

Redmine - Patch #14735

Allow negative time entries

2013-08-21 14:23 - Jérôme BATAILLE

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Time tracking	Estimated time:	0.00 hour
Target version:			
Description			

History

#1 - 2013-08-21 14:49 - Rahil Rafiq

- File *zakatcalc.xls* added

#2 - 2013-08-21 14:56 - Jérôme BATAILLE

- File *RM_14735_allow_negative_time_entries.diff* added

Here is the first patch (Redmine::CoreExtensions::String::Conversions.to_hours).

I will furnish the last one (TimeEntry model validations) soon.

#3 - 2013-08-21 15:31 - Etienne Massip

- File *deleted (zakatcalc.xls)*

#4 - 2014-01-11 14:31 - Jean-Philippe Lang

Why should we allow negative time entries BTW?

#5 - 2014-01-11 16:12 - Daniel Felix

Well I'm not sure either.

Maybe for a different use case like planning time savings. But I don't need this.

#6 - 2014-01-11 18:25 - Jérôme BATAILLE

Jean-Philippe Lang wrote:

Why should we allow negative time entries BTW?

It's very simple, time spent are reported in our billing software each month.
When the month is validated, we must not changed previous time spent.
So if we want to make fixes of previous months, but stay in line between Redmine and other software, we can use negative hours to cancel previous wrong time entries.
Thus time spent if fixed on the next month, with keeping previously entered values.

We have tested negative hours and it works pretty well.

#7 - 2014-01-11 18:29 - Jérôme BATAILLE

- File *RM_14735_allow_negative_time_entries.diff* added

By the way, the previous patch has issues.
Here is a more correct version.

#8 - 2014-01-13 14:20 - Etienne Massip

- Category set to *Time tracking*

Jérôme BATAILLE wrote:

Jean-Philippe Lang wrote:

Why should we allow negative time entries BTW?

It's very simple, time spent are reported in our billing software each month.

When the month is validated, we must not changed previous time spent.

So if we want to make fixes of previous months, but stay in line between Redmine and other software, we can use negative hours to cancel previous wrong time entries.

Thus time spent if fixed on the next month, with keeping previously entered values.

We have tested negative hours and it works pretty well.

Not sure either it's a good idea to store adjustments in Redmine which is not a billing system, maybe a plugin could do it?

#9 - 2014-01-13 15:07 - Jérôme BATAILLE

Etienne Massip wrote:

Not sure either it's a good idea to store adjustments in Redmine which is not a billing system, maybe a plugin could do it?

It's just a question of extending the possible values.

for now, it's : $0 \leq \text{spent_hours} < 1000$

It's only an enhancement to allow $-1000 < \text{spent_hours} < 1000$

The only places to modify are :

- the display helpers :
 - to_hours
 - l_hours
- the validation

Pretty simple.

We have done the missing modifications in our plugin, but it would be very great to have no need to put these in it.

To explain our Redmine use :

This enhancement is properly to address the issue of software separation.

In our company we had a lot of demands to extend Redmine to satisfy constraints :

- time_spent in days
- hours by day management
- etc.

We choiced to keep the way Redmine works, Redmine gathers hours that's all.

We are now modifying our business processes to track time with Redmine, and I suppose it's the normal choice that finish to appear, for companies that use Redmine as their main tracking software.

If you use it to be able to tell what are people are doing precisely, it's the right place to gather the duration of their actions.

The direct consequence is that collected time spent has to be exported each month (or more often) in another software with budget tracking constraints.

Thus time spent has to be **exported**, and **then adapted** to budget and billing constraints.

But allowing negative values is a simple way to allow Redmine to cope with input errors, while keeping a trace of the first input. This issue is linked too, to the fact that time_entries modifications are not journalized.

In fact we have even developed a big enhancement to be able to track too the budget and remaining time, and thus manage follow-up on our projects. Same choice, these data are exported and then adapted to budget and billing constraints.

So Redmine is really adapted to the buisness processes of a big company, but it has only to be tweaked on some very few and minor points.

#10 - 2014-05-14 13:31 - Jérôme BATAILLE

Hi, any chance that this enhancement can be examined ?

The patch is very simple and straightforward. We have been using it for more than 2 years and it works perfectly.

#11 - 2014-09-25 13:11 - Edilberto Monje

Hi, In the patch (.diff) have a little error.

In the code of the patch miss a i letter. Check the next line:
s.gsub!(/r{^((2d+))e*(h|hours?)?e*((d+))e*(m|min)??)\$}) { |m| (\$1 || \$4) ? (\$2.to_i + \$5.to_i / 60.0 * (\$2.to_i > 0 ? 1 : -1)) : m⁰ } <<WRONG
~~s.gsub!(/r{^((2d+))e*(h|hours?)?e*((d+))e*(m|min)??)\$}) { |m| (\$1 || \$4) ? (\$2.to_i + \$5.to_i / 60.0 * (\$2.to_i > 0 ? 1 : -1)) : m⁰ } <<CORRECT~~

And it's necessary change the file /app/models/time_entry.rb :

```
.....
def validate_time_entry
  errors.add :hours, :invalid if hours && (hours <= -1000 || hours >= 1000)
  errors.add :project_id, :invalid if project.nil?
  errors.add :issue_id, :invalid if (issue_id && !issue) || (issue && project!=issue.project)
end
.....
```

In order to admit values negatives.

Best.

#12 - 2014-09-26 11:35 - Edilberto Monje

One bug more: Report don't show negative values.

We can correct this changing the following lines in the files:

```
./app/views/timelog/_report_criteria.html.erb: <td class="hours"><%= html_hours("%.2f" % sum) if sum > 0 ></td>
./app/views/timelog/_report_criteria.html.erb: <td class="hours"><%= html_hours("%.2f" % total) if total > 0 ></td>
./app/views/timelog/report.html.erb: <td class="hours"><%= html_hours("%.2f" % sum) if sum > 0 ></td>
./app/views/timelog/report.html.erb: <td class="hours"><%= html_hours("%.2f" % total) if total > 0 %></td>
```

Change this lines inside the files as in the next:

```
./app/views/timelog/_report_criteria.html.erb: <td class="hours"><%= html_hours("%.2f" % sum) if sum > -1000 ></td>
./app/views/timelog/_report_criteria.html.erb: <td class="hours"><%= html_hours("%.2f" % total) if total > -1000 ></td>
./app/views/timelog/report.html.erb: <td class="hours"><%= html_hours("%.2f" % sum) if sum > -1000 ></td>
./app/views/timelog/report.html.erb: <td class="hours"><%= html_hours("%.2f" % total) if total > -1000 %></td>
```

Best.

#13 - 2014-12-04 14:41 - Jérôme BATAILLE

Edilberto, all your remarks about the other places to modify are correct.

what is the aim to add an i to the regular expression, we use it on our side without it.

#14 - 2014-12-04 15:08 - Jérôme BATAILLE

I have not found any mention of this i on the internet, and it's not used with the others regexp, could it be a typo ?

#15 - 2015-03-10 15:09 - Joachim Müller

+1 for allowing negative time entries

#16 - 2018-12-18 12:10 - Sebastian Paluch

+1

Jean-Philippe Lang wrote:

Why should we allow negative time entries BTW?

Editing spent time does not leave a trace in the journals, so, we don't allow to do that. Adding negative values allows to make corrections but keep the history.

Files			
RM_14735_allow_negative_time_entries.diff	909 Bytes	2013-08-21	Jérôme BATAILLE
RM_14735_allow_negative_time_entries.diff	1.02 KB	2014-01-11	Jérôme BATAILLE