

Philo - Bug #14

Template errors prevent page saving and opening

08/09/2010 03:24 PM - Stephen Burrows

Status:	Resolved	Start date:	08/09/2010
Priority:	Normal	Due date:	
Assignee:	Stephen Burrows	% Done:	100%
Category:	Core	Estimated time:	0.00 hour
Target version:			
Description			
If there is an error in a philo template, then when the page using that template checks for containers - for example when it's saved or when the admin interface tries to open it for changing - the error will be raised and block whatever action should have been taken.			
Preferred behavior:			
a) If there is a template error, treat it as "no containers" in the admin.			
b) Additionally, add a message - use the message framework? or is there a different way?			

Associated revisions

Revision 92e6313a - 08/17/2010 04:16 PM - Stephen Burrows

Added template code validation to the admin. This essentially addresses bug #14 by catching errors with validation on templates instead of letting it show up as a server error later. (Note that there should still be better validation on Pages, as deletion of a template say on the filesystem could render a db-driven template invalid.)

Revision e06925fc - 08/20/2010 09:21 PM - Joseph Spiros

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

- 'master' of git://github.com/melinath/philo:
 - Re-tweaked page add form to preserve 'continue on normal save' functionality. This is implemented with javascript and will probably break some time in the future, but there doesn't seem to be a better way to do it at the moment. Fixes bug #8.
 - Tweaked page add_form template. Fixes bug #8.
 - Moved templateform to forms.py in preparation for experimentation with page forms.
 - Added collapse classes to attribute and relationship inlines on entities.
 - Added template code validation to the admin. This essentially addresses bug #14 by catching errors with validation on templates instead of letting it show up as a server error later. (Note that there should still be better validation on Pages, as deletion of a template say on the filesystem could render a db-driven template invalid.)
 - Set TreeModel slug field length to 255 to match Titled

Revision 0133fd1a - 10/19/2010 02:54 PM - Stephen Burrows

Added model validation to Pages; for a page to be valid, its related template must also be valid. Ordinarily, the template would need to be valid in order to exist; however, this should help catch cases where, for example, changes on a filesystem template affect the validity of a database template. Addresses the remaining issues with bug #14.

Revision 9592f8df - 10/21/2010 07:59 PM - Joseph Spiros

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

- 'master' of git://github.com/melinath/philo:
 - Added support for 'editable' to ProxyFields. Tweaked AttributeFieldDescriptor base class to not raise an AttributeError on access. This allows the descriptor to be detected by Django admin validation, allowing the proxy fields to be used (for example) in a ModelAdmin's readonly_fields and list_display fields. This may address the remaining issues in feature #16.
 - Tweaked philo migrations to force contenttype creation after migration 0005 so the contenttypes are available in migration 0006. Ordinarily contenttypes are only synced after a complete set of migrations. Also implemented default values on proxy fields; partially addresses feature #16.
 - Added unittest for node_url templatetag. Fixed issues uncovered with unittests. Added test fixture. Seems bugfree - as such, addresses feature #12.
 - Initial node_url_refactor commit. Revises node_url templatetag to accept args/kwargs like urlnode, using the current or given node's view as the urlconf. with <obj> is retained as a convenient shortcut.
 - Initial node_middlewares commit. Implements an attribute on the HttpRequest class which handles lazy node evaluation - i.e. request.node is available for all requests, but is only evaluated on first use. Removed all references I could find to node, path, and subpath in Views, which may make for a simpler API. Path and subpath were generally unused and can be extrapolated in other ways; however, should it come to light that they are necessary, they can be added back in.
 - Added model validation to Pages; for a page to be valid, its related template must also be valid. Ordinarily, the template would need to be valid in order to exist; however, this should help catch cases where, for example, changes on a filesystem template affect the validity of a database template. Addresses the remaining issues with bug #14.

Genericized penfield's migrations to allow for variations in the PHILO_PERSON_MODULE setting by adding a central frozen reference to the person module appropriate for use in a South Migration.

Tweaked QuerySetMapper to return the value's value, in line with the changes to flexible attributes. Adjusted attribute fields to compensate. Shunted the responsibility in Entity proxy models for setting an Attribute's value onto the field that manages the model's relationship to that attribute.

Added migrations to handle transfer of structure and information to and from flexible attributes.

Added AttributeInlineFormSet to force new Attribute instances to be run through the form's save method for extra processing.

Added generic relation from attribute values back to Attribute.

Added migrations from ithinksw/master to melinath/master

Added search field to tags; added db_index=True to newsletter article full_text to expediate searching.

Polished AttributeForm to better handle changes to an attribute instance. Introduced apply_data to AttributeValue models to further allow customization. Made an AttributeValue class which must be subclassed to create new attribute values.

Added AttributeForm to elegantly handle all the fields that should be displayed in the admin related to the instance's value. Tweaked the tabular_collapse template (which was only used for attributes) to display the extra fields. Added the "value_formfield" method to Attribute Value models to allow for the possible future addition of value types.

Initial flexible attributes commit. Removed Relationship model in favor of a flexible Attribute model with generic foreign keys to an entity and a value type. This should allow for easy addition of attribute types: for example, an attribute with a ManyToMany value.

Created JSON field, descriptor, and formfield to handle json storage on models uniformly. Addresses feature #26.

History

#1 - 08/09/2010 03:52 PM - Stephen Burrows

- Subject changed from *Template errors prevent page saving* to *Template errors prevent page saving and opening*

#2 - 08/17/2010 04:48 PM - Stephen Burrows

- Status changed from *New* to *In Progress*

- Assignee set to *Stephen Burrows*

- % Done changed from *0* to *50*

#3 - 08/17/2010 04:49 PM - Stephen Burrows

(50% - see commit 92e6313ae3fbcf8f7c381aa91d53f2e32714f2c3)

#4 - 10/01/2010 02:38 PM - Stephen Burrows

- Status changed from *In Progress* to *Feedback*

- Assignee changed from *Stephen Burrows* to *Joseph Spiros*

I think what I was getting at was that Pages should make sure the db template is valid before accepting it... but that might be overkill. Even as it stands, there's no good way to validate filesystem templates that are included by database templates, and if a filesystem template was rendered invalid after the db template which extends it was already tied to the page in question... boy, that was long... then any efforts we may have tried to make would be in vain.

#5 - 10/19/2010 03:03 PM - Stephen Burrows

- Category set to *Core*

- Status changed from *Feedback* to *Resolved*

- Assignee changed from *Joseph Spiros* to *Stephen Burrows*

Resolved with commit 0133fd1af27b193ce3cb39073e67086561b787c3

#6 - 10/19/2010 03:03 PM - Stephen Burrows

- % Done changed from *50* to *100*