting code snippets. If you are using the additionals plugin for your Redmine, you can implement your Github Gist by using the following wiki macro.

`{{gist(gist)}}`

show Github gist

Parameters **`gist`** (*string*) – gist to display. With or without Github username.

Scope

This macro works in all text fields with formatting support.

Examples

show Github gist `6737338` (without user name)

```
{{gist(6737338)}}
```

Show Github gist `plentz/6737338` (with user name)

```
{{gist(plentz/6737338)}}
```

9.3.6 Google Map

Google map wiki macro for Redmine to implement a street map. There are various configuration options for presenting the GMap.

`{{gmap([q=QUERY, mode=MODE, width=WIDTH, height=HEIGHT])}}`

Show Google map

Parameters

- **q** (*string*) – query, e.g. a city or location
- **mode** (*string*) – place, directions, search, view oder streetview (default: search)
- **width** (*int*) – widget width
- **height** (*int*) – widget height

Scope

This macro works in all text fields with formatting support.

Examples

The simplest way is to show a widget for Munich with `Munich`

```
{{gmap(Munich)}}
```

Another option is to show a widget for direction for example from Munich to Arco (Italy)

```
{{gmap(mode=directions, origin=Munich+Implerstr, destination=Arco)}}
```

9.3.7 Google Docs

Google Docs wiki macro for Redmine. Google Docs is an online word processor that you can use to create, format, and collaborate on documents. The macro helps you to integrat a Google Docs document as iframe into a Redmine text field.

`{{google_docs(link[, width=WIDTH, height=HEIGHT, edit_link=URL])}}`

Show Google Docs embedded

Parameters

- **`link`** (*`string`*) – Embedded Google docs link
- **`width`** (*`int`*) – width (if not specified, 100% is used)
- **`height`** (*`int`*) – height (if not specified, 485 is used)
- **`edit_link`** (*`int`*) – Link to edit page

Scope

This macro works in all text fields with formatting support (e.g. Redmine Wiki).

Examples

Google docs `https://docs.google.com/spreadsheets/d/e/2PACX-1vQL__Vgu0Y0f-P__GJ9kpUmQ0S-HG56ni_` `pubhtml` (iframe) and default size 100% x 485

```
{{google_docs(https://docs.google.com/spreadsheets/d/e/2PACX-1vQL__Vgu0Y0f-P__
↪GJ9kpUmQ0S-HG56ni_b-x4WpWxzGIGXh3X6A587SeqvJDpH42rDmWVZoUN07VGE/pubhtml)}}
```

Google docs with size 1200 x 1000

```
{{slideshare(https://docs.google.com/spreadsheets/d/e/2PACX-1vQL__Vgu0Y0f-P__
↪GJ9kpUmQ0S-HG56ni_b-x4WpWxzGIGXh3X6A587SeqvJDpH42rDmWVZoUN07VGE/pubhtml, width=1200,
↪ height=1000)}}
```

9.3.8 Members, Group

Group users macro for Redmine.

`{{group_users(group_name)}}`

Show list of users in a user group (according the respective permissions)

Parameters **`group_name`** (*`string`*) – group name

Scope

This macro works in all text fields with formatting support.

Examples

List all users of user group `Team A`

```
{{group_users(Team A)}}
```

9.3.9 Iframe

Iframe macro for Redmine. The iframe is a square HTML element. It can be embedded e.g. within a Wiki page of Redmine. Via an iframe, content from other Internet pages can be integrated just like via a small integrated browser window.

For example you can integrate a CodiMD Pad page into your Wiki pages.

`{{iframe(url [, width=INT, height=INT, with_link=BOOL])}}`

Include an Iframe into Redmine. If your Redmine is running with HTTPS, only iframes with HTTPS are accepted by this macro.

Parameters

- **`url`** (*`string`*) – URL to website
- **`width`** (*`int`*) – width (default is 100%)
- **`height`** (*`int`*) – height (default is 485)
- **`with_link`** (*`bool`*) – true or false (if link to url should be displayed below iframe)

Note 1: you can only include an iframe, if the website of the iframe url does allow it. If not, you will get a empty page with the HTTP header info:

`Load denied by X-Frame-Options: https://your-target-url.com/ does not permit cross-origin framing.`

Note 2: Your Redmine webserver can also block your iframe inclusion, check your Content Security Policy (CSP)

Scope

This macro works in all text fields with formatting support.

