

## Redmine - Feature #6885

### Make activity.atom feed usable for detecting redmine changes

2010-11-13 12:24 - Grigory Petrov

<b>Status:</b>	New	<b>Start date:</b>	2010-11-13
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Feeds	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Resolution:</b>			
<b>Description</b>			
<p>Currently, activity.atom feed only tracks issue creation/removal/status and comments. This limits feed for only human reader. But there are numerous automation tasks this feed can be very useful.</p> <p>Right now i'm trying to create something like Jira Client (<a href="http://almworks.com/jiraclient">http://almworks.com/jiraclient</a>) for redmine. This is my personal hobby project that has a goal to make me a better programmer. I have already performed various tests, created architecture sketch and performed a lot of research. The only serious problem i can see right now is "how API client will notice that something changed in redmine".</p> <p>The intuitive way is to make API call like "show all issues that has changes since last time i checked" - but this is completely impossible in redmine since only saved filters can be used in issues API and where is no such thing possibel in saved filter.</p> <p>The more interesting way i'm researching is to use activity feed. Modified a little it can be used to detect ANY change on redmine server so API client can fetch changed issues/projects/users/whatever and do it's bidding. The modifications i can propose:</p> <ol style="list-style-type: none"><li>1. Add more information to activity feed. Ideally, any redmine change can lead to record in activity feed. But this is not need to be implemented "all and right now" and can be added step-by-step. For example, right now records for issues priority and assigneed change can be added :).</li><li>2. Currently, activity feed only shows last N records. This is good for automatic aggregator that runs 24/7 but is not very good for human/software that checks state periodically. For example, i'm a program that s installed in laptop and synchronize itself with redmine. Periodically, since sometime my owner closes down laptop lid and sometimes where is no network connection. I has synchronized and network dissapears. After a few hours network goes back again and i poll redmine for activity feed - and see 15 new feeds. So now i have a problem - how many activities it was between my last sync and current time? How i can get this activities? The possible solution is to add atom feed arguments like "GET <a href="http://www.redmine.org/activity.atom?from=15.16.16.20.10.2010">http://www.redmine.org/activity.atom?from=15.16.16.20.10.2010</a>" where "15.16.16.20.10.2010" is time and date.</li><li>3. Adding something like ?from to activity.atom is very good and solves all problems, but is not very bullet-proof since granularity is 1 second and redmine can be changed same seconds last item was fetched, making this change time equal to "last fetch" time. The bullet-proof solution is no introduce something like subversion 'revision' to redmine. That's very simple: add a counter to database and every change increments it. So, activities feed user can just request "show me activities after specified revision", like this: "GET <a href="http://www.redmine.org/activity.atom?after=123">http://www.redmine.org/activity.atom?after=123</a>".</li></ol>			

#### History

##### #1 - 2010-11-13 14:46 - Jean-Philippe Lang

I think you should better use the API rather than feeds. You could simply iterate on the following URI until you reach the last update you've already retrieved: `/issues.xml?sort=updated_on:desc&page=[1..n]`

##### #2 - 2010-11-13 15:35 - Grigory Petrov

Jean-Philippe Lang, it's a good idea, but amount of traffic for polling issues, tasks, users and worklogs via pagination will be REALLY HUGE :). Just try [http://www.redmine.org/issues.xml?sort=updated\\_on:desc&page=1](http://www.redmine.org/issues.xml?sort=updated_on:desc&page=1) for yourself.

BTW, how do you know that issues.xml can respond to "sort"? I never soo this on API wiki page.

##### #3 - 2010-11-14 14:48 - Jean-Philippe Lang

- Category set to Feeds

Grigory Petrov wrote:

Jean-Philippe Lang, it's a good idea, but amount of traffic for polling issues, tasks, users and worklogs via pagination will be REALLY HUGE :). Just try [http://www.redmine.org/issues.xml?sort=updated\\_on:desc&page=1](http://www.redmine.org/issues.xml?sort=updated_on:desc&page=1) for yourself.

I don't know what you're trying to pull exactly but you should definitely rely on the API. Atom feeds only provide human readable content about updates.

BTW, how do you know that issues.xml can respond to "sort"? I never saw this on API wiki page.

This is something to add to the API documentation :-)

#### **#4 - 2010-11-14 17:21 - Grigory Petrov**

I don't know what you're trying to pull exactly but you should definitely rely on the API. Atom feeds only provide human readable content about updates.

AFAIK, atom and rss feeds are general notification mechanics. It can't be readed by humans 'as is' :). It's always an RSS/Atom reader that displays it in suitable form. But feeds are widely used as notification middleware and it is absolutely ok to use it for polling redmine changes. The only thing to add is widely used 'from' option (that is used by news readers too).

#### **#5 - 2011-02-02 19:06 - Mark Anderson**

Hello friends!

I'll add to the "human readable" aspect. I now have an iPad and get some of my Redmine feeds. the User Interface approach to the various readers on the iPad are different than we might see via a standard web browser. So the feeds are not really read by "me" but by my reader's software.

Mark

#### **#6 - 2011-02-03 08:43 - Grigory Petrov**

Recently, i have studied an "REST in Practice" book that states a few interesting things:

1. Authors claims that atom feeds are perfect for middleware and asynchronous client-server event notification.
2. Using special ATOM tags it's possible to embed "previous", "first" and "next" links into ATOM feed and introduce so-called "archive" that contains chunks of previous feeds. Such feed organization seems to me a much better than "?from" request i proposed previously. It allow redmine clients to flawlessly get notifications about redmine events regardless of network connectivity.

I'm currently investigating a subject to find a solution that is technically correct. If some experienced Redmine programmer can help, i can pay some amount of money for implementing a redmine atom feed patch (i'm a good programmer myself, but i'm not experienced with Ruby on Rails architecture).