Redmine - Feature #33418

Bulk addition of related issues

2020-05-08 12:40 - Dmitry Makurin

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Marius BĂLTEANU	**% Done:**	0%
Category:	Issues	**Estimated time:**	0.00 hour
Target version:	4.2.0		
Resolution:	Fixed		

Description

Sometimes you need to add a lot of relations to an issue and it's getting really hard when there're 50 or more relations.

The patch provides bulk create for relations. Fill an input field with ids (delimited with comma) and press form's button.

exampleexample

i7kB9J3qsk2.gif

Related issues:

Related to Redmine - Feature #29272: Create several relations at the same tim...	**New**
Related to Redmine - Defect #35039: API create issue relation method returns ...	**Closed**
Related to Redmine - Defect #35655: Create duplicated follows relations fails...	**Closed**

Associated revisions

Revision 20689 - 2020-12-25 03:25 - Go MAEDA

Bulk addition of related issues (#33418).

Patch by Dmitry Makurin and Marius BALTEANU.

Revision 20690 - 2020-12-25 03:31 - Go MAEDA

Allow issue relation autocomplete to select multiple values (#33418).

Patch by Marius BALTEANU.

History

#1 - 2020-05-09 21:51 - Robert Korulczyk

+1. I miss this feature so much...

#2 - 2020-05-16 04:20 - Go MAEDA

- Target version set to Candidate for next major release

#3 - 2020-06-25 15:08 - Go MAEDA

- Related to Feature #29272: Create several relations at the same time via Issue Relations API added

#4 - 2020-09-13 04:15 - Go MAEDA

- Category set to Issues

#5 - 2020-09-29 10:23 - Go MAEDA

I tried out the patch and found it really nice.

The only one small issue of the patch is that issue auto-complete feature does not work. But I think we can resolve it by using Tribute.js which is used for inline auto complete feature (#31989).

#6 - 2020-10-20 06:24 - Go MAEDA

Go MAEDA wrote:

> The only one small issue of the patch is that issue auto-complete feature does not work. But I think we can resolve it by using Tribute.js which is used for inline auto complete feature (#31989).

I realized that implementing the feature with Tribute.js is not so easy because the auto-complete should be triggered without typing a preceding "#" in the related issues section.

Now I think it is OK even if the feature does not support auto-comple for multiple issue ids because it is a trivial matter compared to the usefulness of this feature.

#7 - 2020-12-06 01:37 - Marius BĂLTEANU

- Assignee set to Marius BĂLTEANU

#8 - 2020-12-12 12:08 - Marius BĂLTEANU

- File 0002-Allow-issue-relation-autocomplete-to-select-multiple.patch added

- File 0001-Rework-patch-from-33418.patch added

Dmitry, thanks for posting the patch. I find this feature really useful.

Regarding your proposed solution, I don't think that it's a good ideea to remove the validation from the model because it will allow setting a non visible issue if you by-pass the controller (from a plugin, for example).

I'm attaching two patches:

1. 0001-Rework-patch-from-33418.patch
Based on the existing patch, but keeping the validation in the model and also fixes some edge-case bugs (for ex: no error if you press Add without typing the issue id).

2. 0002-Allow-issue-relation-autocomplete-to-select-multiple
Adds a new JS autocomplete function that allows searching for multiple values. If the solution is good enough, I'll add a system test for it.

#9 - 2020-12-14 03:49 - Go MAEDA

Marius, thank you for improving the patch. I tried out the patches #33418#note-8 and found they are really nice.

Marius BALTEANU wrote:

> **2. 0002-Allow-issue-relation-autocomplete-to-select-multiple**
> Adds a new JS autocomplete function that allows searching for multiple values. If the solution is good enough, I'll add a system test for it.

I didn't feel strange compared to the autocomplete in Redmine 4.1.0.

#10 - 2020-12-17 00:15 - Marius BĂLTEANU

- File deleted (0002-Allow-issue-relation-autocomplete-to-select-multiple.patch)

#11 - 2020-12-17 00:16 - Marius BĂLTEANU

- File 0002-Allow-issue-relation-autocomplete-to-select-multiple.patch added

- Assignee deleted (Marius BĂLTEANU)

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

Updated the second patch to include a test.

Let's discuss this for 4.2.0.

#12 - 2020-12-25 03:32 - Go MAEDA

- Tracker changed from Patch to Feature

- Subject changed from Issue relation bulk create to Bulk addition of related issues

- Status changed from New to Closed

- Assignee set to Go MAEDA

- Resolution set to Fixed

Committed the patch. Thank you for improving Redmine.

