

Philo - Bug #64

Rendering template fields within templates

11/16/2010 03:07 PM - Stephen Burrows

Status:	Resolved	Start date:	11/16/2010
Priority:	Normal	Due date:	
Assignee:	Stephen Burrows	% Done:	0%
Category:	Penfield	Estimated time:	0.00 hour
Target version:			
Description			
Currently, NewsletterArticles contain TemplateFields - basically solely so that they can use image embedding. However, there's no way to interpret the field as a template in the context of rendering a template that the article appears in. A few possible solutions include: <ol style="list-style-type: none">1. provide some sort of render method on all models using template field by fiddling with contribute_to_class2. use a custom filter3. use a custom block tag so context can also be passed in.4. leave things as they are and let people either hack template parsing into views or define rendering methods on proxy models.			

History

#1 - 11/16/2010 03:53 PM - Stephen Burrows

- Status changed from New to Feedback

I really like the idea of providing a render method. The only catch is that this means there's no shared context between the template and the article - which means that any definition of which template to use for embedding should be inside the article - which defeats the whole "one article can embed different photos and they will be displayed appropriately according to the template" aspect of this feature. It might be better, then, to use a custom block tag.

#2 - 11/16/2010 09:00 PM - Joseph Spiros

A template tag or filter would be ideal. It would be *really* nice if we didn't need to use a tag, and if simply doing `{{ obj.template_field }}` somehow did the magic, but oh well. At any rate, whatever we use should be similar to using the `{% include %}` tag, meaning that context should be forwarded to the template field as it is rendered.

#3 - 11/17/2010 04:44 PM - Stephen Burrows

Wouldn't it be great if we could just use the include tag? i.e. `{% include article.full_text %}`

I've made a branch that implements this - it's actually quite simple and feels less complicated to me than a template tag. It just involves having template fields store their values as template strings that track their origin, and then adding a template loader that checks to see if the "template name" is actually a template.

Initial commit: `ba55bf97c1f1fb44ab2c9150d63c2b4256731bf7`

#4 - 11/18/2010 04:19 PM - Stephen Burrows

Template tag is also a simple enough solution. See commit `86afe7e80aa9d4ede8a7d309a97c068d8ef6a65a`.

#5 - 11/30/2010 04:49 PM - Stephen Burrows

- Status changed from Feedback to Resolved

Template tag solution chosen. Merged into master with commit `caafce88e26fef7f0cd5be751c1f0fa8bd0b5a06`. Could potentially be improved by allowing the syntax `{% include_string <string> [from <origin>] %}` for debugging purposes.