

Philo - Feature #70

Multiple navigations on one page

01/14/2011 05:21 PM - Stephen Burrows

Status:	Resolved	Start date:	01/14/2011
Priority:	Normal	Due date:	
Assignee:	Stephen Burrows	% Done:	0%
Category:	Shipherd	Estimated time:	0.00 hour
Target version:			
Description			
<p>Shipherd currently has a foreignkey from navigation to the hosting node. I'm not sure this is the best way to handle things. For example, I may want to have a main site navigation, a sidebar navigation, and a section navigation all on the same page. This isn't possible without major hacks in the template right now. Ideally, I'd want to be able to give each navigation section a different access key - i.e. node.navigation['sidebar'] vs node.navigation['main'].</p> <p>Ways to handle this:</p> <ol style="list-style-type: none">1. Attributes2. Collections (+ attributes or a simpler connective model)3. interpret the first layer of navigation as a connective model to the actual navigation. <p>Attributes would give the most flexibility, and allow for more nuanced inheritance of individual branches; however, there wouldn't be any way to cache navigation, so a lot of queries would be added. Also, attributes are meant primarily for tacked-on data, to my understanding, whereas navigation, if used, would be fairly standard for a node to contain.</p> <p>Option 2 might work well, but it would probably make more sense to just make a model that had a foreignkey to nodes and a manytomany with navigation... or that navigation had a foreignkey to.</p> <p>Option 3 would be fine, except that it would be confusing.</p> <p>Perhaps change "Navigation" to "NavigationItem" and add a Navigation model that connects Nodes to NavigationItems?</p>			

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 shipherd 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.

...

History

#1 - 01/18/2011 10:39 PM - Stephen Burrows

- *Status changed from New to Resolved*

Addressed by commit d807884e8aa5b94a79e5f130fd2ce711e287eb3a.