

## Ansible Plugin - Story #5522

Story # 5517 (NEW): [EPIC] Automation Hub Release Blockers

### As a user, I can filter ImportCollection response messages by date to receive limit the set of messages returned

09/30/2019 11:25 PM - bmbouter

<b>Status:</b>	CLOSED - CURRENTRELEASE	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	daviddavis	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0:00 hour
<b>Sprint/Milestone:</b>		<b>Tags:</b>	
<b>Platform Release:</b>		<b>Sprint:</b>	Sprint 60
<b>Groomed:</b>	Yes	<b>Quarter:</b>	
<b>Sprint Candidate:</b>	No		

**Description**

**Problem**

Each time the user fetches logging output from GALAXY\_API\_ROOT/v3/imports/collections/<uuid:pk>/ it comes from [this viewset](#)

The issue is that the user wants to receive new log messages, but they get all logs every time.

**Solution**

Rename the [CollectionImport](#) Model by:

- removing the "messages" attribute
- creating three new attributes
  - message <--- long string, required.
  - level. <--- this should be a choice field w/ the standard [Python logging levels](#) enumerated
  - time. This one can be configured to auto-record at save using the django ORM syntax for that.
- The "add\_log\_record" method should be removed
- Port the [logutils.CollectionImportHandler](#) to create a CollectionImport for each call to emit()
- Rename CollectionImport to CollectionImportMessage, and update migrations accordingly.

#### Associated revisions

Revision 395ba61e - 10/14/2019 06:44 PM - daviddavis

Add since filter for collection import messages

fixes #5522 <https://pulp.plan.io/issues/5522>

#### History

#1 - 09/30/2019 11:26 PM - bmbouter

- Parent task set to #5517

#2 - 10/03/2019 11:18 PM - bmbouter

- Subject changed from As a user, I can provide a 'since' attribute to filter response data for ImportCollection to As a user, I can filter ImportCollection response messages by date to receive limit the set of messages returned

- Description updated

#3 - 10/03/2019 11:25 PM - bmbouter

- Description updated

#4 - 10/03/2019 11:25 PM - daviddavis

- Groomed changed from No to Yes

- Sprint set to Sprint 60

**#5 - 10/03/2019 11:26 PM - daviddavis**

- Status changed from NEW to ASSIGNED

- Assignee set to daviddavis

**#6 - 10/04/2019 09:27 PM - daviddavis**

For the time field, it looks like we currently use the log record's created field[0]. Will using the db record creation timestamp cause different values/ordering?

[0] [https://github.com/pulp/pulp\\_ansible/blob/055305d7b928f9cef84bacb6b086ea9a41b23a89/pulp\\_ansible/app/models.py#L70](https://github.com/pulp/pulp_ansible/blob/055305d7b928f9cef84bacb6b086ea9a41b23a89/pulp_ansible/app/models.py#L70)

**#7 - 10/04/2019 10:15 PM - bmbouter**

daviddavis wrote:

For the time field, it looks like we currently use the log record's created field[0]. Will using the db record creation timestamp cause different values/ordering?

[0] [https://github.com/pulp/pulp\\_ansible/blob/055305d7b928f9cef84bacb6b086ea9a41b23a89/pulp\\_ansible/app/models.py#L70](https://github.com/pulp/pulp_ansible/blob/055305d7b928f9cef84bacb6b086ea9a41b23a89/pulp_ansible/app/models.py#L70)

It likely will change the values but not the ordering (I think). Messages created one after the other would still be time-sorted correctly if they are sorted on that field by default.

The values changing I think is ok because two representations of the same moment in time could be translated. I'm making this up, but say the logger uses epoch, but the DB uses a strftime() based timestamp, I think the user would submit the type required from the API and the internals would use it's datetime equivalent as it queries the db, so it should "just work".

Maybe I'm missing something though, not working on the code directly.

**#8 - 10/07/2019 07:07 PM - daviddavis**

Follow up question: do we need a time field since it duplicates the pulp\_created field?

**#9 - 10/07/2019 07:32 PM - bmbouter**

daviddavis wrote:

Follow up question: do we need a time field since it duplicates the pulp\_created field?

I don't think we need it, it would be redundant.

**#10 - 10/10/2019 01:42 PM - daviddavis**

This is a bit more complex than I thought. Currently, there is a collection imports endpoint[0]. I thought I could "fake" a CollectionImport in the API by using the collection import Task instead. However, the design is complex and not very straightforward. Also, it would mean that if galaxy ever wanted to store collection import data, they would have no place to store it. I think it might be better to keep the CollectionImport model instead and create a new CollectionImportMessage model.

[0] [https://github.com/pulp/pulp\\_ansible/blob/078942a1e695135af09fcd0f4759824d8a3df498/pulp\\_ansible/app/urls.py#L67-L71](https://github.com/pulp/pulp_ansible/blob/078942a1e695135af09fcd0f4759824d8a3df498/pulp_ansible/app/urls.py#L67-L71)

**#11 - 10/11/2019 09:30 PM - daviddavis**

I added a since filter to just filter the messages as they are now. I think this is probably the easiest/simplest solution:

[https://github.com/pulp/pulp\\_ansible/pull/229](https://github.com/pulp/pulp_ansible/pull/229)

**#12 - 10/14/2019 06:44 PM - daviddavis**

- Status changed from ASSIGNED to MODIFIED

- % Done changed from 0 to 100

Applied in changeset [pulp\\_ansible|395ba61ed0d499285e2fe204d51307ca7c9f9251](https://github.com/pulp/pulp_ansible/pull/229).

**#13 - 09/23/2020 10:01 PM - bmbouter**

- Status changed from MODIFIED to CLOSED - CURRENTRELEASE