Examples

Show iframe of URL `https://www.google.com/`

```
{{iframe(https://www.google.com/)}}
```

Show iframe of URL https://www.google.com/ and show link to it

```
{{iframe(https://www.google.com/, with_link: true)}}
```

9.3.10 Issue

Issue wiki macro for Redmine.

`{{issue(url [, format=FORMAT, id=ISSUE_ID, note_id=COMMENT_ID])}}`

Display a link to issue with subject (optional with an issue note)

Parameters

- **`url`** (*`string`*) – URL to an issue with issue id (and note_id)
- **`format`** (*`string`*) – custom format of link name. Possible values: full, text, short or link. If not specified 'link' is used as default.
- **`id`** (*`int`*) – issue id (if this is defined, it will be always used prioritized - before the parameter for URL)
- **`note_id`** (*`int`*) – comment id (if this is defined, it will be always used prioritized - before the parameter for URL)

Scope

This macro works in all text fields with formatting support.

Examples

Link to issue with id and subject

```
{{issue(1)}}
```

Link to issue with id and subject by using the URL

```
{{issue(http://myredmine.url/issues/1)}}
```

Link to issue with id and subject and show comment 3

```
{{issue(http://myredmine.url/issues/1#note-3)}}
```

Link to issue with subject (without id)

```
{{issue(1, format=short)}}
```

Link to issue with tracker, subject and id

```
{{issue(1, format=full)}}
```

Display subject of issue

```
{{issue(1, format=text)}}
```

9.3.11 Last updated at

Last updated at wiki macro for Redmine.

{{last_updated_at}}
show the timestamp when the wiki page was updated

Scope

This macro only works in wiki page contexts.

9.3.12 Last updated by

Last updated by wiki macro for Redmine.

{{last_updated_by}}
show the name of user who updated the wiki page

Scope

This macro only works in wiki page contexts.

9.3.13 Meteoblue

Meteoblue wiki macro for Redmine. A macro to show weather forecast for a specific place. There are various configuration options for presenting the weather widget.

`{{meteoblue(location [, days=DAYS, width=WIDTH, height=HEIGHT, color=BOOL, flags...])}}`
Show weather widget from Meteoblue

Parameters

- **`location`** (*`string`*) – Weather location, e.g. münchen_deutschland_2867714. This is the part after https://www.meteoblue.com/en/weather/forecast/week/
- **`width`** (*`int`*) – widget width
- **`height`** (*`int`*) – widget height
- **`color`** (*`bool`*) – color or monochrome
- **`pictoicon`** (*`bool`*) – pictoicon
- **`maxtemperature`** (*`bool`*) – maxtemperature
- **`mintemperature`** (*`bool`*) – mintemperature
- **`windspeed`** (*`bool`*) – windspeed
- **`windgust`** (*`bool`*) – windgust
- **`winddirection`** (*`bool`*) – winddirection
- **`uv`** (*`bool`*) – uv
- **`humidity`** (*`bool`*) – humidity
- **`precipitation`** (*`bool`*) – precipitation
- **`precipitationprobability`** (*`bool`*) – precipitationprobability
- **`spot`** (*`bool`*) – spot

Scope

This macro works in all text fields with formatting support.

Examples

Show widget for Munich with `münchen_deutschland_2867714` (iframe)

```
{{meteoblue(münchen_deutschland_2867714)}}
```

Show widget for Munich with `münchen_deutschland_2867714` (iframe) and uv information

```
{{meteoblue(münchen_deutschland_2867714, uv=true)}}
```

9.3.14 Members, Project

Project members macro for Redmine.

`{{members([project_name, title=TITLE, role=ROLE, with_sum=BOOL])}}`
Show list of project members

Parameters

- **`project_name`** (*string*) – can be project identifier, project name or project id
- **`title`** (*string*) – title to use for member list
- **`role`** (*string*) – only list members with this role. If you want to use multiple roles as filters, you have to use a | as separator.
- **`with_sum`** (*bool*) – show amount of members.

Scope

This macro works in all text fields with formatting support.

