

Redmine - Defect #41819

Time entry API returning `hours` as Rational instead of Float

2024-11-21 09:07 - Łukasz Szczepański

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Go MAEDA	% Done:	0%
Category:	REST API	Estimated time:	0.00 hour
Target version:	6.0.2	Affected version:	6.0.1
Resolution:	Fixed		

Description

Before Redmine 6, the "hours" field returned data in double (float) format, like 0.25. Currently, Redmine 6 returns this field in weird fraction format 9/1 (as 9 hours).

This change wasn't stated in changelog. Is this an intentional change? It breaks integration that uses REST API from Redmine.

Response from Redmine 5.0.11:

```
{
  "id": 57818,
  "project": {
    "id": 120,
    "name": "Project Name"
  },
  "issue": {
    "id": 50271
  },
  "user": {
    "id": 534,
    "name": "User Name"
  },
  "activity": {
    "id": 27,
    "name": "Implementation"
  },
  "hours": 0.25,
  "comments": "Comment",
  "spent_on": "2024-10-23",
  "created_on": "2024-10-23T08:03:14Z",
  "updated_on": "2024-10-23T08:03:14Z"
},
```

Response from Redmine 6.0.1

```
{
  "id": 153263,
  "project": {
    "id": 524,
    "name": "Project Name"
  },
  "issue": {
    "id": 56753
  },
  "user": {
    "id": 636,
    "name": "User Name"
  },
  "activity": {
    "id": 10,
    "name": "Implementation"
  },
  "hours": "9/1"
},
```

```
"hours": "8/1",
"comments": "",
"spent_on": "2022-04-13",
"created_on": "2022-04-27T10:55:22Z",
"updated_on": "2022-04-27T10:55:22Z"
},
```

Related issues:

Related to Redmine - Defect #40914: Fix precision issues in TimeEntry#hours c...

Closed

Associated revisions

Revision 23304 - 2024-11-24 08:04 - Go MAEDA

Fix: Time entry API returning `hours` as Rational instead of Float (#41819).

Patch by Go MAEDA (user:maeda).

Revision 23310 - 2024-11-25 15:39 - Go MAEDA

Merged r23304 from trunk to 6.0-stable (#41819).

History

#1 - 2024-11-21 10:03 - Łukasz Szczepański

It's looks like this change has something to do with that. It's a diff between 5.1.4 and 6.0.1.

```
diff --git a/lib/redmine/i18n.rb b/lib/redmine/i18n.rb
index 0fe080f4b..a9cd1dd0d 100644
--- a/lib/redmine/i18n.rb
+++ b/lib/redmine/i18n.rb
@@ -21,6 +21,8 @@ require 'redmine'

module Redmine
  module I18n
+   include ActionView::Helpers::NumberHelper
+
    def self.included(base)
      base.extend Redmine::I18n
    end
@@ -43,17 +45,17 @@ module Redmine
  end

  def l_or_humanize(s, options={})
-   k = "#{options[:prefix]}#{s}".to_sym
+   k = :#{options[:prefix]}#{s}"
    ::I18n.t(k, :default => s.to_s.humanize)
  end

  def l_hours(hours)
-   hours = hours.to_f
+   hours = hours.to_f unless hours.is_a?(Numeric)
    l((hours < 2.0 ? :label_f_hour : :label_f_hour_plural), :value => format_hours(hours))
  end

  def l_hours_short(hours)
-   l(:label_f_hour_short, :value => format_hours(hours.to_f))
+   l(:label_f_hour_short, :value => format_hours(hours.is_a?(Numeric) ? hours : hours.to_f))
  end

  def ll(lang, str, arg=nil)
@@ -90,15 +92,27 @@ module Redmine
  def format_hours(hours)
    return "" if hours.blank?

+   minutes = (hours * 60).round
    if Setting.timespan_format == 'minutes'
-     h = hours.floor
-     m = ((hours - h) * 60).round
+     h, m = minutes.divmod(60)
      "%d:%02d" % [h, m]
    else
      "%.2f" % hours.to_f
    end
  end
end
```

```

+     number_with_delimiter(sprintf('%0.2f', minutes.fdiv(60)), delimiter: nil)
+     end
+   end
+
+   # Will consider language specific separator in user input
+   # and normalize them to a unified format to be accepted by Kernel.Float().
+   #
+   # @param value [String] A string representation of a float value.
+   #
+   # @note The delimiter cannot be used here if it is a decimal point since it
+   #       will clash with the dot separator.
+   def normalize_float(value)
+     separator = ::I18n.t('number.format.separator')
+     value.gsub(/[#{separator}]/, separator => '.')
+   end
+
+   def day_name(day)
+     ::I18n.t('date.day_names')[day % 7]
+   end

```

#2 - 2024-11-21 10:09 - Marius BĂLTEANU

I think it is related to the changes made in [#40914](#) or [#22024](#).

#3 - 2024-11-21 10:10 - Łukasz Szczepański

The timespan setting in Redmine is set to float (0.75) in both instances.

#4 - 2024-11-21 14:39 - Go MAEDA

- Status changed from New to Confirmed

#5 - 2024-11-21 14:50 - Go MAEDA

- Assignee set to Go MAEDA

- Target version set to 6.0.2

#6 - 2024-11-21 15:26 - Go MAEDA

- File 41819.patch added

- Assignee deleted (Go MAEDA)

The attached patch should fix the issue.

#7 - 2024-11-24 08:05 - Go MAEDA

- Subject changed from Unexpected hours format in time entries response to Time entry API returning `hours` as Rational instead of Float

- Status changed from Confirmed to Resolved

- Assignee set to Go MAEDA

- Resolution set to Fixed

Committed the fix in [r23304](#). Thank you for reporting this issue.

#8 - 2024-11-25 15:40 - Go MAEDA

- Status changed from Resolved to Closed

Merged the change into the 6.0-stable branch in [r23310](#).

#9 - 2024-12-01 07:23 - Go MAEDA

- Related to Defect #40914: Fix precision issues in TimeEntry#hours calculation by returning Rational instead of Float added

Files

File Name	Size	Date	Author
41819.patch	2.84 KB	2024-11-21	Go MAEDA