Redmine - Feature #3322

Setting to restrict spent times on future dates

2009-05-08 14:16 - Nanda P

Status:	Closed	**Start date:**	2009-05-08
Priority:	Normal	**Due date:**	
Assignee:	Go MAEDA	**% Done:**	0%
Category:	Time tracking	**Estimated time:**	0.00 hour
Target version:	4.1.0		
Resolution:	Fixed		

Description

When entering Spent time the date field allows future date.

Can we have a validation & a warning message?

Related issues:

Related to Redmine - Feature #13596: Add setting to allow/deny time logging o...	**Closed**
Related to Redmine - Feature #13244: Restrict log time for old days	**Reopened**
Has duplicate Redmine - Defect #14840: Time logging shouldn't be possible for...	**Closed**

Associated revisions

Revision 17719 - 2018-12-11 04:55 - Go MAEDA

Setting to restrict spent times on future dates (#3322).

Patch by Marius BALTEANU.

Revision 17720 - 2018-12-11 05:22 - Go MAEDA

Update locales (#3322).

History

#1 - 2009-05-12 18:52 - Jean-Philippe Lang

If I had this validation, I'm pretty sure that some users will complain about it.
Any other opinions?

#2 - 2009-05-14 21:04 - Olafur Gislason

One possibility is a checkbox in project settings page that would turn on/off that validation.

This could in my opinion also apply to logging time on a closed issue. (The checkbox).

#3 - 2009-05-19 20:29 - Eric Davis

I don't like the idea of adding another option to the project settings page. What if when a time was logged that is in the future, we display a message along the lines of "Time added successfully. This time was logged to a future date". That would at least make it visible to the user that they might have made a typo.

#4 - 2009-05-19 20:53 - Nanda P

I agree with the warning message.

#5 - 2010-12-30 05:35 - Mischa The Evil

- Tracker changed from Defect to Feature

- Subject changed from Spent time allows Future date" to Spent time allows Future date

#6 - 2016-08-27 07:04 - Go MAEDA

- Is duplicate of Feature #13244: Restrict log time for old days added

#7 - 2016-08-27 07:06 - Go MAEDA

- Is duplicate of deleted (Feature #13244: Restrict log time for old days)

#8 - 2016-08-27 07:06 - Go MAEDA

- Has duplicate Feature #13244: Restrict log time for old days added

#9 - 2016-08-27 07:07 - Go MAEDA

- Has duplicate Defect #14840: Time logging shouldn't be possible for the future periods added

#10 - 2016-08-29 08:05 - Toshi MARUYAMA

- Related to Feature #13596: Add setting to allow/deny time logging on closed issues added

#11 - 2016-09-09 12:12 - Toshi MARUYAMA

#13244#note-16 has patch.

#12 - 2016-09-10 09:24 - Marius BĂLTEANU

Jean-Philippe Lang wrote:

> If I had this validation, I'm pretty sure that some users will complain about it.
> Any other opinions?

I do not see an use case for adding spent times on future dates. In my opinion, a spent time added for tomorrow (for example) is just an estimation.

#13 - 2017-05-24 18:08 - Kirill Marchuk

first of all, how is it that this FR duplicates #13244 ?

These are totally different things!

This issue goes about restricting time logging in FUTURE, to avoid cheating or mistakes.

Issue #13244 has it about restricting time logged on days, that are X and more days in the past, to ensure employees discipline and to avoid "forgetting to log time" issues

what can I as a Redmine user do, to have these 2 issues decoupled and let #13244 pass thru the pipeline and get merged into upstream Redmine ?

#14 - 2017-06-23 14:58 - Toshi MARUYAMA

- Has duplicate deleted (Feature #13244: Restrict log time for old days)

#15 - 2017-06-23 14:58 - Toshi MARUYAMA

- Related to Feature #13244: Restrict log time for old days added

#16 - 2018-04-02 21:56 - Marius BĂLTEANU

- File 0001-setting-to-not-allow-spent-times-on-future-dates.patch added

I made a patch that adds a new setting to the Time Tracking tab. Admins can now allows / or do not allows time logs on future dates. The default setting is to allow in order to not change the current behaviour.

