

Pulp - Story #3560

As a plugin writer, I can write views (at arbitrary endpoints) which are discovered and registered with pulpcore.

04/10/2018 04:11 AM - amacdona@redhat.com

Status:	CLOSED - CURRENTRELEASE	Start date:	
Priority:	Normal	Due date:	
Assignee:	amacdona@redhat.com	% Done:	100%
Category:		Estimated time:	0:00 hour
Sprint/Milestone:	3.0.0	Tags:	
Platform Release:		Sprint:	Sprint 35
Groomed:	Yes	Quarter:	
Sprint Candidate:	Yes		
Description			
Plugin ViewSets are automatically registered with the router. There needs to be a similar mechanism for views. Modify PulpPluginAppConfig to register named_views. https://github.com/pulp/pulp/blob/3.0-dev/pulpcore/pulpcore/app/apps.py#L46-L112 The urls.py is able to retrieve the app configs with named_views. https://github.com/pulp/pulp/blob/3.0-dev/pulpcore/pulpcore/app/urls.py#L102			
Related issues:			
Related to Pulp - Story #3473: As a plugin writer, I have documentation on ho...		CLOSED - WONTFIX	

Associated revisions

Revision 69954d63 - 04/16/2018 07:03 PM - amacdona@redhat.com

Register plugin provided urls.py

If plugins define a urls.py, register it. This allows the plugin writer to register views at arbitrary endpoints, allowing them to implement a Live API.

fixes #3560 <https://pulp.plan.io/issues/3560>

Revision 69954d63 - 04/16/2018 07:03 PM - amacdona@redhat.com

Register plugin provided urls.py

If plugins define a urls.py, register it. This allows the plugin writer to register views at arbitrary endpoints, allowing them to implement a Live API.

fixes #3560 <https://pulp.plan.io/issues/3560>

History

#1 - 04/10/2018 04:12 AM - amacdona@redhat.com

- Tags Pulp 3, Pulp 3 MVP added

#2 - 04/10/2018 06:33 PM - daviddavis

I'm interested to know how this fits with [#3473](#) because this was the use case I was trying to capture initially in [#3473](#).

#3 - 04/11/2018 03:28 PM - amacdona@redhat.com

[daviddavis](#), I read [#3473](#) to be primarily about preventing endpoint collisions, but it does also include registering Views. I'll relate that issue to this one. This story will be the mechanism, [#3743](#) will be docs and endpoint conventions to prevent collisions.

This story is to write the mechanism to register Views, but does not include collision prevention. Currently in the code there is a mechanism for plugin writers to register ViewSets, but there is no mechanism to register Views. With ViewSets, plugin writers can implement arbitrary object CRUD, but if their Live API is not based on resource CRUD, they will need Views.

#4 - 04/11/2018 03:32 PM - amacdona@redhat.com

- Related to Story #3473: As a plugin writer, I have documentation on how to create live api endpoints responsibly added

#5 - 04/11/2018 05:10 PM - amacdona@redhat.com

- Groomed changed from No to Yes
- Sprint Candidate changed from No to Yes
- Sprint set to Sprint 35

#6 - 04/11/2018 06:55 PM - amacdona@redhat.com

Q: "Why do plugin writers need Views in addition to ViewSets?"

A: ViewSets are associated with Django Models. If a plugin's Live API only involves object CRUD, ViewSets are fine. Endpoints that do not involve CRUD of a Model need a View.

#7 - 04/13/2018 07:44 PM - amacdona@redhat.com

A downside to registering views like we do with viewsets is that the viewsets have extra code that registers them based on class attributes.

<https://github.com/pulp/pulp/blob/3.0-dev/pulpcore/pulpcore/app/urls.py#L70>
<https://github.com/pulp/pulp/blob/3.0-dev/pulpcore/pulpcore/app/viewsets/base.py#L161-L198>

This seems like a lot of unnecessary cruft for plugin-provided views. A more standard way (Django, not DRF) to do it would be to allow each plugin to implement a urls.py which is dynamically imported. This is similar to how we do viewsets, but the difference is that core would only have to register each plugin's singular urls.py, which itself would contain the structure of their views. This seems nice because the plugin writer could structure their views in one place, and core wouldn't have to guess their needs.

<https://stackoverflow.com/questions/19833393/django-dynamically-add-apps-as-plugin-building-urls-and-other-settings-automat>

#8 - 04/13/2018 07:56 PM - amacdona@redhat.com

Concerns: this story is to register **any endpoint, without namespace**. We should **suggest** a namespace, but not enforce it.

If an endpoint is registered twice, who wins? [first | last | raise] (Does it make a difference if we are including a urls.py or registering a viewset with a router?)

If a plugin urls and a viewset (pulpcore or plugin) register the same endpoint, who **should** win? [plugin | pulpcore | raise]

Is it **possible** (the mvp is clear that this shouldn't be required) for a plugin use a different port to give its live API all the room it needs?

Does django allow us to ``include`` more than one urls.py at "/"?

<https://docs.djangoproject.com/en/2.0/ref/urls/#include>

#9 - 04/13/2018 08:21 PM - bmbouter

amacdona@redhat.com wrote:

Concerns: this story is to register **any endpoint, without namespace**. We should **suggest** a namespace, but not enforce it.

