

## Python Support - Story #4690

### Implement package locking / digest specification

04/15/2019 07:53 PM - dalley

<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>
<b>Platform Release:</b>	<b>Sprint:</b>
<b>Target Release - Python:</b>	<b>Quarter:</b>
<b>Groomed:</b> No	
<b>Sprint Candidate:</b> No	

**Description**

There are two primary ways that I can foresee implementing this.

1. Either make a self-contained object that can be created from pipenv.lock, or something similar, which contains a full listing of all packages, their versions, and their digests being used, and allow it to be linked to the remote and supplement or take the place of the remote's own include and exclude settings when being used, or
2. Make a separate field called "locked\_versions", that would supplement or take the place of the includes or excludes fields when being used

Both of these designs would:

- Simplify the sync codepaths **dramatically**
- Enforce the fact that digests only make sense in the context of a single release
- Provide more guarantees to the repo owner that when they are using digests for reproducibility, they have a solid guarantee that the entire contents of the repository is the reproducible set

They would also:

- Make it possible to simplify the format of "includes" and "excludes".
  - We could let them accept an array of requirements.txt-style strings instead of dictionaries. e.g. includes=["django>=1,<=2.1,==2.3", "flask==2.0.1", "bottle~=3.2"]
  - This might be a little harder to validate so maybe we wouldn't do this, but it's possible and I think it would make Pulp feel a lot more familiar
- Open up the possibility for adding various strict modes
  - maybe in strict mode, you're only allowed to have one release per project (with or without providing digests)
  - maybe in strict mode, you're **only** allowed to have locked versions on the remote (no using includes/excludes), and providing hashes is enforced for all locked versions, similar to how pip behaves in hash-checking mode

**Related issues:**

Related to Python Support - Story #4711: As a user, I can specify Remote.incl...	<b>NEW</b>
Related to Python Support - Story #4713: As a user, I can specify Remote.incl...	<b>NEW</b>
Related to Python Support - Story #6929: As a User, I can configure Pulp Pyth...	<b>MODIFIED</b>

#### History

#1 - 04/15/2019 07:55 PM - dalley

- Description updated

**#2 - 04/15/2019 07:59 PM - dalley**

- Description updated

**#3 - 04/15/2019 11:21 PM - dalley**

- Subject changed from *Change the way digest specification works to better match how the ecosystem works to Implement package locking / digest specification*

- Description updated

**#4 - 04/26/2019 10:30 PM - bmbouter**

- Tags deleted (Pulp 3)

**#5 - 06/02/2020 06:32 AM - dalley**

- Related to Story #6875: *[Meta] Integrate with Bandersnatch added*

**#6 - 06/02/2020 06:33 AM - dalley**

- Related to Story #4711: *As a user, I can specify Remote.includes with a requirements.txt added*

**#7 - 06/02/2020 06:33 AM - dalley**

- Related to Story #4713: *As a user, I can specify Remote.includes with a pipenv.lock files added*

**#8 - 06/08/2020 10:45 PM - dalley**

- Related to deleted (Story #6875: *[Meta] Integrate with Bandersnatch*)

**#9 - 06/08/2020 10:47 PM - dalley**

- Related to Story #6929: *As a User, I can configure Pulp Python remotes using a Bandersnatch config file added*