

## Pulp - Task #2175

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

### Make DRF tools that can represent our master/detail relationships via the API

08/16/2016 04:16 PM - semyers

<b>Status:</b>	CLOSED - CURRENTRELEASE	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	semyers	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0:00 hour
<b>Sprint/Milestone:</b>	3.0.0	<b>Tags:</b>	
<b>Platform Release:</b>		<b>Sprint:</b>	
<b>Groomed:</b>	No	<b>Quarter:</b>	
<b>Sprint Candidate:</b>	No		
<b>Description</b>			
Our master/detail modeling in pulp 3 makes generating API endpoints a little interesting, particularly given the stated needs of the API, which include automatic documentation using docstrings in code, and autogenerated API clients. Implementing this will require creating some slightly customized DRF components that are "master/detail" aware.			
This is part of the effort for <a href="#">#1873</a> that I'm opting to track in a separate task.			

#### History

##### #1 - 08/16/2016 04:16 PM - semyers

- Tracker changed from Issue to Task

##### #2 - 08/16/2016 04:21 PM - semyers

Since the docstrings are used to generate the API documentation, I've done some experiments with combining docstrings from superclasses as one potential way to combine master and detail docstrings. I believe DRF also provides an interface for providing the view string, so that will likely be the more robust and reasonable approach.

I'll revisit that, but right now I'm moving on to creating the master-detail-aware viewsets. These will be responsible for creating the detail URL routes, with my initial approach being on exposing only the detail model endpoints. This will also require creating supporting components that are master-detail-aware, such as URL generators for related fields.

##### #3 - 09/02/2016 05:57 PM - semyers

- Checklist item [x] Come up with a way (or ways) to combine master and detail docstrings to make writing docstrings easier set to Done  
Checklist item [x] Create a type of DRF ViewSet that correctly creates endpoints for detail model serializers when given the master model class set to Done

While the checklist items are done, I created these cool tools on my rel-pulp fork. I've been working to "rebase" them onto pulp 3.0-dev, but I've had a lot of excellent small meetings this week that have been distracting.

Also, the checklist item about doing magical things when handed a model class has proven (to my satisfaction, at least), to be a bit foolish. The API is entirely made up of ViewSets, and it proved much more reasonable/usable to explicitly declare all serializers and viewsets in plugins, and make a mechanism available in platform for the plugins to uniformly register their viewsets with the platform for exposure to the API.

I went with a pretty django-standard "register" decorator model, as seen in the django contrib admin app, and some others.

**#4 - 10/05/2016 03:41 PM - semyers**

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

After getting feedback on the initial design, instead of generating endpoints based on ViewSet Models, we opted to explicitly declare the endpoint name on ViewSets directly, and have the ViewSet class correctly combine them in the case of master/detail models.

**#5 - 10/05/2016 03:45 PM - semyers**

- Status changed from ASSIGNED to CLOSED - CURRENTRELEASE

**#6 - 04/25/2019 06:47 PM - daviddavis**

- Sprint/Milestone set to 3.0.0

**#7 - 04/26/2019 10:39 PM - bmbouter**

- Tags deleted (Pulp 3)

**#8 - 06/16/2020 10:54 PM - bmbouter**

- Category deleted (14)

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