Examples

List all members for all projects (with the current user permission)

```
{{members}}
```

List all members for all projects and show amount of members

```
{{members(with_sum=true)}}
```

List all members for the project with the identifier of `myproject`

```
{{members(myproject)}}
```

List all members for the project with the identifier of `myproject`, which have the role `Manager`

```
{{members(myproject, role=Manager)}}
```

List all members for the project with the identifier of `myproject`, which have the role `Manager` or `Team`

```
{{members(myproject, role=Manager|Team)}}
```

List all members for the project with name `My project title` and with box title `My member list`

```
{{members(My project title, title=My member list)}}
```

9.3.15 New issue

Create a link for "New issue" for the current user.

`{{new_issue([project_name, name=NAME])}}`

Show link to create new issue

Parameters

- **`project_name`** (*string*) – can be project identifier, project name or project id If no project_name is specified, first project is used, which the current user has permission to create an issue.
- **`name`** (*string*) – name to use for link. If not specified, "New issue" is used. You can use all language as suffix, eg. name_de, name_it, button_es

Scope

This macro works in all text fields with formatting support.

Examples

Link to create new issue in first available project

```
{{new_issue}}
```

Link to create new issue in project with the identifier of `myproject`

```
{{new_issue(myproject)}}
```

Link to create new issue in project with the identifier of `myproject` and with displayed link name `New issue for broken displays`

```
{{new_issue(myproject, title=New issue for broken displays)}}
```

Link to create new issue in first available project and overwrite name of link only for German users

```
{{new_issue(name_de: Ticket hier bitte anlegen)}}
```

9.3.16 Projects

Projects wiki macro for Redmine.

`{{projects([title=TITLE, with_create_issue=BOOL])}}`

Lists projects of the current user

Parameters

- **`title`** (*`string`*) – title to use for project list
- **`with_create_issue`** (*`bool`*) – show link to create new issue, if true

Scope

This macro works in all text fields with formatting support.

Examples

List all projects of the current users

```
{{projects}}
```

List all projects of the current users and adds the heading `My project list`

```
{{projects(title=My project list)}}
```

List all project with link to create new issue, which I am member of

```
{{projects(with_create_issue=true)}}
```

9.3.17 Recently updated

Recently updated wiki macro for Redmine.

`{{recently_updated([days])}}`

show the list of the pages that were changed recently.

Parameters **`days`** (*`int`*) – number of days, which should be used. Default is 5.

Scope

This macro only works in wiki page contexts.

Examples

List last updated pages (of the last 5 days)

```
{{recently_updated}}
```

List last updated pages of the last 15 days

```
{{recently_updated(15)}}
```

9.3.18 Reddit

Reddit is an English-language website for social media news, web content rating and discussions. As a registered member, you can submit content, links or images and present them to other members. Link to your Reddit account or other content within Redmine by using the additionals plugin macro.

Reddit wiki macro for Redmine.

`{{reddit(name)}}`

show link to Reddit

Parameters **`name`** (*`string`*) – Reddit subject or user name

Scope

This macro works in all text fields with formatting support.

Examples

Show link to reddit subject `r/redmine`

```
{{reddit(redmine)}}
```

or

```
{{reddit(r/redmine)}}
```

Show link to reddit user profile `u/redmine`

```
{{reddit(u/redmine)}}
```

9.3.19 Redmine.org Issue

Redmine.org issue wiki macro for Redmine.

{{redmine_issue(url [, title=TITLE])}}
Display a link to an issue on redmine.org

Parameters

- **url** (*string*) – this can be an absolute path to an redmine.org issue or an issue id
- **title** (*string*) – title of link to display

Scope

This macro works in all text fields with formatting support.

Examples

Link to redmine.org issue with issue id

```
{{redmine_issue(1333)}}
```

Link to redmine.org issue with issue id and anchor

```
{{redmine_issue(1333#note-6)}}
```

Link to redmine.org issue with absolute url

```
{{redmine_issue(http://www.redmine.org/issues/12066)}}
```

9.3.20 Redmine.org Wiki page

Redmine.org wiki page macro for Redmine.

