

Redmine - Feature #33102

Import user accounts from CSV file

2020-03-06 07:49 - Go MAEDA

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	Importers	Estimated time:	0.00 hour
Target version:	4.2.0		
Resolution:	Fixed		
Description			
<p>It is a time-consuming task to add dozens of user accounts when launching a new Redmine server.</p> <p>I think it can be resolved if Redmine has a feature to import user accounts from a CSV file. I often get requests for this feature from my customers.</p>			

Associated revisions

Revision 19799 - 2020-05-30 06:53 - Go MAEDA

Import user accounts from CSV file (#33102).

Patch by Takenori TAKAKI.

Revision 19800 - 2020-05-30 07:09 - Go MAEDA

Update locales (#33102).

Revision 19802 - 2020-06-02 16:29 - Go MAEDA

Add fixtures for UserImportTest (#33102).

History

#1 - 2020-04-07 18:02 - Takenori TAKAKI

- File `csv_user_import.patch` added

I made a patch to achive this feature in the same way as Importing Issue, Importing TimeEntry.

Admin users will be able to import the following information from CSV.

- Login
- Firstname, Lastname
- Email
- Language (e.g. 'en')
- Administrator (Yes / No)
- Authentication mode
- Password
- Must change password at next logon (Yes / No)
- Status (active / registered / locked)
- Custom Fields

#2 - 2020-04-28 17:12 - Go MAEDA

- Target version set to Candidate for next major release

#3 - 2020-05-04 06:37 - Go MAEDA

- Target version changed from Candidate for next major release to 4.2.0

Setting the target version to 4.2.0.

#4 - 2020-05-07 04:08 - Go MAEDA

- Target version changed from 4.2.0 to Candidate for next major release

Thank you for the patch but imports_test.rb fails after applying the patch. Could you look into this?

Failure:

RoutingImportsTest#test_imports [/Users/maeda/redmines/trunk/test/test_helper.rb:313]:

The generated path </issues/imports/new> did not match </users/imports/new>.

Expected: "/users/imports/new"

Actual: "/issues/imports/new"

bin/rails test test/integration/routing/imports_test.rb:23

#5 - 2020-05-08 05:11 - Takenori TAKAKI

- File csv_user_import-002.patch added

Thank you, Mr. Maeda, for pointing out the bug.

The test code I tried during development was still in the patch...

I will re-post a patch that removes the unnecessary test code.

#6 - 2020-05-08 08:09 - Go MAEDA

- Target version changed from Candidate for next major release to 4.2.0

#7 - 2020-05-08 09:31 - Marius BALTEANU

Nice feature, I've only one question: Why do you rescue the templates (rescue nil)?

#8 - 2020-05-09 06:09 - Takenori TAKAKI

Marius BALTEANU wrote:

| *Nice feature, I've only one question: Why do you rescue the templates (rescue nil)?*

For this feature, I thought there was no information that should be displayed using the sidebar partial.

So I added `rescue nil` to make the template work with or without the sidebar partial.

#9 - 2020-05-12 04:52 - Go MAEDA

Takenori TAKAKI wrote:

Marius BALTEANU wrote:

Nice feature, I've only one question: Why do you rescue the templates (rescue nil)?

For this feature, I thought there was no information that should be displayed using the sidebar partial.

So I added `rescue nil` to make the template work with or without the sidebar partial.

How about avoiding catching all exceptions by checking the existence of the partial with [ActionView::LookupContext::ViewPaths#exists?](#) ?

I have confirmed that the following experimental code works as expected. Perhaps we can define a helper method like `partial_exists?` or `render_if_exists`.

```
diff --git a/app/views/imports/new.html.erb b/app/views/imports/new.html.erb
index e7ca82428..4e9b89ffb 100644
--- a/app/views/imports/new.html.erb
+++ b/app/views/imports/new.html.erb
@@ -12,4 +12,4 @@
   <p><%= submit_tag l(:label_next).html_safe + " &#187;".html_safe, :name => nil %></p>
   <% end %>

   <%= render :partial => "#{import_partial_prefix}_sidebar" %>
+   <%= render :partial => "#{import_partial_prefix}_sidebar" if lookup_context.exists?("#{import_partial_prefix}_sidebar",
lookup_context.prefixes, true) %>
```

#10 - 2020-05-13 08:44 - Takenori TAKAKI

- File `csv_user_import-003.patch` added

Go MAEDA wrote:

How about avoiding catching all exceptions by checking the existence of the partial with [ActionView::LookupContext::ViewPaths#exists?](#) ?

I have confirmed that the following experimental code works as expected. Perhaps we can define a helper method like `partial_exists?` or `render_if_exists`.

I think the proposed code is a more polite implementation.

I created a patch that defines `ImportsHelper#render_if_exists`, how do you like it?

#11 - 2020-05-17 15:43 - Marius BALTEANU

The `render_if_exists` method is much better than `rescue nil`, but to be honest, I don't see a real value for users to show the issues/time entries sidebar during import wizard, especially after we removed the navigations links from the sidebar (#30294, #315980). My recommendation is just to remove the sidebar, it's simplify the code.

But if we want to keep the sidebar, then let's render the administration sidebar in order to be consistent with issues/time entries.

Also, the new route added by the patch should be covered by a test in test/integration/routing/imports_test.rb.

Regarding the functionality, I'm wondering If we should support "Generate password" and "Send information to users". From a security/privacy point of view, I think it will nice to import users without defining a default password for them (even if you are an administrator). This can be added later if we agree.

