

Foreman - Bug #6299

POST to /api/hosts fails when host has interfaces array

06/19/2014 07:09 PM - Aaron Stone

Status:	New	
Priority:	Normal	
Assignee:		
Category:	API	
Target version:		
Difficulty:		Fixed in Releases:
Triaged:		Found in Releases:
Bugzilla link:		Red Hat JIRA:
Pull request:		

Description

Using Foreman 1.5.1, if I POST to /api/hosts with a JSON body that contains an interfaces array, it fails with the following message in the log file. My goal is to create a host and its dependent IPMI interface entry in a single query.

I see a nearly identical message with 0 replies in the mailing list from April, that fellow was using Ruby 1.9. So this is a current bug and independent of the Ruby version.

My hunch is that something about the nested resources is not declared right in the Hosts model, or not working right at the ActiveRecord level.

Completed 500 Internal Server Error in 26.2ms (Views: 1.9ms | ActiveRecord: 3.2ms)
Started POST "/api/hosts" for 127.0.0.1 at Thu Jun 19 18:54:41 +0000 2014
Processing by Api::V2::HostsController#create as JSON
Parameters: {"apiv"=>"v2", "host"=>{"build"=>"false", "name"=>"hostname", "interfaces"=>{"name"=>"", "domain_id"=>"1", "ip"=>"1.2.3.4", "type"=>"Nic::BMC", "username"=>"ipmiuser", "mac"=>"AA:BB:CC:DD:EE:FF", "subnet_id"=>"26", "provider"=>"IPMI", "password"=>"[FILTERED]"}, "domain_id"=>"1", "model_id"=>"4", "environment_id"=>"1", "ip"=>"2.3.4.5", "mac"=>"FF:EE:DD:CC:BB:AA", "ptable_id"=>"11", "subnet_id"=>"10", "hostgroup_id"=>"3", "operatingsystem_id"=>"1"}}
Expire fragment views/tabs_and_title_records-1 0.2ms
Authorized user astone(Aaron Stone)
Nic::Base(#70287117238000) expected, got Array(#70287225667100) (ActiveRecord::AssociationTypeMismatch)
/usr/share/foreman/vendor/ruby/1.8/gems/activerecord-3.2.18/lib/active_record/associations/association.rb:210:in `raise_on_type_mismatch'
/usr/share/foreman/vendor/ruby/1.8/gems/activerecord-3.2.18/lib/active_record/associations/collection_association.rb:310:in `replace'
/usr/share/foreman/vendor/ruby/1.8/gems/activerecord-3.2.18/lib/active_record/associations/collection_association.rb:310:in `each'
/usr/share/foreman/vendor/ruby/1.8/gems/activerecord-3.2.18/lib/active_record/associations/collection_association.rb:310:in `replace'
/usr/share/foreman/vendor/ruby/1.8/gems/activerecord-3.2.18/lib/active_record/associations/collection_association.rb:41:in `writer'
/usr/share/foreman/vendor/ruby/1.8/gems/activerecord-3.2.18/lib/active_record/associations/builder/association.rb:51:in `interfaces='
/usr/share/foreman/vendor/ruby/1.8/gems/activerecord-3.2.18/lib/active_record/attribute_assignment.rb:94:in `send'
/usr/share/foreman/vendor/ruby/1.8/gems/activerecord-3.2.18/lib/active_record/attribute_assignment.rb:94:in `assign_attributes'
/usr/share/foreman/vendor/ruby/1.8/gems/activerecord-3.2.18/lib/active_record/attribute_assignment.rb:93:in `each'
/usr/share/foreman/vendor/ruby/1.8/gems/activerecord-3.2.18/lib/active_record/attribute_assignment.rb:93:in `assign_attributes'
/usr/share/foreman/vendor/ruby/1.8/gems/activerecord-3.2.18/lib/active_record/base.rb:498:in `initialize'
/usr/share/foreman/vendor/ruby/1.8/gems/actionpack-3.2.18/lib/action_dispatch/routing/url_for.rb:103:in `initialize'
/usr/share/foreman/app/models/concerns/foreman/sti.rb:22:in `new_without_cast'
/usr/share/foreman/app/models/concerns/foreman/sti.rb:22:in `new'
/usr/share/foreman/app/models/host.rb:16:in `send'
/usr/share/foreman/app/models/host.rb:16:in `method_missing'
/usr/share/foreman/app/controllers/api/v2/hosts_controller.rb:70:in `create'

History

#1 - 06/19/2014 10:38 PM - Aaron Stone

This stick out at me:
/usr/share/foreman/app/models/concerns/foreman/sti.rb:22:in `new_without_cast'
/usr/share/foreman/app/models/concerns/foreman/sti.rb:22:in `new'

It means that in the Single Table Inheritance helper, the piece of code that handles the type field is not triggering, and instead falling through to the default `new_without_cast`. See <https://github.com/theforeman/foreman/blob/1.5-stable/app/models/concerns/foreman/sti.rb#L22>

#2 - 06/23/2014 07:03 PM - Aaron Stone

Is it as simple as changing from

```
{"host": { "interfaces": [ ... ] } }
```

to

```
{"host": { "interfaces_attributes": { "foo_1": { ip, mac, etc } } } }
```

As seen here: https://github.com/theforeman/hammer-cli-foreman/blob/master/lib/hammer_cli_foreman/host.rb#L83

The "foo_1" intermediate hash was really confusing. Turns out this is a bleed-through of how Rails handled forms, where each input would be something like:

```
<input name="host[interfaces_attributes][new_1][ip]">
<input name="host[interfaces_attributes][new_1][mac]">
```

In a JSON API, you don't need the intermediate hash because all of your data is in the same structure, but that is unfortunately not how it was implemented yet.

Let's update the durned docs!!

#3 - 06/24/2014 07:28 AM - Dominic Cleal

- *Category set to API*