

Pulp - Story #7762

Need synchronous REST endpoint for (RPM) distribution update

10/28/2020 02:33 PM - Anonymous

Status:	CLOSED - WONTFIX	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0:00 hour
Sprint/Milestone:		Tags:	
Platform Release:		Sprint:	
Groomed:	No	Quarter:	
Sprint Candidate:	No		

Description

In a typical RHUI setup with 400+ repos and similar number of pending repository sync tasks, it would be of a great help to have ability to "publish" (ie. update their distribution with a publication that is already available) those repos that already have the latest metadata compiled without waiting for the sync task queue to drain. Alternatively, a way to "push" newly created distribution update onto the start of the task queue. In my understanding, updating a distribution looks like a lightweight task compared to sync or metadata generation tasks.

History

#1 - 11/03/2020 08:19 PM - bmbouter

We already have [an endpoint that provides this](#) just not synchronously. From a REST perspective how can we make the existing endpoint return a 200 in cases where the tasking system doesn't need to be involved, and a 202 where it does? My concern is that we're going to make two ways to do one thing? My other concern is that another API wouldn't be RESTful and would instead include some kind of verb since the noun resource, i.e. Distribution, is already using in this async API.

What do you all think our options are to have one viewset return different response types for different situations?

#2 - 11/03/2020 09:07 PM - daviddavis

I'm not quite sure I follow. You're proposing that the distribution update endpoint returns a 202 with a task or a 200 depending on what fields are being updated?

#3 - 11/03/2020 09:11 PM - bmbouter

daviddavis wrote:

I'm not quite sure I follow. You're proposing that the distribution update endpoint returns a 202 with a task or a 200 depending on what fields are being updated?

That's what I'm asking about yes.

#4 - 11/03/2020 09:14 PM - daviddavis

bmbouter wrote:

daviddavis wrote:

I'm not quite sure I follow. You're proposing that the distribution update endpoint returns a 202 with a task or a 200 depending on what fields are being updated?

That's what I'm asking about yes.

A couple thoughts off the top of my head: will the bindings handle different responses?

The other concern would be semantic versioning (ie if I update the publication field on a distribution today, I get back a 202 and task). I think we'll have to probably add a parameter or something to avoid breaking semver.

#5 - 11/03/2020 09:25 PM - bmbouter

daviddavis wrote:

A couple thoughts off the top of my head: will the bindings handle different responses? I don't know this is what stopped me as well. Webservers can respond with a lot of error codes, and I do see "reponses" listed in the browseable docs API, but I am not sure what openAPI provides for this.

The other concern would be semantic versioning (ie if I update the publication field on a distribution today, I get back a 202 and task). I think we'll have to probably add a parameter or something to avoid breaking semver.

I agree. It's a little strange admittedly but it still keeps one API endpoint which I think is important, and it's still RESTful.

#6 - 04/20/2021 03:54 PM - mdellweg

My concern is that whenever we change the plugin api by a change in pulpcore, we need to think about the compatibility of the plugin (in some version) with it's bindings (in the same version). In this particular case, the change may actually be additive. So it might be sufficient to add the possible 200 response to the openapi schema in one release (of pulpcore) and implement the actual change in the next one. That way, there would be no need to coordinate code with the plugins.

#7 - 04/20/2021 06:31 PM - dalley

In a typical RHUI setup with 400+ repos and similar number of pending repository sync tasks, it would be of a great help to have ability to "publish" (ie. update their distribution with a publication that is already available) those repos that already have the latest metadata compiled without waiting for the sync task queue to drain.

@sskracic, the auto-publish feature we've just added provides a way to do almost exactly this. We implemented it as a hook that is called any time a new repository version is created, and executes in the same task.

https://github.com/pulp/pulp_rpm/blob/edd3daa63d62820296db97ee3715459ef9fe8ce3/pulp_rpm/app/models/repository.py#L268-L296

What this code does:

- If the repository has "autopublish" flag set, it will automatically create a publication for new repository versions in the repo
- ***If any distributions are pointing at this repository (as opposed to just pointing to a specific publication we added a new field for this), then we automatically update those with the new publication***

What it doesn't do:

- Provide a way to update the distributions all at once after the publications have been created, it happens as each individual repository is updated
- Solve the problem "generally" if you want to roll back many repositories at once or do some other operation which isn't using the very latest publications - but in this instance your queue is likely not so full

Does this diminish your need for this feature? Are there use cases that we should consider beyond what we've done thus far? The great thing about this approach is that it's easy to update / customize the behavior.

#8 - 04/26/2021 10:58 AM - sskracic@redhat.com

[dalley](#) - yes, that would alleviate the need for the newly proposed REST endpoint, so I suggest you to close this issue. Thanks.

#9 - 04/26/2021 02:13 PM - dalley

- Status changed from NEW to CLOSED - WONTFIX