{{redmine_wiki(url [, name=NAME, title=TITLE])}}
Display a link to an issue on redmine.org

Parameters

- **url** (*string*) – this can be an absolute path to an redmine.org issue or an issue id
- **name** (*string*) – name to display for link, if not specified, wiki page name is used
- **title** (*string*) – title of link to display

Scope

This macro works in all text fields with formatting support.

Examples

Link to redmine.org wiki page with page name

```
{{redmine_wiki(RedmineInstall)}}
```

Link to redmine.org wiki page with page name and anchor

```
{{redmine_wiki(FAQ#How-do-I-create-sub-pages-parentchild-relationships-in-the-wiki)}}
```

Link to redmine.org wiki page with absolute url

```
{{redmine_wiki(https://www.redmine.org/projects/redmine/wiki/RedmineInstall)}}
```

9.3.21 Slideshare

Slideshare wiki macro for Redmine.

`{{slideshare(key [, width=WIDTH, height=HEIGHT, slide=SLIDE])}}`

Show slideshare embedded slide

Parameters

- **`key`** (*`string`*) – Slideshare embedded key code, e.g. 57941706. This is the part is the last number in url: http://de.slideshare.net/AnimeshSingh/caps-whats-best-for-deploying-and-managing-openstack-chef-vs-ansible-vs-puppet-vs-salt-57941706
- **`width`** (*`int`*) – width
- **`height`** (*`int`*) – height
- **`slide`** (*`int`*) – Slide page

Scope

This macro works in all text fields with formatting support.

Examples

Slideshare slide for slide `57941706` (iframe) and default size 595x485

```
{{slideshare(57941706)}}
```

Slideshare slide with size 514x422

```
{{slideshare(57941706, width=514, height=422)}}
```

Slideshare slide and start with page 5

```
{{slideshare(57941706, slide=5)}}
```

9.3.22 TradingView

TradingView wiki macro for Redmine.

See https://www.tradingview.com/widget/ for more information.

`{{tradingview(options)}}`
show TradingView chart

Parameters

- **`width`** (*int*) – default 640
- **`height`** (*int*) – default 480
- **`symbol`** (*string*) – default NASDAQ:AAPL
- **`interval`** (*string*) – default W
- **`timezone`** (*string*) – default Europe/Berlin
- **`theme`** (*string*) – default White
- **`style`** (*int*) – default 2
- **`locale`** (*string*) – default de
- **`toolbar_bg`** (*string*) – default #f1f3f6
- **`enable_publishing`** (*bool*) – default false
- **`allow_symbol_change`** (*bool*) – default true
- **`hideideasbutton`** (*bool*) – default true

Scope

This macro works in all text fields with formatting support.

Examples

Show chart with symbol `NASDAQ:AMZN` and use English locale

```
{{tradingview(symbol=NASDAQ:AMZN, locale=en)}}
```

9.3.23 Twitter

Twitter wiki macro for Redmine.

`{{twitter(profile)}}`
show link to Twitter profile

Parameters **`profile`** (*string*) – Twitter profile name with @. E.g. alphanodes

Scope

This macro works in all text fields with formatting support.

Examples

Show link to twitter profile `@alphanodes`

```
{{twitter(alphanodes)}}
```

or

```
{{twitter(@alphanodes)}}
```

Show link to hashtag `#redmine`

```
{{twitter(#redmine)}}
```

9.3.24 User

User wiki macro for Redmine.

`{{user(user [, format=FORMAT, text=BOOL, avatar=BOOL])}}`

Display link to user profile

Parameters

- **`user`** (*`string`*) – user name (login name) or user id of the user. If current user is used as login name the currently logged in user will be used.
- **`format`** (*`string`*) – custom format of link name. If not specified system settings will be used. You can use format with the same options as for system settings.
- **`text`** (*`bool`*) – show text only (without link), if true (default: false)
- **`avatar`** (*`bool`*) – show avatar, if true (default: false)

Scope

This macro works in all text fields with formatting support.

