Philo - Bug #160

Shipherd shouldn't require 'request' in the context for {% recursenavigation %}

06/15/2011 03:50 PM - Stephen Burrows

Status:	New		**Start date:**	06/15/2011
Priority:	Normal		**Due date:**	
Assignee:			**% Done:**	0%
Category:	Shipherd		**Estimated time:**	0.00 hour
Target version:				

Description

Right now, if 'request' isn't in the context (i.e. if the request context processor isn't installed) shipherd's {% recursenavigation } templatetag will just not work - it'll always render as an empty string. The reasoning behind this was that there's no way to tell if a navigation item is active if the request is not provided. Really, though, { recursenavigation %} could still work without a request, just always returning False for "active". This behavior would make more sense, I think.

Associated revisions

Revision 1cd3ce0a - 06/15/2011 09:50 PM - Stephen Burrows

Added information about request context processor to shipherd docs. See issue #160.

Revision 2a3bb0d2 - 06/22/2011 05:07 PM - Joseph Spiros

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

- 'develop' of git://github.com/melinath/philo:
  Added winer doc file. Improved the docstrings for FeedView from the penfield version.
  Renamed syndication to winer.
  Added information about request context processor to shipherd docs. See issue #160.
  Minor correction to shipherd howto.
  Moved a migration after the master migrations.
  Tweaked FeedView to more correctly generate links vs feed urls.
  Improved penfield docstrings. Added urlpatterns and support for all registered feed types. Resolves issue #141.
  Initial implementation of a separate syndication contrib app based on penfield's FeedView, updated to use a registry instead of a dict.
  Moved sobol's search registry into core as a generic registry utility. Added support for RegistryIterators directly to SlugMultipleChoiceField since it is a common use case.
  Made views optional on nodes; nodes without views will raise a 404 when rendering and will return False for accepting and handling subpaths.
  Implements feature #131.