Pulp - Task #2327

Add apidoc to the docs script

10/06/2016 04:48 PM - bmbouter

Status:	CLOSED - CURRENTRELEASE	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	semyers	**% Done:**	100%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:	3.0		
Platform Release:		**QA Contact:**	
Blocks Release:		**Complexity:**	
Backwards Incompatible:	No	**Smash Test:**	
Groomed:	Yes	**Verified:**	No
Sprint Candidate:	Yes	**Verification Required:**	No
Tags:		**Sprint:**	Sprint 9

Description

With Pulp3 we will be using apidoc. The docs builder needs this enabled.

Associated revisions

Revision 8c23b326 - 11/09/2016 05:51 PM - semyers

Build API docs using sphinx autodoc

This requires turning off "nit-picky" mode due to references to types unknown
to sphinx without getting intersphinx involved (e.g. str, django.db.models.Model,
etc.)

I added a little extension to napoleon to support our "Fields" and "Relations"
sections that we use in our Django models, which made it easy to test different
Sphinx methods of exposing class attributes, and also lets us use napoleon. :)

closes #2327
https://pulp.plan.io/issues/2327

Revision 8c23b326 - 11/09/2016 05:51 PM - semyers

Build API docs using sphinx autodoc

This requires turning off "nit-picky" mode due to references to types unknown
to sphinx without getting intersphinx involved (e.g. str, django.db.models.Model,
etc.)

I added a little extension to napoleon to support our "Fields" and "Relations"
sections that we use in our Django models, which made it easy to test different
Sphinx methods of exposing class attributes, and also lets us use napoleon. :)

closes #2327
https://pulp.plan.io/issues/2327

Revision 8c23b326 - 11/09/2016 05:51 PM - semyers

Build API docs using sphinx autodoc

This requires turning off "nit-picky" mode due to references to types unknown to sphinx without getting intersphinx involved (e.g. str, django.db.models.Model, etc.)

I added a little extension to napoleon to support our "Fields" and "Relations" sections that we use in our Django models, which made it easy to test different Sphinx methods of exposing class attributes, and also lets us use napoleon. :)

closes #2327
https://pulp.plan.io/issues/2327

Revision ca014a7e - 11/09/2016 09:34 PM - semyers

Update docs jobs and builder to install pulp for apidocs

re #2327
https://pulp.plan.io/issues/2327

Revision 0752dd3e - 11/15/2016 10:28 PM - semyers

Restore platform API docs

After the work in https://pulp.plan.io/issues/2327, we did a docs restructure that didn't include that work. This restores and improves upon it, by including the entire platform API as it stands today.

Minor changes had to be made to some modules to get the docs to build, including correcting sphinx syntax errors and fixing some imports that are no longer valid.

re #2327

Revision 12bf1305 - 11/15/2016 10:41 PM - semyers

Restore platform API docs

After the work in https://pulp.plan.io/issues/2327, we did a docs restructure that didn't include that work. This restores and improves upon it, by including the entire platform API as it stands today.

Minor changes had to be made to some modules to get the docs to build, including correcting sphinx syntax errors and fixing some imports that are no longer valid.

re #2327

Revision 12bf1305 - 11/15/2016 10:41 PM - semyers

Restore platform API docs

After the work in https://pulp.plan.io/issues/2327, we did a docs restructure that didn't include that work. This restores and improves upon it, by including the entire platform API as it stands today.

Minor changes had to be made to some modules to get the docs to build, including correcting sphinx syntax errors and fixing some imports that are no longer valid.

re #2327

Revision 12bf1305 - 11/15/2016 10:41 PM - semyers

Restore platform API docs

After the work in https://pulp.plan.io/issues/2327, we did a docs restructure that didn't include that work. This restores and improves upon it, by including the entire platform API as it stands today.

Minor changes had to be made to some modules to get the docs to build, including correcting sphinx syntax errors and fixing some imports that are no longer valid.

re #2327

History

#1 - 10/06/2016 04:50 PM - semyers

Do we want to use http://www.sphinx-doc.org/en/stable/man/sphinx-apidoc.html or explicitly call out to http://www.sphinx-doc.org/en/stable/ext/autodoc.html#module-sphinx.ext.autodoc?

#2 - 10/06/2016 04:59 PM - bmbouter

I don't have much experience with these things, but I think the integrated one with the commands in the .rst structure will be the best. I think that would be this[0]. The binary one would have us need to be more involved in organization of the produced content. I don't know much about these things.

Will there need to be a conf.py change? Maybe we need to enable the apidoc module? If so, FYI this conf.py[1] is the one for the hosted site. Each plugin has its own conf.py, which would need to be updated to build the docs locally without errors when apidoc commands are encountered.

