

Tapestry - Story #2552

Configurable mini-consent form and validations

04/08/2014 12:13 PM - Phil Hodgson

Status:	New	Start date:	04/08/2014
Priority:	Normal	Due date:	
Assigned To:	Phil Hodgson	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	Testing and Upgrading + PGP.ca features		
Description			

History

#1 - 06/10/2014 10:22 AM - Phil Hodgson

These comments/questions from [#2553](#) are more pertinent to this issue:

<https://arvados.org/issues/2553#note-2>

<https://arvados.org/issues/2553#note-3>

Anyway I'll repaste them here...

#2 - 06/10/2014 10:23 AM - Phil Hodgson

(Pasted from comments from [#2553](#))

I decided just to first get rid of the internal references, column names and variable names, favouring the name `citizen_or_resident` over `us_citizen_or_resident`. It was painless refactoring and seemed a good first step. Then I got to the part where the validations themselves become configurable. I got stopped up there because the validation is done in a two-step process (first save the answers then go to the next enrollment step which is reviewing the answers). So my initial instinct to use the new `SiteSpecific::Validations` override mechanism couldn't work without some modifications.

I see a few choices here:

1. Leave it as is and let different sites use a fork of the code and actually modify the model itself.
2. Develop some kind of mechanism akin to the `SiteSpecific::Validations` business but which overrides model instance methods such as `ScreeningSurveyResponse#passed?` so that site-specific changes remain outside of the main source code tree as much as possible. I am wary of this. It would be powerful, but perhaps too subject to abuse. The `SiteSpecific::Validations` is more contained in how it is meant to work, which is why we feel as if it is safer to unleash on the site programmers.
3. Change the way `ScreeningSurveyResponse` is validated so that it can use a real ActiveRecord model validation that can be overridden with the existing `SiteSpecific::Validations` mechanism. I might suggest adding a new column to `ScreeningSurveyResponse` called `passed`, and the way to find out if something has passed is to check if that column is set to true. If it is not then the way to test is a screening survey response has passed would be to attempt to set the `passed` column to true, triggering a validation failure if it cannot be set to true because of the conditions that are considered to be a "pass". This would fit into our existing framework.

I also came across some code about being able to export lists of people who failed the screening survey based on a previous version of the survey, or something. It uses an alternate scope in `ScreeningSurveyResponse` called `failed` but which isn't just the opposite of the `passed` scope but rather a whole string of reversed scopes. It's weird and really overcomplicates trying to customize what "passing" means, mostly because it's so unDRY...

#3 - 06/10/2014 10:23 AM - Phil Hodgson

(Pasted from comments from [#2553](#))

There's also, though I don't know if I should suggest it, the idea of not having a "review" step in the enrollment process and doing a more straightforward validation of the responses, moving all of the various enrollment steps towards a kind of basic, customizable form with a standard kind of validation...

On the other hand, we may want that standard kind of validation to have the ability to be part of a two-step process, because other stages of enrollment use a "review" step too... I guess I'd push for something like solution #3 above then.

Either way, it seems to me that every conceivable enrollment step *should* work with the same set of rules and mechanisms. These should be flexible enough at least to handle the way all of the current enrollment steps work. The idea would be to have *one* controller in total rather than *one controller for every step*, and needing to add a new controller for every new, custom step. I see this as less powerful but easier for different sites to deal with because it is more formalized.

#4 - 06/10/2014 10:27 AM - Phil Hodgson

- Parent task deleted (#2549)

#5 - 06/10/2014 10:28 AM - Phil Hodgson

- Tracker changed from Task to Story