

Redmine - Feature #31989

Inline issue auto complete (#) in fields with text-formatting enabled

2019-08-31 14:36 - Marius BĂLTEANU

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	UI	Estimated time:	0.00 hour
Target version:	4.1.0		
Resolution:			
Description			
The attached patch adds support for inline issue auto complete in fields with text-formatting enabled: inline_issue_autocomplete.png As library, I propose tribute.js which is lightweight and has no dependencies.			
Related issues:			
Related to Redmine - Feature #13919: Mention user on issues and wiki pages us...		Closed	
Related to Redmine - Feature #5840: Live lookup of the existing issues based ...		New	2010-07-08
Related to Redmine - Patch #31994: Allow issue auto complete to return 10 iss...		Closed	
Related to Redmine - Feature #31908: Do not load all projects in project jump...		New	
Related to Redmine - Feature #32052: Auto-complete issues #id in search form		Closed	
Related to Redmine - Defect #32450: Attempts to get "tribute.min.js.map" by C...		Closed	
Related to Redmine - Defect #33183: Unable to edit user or group that has cus...		Closed	
Related to Redmine - Feature #33352: Enable inline issue autocomplete in issu...		New	
Related to Redmine - Defect #33846: Inline issue auto complete doesn't saniti...		Closed	
Related to Redmine - Feature #33820: Auto complete wiki page links		Closed	
Related to Redmine - Feature #33425: Auto complete wiki macros		New	
Has duplicate Redmine - Feature #4325: Quick Issue Reference		Closed	2009-12-03
Has duplicate Redmine - Feature #1461: Issues linking by a combo box or similar		Closed	2008-06-16

Associated revisions

Revision 18444 - 2019-09-09 23:49 - Go MAEDA

Inline auto complete for issue (#) in wiki-edit fields (#31989).

Patch by Marius BALTEANU.

Revision 19331 - 2019-12-03 10:15 - Go MAEDA

Merged r19221 from trunk to 4.1-stable (#32450, #31989).

History

#1 - 2019-09-01 05:22 - Go MAEDA

- Related to Feature #13919: Mention user on issues and wiki pages using @user with autocomplete added

#2 - 2019-09-01 06:13 - Go MAEDA

- Related to Feature #5840: Live lookup of the existing issues based on the keywords entered in the title for new issue added

#3 - 2019-09-01 08:49 - Go MAEDA

Thank you for posting the nice patch.

The feature is interesting, but I think many users don't find the feature because they don't type other than digits after '#' sign.

I am not sure it is OK to add tribute.js to implement the hidden trick. But I think it is OK to add if tribute.js is used for other features such as [#13919](#) as well.

#4 - 2019-09-01 15:11 - Marius BĂLTEANU

Go MAEDA wrote:

Thank you for posting the nice patch.

Thanks for reviewing the patch.

The feature is interesting, but I think many users don't find the feature because they don't type other than digits after '#' sign.

That is a problem because we force the users to remember the ID of the issue which could be quite a big number for some instances (in my case, last issue id is 170787). When I'm creating or reading an issue, I'll remember the subject (or at least some keywords from the subject) and not the ID. In most of the cases, when I want to reference an issue, I open a new tab, I search for the issue and then I copy the ID (it requires multiple clicks/steps). I'm sure that users will find this feature very useful and also, they still can use the existing behaviour.

I am not sure it is OK to add `tribute.js` to implement the hidden trick. But I think it is OK to add if `tribute.js` is used for other features such as [#13919](#) as well.

I won't call it hidden trick, it's very common now days to have an inline auto complete to help you find references (all major platforms have GitHub, GitLab, Jira, Twitter, Facebook, etc). And yes, I'm already working on adding support for user auto completes (this was the initial use case I had in mind, but needs a little bit more work and I preferred to post for issues in order to get some feedback).

#5 - 2019-09-01 15:44 - Go MAEDA

- *Target version set to Candidate for next major release*

Marius BALTEANU wrote:

I won't call it hidden trick, it's very common now days to have an inline auto complete to help you find references (all major platforms have GitHub, GitLab, Jira, Twitter, Facebook, etc)

Oh, I didn't know that GitHub has '#' + keyword auto-complete feature! I have just confirmed that GitHub has the same feature as this patch. I withdraw the words "hidden trick".

I think there is no problem to add a new JS library only for this patch if users can easily find this feature.

#6 - 2019-09-01 15:57 - Go MAEDA

In order to let users know this feature, I think it would be even better if some issues are displayed just after typing '#', as GitHub does.