#17 - 2018-04-03 08:36 - Go MAEDA

Marius BALTEANU wrote:

> I made a patch that adds a new setting to the Time Tracking tab. Admins can now allows / or do not allows time logs on future dates. The default setting is to allow in order to not change the current behaviour.

Fine improvement! But I think it will be even better if the error is more specific, for example, "Future date is not allowed" instead of "Date invalid".

#18 - 2018-04-03 09:39 - Marius BĂLTEANU

- File 0001-setting-to-not-allow-spent-times-on-future-dates.patch added

Go MAEDA wrote:

> Fine improvement! But I think it will be even better if the error is more specific, for example, "Future date is not allowed" instead of "Date invalid".

Totally agree, I had this in mind after I retested my patch in this morning. Here is the updated one.

#19 - 2018-04-03 09:39 - Marius BĂLTEANU

- File deleted (0001-setting-to-not-allow-spent-times-on-future-dates.patch)

#20 - 2018-04-03 09:46 - Go MAEDA

- Target version set to Candidate for next major release

#21 - 2018-04-03 14:12 - Mischa The Evil

Marius Ionescu and @Go: please hold the presses on this issue/patch. I'll elaborate below.

Over the last year (!), I've been working – silenty, and in tiny iterations – on an extensive (last count: 124 local development commits, 1200+ diff-LOC) patch series for a third-party, covering four whole new timelog restrictions (among one covers this particular issue) and extensions over the two restrictions already implemented for #24005, in one coherent, consistent and tested implementation, where each restriction comes with its own project, user, group, role and administrator exclusion/inclusion settings.
This work-in-progress is currently (and finally) in its final stages and thus nearing completion. This would also come with proper, detailed documentation.

I think it's only a matter of weeks utmost before all the remaining 'work' (it's already feature complete) will be completed. Therefore, I'd like to ask you all sincerely to wait with any potential core integration of any patch for this issue (or issue's #13244 and #13596, for that matter) until I've been able to open-source (i.e. share) the finished patch series in a proper manner, here on redmine.org publicly.

Thanks in advance...

P.S.: Marius Ionescu, this is in no way meant as a review of your patch(es)... ;)
P.P.S.: off-topic note to self: review and/or change future strategy regarding issue-assignment to prevent other users, repeatedly, working simultaneously on the same issues as I am, thus wasting my or their time in the end.

#22 - 2018-04-03 16:20 - Marius BĂLTEANU

Mischa The Evil wrote:

> Marius Ionescu and @Go: please hold the presses on this issue/patch. I'll elaborate below.
>
> Over the last year (!), I've been working – silenty, and in tiny iterations – on an extensive (last count: 124 local development commits, 1200+ diff-LOC) patch series for a third-party ...

This is why Waterfall is so bad :)

> This work-in-progress is currently (and finally) in its final stages and thus nearing completion. This would also come with proper, detailed documentation.
>
> I think it's only a matter of weeks utmost before all the remaining 'work' (it's already feature complete) will be completed. Therefore, I'd like to ask you all sincerely to wait with any potential core integration of any patch for this issue (or issue's #13244 and #13596, for that matter) until I've been able to open-source (i.e. share) the finished patch series in a proper manner, here on redmine.org publicly.
>
> Thanks in advance...
>
> P.S.: Marius Ionescu, this is in no way meant as a review of your patch(es)... ;)
> P.P.S.: off-topic note to self: review and/or change future strategy regarding issue-assignment to prevent other users, repeatedly, working simultaneously on the same issues as I am, thus wasting my or their time in the end.

I've invested only one hour in developing this patch, so is not a big waste from my point of view. I'm happy that you post your plan here because i's also in my plan for this week to develop the restriction for #13244. My main problem here is that we really need these 2 features internally next week, so I can't wait too long. Can we continue this discussion in private? Maybe we can find together a better solution.

#23 - 2018-04-03 17:06 - Go MAEDA

I am very looking forward Mischa's work. Aside from it, I found that Marius's patch may not work depending on time zones.

Suppose that time-zone of the user is set to Tokyo (+0900) and the server is set to UTC. If the user tries to log time for today at 08:00 on April 1 in Tokyo (23:00 UTC on March 31), the user cannot log time and see "Cannot log time on a future date" error. This is because TimeEntry#spent_on is April 1 and the server's date is March 31. In this situation, TimeEntry#spent_on.future? returns true and the user sees the error message.

