

Redmine - Defect #9598

JSON REST API : PUT returns one space character

2011-11-18 12:32 - Francis West

Status:	Closed	Start date:	2011-11-18
Priority:	High	Due date:	
Assignee:		% Done:	0%
Category:	REST API	Estimated time:	0.00 hour
Target version:		Affected version:	1.2.2
Resolution:	Invalid		

Description

I am trying to build a python class to support creating, updating and closing tickets to be used in my deploy system. It looks something like this:

```
class Issues(Resource):
    def __init__(self, **kwargs):
        '''
        init Issues
        '''
        redmine_url = "<URL>"
        self.key = None
        self.project = None
        super(Issues, self).__init__(redmine_url,
                                     follow_redirect=True,
                                     max_follow_redirect=10,
                                     **kwargs)

    def set_key(self, key):
        self.key = key

    def set_project(self, project):
        self.project = project

    def tickets(self):
        if (self.key == None) and (self.project == None):
            resp = dict()
        else:
            resp = self.get('issues.json', project=self.project, key=self.key)
        return resp

    def get_ticket(self, ticket_id):
        resp = self.get('issues/%s.json' % ticket_id,
                       project=self.project, key=self.key)
        return resp

    def update_ticket(self, ticket_update, ticket_id):
        json_ticket = json.dumps(ticket_update)
        resp = self.put('issues/%s.json' % ticket_id,
                       json_ticket,
                       headers={'Content-type': 'application/json', },
                       params_dict={'key': self.key, })

        return resp

    def new_ticket(self, ticket):
        json_ticket = json.dumps(ticket)
        resp = self.post('issues.json',
                       json_ticket,
                       headers={'Content-type': 'application/json',
                                'key': self.key,
                                'project': self.project},
                       params_dict={'key': self.key,
                                    'project': self.project})
```

```

    )
    return resp

def request(self, *args, **kwargs):
    resp = super(Issues, self).request(*args, **kwargs)
    self.body_string = resp.body_string()
    if len(self.body_string) == 1:
        return dict()
    else:
        return json.loads(self.body_string)
    return self.body_string

```

a quick test harness

```

from redmine import Issues
from pprint import pprint
issue = Issues()
issue.set_key('<what ever>')
issue.set_project('archiect')

# create a ticket
a = {'issue': {'project_id': 'Project_X', 'description': 'information description', 'subject':
'something test', 'tracker_id': 4}}
z = issue.new_ticket(a)
pprint(z)
ticket_id = z['issue']['id']

z['issue']['notes'] = u"Changing the subject"
z['issue']['status'] = {'id': 3}
z['issue']['done_ratio'] = 100
z['issue']['subject'] = u"Example issue (was: %s)" % z['issue']['subject']
pprint(z)
asd = issue.update_ticket(z,ticket_id)
pprint(asd)

```

So the print out from z looks very sensible and with the right permissions the ticket updates!

The problem is with the return value of the update which breaks the json.loads on the request method. This is because I do not get some JSON returned, just a ''.

I think the ticket should be returned with its updates! also if there were any errors could they be returned in ['errors'] for example?

Yes, right now I could get the ticket again to check if the updates have happened. However this is not a good solution.

Related issues:

Related to Redmine - Defect #11388: Updating a version through rest API retur... **Closed**

History

#1 - 2011-11-19 12:03 - Jean-Philippe Lang

- Status changed from New to Closed
- Resolution set to Invalid

This is a common behaviour of the API. When you update an object:

- you get a simple 200 OK if the update succeeded.
- you get a 422 Unprocessable Entity with the errors in the response body if the update failed.