[0]: http://www.sphinx-doc.org/en/stable/ext/autodoc.html#module-sphinx.ext.autodoc
[1]: https://github.com/pulp/pulp/blob/81ea84bb73364b39b398d91ada61c87595cd6267/docs/conf.py

#3 - 10/06/2016 05:04 PM - semyers

- Groomed changed from No to Yes

bmbouter wrote:

> I don't have much experience with these things, but I think the integrated one with the commands in the .rst structure will be the best. I think that would be this[0]. The binary one would have us need to be more involved in organization of the produced content. I don't know much about these things.

I agree. I also like the integrated (autodoc commands) approach because it's explicit, there's no mystery about where the api docs come from.

> Will there need to be a conf.py change? Maybe we need to enable the apidoc module? If so, FYI this conf.py[1] is the one for the hosted site. Each plugin has its own conf.py, which would need to be updated to build the docs locally without errors when autodoc commands are encountered.

Yeah, we'll need to add sphinx.ext.autodoc to the extensions list in the conf. I'm personally okay with the thought that any plugin itself using autodoc commands would need to name autodoc among its extensions as well.

I'm tinkering with this right now since I hate having an open PR that breaks the docs; I might sorta accidentally do this.

#4 - 10/06/2016 05:56 PM - bmbouter

Having the autodoc enabled in plugins as-needed sounds fine.

+1 to you maybe sorta accidentally doing this. Should we add this to the sprint?

#5 - 10/06/2016 06:07 PM - semyers

- Status changed from NEW to ASSIGNED

- Assignee set to semyers

- Sprint/Milestone set to 27

Since my PR for #1873 is the one that brings this about and it should be a relatively small task, I'll pick it up and test it against my breaking PR.

#6 - 10/07/2016 01:06 AM - semyers

This went fairly easily, except for one pretty major problem: We can't compile docs strictly (warnings become errors) due to a bug in sphinx[0].

The bug references :ivar:, but the same problem also affects :cvar: (which we use extensively), and I assume also :var:.

I personally like what we're doing with fields and relations called out explicitly with :cvar:, but it looks like if we want to keep it, our options are (in order of best-ness):

1. Invest the time and either fix this issue upstream or otherwise move the issue forward by helping in some way. Build docs with warnings disabled in the meantime.
2. Build docs with warnings disabled.
3. Stop using :*var: to represent object attributes. I don't like this because it makes the docs a lot harder to parse (human-parse, I mean).
4. Don't use autodoc to document our modules, instead documenting our modules separately from their code.
5. Never use the same model field name more than once in the project. (definitely least best, but only barely).

So assuming there are no major disagreements with what I think is the best option, I'll take a little bit of time and poke around in sphinx's guts and see about putting together a PR to stop these crazy warnings and allow us to build with -W again in a future version of sphinx. (**sadface**)

As a crazy side-effect of this investigation, I made a customized version of napoleon[1] that made it really easy to switch different rendering methods for Django-specific object attributes, and demonstrate how crappy everything that isn't :cvar: looks.

[0]: https://github.com/sphinx-doc/sphinx/issues/2549
[1]: http://www.sphinx-doc.org/en/stable/ext/napoleon.html

#7 - 10/07/2016 04:27 PM - bmbouter

@smyers thank you for such a great investigation.

+1 to your top proposal which is to fix the issue upstream and temporarily disable strict Sphinx builds. What needs to be done to fix sphinx? Any ideas there?

#8 - 10/07/2016 05:56 PM - semyers

Yeah, I have some ideas. For starters, it helps that there's already an open bug to ref.

Here's the sphinx code throwing the warnings that I'm seeing:
https://github.com/sphinx-doc/sphinx/blob/ea86d23845c9179d35420e0f353b47fa83c780f0/sphinx/domains/python.py#L738-L759

I think there's enough info local to that function (or its caller) to know whether or not to xref based on the current directive. I think it's entirely inappropriate for this xref function to be called in the case of :var:, :ivar:, and :cvar: since their role is explicitly local to their container, so either handling that case in the linked function or finding a place to bail out before it is called is a likely solution.

#9 - 10/07/2016 10:06 PM - semyers

I think the upstream issue is here:
https://github.com/sphinx-doc/sphinx/blob/ea86d23845c9179d35420e0f353b47fa83c780f0/sphinx/domains/python.py#L154-L157 (current master ref at time of this post)

The role for :var:, :ivar:, and :cvar: is "obj", which is a cross-referenceable type. Changing this bit to set the role to 'attr' make sphinx correctly handle them as the non-cross-referenced attribute type, and makes all the ugly warnings go away.

So we'll run without -W for now, and I'll get this fix in upstream ASAP.

