

Foreman - Refactor #16172

Remove base_class override in template subclasses

08/18/2016 10:49 AM - Dominic Cleal

Status: Closed	
Priority: Normal	
Assignee: Dominic Cleal	
Category: Rails	
Target version: 1.13.0	
Difficulty:	Fixed in Releases:
Triaged:	Found in Releases:
Bugzilla link:	Red Hat JIRA:
Pull request: https://github.com/foreman/foreman/pull/3745	
Description	
<p>Both ProvisioningTemplate and Ptable contain overrides of the Rails base_class method, indicating that the STI base class of ProvisioningTemplate (etc) is ProvisioningTemplate, rather than Template.</p> <pre># we have to override the base_class because polymorphic associations does not detect it correctly , more details at # http://apidock.com/rails/ActiveRecord/Associations/ClassMethods/has_many#1010-Polymorphic-has-many-within-inherited-class-gotcha def self.base_class self end # not sure why but this changes table_name so we set it explicitly self.table_name = 'templates'</pre> <p>This causes an issue with the Template model instantiation under Rails 5.0, as ActiveRecord performs casting of new models, much like Foreman::STI does. This relies on base_class returning the actual base class, else the inheritance appears inverted and causes this exception:</p> <pre>DomainTest#test_0008_should update total_hosts on changing primary interface domain: ActiveRecord::SubclassNotFound: Invalid single-table inheritance type: Template is not a subclass of Ptable activerecord (5.0.0) lib/active_record/inheritance.rb:189:in `find_sti_class' activerecord (5.0.0) lib/active_record/inheritance.rb:209:in `subclass_from_attributes' activerecord (5.0.0) lib/active_record/inheritance.rb:58:in `new'</pre> <p>Removing base_class causes the problem noted in the comment above, that polymorphic associations to these models break - in fact, they search for the base class rather than the specific class. https://github.com/rails/rails/blob/v5.0.0/activerecord/lib/active_record/associations.rb#L780 notes that in actual fact, the <i>base class</i> should be stored in the type column for polymorphic associations, not the derivative.</p> <p>This should be changed to remove the overrides and store the base class when dealing with polymorphic STI associations (both Taxonomy and Audit are associated to Template derivatives).</p>	
Related issues:	
Related to Foreman Remote Execution - Refactor #16195: Remove base_class over...	New 08/19/2016

Associated revisions

Revision 5908bf41 - 08/22/2016 03:39 AM - Dominic Cleal

fixes #16172 - remove base_class override in template subclasses

On Rails 5.0, the base_class is used during model instantiation to validate STI relationships as it now casts similarly to Foreman::STI, causing failures when base_class returns the subclass rather than parent class.

Removing the overriding of `base_class` in Template subclasses

necessitates changing polymorphic associations to store the base class in the table's type field rather than the subclass name because the `has_many` associations will search by ``base_class``. This is recommended by ActiveRecord when using STI with polymorphic associations:
https://github.com/rails/rails/blob/v5.0.0/activerecord/lib/active_record/associations.rb#L780

History

#1 - 08/18/2016 10:59 AM - The Foreman Bot

- Status changed from *Assigned* to *Ready For Testing*
- Pull request <https://github.com/foreman/foreman/pull/3745> added

#2 - 08/19/2016 04:04 AM - Marek Hulán

- Related to Refactor #16195: *Remove base_class override in job_template* added

#3 - 08/22/2016 03:40 AM - Marek Hulán

- translation missing: *en.field_release* set to 160

#4 - 08/22/2016 04:02 AM - Dominic Cleal

- Status changed from *Ready For Testing* to *Closed*
- % Done changed from 0 to 100

Applied in changeset [5908bf418d28c57dcdc4d2e01ccf3253c0e8a81a](#).