

Improvement proposal to optimise rendering of the sprint board

Status:	New	Estimated time:	0.00 hour
Priority:	Low		
Assignee:			
Blocked:	No		

Update the view `app/views/sprints/show.html.erb` by adding a 'cache' markup

then update the IssuePatch model by adding

```

      issue.project.scrum? and issue.status_id_changed? and issue.is_task?
    }
    before_save :update_assigned_to, :if => lambda { |issue|
      issue.project.scrum? and issue.status_id_changed? and issue.is_task?
    }
    before_save :update_parent_pbi, :if => lambda { |issue|
      issue.project.scrum? and Scrum::Setting.auto_update_pbi_status and
      (issue.status_id_changed? or issue.new_record?) and
      issue.is_task? and !issue.parent_id.nil?
    }
    before_save :touch_sprint, :if => lambda { |issue|
      issue.project.scrum?
    }
  }

```

here's the touch_sprint implementation

```

def touch_sprint
  if sprint
    sprint.touch
    sprint.save!
  end
  if (old_sprint = Sprint.find_by_id(sprint_id_was))
    old_sprint.touch
    old_sprint.save!
  end
end

```

create a new lib/scrums/time_entry_patch.rb file

```

require_dependency 'time_entry'

module Scrum
  module TimeEntryPatch
    def self.included(base)
      base.class_eval do

        before_save :touch_issue, :if => lambda { |issue|
          issue.project.scrum?
        }

      private

        def touch_issue
          if issue
            issue.touch
            issue.save!
          end
        end

      end
    end
  end
end

```

and update the init.rb file by adding the new patch

```

Query.send(:include, Scrum::QueryPatch)
Tracker.send(:include, Scrum::TrackerPatch)
User.send(:include, Scrum::UserPatch)
TimeEntry.send(:include, Scrum::TimeEntryPatch)

```

History

#1 - 2017-01-06 18:25 - Angelinsky7 Angelinsky7

I know this is not the best way to do it, but like that navigating in the sprint board is now good.
everytime something change (for now, an issue, a time track or the sprint), everything is recache...
This could be improved by caching with more granularity (sprint -> pbi_row -> task)... (but need someone with a lot more knowing of ruby than me)
this is no 100% perfect, i found that sometime, a "parent" object is not touch (for whatever reason)
Hope that could help someone or this amazing project.
related to [#1160](#)

#2 - 2017-01-06 18:42 - Angelinsky7 Angelinsky7

i made a mistake when copy/paste...
the <% end %> tag is just behind the end of the table tag

```
<tbody id="<%= sprint_board_id %>" class="sprint-board">
  <%- @sprint.pbis.each do |pbi| -%>
    <%= render :partial => 'post_its/sprint_board/pbi_row', :formats => [:html],
      :locals => {:project => @project, :sprint_board_id => sprint_board_id, :pbi => pbi,
        :task_statuses => task_statuses} %>

    <%- end -%>
  </tbody>
</table>
<% end %>

<%- if User.current.allowed_to?(:add_issues, @project) and
  User.current.allowed_to?(:edit_sprint_board, @project) and
  @sprint.open? -%>
  <div>
```

#3 - 2017-01-06 18:46 - Angelinsky7 Angelinsky7

- File *scrum_16_2.patch* added

here's a patch file against 16_2 for this

#4 - 2017-01-10 16:06 - Angelinsky7 Angelinsky7

- File *scrum_cache_16_2.patch* added

rewrite some of it (and added more layer)
now the cache is on

- 1. sprint table
- 2. pbi_row (by status)
- 3. task (each post-it)

it's working well, and each time a task is changed only the parent pbi row is rerendered...

here's the patch file

Files			
scrum_16_2.patch	3.3 KB	2017-01-06	Angelinsky7 Angelinsky7
scrum_cache_16_2.patch	5.55 KB	2017-01-10	Angelinsky7 Angelinsky7