#12 - 2020-05-17 17:55 - Takenori TAKAKI

Marius, thank you for your feedback.

Marius BALTEANU wrote:

The render_if_exists method is much better than rescue nil, but to be honest, I don't see a real value for users to show the issues/time entries sidebar during import wizard, especially after we removed the navigations links from the sidebar (#30294, #315980). My recommendation is just to remove the sidebar, it's simplify the code.

I agree with Marius on the value of the sidebar.

But if we want to keep the sidebar, then let's render the administration sidebar in order to be consistent with issues/time entries.

I use the 'layout/admin' on the patch because this feature is for admins. Therefore, the current patch already renders the administration sidebar.

#13 - 2020-05-17 18:39 - Marius BALTEANU

Takenori TAKAKI wrote:

I agree with Marius on the value of the sidebar.

Let's wait for Go Maeda feedback on this.

I use the 'layout/admin' on the patch because this feature is for admins. Therefore, the current patch already renders the administration sidebar.

You're right, sorry for the mistake.

#14 - 2020-05-18 17:25 - Go MAEDA

Marius BALTEANU wrote:

The render_if_exists method is much better than rescue nil, but to be honest, I don't see a real value for users to show the issues/time entries sidebar during import wizard, especially after we removed the navigations links from the sidebar (#30294, #315980). My recommendation is just to remove the sidebar, it's simplify the code.

I agree that it is hard to imagine the situation that users need the content in the sidebar while using the CSV Import Wizard. However, I am not sure if we can remove the sidebar because some third-party plugins may suppose the existence of sidebar or use hooks such as view_issues_sidebar_planning_bottom and view_issues_sidebar_queries_bottom.

#15 - 2020-05-28 06:46 - Takenori TAKAKI

Go MAEDA wrote:

I agree that it is hard to imagine the situation that users need the content in the sidebar while using the CSV Import Wizard. However, I am not sure if we can remove the sidebar because some third-party plugins may suppose the existence of sidebar or use hooks such as `view_issues_sidebar_planning_bottom` and `view_issues_sidebar_queries_bottom`.

I understand that there is a need to discuss the deletion of the sidebar during import wizard. On the other hand, I thought it would be good to discuss separately from the feature addition(Import user accounts from CSV file).

Go Maeda, what do you think?

#16 - 2020-05-28 09:09 - Marius BALTEANU

Takenori TAKAKI wrote:

Go MAEDA wrote:

I agree that it is hard to imagine the situation that users need the content in the sidebar while using the CSV Import Wizard. However, I am not sure if we can remove the sidebar because some third-party plugins may suppose the existence of sidebar or use hooks such as `view_issues_sidebar_planning_bottom` and `view_issues_sidebar_queries_bottom`.

I understand that there is a need to discuss the deletion of the sidebar during import wizard. On the other hand, I thought it would be good to discuss separately from the feature addition(Import user accounts from CSV file).

Go Maeda, what do you think?

You're right, let's move the discussion about the sidebar in another ticket.

Regarding your patch, it looks good to me, one minor change that we can do is to move the new method to ApplicationHelper because the method is quite general and it can be used in other pages as well.

#17 - 2020-05-28 12:06 - Takenori TAKAKI

- File `csv_user_import-004.patch` added

Marius BALTEANU wrote:

You're right, let's move the discussion about the sidebar in another ticket.

Thanks for your comment and agreement!

Regarding your patch, it looks good to me, one minor change that we can do is to move the new method to ApplicationHelper because the method is quite general and it can be used in other pages as well.

I think it's better that way.

I made a patch that moved the new method to ApplicationHelper.

#18 - 2020-05-28 15:00 - Go MAEDA

Thank you for making the patch even more sophisticated. I think this patch should be merged into the core.

#19 - 2020-05-30 06:54 - Go MAEDA

- Status changed from New to Closed

- Assignee set to Go MAEDA

- Resolution set to Fixed

Committed the patch. Thank you for writing the patch for this feature.

#20 - 2020-06-02 16:01 - Go MAEDA

- Status changed from Closed to Reopened

user_import_test.rb randomly fails due to missing fixtures. It can be fixed with the following patch.

```
diff --git a/test/unit/user_import_test.rb b/test/unit/user_import_test.rb
```

```
index fe1201dff..b790aeb98 100644
```

```
--- a/test/unit/user_import_test.rb
```

```
+++ b/test/unit/user_import_test.rb
```

```
@@ -20,6 +20,8 @@
```

```
require File.expand_path('../test_helper', __FILE__)
```

```
class UserImportTest < ActiveSupport::TestCase
```

```
+ fixtures :users, :auth_sources, :custom_fields
```

```
+ 
```

```
include Redmine::I18n
```

```
def setup
```

#21 - 2020-06-03 15:13 - Go MAEDA

- Status changed from Reopened to Closed

Go MAEDA wrote:

| *user_import_test.rb randomly fails due to missing fixtures. It can be fixed with the following patch.*

Committed the fix in r19802.

Files

csv_user_import.patch

17.8 KB

2020-04-07

Takenori TAKAKI

csv_user_import-002.patch	17.2 KB	2020-05-08	Takenori TAKAKI
csv_user_import-003.patch	19.5 KB	2020-05-13	Takenori TAKAKI
csv_user_import-004.patch	18.8 KB	2020-05-28	Takenori TAKAKI