Pulp - Issue #4185

Content summary returns un-qualified plugin type names and counts

11/26/2018 08:18 PM - dalley

Status:	CLOSED - CURRENTRELEASE	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	ttereshc	**Estimated time:**	0:00 hour
Category:			
Sprint/Milestone:	3.0.0		
Severity:	2. Medium	**Groomed:**	Yes
Version:		**Sprint Candidate:**	Yes
Platform Release:		**Tags:**	
OS:		**Sprint:**	Sprint 46
Triaged:	Yes	**Quarter:**	

Description

The detail view of a repository version provides a "content summary". This looks like the following:

```
"content_summary": {
    "package": 50,
    "update": 2,
    "file": 5
}
```

The problem is that these names come from the TYPE field on the models, and aren't namespaced by plugin. "package" comes from the RPM plugin, but that name is super generic. If another plugin (hypothetically, let's say the Python plugin", defined a content type named "package", the RPM "package"s and the Python "package"s would be counted together as if they were the same content type.

A solution might be to namespace "type" by plugin somehow like so:

```
"content_summary": {
    "pulp_rpm.package": 50,
    "pulp_rpm.update": 2,
    "pulp_file.file": 5
}
```

I would discourage a convention-based solution to this problem since with enough plugins, this will eventually become a problem. I would say it's already a problem for understandability if not correctness.

Related issues:

Blocks Pulp - Task #4067: Commit migrations to source control	**CLOSED - CURRENTRELEASE**

Associated revisions

Revision 1fcd1e57 - 12/19/2018 05:52 PM - ttereshc

Fix FILE_CONTENT_NAME

re #4185 https://pulp.plan.io/issues/4185

Required PR: https://github.com/pulp/pulp/pull/3801

Revision adf83ff4 - 12/20/2018 01:04 PM - ttereshc

Prepend type for master/detail models with a plugin app label

closes #4185 https://pulp.plan.io/issues/4185

Required PR: https://github.com/pulp/pulp_file/pull/150

Revision adf83ff4 - 12/20/2018 01:04 PM - ttereshc

Prepend type for master/detail models with a plugin app label closes #4185 https://pulp.plan.io/issues/4185

Required PR: https://github.com/pulp/pulp_file/pull/150

Revision eb0a1e0f - 01/06/2019 07:09 AM - dalley

Change expected names of content types

re: #4185 https://pulp.plan.io/issues/4185

History

#1 - 11/26/2018 08:40 PM - dalley

- Description updated

#2 - 11/27/2018 04:34 PM - CodeHeeler

- Triaged changed from No to Yes

#3 - 11/27/2018 04:36 PM - CodeHeeler

- Groomed changed from No to Yes

#4 - 11/28/2018 04:19 PM - dalley

- Sprint Candidate changed from No to Yes

- Tags Pulp 3 RC Blocker added

#5 - 11/28/2018 05:26 PM - jortel@redhat.com

- Sprint set to Sprint 46

#6 - 12/05/2018 09:36 PM - daviddavis

- Blocks Task #4067: Commit migrations to source control added

#7 - 12/07/2018 12:12 AM - ttereshc

- Status changed from NEW to ASSIGNED

- Assignee set to ttereshc

#8 - 12/10/2018 04:26 PM - dalley

Just a quick update that a naming collision would likely cause problems long before it reached the content summary, since the name stored in the 'type' field is used for Master/Detail casting. So the bug described in the issue couldn't happen, as it would blow up before reaching that point.

However, we still have the general problem of possibly having colliding names in the 'type' field, and the general problem of that name not being very descriptive for many plugins (e.g. 'package', 'update'), so namespacing (somehow) is still a good idea.

Maybe we could configure our content viewsets to follow a naming more like this:

/api/v3/content/file/files/ --> /api/v3/content/pulp_file/files/
/api/v3/content/rpm/updates/ --> /api/v3/content/pulp_rpm/updates/
/api/v3/content/rpm/packages/ --> /api/v3/content/pulp_rpm/packages/

This would mean we would want to not define that part of "endpoint_name" manually and inject it in through some mechanism in core somehow?

https://github.com/pulp/pulp_rpm/blob/master/pulp_rpm/app/viewsets.py#L70

#9 - 12/10/2018 04:51 PM - bmbouter

dalley wrote:

> /api/v3/content/rpm/updates/ --> /api/v3/content/pulp_rpm/updates/

Where does 'rpm', come from today?
Where does 'pulp_rpm' come from when we switch this?
In this example is TYPE='updates'?

#10 - 12/10/2018 04:59 PM - dalley

Plugins currently write out 'rpm' manually as part of 'endpoint_name' when they write their viewsets. We do some funky things to break that apart inside of "NamedModelViewSet". Although, as I think about this, it then provides an inconsistency when compared against "/rpm/remotes/ and so forth, and I don't know that "/pulp_rpm/remotes/" would make as much sense.

The namespacing, somehow, needs to make it into the 'type' field. So "rpm.update" or "rpm.package" or "pulp_rpm.package" or something instead of just "package", to prevent collisions across plugins. And this would "just work" when it comes down to the content summary because 'type' is what content_summary is using under the hood. We could make this a documented convention, and maybe that's the easiest/best thing to do. I'd like to find an transparent and automatic way to make that happen but it might be too difficult or complex.