#7 - 2019-09-01 16:04 - Marius BALTEANU

Go MAEDA wrote:

In order to let users know this feature, I think it would be even better if some issues are displayed just after typing '#', as GitHub does.

Yes, totally agree, it is in progress as well, sorry for not saying all the features that I've in plan from the begging.

#8 - 2019-09-02 01:56 - Mizuki ISHIKAWA

+1

I think this feature is great!!!

If the `tribute.js` library is introduced, it seems that it can be used for many improvements such as autocomplete of mentions.

#9 - 2019-09-02 07:33 - Go MAEDA

- *Related to Patch #31994: Allow issue auto complete to return 10 issues when there is not search term provided added*

#10 - 2019-09-02 11:16 - Bernhard Rohloff

+1 Very nice improvement. This could also lead to improvements for the project jump box like the one requested in [#31908](#).

#11 - 2019-09-02 11:16 - Bernhard Rohloff

- *Related to Feature #31908: Do not load all projects in project jump box added*

#12 - 2019-09-02 11:41 - Go MAEDA

- *Target version changed from Candidate for next major release to 4.1.0*

Go MAEDA wrote:

In order to let users know this feature, I think it would be even better if some issues are displayed just after typing '#', as GitHub does.

We can use this feature by committing [#31994](#). Great!

#13 - 2019-09-02 19:30 - Marius BĂLTEANU

- File *0001-Inline-auto-complete-for-issue-and-user-in-wiki-edit.patch* added

- File *user.png* added

- File *issue.png* added

- Subject changed from *Inline issue auto complete in fields with text-formatting enabled* to *Inline issue and user auto complete in fields with text-formatting enabled*

Attached an updated patch that implements issue and user auto complete in text-formatting fields. My initial patch for issue is not 100% fully functional because it doesn't send the `project_id` param to the auto complete endpoint. In order to fix this, is require to have a method to pass the `auto_complete_issues_path(:project_id => <id>)` (or other auto complete endpoints) to JavaScript and the method chosen by me was to set those endpoints in `textarea` field as data attributes. This is way the patch touch multiple views. Another option is to pass those variables through `wikitoolbar_for`, but then the feature will depend on JSToolbar.

Any feedback on the proposed solution is welcome.

Screenshots:

`issue.png`

`user.png`

#14 - 2019-09-03 02:58 - Go MAEDA

- How about getting the project identifier from the `body["class"]` element (`<body class="theme-Default project-ecookbook has-main-menu ...">`)? We don't have to avoid updating multiple views?
- I think `user user` auto-complete should be implemented along with [#13919](#) because it is a typical UI of user mentioning feature. Users may misunderstand that they can mention users by using `user user` auto-complete if `user user` auto-complete is implemented

#15 - 2019-09-03 08:34 - Marius BĂLTEANU

- File *0001-Inline-auto-complete-for-issue-and-user-in-wiki-edit.patch* added

Go MAEDA wrote:

- How about getting the project identifier from the `body["class"]` element (`<body class="theme-Default project-ecookbook has-main-menu ...">`)? We don't have to avoid updating multiple views?

I wish to be so easy, but it isn't. The project can change inside the form, you can take as example a project with subprojects where the user can change the project from the dropdown or the global view. Of course, we can do some kind of magic from JS to look for the project first inside the form (if exists) and then in body class or in a global JS variable, but it is no an elegant solution from my point of view and it wasn't used in any other implementation (observeAutocompleteField or preview tab). We can do in the future some refactorization to avoid touching all the views, but it is another story.

- I think `user user` auto-complete should be implemented along with [#13919](#) because it is a typical UI of user mentioning feature. Users may misunderstand that they can mention users by using `user user` auto-complete if `user user` auto-complete is implemented

Indeed, it'll be very nice to deliver the features together, but not a must because `User First Name User Last Name` syntax is already available and [#13919](#) was proposed without an auto-complete feature. IMHO, the confusion exists and in the worst case, users report more comments or defects on this topic. Anyway, your call on this, I can split again the patches, just let me know. Regarding [#13919](#), I have in plan to review the technical solution, but I'm not sure if I can do it this week.

In the meantime, I've slightly updated the patch in order to call `inlineAutoComplete` when user focuses a field with attribute `[data-auto-complete=true]`. In this way, it is no longer required to call `inlineAutoComplete` after new elements are added in the page dynamically through Ajax.