#24 - 2018-04-07 13:05 - Marius BĂLTEANU

- File deleted (0001-setting-to-not-allow-spent-times-on-future-dates.patch)

#25 - 2018-04-07 13:15 - Marius BĂLTEANU

- File 0001-setting-to-not-allow-spent-times-on-future-dates.patch added

Go MAEDA wrote:

> I am very looking forward Mischa's work. Aside from it, I found that Marius's patch may not work depending on time zones.
>
> Suppose that time-zone of the user is set to Tokyo (+0900) and the server is set to UTC. If the user tries to log time for today at 08:00 on April 1 in Tokyo (23:00 UTC on March 31), the user cannot log time and see "Cannot log time on a future date" error. This is because TimeEntry#spent_on is April 1 and the server's date is March 31. In this situation, TimeEntry#spent_on.future? returns true and the user sees the error message.

I've fixed the issue, thanks for feedback.

I'm attaching the patch here for those who need this feature until Mischa's work is public and ready. Also, I don't see any problem to have this committed for the next version and override by Mischa patches in the future.

#26 - 2018-04-08 13:49 - Marius BĂLTEANU

- File deleted (0001-setting-to-not-allow-spent-times-on-future-dates.patch)

#27 - 2018-04-08 14:14 - Marius BĂLTEANU

- File 0001-setting-to-not-allow-spent-times-on-future-dates.patch added

Another update with two fixes:
- take into account the time zone of the time entry author and not of the current user.
- apply the validation only when the spent on date was changed (to follow the implementation from #24005).

#28 - 2018-09-14 07:12 - Marius BĂLTEANU

Go Maeda, I would suggest to propose this feature for the next Redmine version and when Mischa’s work is ready, we can apply it on top of this. A basic implementation (like mine) is better than nothing. What do you think?

#29 - 2018-09-14 07:25 - Go MAEDA

Hi Mischa, could you tell us the current status of your work? You wrote in #3322#note-21 that you were about to post patches. We are very looking forward to seeing your work.

#30 - 2018-09-14 07:32 - Marius BĂLTEANU

Sorry for not asking Mischa, but I’ve observed that he was no longer active in the last months.

#31 - 2018-12-04 00:38 - Marius BĂLTEANU

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

#32 - 2018-12-05 00:40 - Go MAEDA

- Assignee set to Marius BĂLTEANU

Marius, could you check the patch? The test fails with the trunk r17686.

```
$ bin/rails test test/unit/time_entry_test.rb:129
Run options: --seed 16073

# Running:

F

Failure:
TimeEntryTest#test_should_not_accept_future_dates_if_disabled [/Users/maeda/redmines/redmine-trunk/test/unit/time_entry_test.rb:134]:
Expected false to be truthy.

bin/rails test test/unit/time_entry_test.rb:129

Finished in 0.735537s, 1.3596 runs/s, 1.3596 assertions/s.
1 runs, 1 assertions, 1 failures, 0 errors, 0 skips
```

#33 - 2018-12-06 07:42 - Marius BĂLTEANU

Go MAEDA wrote:

> Marius, could you check the patch? The test fails with the trunk r17686.

> [...]

Strange. The tests pass on my local enviornmnet:

```
oot@45ec3a6558b0:/work# ruby test/unit/time_entry_test.rb
DEPRECATION WARNING: `secrets.secret_token` is deprecated in favor of `secret_key_base` and will be removed in
 Rails 6.0. (called from <top (required)> at /work/config/environment.rb:14)
Run options: --seed 7768

# Running:

....................

Finished in 1.356831s, 14.7402 runs/s, 37.5876 assertions/s.
20 runs, 51 assertions, 0 failures, 0 errors, 0 skips
```

And to double check, I've pushed the branch also on GitLab and the tests pass also there:
https://gitlab.com/marius-balteanu/redmine/pipelines/39104859

Can you check again your environment, please?

#34 - 2018-12-06 08:07 - Go MAEDA

Perhaps a timezone issue?

I ran the test around 8:30(JST/UTC+0900) on December 4th. But at that time, it was still December 3rd in UTC.

It is around 16:05(JST) now, the test passes.