Examples

Link to user profile with id 1

```
{{user(1)}}
```

Link to user profile with id 1 and show user avatar

```
{{user(1, avatar=true)}}
```

Link to user profile with login name `admin` and show user avatar

```
{{user(admin, avatar=true)}}
```

Link to user profile with login name `admin` with username as link text

```
{{user(admin, format=username)}}
```

Link to user profile with login name `admin` with first name as link text

```
{{user(admin, format=firstname)}}
```

Link to user profile with login name `admin` with last name as link text

```
{{user(admin, format=lastname)}}
```

Link to user profile with login name `admin` with first name and last name as link text

```
{{user(admin, format=firstname_lastname)}}
```

Display name of currently logged in user with username as text

```
{{user(current_user, text=true)}}
```

Display name of currently logged in user with username

```
{{user(current_user)}}
```

9.3.25 Vimeo

Vimeo wiki macro for Redmine.

`{{vimeo(video [, width=WIDTH, height=HEIGHT, autoplay=BOOL])}}`

Show Vimeo embedded video

Parameters

- **`video`** (*string*) – Vimeo video code, e.g. 142849533. This is the part after https://vimeo.com/
- **`width`** (*int*) – video width
- **`height`** (*int*) – video height
- **`autoplay`** (*bool*) – auto play video, if true

Scope

This macro works in all text fields with formatting support.

Examples

Vimeo video with video `142849533` (iframe) and default size 640x360

```
{{vimeo(142849533)}}
```

Vimeo video with size 853x480

```
{{vimeo(142849533, width=853, height=480)}}
```

Vimeo video and autoplay

```
{{vimeo(142849533, autoplay=true)}}
```

9.3.26 Youtube

Youtube wiki macro for Redmine.

`{{youtube(video [, width=WIDTH, height=HEIGHT, autoplay=BOOL])}}`

Show Youtube embedded video

Parameters

- **`video`** (*string*) – Youtube video code, e.g. wvsboPUjrGc. This is the part after https://www.youtube.com/watch?v=
- **`width`** (*int*) – video width
- **`height`** (*int*) – video height
- **`autoplay`** (*bool*) – auto play video, if true

Scope

This macro works in all text fields with formatting support.

Examples

Youtube video with video `wvsboPUjrGc` (iframe) and default size 640x360

```
{{youtube(wvsboPUjrGc)}}
```

Youtube video with size 853x480

```
{{youtube(wvsboPUjrGc, width=853, height=480)}}
```

Youtube video with auto play

```
{{youtube(wvsboPUjrGc, autoplay=true)}}
```

9.4 Tasks

Additionals comes with some rake tasks, which can be used for administration.

9.4.1 drop_settings

`rake redmine:additionals:drop_settings plugin=PLUGIN`

Drop settings for a plugin

Parameters **`plugin`** (*string*) – name of the plugin

Examples

Remove plugin settings for redmine plugin named `redmine_plugin_example`

```
bundle exec rake redmine:additionals:drop_settings RAILS_ENV=production plugin=
↪"redmine_plugin_example"
```

9.4.2 setting_set

rake redmine:additionals:setting_set name=NAME setting=SETTING value=VALUE

Set settings for redmine or plugin

Parameters

- **name** (*string*) – name of the plugin or redmine (this is optional, if not defined redmine is used)
- **setting** (*string*) – name of setting
- **value** (*string*) – value for setting
- **values** (*string*) – list of values (seperator is ,) to generate value array automaticaly

Examples

Set application title for Redmine

```
bundle exec rake redmine:additionals:setting_set RAILS_ENV=production setting="app_
↪title" value="Redmine test instance"
```

Set default modules for new projects

```
bundle exec rake redmine:additionals:setting_set RAILS_ENV=production setting=
↪"default_projects_modules" values="issue_tracking,time_tracking,wiki"
```

Set plugin setting `open_external_urls` for plugin additionals to value 2

```
bundle exec rake redmine:additionals:setting_set RAILS_ENV=production name=
↪"additionals" setting="open_external_urls" value="1"
```

9.4.3 setting_get

rake redmine:additionals:setting_get name=NAME setting=SETTING

Get a setting of redmine or a plugin

Parameters

- **name** (*string*) – name of the plugin or redmine (this is optional, if not defined redmine is used)
- **setting** (*string*) – name of setting

Examples

Get setting for `open_external_urls` of the plugin additionals

```
bundle exec rake redmine:additionals:setting_get RAILS_ENV=production name=
↪"additionals" setting="open_external_urls"
```

Get `app_title` of redmine

```
bundle exec rake redmine:additionals:setting_get RAILS_ENV=production name="redmine"␣
↪setting="app_title"
```

Get `app_title` of redmine

```
bundle exec rake redmine:additionals:setting_get RAILS_ENV=production setting="app_
↪title"
```

9.5 You need a feature

‘Additionals‘_ is Open-source and it is available at Github

If you want to implement new features in it or if you want to change something, you can provide a pull request.