#16 - 2019-09-05 13:54 - Go MAEDA

I think it is better to add `requireLeadingSpace: true` in order to prevent the auto-complete menu from being displayed while typing the second '#' in "#123#note-4" or "@" in "[foo@example.com](#)".

```
diff --git a/public/javascripts/application.js b/public/javascripts/application.js
index 9579ee1f2..9d874b5ed 100644
--- a/public/javascripts/application.js
```

```
+++ b/public/javascripts/application.js
@@ -1050,6 +1050,7 @@ function inlineAutoComplete(element) {
    },
    lookup: 'label',
    fillAttr: 'label',
+   requireLeadingSpace: true,
    selectTemplate: function (issue) {
        return '#' + issue.original.id;
    }
@@ -1067,6 +1068,7 @@ function inlineAutoComplete(element) {
    menuItemTemplate: function (user) {
        return user.original.name;
    },
+   requireLeadingSpace: true,
    selectTemplate: function (user) {
        return '@' + user.original.login;
    }
}
```

#17 - 2019-09-06 04:24 - Go MAEDA

I am hesitating to commit the patch because auto-complete feature for users was added. If the patch had not had the auto-complete for users, I had committed the patch without any hesitation.

As I wrote in [#31989#note-14](#), I am afraid that the feature looks as if users can send mentions because the UI is a typical interface to send mentions in other apps.

Marius, I think the feature is great. But, IMHO, it is a bit early to deliver. Could you separate the feature from the patch? It is ideal if we can deliver the feature together with [#13919](#).

#18 - 2019-09-06 04:37 - Mizuki ISHIKAWA

Go MAEDA wrote:

It is ideal if we can deliver the feature together with [#13919](#).

I agree with this opinion.

I also associate mention feature from [user user](#) format as Go MAEDA says.

#19 - 2019-09-09 20:04 - Marius BĂLTEANU

- File *0001-Inline-auto-complete-for-issue-in-wiki-edit-fields.patch* added

- Subject changed from *Inline issue and user auto complete in fields with text-formatting enabled* to *Inline issue auto complete (#) in fields with text-formatting enabled*

I've removed the user autocomplete feature from the patch.

Go MAEDA wrote:

I think it is better to add `requireLeadingSpace: true` in order to prevent the auto-complete menu from being displayed while typing the second "#" in "#123#note-4" or "@" in "[foo@example.com](#)".

Agree, added to the patch.

#20 - 2019-09-09 23:50 - Go MAEDA

- Status changed from *New* to *Closed*

- Assignee set to *Go MAEDA*

Committed the patch. Thank you for contributing such a great enhancement!

#21 - 2019-09-12 12:07 - Go MAEDA

- Related to Feature [#32052](#): *Auto-complete issues #id in search form* added

#22 - 2019-11-14 00:35 - Go MAEDA

- Related to Defect [#32450](#): *Attempts to get "tribute.min.js.map" by Chrome DevTool causes RoutingError.* added

#23 - 2019-12-20 10:10 - Jean-Philippe Lang

- Tracker changed from *Patch* to *Feature*

#24 - 2020-01-13 06:44 - Go MAEDA

- Has duplicate Feature #4325: Quick Issue Reference added

#25 - 2020-03-23 09:50 - Go MAEDA

- Related to Defect #33183: Unable to edit user or group that has custom fields with text formatting enabled added

#26 - 2020-04-25 10:13 - Mischa The Evil

- Related to Feature #33352: Enable inline issue autocomplete in issue id query filter fields added

#27 - 2020-10-05 00:49 - Marius BĂLTEANU

- Related to Defect #33846: Inline issue auto complete doesn't sanitize HTML tags added

#28 - 2020-10-13 07:39 - Marius BĂLTEANU

- Related to Feature #33820: Auto complete wiki page links added

#29 - 2020-10-13 07:39 - Marius BĂLTEANU

- Related to Feature #33425: Auto complete wiki macros added

#30 - 2022-07-10 09:32 - Go MAEDA

- Has duplicate Feature #1461: Issues linking by a combo box or similar added

Files

0001-Inline-issue-auto-complete-in-wiki-edit-fields.patch	38 KB	2019-08-31	Marius BĂLTEANU
inline_issue_autocomplete.png	253 KB	2019-08-31	Marius BĂLTEANU
0001-Inline-auto-complete-for-issue-and-user-in-wiki-edit.patch	57.2 KB	2019-09-02	Marius BĂLTEANU
user.png	12.3 KB	2019-09-02	Marius BĂLTEANU
issue.png	22.4 KB	2019-09-02	Marius BĂLTEANU
0001-Inline-auto-complete-for-issue-and-user-in-wiki-edit.patch	56.8 KB	2019-09-03	Marius BĂLTEANU
0001-Inline-auto-complete-for-issue-in-wiki-edit-fields.patch	48.3 KB	2019-09-09	Marius BĂLTEANU