#35 - 2018-12-06 08:35 - Marius BĂLTEANU

Go MAEDA wrote:

> Perhaps a timezone issue?
>
> I ran the test around 8:30(JST/UTC+0900) on December 4th. But at that time, it was still December 3rd in UTC.
>
> It is around 16:05(JST) now, the test passes.

Yes, could be, what do you think if I replace in the patch the method Date.tomorrow with User.current.today + 1 ?

```
diff --git a/test/unit/time_entry_test.rb b/test/unit/time_entry_test.rb
index e94c64b..2551c6f 100644
--- a/test/unit/time_entry_test.rb
+++ b/test/unit/time_entry_test.rb
@@ -121,7 +121,7 @@ class TimeEntryTest < ActiveSupport::TestCase

   def test_should_accept_future_dates
     entry = TimeEntry.generate
-    entry.spent_on = Date.tomorrow
+    entry.spent_on = User.current.today + 1

     assert entry.save
   end
@@ -129,7 +129,7 @@ class TimeEntryTest < ActiveSupport::TestCase
   def test_should_not_accept_future_dates_if_disabled
     with_settings :timelog_accept_future_dates => '0' do
       entry = TimeEntry.generate
-      entry.spent_on = Date.tomorrow
+      entry.spent_on = User.current.today + 1

       assert !entry.save
       assert entry.errors[:base].present?
```

#36 - 2018-12-06 14:05 - Go MAEDA

Marius BALTEANU wrote:

> Yes, could be, what do you think if I replace in the patch the method Date.tomorrow with User.current.today + 1 ?

Thank you for the fix, now it looks good. After applying the patch in #3322#note-35, tests passed without any problem.

December 7th at the local time, December 6th in UTC.

```
laphroaig:redmine-trunk maeda$ date && date -u
Fri Dec  7 01:42:18 NZDT 2018
Thu Dec  6 12:42:18 UTC 2018
```

The test fails.

```
laphroaig:redmine-trunk maeda$ ruby test/unit/time_entry_test.rb
WARNING: Nokogiri was built against LibXML version 2.9.7, but has dynamically loaded 2.9.8
Run options: --seed 7458

# Running:

.........F

Failure:
TimeEntryTest#test_should_not_accept_future_dates_if_disabled [test/unit/time_entry_test.rb:134]:
Expected false to be truthy.

bin/rails test test/unit/time_entry_test.rb:129

..........

Finished in 3.736849s, 5.3521 runs/s, 13.3803 assertions/s.
20 runs, 50 assertions, 1 failures, 0 errors, 0 skips
```

User.current.today returns a date in NZDT (the user's timezone). Date.tommorow returns a date in UTC. Date.tommorow is expected to return "Fri, 08 Dec 2018", but actually it returns "Fri, 07 Dec 2018" in the test code because the current date in UTC is "Fri, 06 Dec 2018".

```
(byebug) User.current.today
Fri, 07 Dec 2018
(byebug) Date.tomorrow
Fri, 07 Dec 2018
(byebug) Date.current
Thu, 06 Dec 2018
(byebug) Time.zone.name
"UTC"
```

Test passes after applying the patch in #3322#note-35.

#37 - 2018-12-06 18:55 - Marius BĂLTEANU

- File deleted (0001-setting-to-not-allow-spent-times-on-future-dates.patch)

#38 - 2018-12-06 18:57 - Marius BĂLTEANU

- File 0001-Setting-to-restrict-spent-times-on-future-dates-3322.patch added

Thank you for making all of those tests. I've updated the patch.

#39 - 2018-12-11 04:57 - Go MAEDA

- Status changed from New to Resolved

- Resolution set to Fixed

Committed the patch. Thank you for improving Redmine.

#40 - 2018-12-11 04:58 - Go MAEDA

- Assignee changed from Marius BĂLTEANU to Go MAEDA

#41 - 2018-12-11 05:22 - Go MAEDA

- Status changed from Resolved to Closed

#42 - 2018-12-11 06:56 - Marius BĂLTEANU

- Subject changed from Spent time allows Future date to Setting to restrict spent times on future dates

Files

0001-Setting-to-restrict-spent-times-on-future-dates-3322.patch	3.81 KB	2018-12-06	Marius BĂLTEANU