The plugin is maintained by AlphaNodes for free as far as possible. In case you want a feature, which is not available and you are not capable of implementing it yourself, you can request this feature from AlphaNodes.

We are an Open-source company from Munich and we are usually getting payed for our time we spent on development. As we know our plugin at its best we are glad to take this job from you. In case the requested plugin changes still fit to the plugin purpose. Please, contact us in case you are interested in plugin development.

9.6 Changelog

9.6.1 3.0.1

- Do not show “Assign to me” if assigned_to is disabled for tracker
- FontAwesome 5.15.1 support
- D3 6.2.0 support
- Mermaid 8.8.2 support
- add current_user as special login name for user macro (which shows current login user)
- add text parameter to user macro (which disable link to user)
- add asynchronous text block

9.6.2 3.0.0

- Introduce dashboards
- Redmine 4.1 or newer is required
- FontAwesome 5.14.0 support
- D3 6.1.1 support
- Mermaid 8.8.0 support
- d3plus to v2.0.0-alpha.29 support
- drop wiki header and footer settings

9.6.3 2.0.24

- FontAwesome 5.13.0 support
- Mermaid 8.4.8 support
- clipboard.js updated to v2.0.6
- fix for spam protection with invisible_captcha
- D3 5.16.0 support
- Ruby 2.4 is required

9.6.4 2.0.23

- members macro now supports with_sum option
- FontAwesome 5.12 support
- FontAwesome ajax search has been added
- Mermaid 8.4.6 support
- D3 5.15.0 support
- Drop nvd3 library
- Drop Chartjs stacked100 library
- Drop d3plus-hierarchy library
- Drop calendar macro
- Support private comments with issue macro
- Google Docs macro has been added
- Fix bug with Rack 2.0.8 or newer
- Drop Redmine 3.4 support
- Add Redmine 4.1 support
- Use view_layouts_base_body_top hook, which is available since Redmine 3.4
- Refactoring new hooks (without template)
- asciinema.org macro has been added - thanks to @kotashiratsuka
- Select2 4.0.13 support

9.6.5 2.0.22

- FontAwesome 5.11.2 support
- Mermaid 8.4.2 support
- Select2 4.0.12 support
- Chart.js 2.9.3 support
- Chart.js Plugin datalabels 0.7.0 support
- d3plus to v2.0.0-alpha.25

- Fix user visibility for members macro
- Fix user visibility for issue reports
- Drop ZeroClipboard library

9.6.6 2.0.21

- fix mail notification if issue author changed
- fix permission bug for closed issues with freezed mode
- Ruby 2.2.x support has been dropped. Use 2.3.x or newer ruby verion
- FontAwesome 5.9.0 support
- remove issue_close_with_open_children functionality, because this is included in Redmine 3.4.x #47 (thanks to @pva)
- add hierarchy support for projects macro #45
- select2 support
- bootstrap-datepicker 1.9.0 support

9.6.7 2.0.20

- support single process rake installation #40
- FontAwesome 5.8.0 support

9.6.8 2.0.19

- mermaid 8.0.0 support
- FontAwesome 5.7.1 support
- fixed close issue without permission
- create correct journal entry if issue status changed from sidebar #37
- create correct journal entry if issue has been ‘assigned to me’ from sidebar

9.6.9 2.0.18

- Performance improvement (#36)
- FontAwesome 5.6.3 support
- Fix problem from migrating from Redmine 3.x to Redmine 4 with lost settings

9.6.10 2.0.17

- Fix bug with undefined constant for tags
- add possibility to use custom date with date macro
- FontAwesome 5.6.0 support

