

Philo - Bug #68

JSONValue objects cannot be imported from fixtures

01/06/2011 10:06 AM - Joseph Spiros

Status:	Resolved	Start date:	01/06/2011
Priority:	Normal	Due date:	
Assignee:	Stephen Burrows	% Done:	0%
Category:	Core	Estimated time:	0.00 hour
Target version:			
Description			

Associated revisions

Revision f50cd63e - 01/28/2011 12:45 PM - Joseph Spiros

Merge branch 'master' of git://github.com/melinath/philo

- 'master' of git://github.com/melinath/philo: (31 commits)
Merged cowell back into core.
Added support for DateTime information being stored in a JSONAttribute. Switched ForeignKeyAttribute and M2MAttribute to store the related model in self.to instead of overwriting self.model.
Initial raw_id_fields support for proxy fields. Involves hacks to bypass model validation for things like raw_id_fields, where validation depends on model._meta.get_field(). Renamed EntityForm to ProxyFieldForm to more accurately reflect its purpose. Removed extraneous code from ProxyFieldForm that didn't belong there.
Ungenericized get_item_queryset and added tag/issue queryset fetching methods to BlogView and NewsletterView, respectively. Re-added the tag_archive_view and issue_archive_view methods and set them up to use these methods.
Added get_item_queryset method to BlogView and NewsletterView so that subclasses can replace this functionality. Moved get_context hook onto the MultiView base class and added a basic_view method which provides a simple way to add a view method to a multiview's urlpatterns for a related View subclass instance.
Added missing import to forms/fields.py
Small tweaks to TreeModel.get_path to take advantage of updates to django-mptt. Updated README accordingly.
Moved container forms and attribute forms into admin. Moved AttributeFields and EntityForm into contrib/cowell/; this more accurately reflects their status as convenient in certain circumstances. Simplified imports according to these changes. Made further tweaks to ManyToManyValue's methods.
Split forms into containers, entities, and fields. Split attribute fields out from other model fields. Revamped the interaction between AttributeForm and the AttributeValue subclasses to more clearly define their roles. Moved all code related to AttributeFields into models.fields.attributes.
Genericized AttributeFieldDescriptor and removed the other descriptor options.
Minor correction to NavigationManager caching.
Reduced number of queries for BlogView.get_entries_by_tag. Corrected waldo's AccountMultiView.has_valid_account() method, which had holdovers from the user_form/profile_form distinction.
Improved blog and newsletter ModelAdmins. Set BlogEntries and NewsletterArticlesup to set their dates on save if it wasn't set manually - gives more accurate publishing times than a datetime.now() default. Set feed creation to use templates for title and description so that projects can customize the display for markdown etc. Limited Attribute.key to word characters to ensure template usability. Ordered Tags by name.
Added a Navigation model to mediate between Nodes and multiple sets of navigation. Updated templatetags accordingly. Addresses issue 70.
Added fix_init_kwarg method to JSONField and connected it to the pre_init signal during contribute_to_class - this effectively mimics what django does for RelatedObject fields in Model.__init__ by allowing a python value to be passed in to the constructor (e.g. JSONValue(value=()) instead of JSONValue(value_json='{}')). This resolves issue 68.
Split shpherd NodeNavigationInline into inlines for hosted/targeting inlines. Added NodeAdmin reregistration to bring the new inlines into play.
Added navigation_host filter.
Minor correction to Navigation.is_active()
updated grappelli_tabular_attribute.html to match changes in grappelli 2.3
Shifted NavigationManager caching to use node instances rather than node pks. Improved (i.e. added) a distinction between having a cache for a node and having a Navigation instance somewhere in the cache. Refactored Navigation.is_active for better clarity. Added cache clearing for previously-uncached, directly-hosted navigation. Adjusted recursenavigation to accept either a node or a queryset/iterable. Added targeting_navigation filter.
Added shortcut is_cached method to Navigation instances and Navigation.objects; updated other methods to use these shortcuts. Added cache clearing for new and changed Navigation instances that have not been cached but are within the range of a hosted navigation instance.
Corrected shpherd Navigation cache clearing and made grappelli attribute inline template compatible with grappelli r1399. Tweaked recursenavigation template tag behavior to set {{ navigation }} in the context instead of node and corrected its documentation. Added has_navigation filter.
...