#11 - 12/10/2018 05:09 PM - ttereshc

Can we use app label for all the plugin specific endpoints? and also for content summary? This is how I read your comment#8, dalley .
I like the idea of doing it for every plugin to maintain unique paths across plugins.

#12 - 12/10/2018 05:12 PM - bmbouter

dalley that sounds like a pretty good plan.

What about another option where the namespace for all models comes from the Django AppConfig for that plugin, e.g.
https://github.com/pulp/pulp_rpm/blob/master/pulp_rpm/app/__init__.py#L10

So TYPE = 'updates' and label = 'pulp_rpm' and the URL would be /api/v3/content/pulp_rpm/updates/

#13 - 12/12/2018 12:14 AM - ttereshc

TL;DR: I think I found a way to automate this namespacing but I believe there is a bug in Django which prevents to do so at the moment. Any suggestions on an acceptable workaround until Django bug(?) is resolved?

My assumptions for this task:
- we want to namespace endpoints for content/ automatically
- we want to namespace content_summary content automatically
- we don't want to namespace endpoints for other master/detail models, like remotes, publishers, etc

There is a new feature in Python 3.6 for customizing subclasses. This can allow us to customize class attributes in plugin subclasses when needed. E.g. TYPE for Content subclasses and endpoints for ContentViewSet subclasses. Something like:

```
diff --git a/pulpcore/app/models/content.py b/pulpcore/app/models/content.py
index 665ecc47d..3b7bd904f 100644
--- a/pulpcore/app/models/content.py
+++ b/pulpcore/app/models/content.py
@@ -252,6 +252,12 @@ class Content(MasterModel, QueryMixin):
         verbose_name_plural = 'content'
         unique_together = ()

+
+    def __init_subclass__(cls, **kwargs):
+        if hasattr (cls, 'TYPE'):
+            cls.TYPE = app_label + str(cls.TYPE)
+        super().__init_subclass__(**kwargs)
+
     @classmethod
     def natural_key_fields(cls):
         """
```

It works for viewsets but not for models.
Django tried to fix it but unsuccessfully, in my opinion. The problem is that they defined their own metaclass for models and they didn't handle __init_subclass__ correctly in their __new__ method. That's my current thinking. I'll file an issue when I have a small reproducer for them.

Any thoughts on the approach?
Until the Django fixes the problem, we can generate endpoint_name attributes as described above but for content_summary there will be some nasty hack, probably here.

My worry is that if we leave this for now as a documentation problem and rely on plugin writer to namespace their plugin properly, I'm not sure there is a way to automate namespacing painlessly later, it will break REST API for plugin specific endpoints + will require a DB migration.

#14 - 12/12/2018 11:45 AM - daviddavis

+1 from me. I looked at the django fix for __init_subclass__ and there's a test that seems to do what we want basically. I'm surprised it doesn't work.

Anyway, +1 to opening a bug against django if we can come up with a reproducer. Good find.

#15 - 12/12/2018 02:28 PM - ttereshc

FWIW, adding __init_subclass__ directly into a model class triggers TypeError in Django model metaclass; using __init_subclass__ in a mixin class (Django test covers only this case) doesn't produce any error (which is tested in Django) ~~but doesn't work either (not tested in Django)~~ :\.

#16 - 12/12/2018 05:24 PM - bmbouter

ttereshc thanks for the great investigation. +1 to continuing with the automatic namespacing and doing the workarounds in pulp while we wait for upstream Django to fix it.

#17 - 12/13/2018 08:11 PM - ttereshc

Update:
I filed a Django bug https://code.djangoproject.com/ticket/30041
My small reproducer showed that mixin class works (it doesn't help us, we can't use it in pulpcore) but the direct usage of __init_subclass__ doesn't.

#18 - 12/16/2018 09:55 PM - ttereshc

- Status changed from ASSIGNED to POST

https://github.com/pulp/pulp/pull/3801

#19 - 12/19/2018 01:01 PM - ttereshc

The Django bug will probably be fixed with Django 2.2, there is already a patch for it https://github.com/django/django/pull/10763 .
It turned out that the bug I filed is only one of the consequences of the way __new__ is defined for models in Django and one of the issues became a release blocker for Django 2.2.

#20 - 12/19/2018 04:22 PM - ttereshc

This issue covers only the type filed in the DB (and content summary as a consequence).
Endpoints will be taken care of in the other issue #4279

#21 - 12/20/2018 04:48 PM - ttereshc

- Status changed from POST to MODIFIED

Applied in changeset pulp|adf83ff43542287ed595560bd5c05737236b0752.

#22 - 04/25/2019 06:44 PM - daviddavis

- Sprint/Milestone set to 3.0.0

#23 - 04/26/2019 10:33 PM - bmbouter

- Tags deleted (Pulp 3, Pulp 3 RC Blocker)

#24 - 12/13/2019 06:10 PM - bmbouter

- Status changed from MODIFIED to CLOSED - CURRENTRELEASE