9.6.11 2.0.16

- CSS fix for project macro
- More compatibility for autocomplete_users with other plugins (like redmine_contacts_helpdesk)
- Compatibility to wiking plugin (macro list)
- Add Spanish translation, thanks to @dktcoding!
- Wiki button for available macros
- replace permission hide-in-memberbox with "hide" as role setting - check your roles, if you used this permission!
- replace multiple current_date macros with one macro called date
- cleanup macros: if no data exists, macros is displayed instead of data (before some macros used error messages other hide message at all)

9.6.12 2.0.15

- FontAwesome 5.5.0 support
- Usability improvement for change author in issue formular
- options and permission for issue requires timelog to use status
- New option to freeze issues on close. With permission "edit closed issue" user can break this rule.
- Fix problem with help menu and other redmine plugins (compatibility problem with other plugins)
- Fix problem with disabled users and changing author for issues

9.6.13 2.0.14

- Change status is now compatible with redmine_agile
- Do not show sidebar for changing status, if edit_closed_issues permission is missing
- FontAwesome 5.3.1 support
- compatibility with plugin redmine_sudo and redmine_base_deface
- FontAwesome wiki macro has been added (called fa)
- Redmine.org issue and wiki page macro has been added
- Show macro list to all logged users at /help/macros
- Help menu, with more links to Redmine help pages (which can be used with other plugins, to assign additional entries)

9.6.14 2.0.13

- FontAwesome 5.2.0 support
- smiley support for markdown text_formatting
- new_issue macro with i18n support
- updated bootstrap-datepicker to v1.8.0

- updated d3plus to v2.0.0-alpha.17
- ruby 2.2.0 is required
- Redmine 3.4 is required
- support sidebar with non default wiki titles (thanks to @danielvijge)

9.6.15 2.0.12

- Provide d3 loader function
- More robust code for dealing with finding data

9.6.16 2.0.11

- i18n methods
- FontAwesome 5.0.13 support

9.6.17 2.0.10

- Remove bootstrap library
- compatibility with https://www.redmine.org/plugins/issue_id
- bug fix: issue and user macro uses absolute url in mailer notification
- Updated marmaid library to version 8.0.0-rc8
- Updated d3 library to 3.5.17
- Updated nvd3 library to latest 1.8.6
- FontAwesome 5.0.12 support
- Set default values for ui-tooltip css class
- ZeroClipboard updated to 2.3.0

9.6.18 2.0.9

- Updated bootstrap library to 4.0.0
- Drop angular_gantt library
- enables deface overwrite directory for all installed plugins (not only additionals)
- Updated d3plus to version v2.0.0-alpha.16
- add “Assign to me” to issues
- add “Status on sidebar” for issues
- add link to create new issue on user profile
- FontAwesome 5.0.8 support
- Add marmaid library

9.6.19 2.0.8

- Provide XLSX helper (and drop XLS helper)
- FontAwesome 5.0.6 support
- add list support for rake task setting_set

9.6.20 2.0.7

- FontAwesome 5.0.2 support
- Switching to SLIM template engine

9.6.21 2.0.6

- add rake tasks: drop_settings, setting_get and setting_set
- Updated nvd3 library to 1.8.6
- Updated angularjs libraries to v2.0.0-rc.1
- Wiki iframe macro integration has been added

9.6.22 2.0.5

- Redmine 3.4 bug fixes
- Helper function fa_icon renamed to font_awesome_icon because of conflicts with redmine_bootstrap_kit
- Cleanups deface overwrites
- add hook for user show
- Traditional Chinese support has been added (thanks to @archonwang)
- Wiki macro for weather with meteoblue has been added
- Wiki macro for google maps has been added
- Wiki macro for issues now supports display a comment and detect issue id and comment id from URL

9.6.23 2.0.4

- Add group_users macro
- Fix bug with REST-API and assigned_id for issues
- Use user name setting for sort order in macros
- Add invisible_captcha spam protection on registration form

9.6.24 2.0.3

- Allow remove watchers without re-adding it (only if author or assigned_user changed)
- Fix sort order of users for change author
- Add uninstall documentation
- Add option to disable autowatch issue at user level
- Fixed bug with recurring_tasks plugin and autowatch issues
- Add more unit tests

9.6.25 2.0.2

- Add option to add involved issue users automatically
- Add change issue author feature
- Fixed bug with Redmine 3.4.x and default assignee settings
- Refactoring patch include and wiki macros

