

Redmine - Patch #35769

Add a Revision (Changeset) API end-point with support for inclusion of associated issues

2021-08-17 01:22 - Jan Catrysse

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	REST API	Estimated time:	0.00 hour
Target version:			
Description			
This is a small patch to add an API call to get details about a revision and the linked issues. It should be reviewed by someone with some more experience.			
EDIT: app/controllers/repositories_controller.rb ADD: app/views/repositories/revision.rsb			
USE: https://www.redminexxx.org/projects/{project_name}/repository/{repository_name}/revisions/{revision_name}.json			

History

#1 - 2021-08-17 07:34 - Mischa The Evil


- Subject changed from Feature: patch to add simple repositories API call to view revision details + related issues to Add a Revision (Changeset) API end-point with support for inclusion of associated issues

This could be a nice feature. Do you have some specific use-cases in mind with this addition? How do you see this being used without a dedicated end-point to fetch all repository revisions (changesets)?

Some notes on the patch(es):

- Please provide a single, unified diff file (*git format-patch* and *git diff* output is ok).
- Do not include commented-out code in your diffs unless there's a specific reason to do so.
- No tests are included; to get your code integrated you'd need to write some test coverage to:
 - show that your code works as expected;
 - show (some) example API-calls with their expectations, given that you're adding to/modifying the API.

What this patch provides:

- 
 - Example response in JSON for a Git commit:

```
{
  "changeset": {
    "revision": {
      "id": 24,
      "repository_id": 2,
      "revision": "git_commit_hash",
      "committer": "Committer name <committer_email>",
      "committed_on": "2021-01-01T00:00:00.000Z",
      "comments": "Commit message",
      "commit_date": "2021-01-01",
      "scmid": "git_commit_hash",
      "user_id": null
    },
    "issues": [
      {
        "id": 6,
        "subject": "Issue subject",
        "root_id": 6,
        "parent_id": 5
      }
    ]
  }
}
```

Some questions about this feature:

- Do we really want to include associated issues like this? It may be better to add an additional include parameter with an issues value to fetch the associated issue data on-demand only.
- Do we really want to name the issues array issues instead of something like related_issues or associated_issues?
- Do we really want to expose all these revision attributes?
- Do we really want to expose the issue parent_id through this end-point?
- Do we really want to expose the issue root_id through this end-point? AFAIK this attribute isn't exposed to any user at all in the UI.

Some possible, future additions/improvements:

- Add an actual listing of files added/modified/copied/renamed/deleted to the output via an include=changes parameter value. An [example of what this could look like](#).

```
{
  "changeset": {
    "revision": {
      "id": 24,
      "repository_id": 2,
      "revision": "git_commit_hash",
      "committer": "Committer name <committer_email>",
      "committed_on": "2021-01-01T00:00:00.000Z",
      "comments": "Commit message",
      "commit_date": "2021-01-01",
      "scmid": "git_commit_hash",
      "user_id": null
    },
    "issues": [
      {
        "id": 6,
        "subject": "Issue subject",
        "root_id": 6
      }
    ],
    "changes": [
      ["added":
        {
          "file": "/app/views/repositories/revision.api.rsb"
        }
      ],
      ["modified":
        {
          "file": "/trunk/app/controllers/repositories_controller.rb",
          "file": "/trunk/app/controllers/application_controller.rb"
        }
      ],
    ]
  }
}
```

#2 - 2021-08-17 12:20 - Jan Catrysse

Thank you for your help, I will look into that when I am back from holiday.

Files

repositories_controller.rb.patch	975 Bytes	2021-08-16	Jan Catrysse
revision.rsb	588 Bytes	2021-08-16	Jan Catrysse