Philo - Bug #27

Handling form submission with Pages

10/19/2010 02:03 PM - Stephen Burrows

Status:	Feedback	**Start date:**	10/19/2010
Priority:	Normal	**Due date:**	
Assignee:	Stephen Burrows	**% Done:**	0%
Category:	Bartleby	**Estimated time:**	0.00 hour
Target version:			

Description

Currently, the only way to handle form submission for a certain page is to intercept the page's about_to_render signal and use the replace_sender_response function to swap in a new HttpResponse. However, this is inelegant, hackish, and will be difficult to maintain - for example, it generally relies on knowing something "immutable" about the page, such as its id or the path to a node that uses it. It would be more productive in the long run to have a clean standardized way to "register" Forms with a Page, then let the Page handle validation.

Possible solutions:
1. Bartleby forms. As these are defined in the database, it would be possible to have a relationship between them and a Page.
2. An "advanced" page field that holds a list of form locations to be validated (i.e. "django.contrib.auth.forms.UserCreationForm").
3. A template tag that defines forms which are used on the page. Similar to containers, would be picked up by the page.

With any of these, the form would need a way to define its validation - what to do if there was a post, what to do if not, what to do if it's valid.

History

#1 - 10/29/2010 09:25 PM - Stephen Burrows

I'm working on solution 1 with bartleby. It seems to be going well... I am considering adding "advanced" fields that can reference python path locations of functions to check if a form should be evaluated for a given request...
This currently works (as of commit 585c315fde9781b72e1bacd430877c1e5b110e90) with middleware which instantiates the forms and attaches them to the view's kwargs or, if there is valid post data, submits the forms and redirects back to the current page.

Although this solution is working well, I am skeptical of its viability for hard-core form handling. Bartleby forms are wired to save their results as result rows. That's the whole point: they are database-driven, not hard-coded. A contact/feedback form could be written in bartleby. So could a comment form. But not in the sense of django.contrib.

Should there be a separate, core way to handle python forms? Should bartleby forms be more empowered?

#2 - 11/02/2010 02:54 PM - Stephen Burrows

I think Bartleby forms should support the following options:
1. Record User / IP Address / Nothing
2. Require login i.e. require user.
3. Allow changes to your own previous (most recent?) submission.
4. Max number of submissions per user / IP - enforced preferentially through #1, or by cookies otherwise.

Bartleby forms could support the following features, but it would be more complicated.
5. Token-based row requesting... i.e. select a user. A row is created and a link generated for that user. The link is sent to that user. This would possibly require selecting a node which can handle it. If users are not being recorded, the user would be set to None on submission, invalidating the token. No... it would need to give you a link to a Page that had the form in an attribute. the token would then somehow need to be passed to the form with the user id, row id, and timestamp - if relevant.
6. Timeout: optionally set a date after which the form cannot be accessed? Or only allow in conjunction w/ #6.... or, if #6 is selected, it's a timeout on the tokens, otherwise a timeout on the form in general.

#3 - 11/05/2010 11:14 PM - Stephen Burrows

It might be useful to handle forms via signals rather than in the save() method directly. Then people could make custom listeners, use attributes for more specific filtering, and do anything - even use the data to create actual models instead of result rows!

#4 - 01/31/2011 06:27 PM - Stephen Burrows

- Category changed from Core to Bartleby

- Assignee set to Stephen Burrows

Could potentially use or model after django.contrib.formtools.FormWizard for multi-page handling. Perhaps make a more differentiated Issue for this?