#10 - 10/10/2016 07:35 PM - semyers

semyers wrote:

> The role for :var:, :ivar:, and :cvar: is "obj", which is a cross-referenceable type. Changing this bit to set the role to 'attr' make sphinx correctly handle them as the non-cross-referenced attribute type, and makes all the ugly warnings go away.

Argh, all the ugly warnings have not gone away, I was just running sphinx in such a way that they weren't being printed. The 'attr' trick didn't actually change anything. :(

The more I learn about this, the more I think that sphinx just shouldn't be trying to link these at all, so I'm getting to know sphinx a little bit better in hopes of figuring out how to make it stop trying to cross-ref them "the sphinx way".

#11 - 10/17/2016 04:50 PM - semyers

This appears to be a working fix, which I'll be submitting upstream soon:

```
diff --git a/sphinx/domains/python.py b/sphinx/domains/python.py
index 1639d82..c04c0b4 100644
--- a/sphinx/domains/python.py
+++ b/sphinx/domains/python.py
@@ -89,8 +89,8 @@ class PyXrefMixin(object):
                     contnode=None):
         result = super(PyXrefMixin, self).make_xref(rolename, domain, target,
                                                     innernode, contnode)
-        result['refspecific'] = True
         if target.startswith(('.', '~')):
+            result['refspecific'] = True
             prefix, result['reftarget'] = target[0], target[1:]
             if prefix == '.':
                 text = target[1:]
@@ -126,7 +126,7 @@ class PyObject(ObjectDescription):
                             'keyword', 'kwarg', 'kwparam'),
                     typerolename='obj', typenames=('paramtype', 'type'),
                     can_collapse=True),
-        PyTypedField('variable', label=l_('Variables'), rolename='obj',
+        PyTypedField('variable', label=l_('Variables'), rolename='attr',
                     names=('var', 'ivar', 'cvar'),
                     typerolename='obj', typenames=('vartype',),
                     can_collapse=True),
```

I'd like to better understand exactly what that ~~'reftarget'~~ 'refspecific' value is intended to do, but if my current understanding is correct, then this change turns attributes into link (reference) targets only when prefixed with '.' or '~'. This might be a breaking change, but it's worth pointing out that when using :vartype: with :*var:, sphinx does still attempt to reference the var type, leaving us in what I think is the desired state: Class/Instance/Module attrs aren't linked by default but can be turned into links explicitly, and links will be generated to their typedef (if possible) when that type is specified.

Edit: 'refspecific' :)

#12 - 10/17/2016 08:33 PM - semyers

- Status changed from ASSIGNED to POST

https://github.com/pulp/pulp/pull/2789

#13 - 11/09/2016 07:53 PM - semyers

- Status changed from POST to MODIFIED

- % Done changed from 0 to 100

Applied in changeset pulp|8c23b326cdb7739c420fdff8538a66e73147e0fb.

#14 - 11/09/2016 08:38 PM - semyers

Despite being MODIFIED, this is not quite done. Here's the problem:

In order to build the apidocs, sphinx needs to be able to import the code. This means, effectively, that the docs builder and everything it depends on need to run in a python3 virtualenv, since we need to install pulp 3 packages somewhere to make them importable by the sphinx builder.

This includes converting some of our building library code to be py3 compatible, which should be fairly easy to do in a cross-compatible way.

#15 - 11/09/2016 10:09 PM - semyers

- Status changed from MODIFIED to POST

Part 2, back to POST:
https://github.com/pulp/pulp_packaging/pull/246

#16 - 11/14/2016 03:15 PM - semyers

- Status changed from POST to CLOSED - CURRENTRELEASE

This required a little bit of crowbar action to get the relevant bits of our build scripts working in both python2 and python3, which the main actions taken being to switch from print statements to print functions, and to decode to utf8 "at the edges", usually when capturing the output of subprocess calls.

After all that, the 3.0-dev docs now include api docs:
https://docs.pulpproject.org/en/3.0/nightly/api/pulp/app/index.html

At the moment, the docs builder job does a better job of installing pulp 3 than our ansible playbook does. My hope is that the work done for #2413 to fix up the pulp 3 dev setup scripts will be reusable by the docs builder jobs for pulp 3+, removing this duplicated functionality.

#17 - 11/14/2016 07:09 PM - bmbouter

@smyers, thank you so much for this. Having 3.0 nightly docs is a big step forward. smyers++

#18 - 03/08/2018 08:11 PM - bmbouter

- Sprint set to Sprint 9

#19 - 03/08/2018 08:11 PM - bmbouter

- Sprint/Milestone deleted (27)

#20 - 04/25/2019 06:47 PM - daviddavis

- Sprint/Milestone set to 3.0

#21 - 04/26/2019 10:39 PM - bmbouter

- Tags deleted (Pulp 3)