

## Pulp - Task #6965

### Remote.get\_downloader() needs to include remote in kwargs for Factory.build()

06/11/2020 05:37 PM - dkliban@redhat.com

<b>Status:</b> NEW	<b>Start date:</b>
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b>	<b>Estimated time:</b> 0:00 hour
<b>Sprint/Milestone:</b>	<b>Tags:</b> Documentation
<b>Platform Release:</b>	<b>Sprint:</b>
<b>Groomed:</b> No	<b>Quarter:</b>
<b>Sprint Candidate:</b> No	

**Description**

The plugin writer docs need to be updated to explain how custom downloaders should be implemented. The documentation should be based on the information provided in comment 2 of this issue.

**Original Description**

Multiple plugins now override Remote.get\_downloader() method in order to add the 'remote' to the kwargs[0].

The pulpcore implementation of Remote.get\_downloader() should pass the 'remote' as a kwarg to the Factory.build() method[1].

[0] [https://github.com/pulp/pulp\\_container/blob/master/pulp\\_container/app/models.py#L251](https://github.com/pulp/pulp_container/blob/master/pulp_container/app/models.py#L251) [1] <https://github.com/pulp/pulpcore/blob/master/pulpcore/download/factory.py#L108>

#### History

##### #1 - 06/18/2020 08:30 PM - bmbouter

I agree we should do something, but it's a little unclear exactly what at the moment. Some IRC or video chat may be helpful.

Having remote added in as the base implementation is a good step forward, but those kwargs ultimately get passed to a HttpDownloader and the BaseDownloader and that would make the base implementation inconsistent with itself if remote is never expected in either of those objects. My initial thought is that we would also have either HttpDownloader or BaseDownloader accept the remote kwarg, which would be similar to what plugins are also doing: [https://github.com/pulp/pulp\\_container/blob/bfb7422c85210a6208c4752d583b454c900b7249/pulp\\_container/app/downloaders.py#L32](https://github.com/pulp/pulp_container/blob/bfb7422c85210a6208c4752d583b454c900b7249/pulp_container/app/downloaders.py#L32)

##### #2 - 06/18/2020 11:27 PM - bmbouter

@dkliban and I talked this over and the summary is that this plugin pattern is not a good pattern and we should do the plugin code differently. The pulpcore code should likely stay the same.

#### The problems with the current plugin code are:

1. It's confusing. You are creating a kwarg in a Remote method and magically it's brought to your downloader and you have to pop it or the code won't run
2. The subclassed downloader in the plugin requires the attributes from the Remote, yet it's been passed in silently as an undeclared kwarg. Even if the kwarg is declared required arguments are typically passed as positional arguments. This is a weakness and usability concern of the subclassed downloaders.

#### The problem with continuing down this road

To have the base implementation of the factory pass along the Remote to a pulpcore provided downloader (not a subclassed one) we would have to add 'remote' as a param to either HttpDownloader and FileDownloader, or BaseDownloader. That would be strange because remote wouldn't even be used. It has no purpose in the base design if we were to do that.

#### Solution (do all the steps below):

1. Have the attributes the subclassed downloader needs from the remote passed into the signature of the subclassed downloader.

2. The downloader historically hasn't had a reference to the remote because the Factory is designed to be the interpreter of a Remote and that's why the Factory has the reference to it already. So generally avoid having the 'remote' passed to the downloader, and instead have the attributes themselves passed in. This keeps the concern of "interpreting the remote and constructing the downloader" in the factory, which is its purpose.
3. subclass the factory so it can pass along the newly required positional args from the Remote to the subclassed downloader it is constructing.
4. If plugins use ^ approach, they won't have to define a get\_downloader() at all instead their get\_download\_factory would implicate using their customized Factory and all will be well.

**#3 - 06/19/2020 05:41 PM - dkliban@redhat.com**

- *Description updated*

- *Tags Documentation added*