Revision 38c44398 - 03/31/2011 03:06 PM - Stephen Burrows

Resolves issue 68 by treating any incorrectly-passed-in values for JSON fields as json strings rather than as python objects.

Revision a917b5dd - 04/06/2011 10:15 PM - Joseph Spiros

Merge branch 'master' of git://github.com/melinath/philo

- 'master' of git://github.com/melinath/philo: (31 commits)

Improved TemplateValidator error reporting to include origins and line numbers in TemplateSyntaxError reports.

Resolves issue 68 by treating any incorrectly-passed-in values for JSON fields as json strings rather than as python objects.

Resolved issue #120 by removing detection of containers which would not be rendered.

Made uuid of event autogenerated. Fully resolves issue #119. Also had to reset julian migrations again.

Julian improvements: Made calendar events optional and made calendar feed uuids auto-generated. Reset the migrations because south wasn't cooperating.

Adjusted 14082 hack again... "fixed" the ModelFormMetaclass.__new__ method and wrote EntityFormMetaclass as if there was no issue.

Moved the EventQuerySet to its own class instead of nesting it in the Event model.

Removed python 2.4 compatibility workaround use, since this is removed in Django 1.3 and philo doesn't support python 2.4 anyway.

Resolves issue #71 for the admin interface by working around the use of modelform_factory. If any code uses modelform_factory elsewhere with an entity form, the same issue will arise. Also compatible with 1.2.X.

Improved proxy-field hiding to be 1.3-proof. Unfortunately, requires a attribute on the proxy admin, but it seems like the best way. Still backwards-compatible with 1.2.X.

Removed shipherd navigation_host filter exception reraising. Tweaked LazyNode - turns out the / needs to be prepended **before** trying to check if the node handles_subpath.

Added some convenient methods to event querysets.

Switched Template.__unicode__ to only return self.name for legibility.

Corrected node_view redirection of trailing slashes to non-trailing slashes to rely on request._cached_node_path instead of request.path.

Tweaked LazyNode to only return a found node if it handles the found subpath.

Added feed_length to BlogViewAdmin.

Solidified AJAX API.

Prettified results page. Added search results templates for grappelli and the normal admin.

Refactored weight code to split the work over Search, ResultURL, and Click models. This is probably somewhat less efficient, but it makes more intuitive sense. It also allows for weight caching on instances. Initial work on a results action/view for the SearchAdmin. Set SearchView to have SearchForm set as an attribute on itself rather than blindly using it.

First attempts at a get_favored_results method to find what people are generally selecting. Some minor aesthetic changes. Changed ajax api template fetch to call get_template with the intent of passing an argument as to whether it should be prepped for ajax.

Updated CalendarView urlpatterns to use the new feed_patterns method.

...

History

#1 - 01/06/2011 10:06 AM - Joseph Spiros

- Assignee set to Stephen Burrows

This bug is either related to the use of the name "value" for the JSONField on the JSONValue model, or the implementation of JSONField.

#2 - 01/06/2011 07:06 PM - Joseph Spiros

- Subject changed from JSONValue objects cannot be imported with loaddata to JSONValue objects cannot be imported from fixtures

#3 - 01/14/2011 08:13 PM - Stephen Burrows

- Category set to Core

- Status changed from New to Resolved

It turns out the issue was that the serializer tries to instantiate JSONValue with a value=<python object> keyword argument. This can be done for example with ForeignKeys, but only because of specialized code in Model.__init__. However, the "value" kwarg can be swapped out with a "value_json" kwarg in the pre_init signalling phase. resolved with commit 4d7f6c42629cb85f001306195c6098b4abec642a.

#4 - 02/01/2011 02:54 PM - Stephen Burrows

- Status changed from Resolved to In Progress

- Priority changed from High to Normal

It seems there is still a slight problem with this: namely, that null values are incorrectly imported as the string ' "null" '

#5 - 03/31/2011 03:08 PM - Stephen Burrows

- Status changed from In Progress to Resolved

Resolved with commit 38c4439877b4551c9197b848b8b4f98b215f659b - for real this time ;-)

Had to give up on interpreting kwargs two ways, though. Alas! This should be revisited if serialization is improved.