#13 - 2020-12-25 05:34 - Go MAEDA

- Related to deleted (Feature #29272: Create several relations at the same time via Issue Relations API)

#14 - 2020-12-25 05:35 - Go MAEDA

- Has duplicate Feature #29272: Create several relations at the same time via Issue Relations API added

#15 - 2020-12-25 05:36 - Go MAEDA

- Has duplicate deleted (Feature #29272: Create several relations at the same time via Issue Relations API)

#16 - 2020-12-25 05:36 - Go MAEDA

- Related to Feature #29272: Create several relations at the same time via Issue Relations API added

#17 - 2020-12-26 14:12 - Marius BĂLTEANU

- File accept_issue_to_id_as_array.patch added

- Status changed from Closed to Reopened

- Assignee deleted (Go MAEDA)

I'm reopening this to discuss about the API part which is not implemented 100% correctly.

1. Request:
Currently, for bulk addition of related issues, the request should contains the ids as comma-delimited string, please see the below example (param issue_to_id):

```
post(
  '/issues/2/relations.xml',
  :params => {:relation => {:issue_to_id => "7, 8", :relation_type => 'relates'}}
)
```

I think we should accept also an array of ids:

```
post(
  '/issues/2/relations.xml',
  :params => {:relation => {:issue_to_id => [7, 8], :relation_type => 'relates'}}
)
```

The attached patch allows this.

2. Response:
Currently, the response contain only the latest relation added:

```
{
  "relation": {
    "id": 107,
    "issue_id": 2,
    "issue_to_id": 8,
    "relation_type": "relates",
    "delay": null
  }
}
```

I think the response should contain all the created relations:

```
{
  "relations": [
    {
      "id": 114,
      "issue_id": 2,
      "issue_to_id": 7,
      "relation_type": "relates",
      "delay": null
    },
    {
      "id": 115,
      "issue_id": 2,
      "issue_to_id": 8,
      "relation_type": "relates",
```

```
      "delay": null
    }
  ]
}
```

but I'm not sure how to handle the case when the response contain both successfully created relations and non-successfully relations. From what I know, we don't have this kind of mixed API response in Redmine.

I see two options:
1. We change the current implementation to be transactional. In this way, we reject the entire request if an error occurs.
2. We respond with succes if at least one relation was created successfully and we append the error messages to the response:

```
{
  "relations": [
    {
      "id": 114,
      "issue_id": 2,
      "issue_to_id": 7,
      "relation_type": "relates",
      "delay": null
    },
    {
      "id": 115,
      "issue_id": 2,
      "issue_to_id": 8,
      "relation_type": "relates",
      "delay": null
    }
  ],
  "errors": [
    "Related issue has already been taken: #3",
  ]
}
```

Any feedback is welcome!

#18 - 2021-03-23 07:28 - Marius BĂLTEANU

- Assignee set to Marius BĂLTEANU

#19 - 2021-03-27 07:55 - Marius BĂLTEANU

- Status changed from Reopened to Closed

We will postpone the fixes for API to another version.

#20 - 2021-04-05 19:02 - Alex Last

hi everyone! I am the owner of redmine-java-api project (https://github.com/taskadapter/redmine-java-api).
I get a server error with Redmine 4.2.0 when trying to create a relation with the request body that previously worked for Redmine 3.x:

```
{"relation"=>{"issue_to_id"=>576, "relation_type"=>"relates"}, "issue_id"=>"575"}
```

the error is:

```
NoMethodError (undefined method `split' for 576:Integer):
```

because **issue_to_id** is now expected to be a string with commas, instead of a number.

is this a permanent change or just an experiment? it would be too expensive to maintain compatibility for multiple redmine versions in the java library.

#21 - 2021-04-05 19:21 - Alex Last

side note - if Redmine REST API supported returning some "server info", the clients could at least choose what json blob to send (for redmine 3.x, 4.x, etc). but since this is not supported, there is no good way to support several redmine REST formats in clients.

#22 - 2021-04-05 21:30 - Marius BĂLTEANU

- Related to Defect #35039: API create issue relation method returns undefined method `split' when issue id is sent as integer added

#23 - 2021-04-05 21:33 - Marius BĂLTEANU

Thanks Alex Last for reporting this issue. It's a defect, I've created #35039 to track this, the fix will be available in 4.2.1.

#24 - 2021-08-01 15:41 - Marius BĂLTEANU

- Related to Defect #35655: Create duplicated follows relations fails with 500 internal error added

#25 - 2024-08-19 15:25 - Mathias Walter

Is there an issue for creating the relationship the other way around (select multiple issues in the issue list and add the relationship via context menu or editing the issues)?

Files

relations_bulk_create.patch	5.18 KB	2020-05-08	Dmitry Makurin
i7kB9J3qsk2.gif	411 KB	2020-05-08	Dmitry Makurin
0001-Rework-patch-from-33418.patch	6.4 KB	2020-12-12	Marius BĂLTEANU
0002-Allow-issue-relation-autocomplete-to-select-multiple.patch	4.18 KB	2020-12-16	Marius BĂLTEANU
accept_issue_to_id_as_array.patch	822 Bytes	2020-12-26	Marius BĂLTEANU