9.6.26 2.0.1

- Simplified Chinese support has been added (thanks to @archonwang)
- Helper function fa_icon has been added
- Help menu item and MyPage menu item does not require application server restart anymore
- Redmine 3.4.x compatibility

9.6.27 2.0.0

- Redmine Tweaks has been renamed to additionals, because to resolve loading order problem of Redmine plugins
- Merge common_libraries plugin into additionals plugin
- Fontawesome support
- Redmine 3.0.x required

9.6.28 1.0.3

- TradingView macro support
- CryptoCompare macro support
- Reddit macro support
- Twitter macro improved with prefix image

9.6.29 1.0.2

- Smiley/Emoji legacy support

9.6.30 1.0.1

- Coding standard cleanups
- ruby 2.1.5 required or newer
- version bump

9.6.31 1.0.0

- user group support for issue auto assign
- optimize deface overwrite path
- drop remove latest projects support (because Redmine 3.2 has dropped latest projects)
- add permission for log time on closed issues - make sure you adjust our permissions!
- code cleanups and bug fixes
- restructure settings
- wiki pdf settings has been added
- updated documentation

9.6.32 0.5.8

- Fixed top menu items permissions for anonymous and non member #29
- Fixed bug with overwriting application handler, which cases problem with other plugins
- Tweaks link added to admin menu
- replaced user macro with {{user}} syntax (old syntax user#id is not supported anymore)
- more formats for user macro and avatar support
- rename list_users to members
- rename list_projects to projects
- new documentation on https://redmine-tweaks.readthedocs.io
- updated bootstrap-datepicker and fixed zh locale problem
- html validation error has been fixed
- remove garfield support (because there is no image source server available)
- slideshare wiki macro has been added
- issue wiki macro has been added
- autoassign issue if no assignee is selected
- n+1 query optimization

9.6.33 0.5.7

- Custom source URL for Garfield source
- Wiki footer bug fixed with missing line break at the end of page
- date period support for calendar macro
- Code cleanups

9.6.34 0.5.6

- Redmine 3.2.x compatibility
- user macro has been added (user#1 or user:admin)
- recently_updated has been added
- lastupdated_by has been added
- lastupdated_at has been added
- calendar macro support
- NoReferrer support has been added
- system information uptime and uname have been added
- twitter macro support
- gist macro support
- vimeo macro support

9.6.35 0.5.5

- dependency with deface (used to overview views)
- fixed garfield caching macro problem
- you can add content to overview page now (top and bottom)
- some content and view optimization (removed wiki_sidebar compatibility problems with other Redmine plugins)
- Code cleanups and refactoring

9.6.36 0.5.4

- issue rule added for closing issue with open sub issues
- issue rule added for status change
- issue rule added for assigned_to change

9.6.37 0.5.3

- Redmine 3.0.x and 3.1.x supported
- “New issue” link with list_projects macro
- Parameter syntax changed for list_users and list_projects macros (sorry for that)

9.6.38 0.5.2

- “Edit closed issue” permission has been added
- Permissions supported for top menu items

9.6.39 0.5.1

- “Hide role in memberbox” has been added

9.6.40 0.5.0

- Redmine 2.6.x compatibility
- URL fixes
- Garfield macro has been added

9.6.41 0.4.9

- added overview text field
- fix style for “goto top”
- added macro overview help page
- fix compatibility problems with sidebar and other plugins

9.6.42 0.4.8

- added youtube macro
- project guide subject can be defined for project overview page

9.6.43 0.4.7

- added jump to top link
- top menu item configuration has been added
- footer configuration (e.g. for imprint url) has been added

9.6.44 0.4.6

- initialize plugins settings now works with other plugins

9.6.45 0.4.5

- option to remove help menu item
- Redmine 2.4.1 required

9.6.46 0.4.4

- installation error fixed
- description update for link handling
- help url now opens in new windows
- sidebar error has been fixed, if no wiki page already exist

9.6.47 0.4.3

- global gantt and calendar bug fix

9.6.48 0.4.2

- no requirements of Wiki extensions plugin anymore

9.6.49 0.4.1

- Fix problem with my page permission

9.6.50 0.4.0

- First public release