Do you mean suggesting a namespace like a dedicated one that they can feel they fully own and have a right to? Got any suggestions?

I like the idea that plugins can contribute views anywhere; at '/' or anywhere. For instance with this capability I believe someone is going to write a UI for Pulp that is a plugin.

If an endpoint is registered twice, who wins? [first | last | raise] (Does it make a difference if we are including a urls.py or registering a viewset with a router?)

Two plugins that are asserting ownership over the same namespace just won't work as expected no matter who wins. A system like that would be broken, so I don't think we need to answer this question. Raise may be a nice to have behavior in the medium/long term.

If a plugin urls and a viewset (pulpcore or plugin) register the same endpoint, who **should** win? [plugin | pulpcore | raise]

Similar answer here as above. It won't work so having someone "win" still leaves Pulp in a broken state. So a raise feature makes the most sense, but also that can come much later (I think).

Is it **possible** (the mvp is clear that this shouldn't be required) for a plugin use a different port to give its live API all the room it needs?

It is possible yes. The reason to still have all APIs to serve on one host is because it makes Pulp much easier to deploy. All deployments are the same no matter what is installed. I was told about this pain point from Satellite installer engineers.

Does django allow us to ``include`` more than one urls.py at "/"?

Two identical url strings can't both be involved in 1 request so the short answer is no. What Django does allow you to do is to put your most specific urls at the top so they are matched first. Then the most general url "/" should be matched last. But having two "/" urls both of them won't match.

<https://docs.djangoproject.com/en/2.0/ref/urls/#include>

#10 - 04/13/2018 08:33 PM - amacdona@redhat.com

bmbouter wrote:

amacdona@redhat.com wrote:

Concerns: this story is to register **any endpoint, without namespace**. We should **suggest** a namespace, but not enforce it.

Do you mean suggesting a namespace like a dedicated one that they can feel they fully own and have a right to? Got any suggestions?

Yeah, it is /plugins/<plugin>/<whatever>/. Since that is convention only, it is covered in the story for documenting how to do this.

<https://pulp.plan.io/issues/3473>

I like the idea that plugins can contribute views anywhere; at '/' or anywhere. For instance with this capability I believe someone is going to write a UI for Pulp that is a plugin.

If an endpoint is registered twice, who wins? [first | last | raise] (Does it make a difference if we are including a urls.py or registering a viewset with a router?)

Two plugins that are asserting ownership over the same namespace just won't work as expected no matter who wins. A system like that would be broken, so I don't think we need to answer this question. Raise may be a nice to have behavior in the medium/long term.

I meant how does Django's ``include`` handle this?

If a plugin urls and a viewset (pulpcore or plugin) register the same endpoint, who **should** win? [plugin | pulpcore | raise]

Similar answer here as above. It won't work so having someone "win" still leaves Pulp in a broken state. So a raise feature makes the most sense, but also that can come much later (I think).

Is it **possible** (the mvp is clear that this shouldn't be required) for a plugin use a different port to give its live API all the room it needs?

It is possible yes. The reason to still have all APIs to serve on one host is because it makes Pulp much easier to deploy. All deployments are the same no matter what is installed. I was told about this pain point from Satellite installer engineers.

If there are collisions, this would be the only way to avoid them. Despite being a pain, I think this will be the safest option, so I bet users will want to do that.

Does django allow us to ``include`` more than one urls.py at "/"?

Two identical url strings can't both be involved in 1 request so the short answer is no. What Django does allow you to do is to put your most specific urls at the top so they are matched first. Then the most general url "/" should be matched last. But having two "/" urls both of them won't match.

I think you misunderstood my question. pulpcore's urls.py will ``include`` each plugin-provided urls.py. Typically, this is done in a namespaced way, so it would be a little strange to include based on "/". Conceptually this is fine, but will this work, like, technically, with Django?

<https://docs.djangoproject.com/en/2.0/ref/urls/#include>

#11 - 04/13/2018 09:41 PM - amacdona@redhat.com

- Status changed from NEW to ASSIGNED

- Assignee set to amacdona@redhat.com

#12 - 04/16/2018 06:40 PM - amacdona@redhat.com

- Status changed from ASSIGNED to POST

<https://github.com/pulp/pulp/pull/3439>

#13 - 04/16/2018 06:41 PM - amacdona@redhat.com

Note: If a plugin attempts to register an endpoint that is used by pulpcore, it looks like the pulpcore endpoint wins. This seems to indicate that "first registration wins".

#14 - 04/18/2018 05:41 PM - amacdona@redhat.com

- Status changed from POST to MODIFIED

- % Done changed from 0 to 100

Applied in changeset [pulp|69954d63328f3dd322235a5282b9a9c83aaa18e5](https://github.com/pulp/pulp/pull/3439).

#15 - 04/19/2018 05:12 PM - dkliban@redhat.com

- Sprint/Milestone set to 3.0.0

#16 - 04/26/2019 10:35 PM - bmbouter

- Tags deleted (Pulp 3, Pulp 3 MVP)

#17 - 12/13/2019 06:30 PM - bmbouter

- Status changed from MODIFIED to CLOSED - CURRENTRELEASE