

Pulp - Task #2238

Task # 1873 (CLOSED - CURRENTRELEASE): Plan REST API for 3.0

Make DRF tools that can represent our generic models via the API

09/07/2016 08:23 PM - semyers

Status:	CLOSED - CURRENTRELEASE	Start date:	
Priority:	Normal	Due date:	
Assignee:	semyers	% Done:	0%
Category:		Estimated time:	0:00 hour
Sprint/Milestone:	3.0.0	Tags:	
Platform Release:		Sprint:	
Groomed:	No	Quarter:	
Sprint Candidate:	No		
Description			
In putting all this API stuff together, I just now realized that I've completely ignored things like the GenericKeyValueMutableMapping that's related to models as the Config/Notes/etc fields. This should be a simple custom field, based on the DRF docs that detail a few different methods of handling generic relations in the api: http://www.django-rest-framework.org/api-guide/relations/#generic-relationships			

History

#1 - 09/07/2016 09:34 PM - semyers

- Parent task set to #1873

#2 - 09/12/2016 06:23 PM - semyers

This should be a simple custom field, based on the DRF docs that detail a few different methods of handling generic relations in the api

Well...sorta?

Since the generic fields we're using right now are exposed as a key/value mapping, I'd really like to see them be exposed that way in the API, as well. DRF support this pretty easily: Make a DictField on your serializers, and then set it's "source" to 'fieldname.mapping', and it then reads its value from the custom mapping field. On the read side, this works perfectly. However, when **writing** to these fields, things get a little trickier. Somewhat understandably, DRF has no idea how to correctly write values to this field. It sanity-checks the "source" value, sees that it's a dotted attribute, and bails out before doing something potentially dangerous.

So, to start, I have created a "GenericKeyValueRelatedField" (or something like that) for us to use when relating to these things. It works well enough in a read_only fashion now, but we'll also need a Serializer base class to eventually support writing to these fields. I'll capture all of this in a task with more detail and some suggestions for resolution, but in terms of "Plan REST API for 3.0", we've got enough to build a plan on so this is probably done depending whether or not the reviewers agree. :)

#3 - 09/14/2016 08:53 PM - semyers

Admitting default and settling for leaving the field read-only bugged the crap out of me, so after letting it simmer for a few days I had an idea for making read-write in a DRF-friendly and sane way. This is now implemented, with the only issue being that the DRF browser-based API views don't support DictField, so the only way to update this field is by using the API with some other content type (i.e. JSON and XML work perfectly). I was also able to cut down on "trickeration" and instead implement the GenericKeyValueRelatedField as a more "normal" DRF field subclass according to their documented extension methods.

tl;dr woot

#4 - 09/14/2016 08:55 PM - semyers

semyers wrote:

the DRF browser-based API views don't support DictField

On this topic, I haven't yet checked to see if there are any upstream issues to deal with this. It's a minor thing, since most API users should be using JSON, but if we could add this to DRF I'm sure they'd appreciate it. I plan to look into this later and see how we can either help DRF out or get help from DRF for this.

#5 - 09/14/2016 09:13 PM - semyers

semyers wrote:

semyers wrote:

the DRF browser-based API views don't support DictField

On this topic, I haven't yet checked to see if there are any upstream issues to deal with this. It's a minor thing, since most API users should be using JSON, but if we could add this to DRF I'm sure they'd appreciate it. I plan to look into this later and see how we can either help DRF out or get help from DRF for this.

And, of course, it's a known issue: <https://github.com/tomchristie/django-rest-framework/issues/2485>

#6 - 10/05/2016 03:39 PM - semyers

- Status changed from ASSIGNED to CLOSED - CURRENTRELEASE

This work is up in a PR in this issue's parent task, [#1873](#).

#7 - 04/25/2019 06:47 PM - daviddavis

- Sprint/Milestone set to 3.0.0

#8 - 04/26/2019 10:39 PM - bmbouter

- Tags deleted (Pulp 3)

#9 - 06/16/2020 10:54 PM - bmbouter

- Category deleted (14)

We are removing the 'API' category per open floor discussion June 16, 2020.