

w.c.s. - Development #71296

Poser un début d'infrastructure de test de formulaire

14 novembre 2022 17:35 - Valentin Deniaud

Statut:	Fermé	Début:	14 novembre 2022
Priorité:	Normal	Echéance:	
Assigné à:	Valentin Deniaud	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Le grand plan est de pouvoir avoir une suite de tests par formulaire, qui serait jouée automatiquement très régulièrement pour avertir l'administrateur fonctionnel au moment où une modif fait planter un test.

L'idée est de ne pas discuter du grand plan ici, mais d'avancer à partir d'un cas simple et concret :

- Un formulaire, un champ `form_var_texte`, une condition de sortie de page `{{ form_var_texte 'a' }}`
- Création d'un test qui vérifie que pour `form_var_texte` valant 'a' on peut bien passer à la page suivante
- Modification de la condition `{{ form_var_texte 'b' }}`
- Le test ne passe plus et on voit une alerte quelque part

L'objectif de ce ticket est d'avoir ça qui fonctionne, ça devrait permettre ensuite d'ajouter ce qu'il faut pour tester les champs/pages conditionnels, les sauts dans les workflows, et d'ouvrir des tickets au fur et à mesure pour ajouter ce qu'il manque.

Révisions associées

Révision a9f63c57 - 27 décembre 2022 10:31 - Valentin Deniaud

testdef: add new table to hold form tests (#71296)

Révision 3916f885 - 27 décembre 2022 10:31 - Valentin Deniaud

testdef: add views to run tests (#71296)

Révision a84b16f4 - 27 décembre 2022 10:31 - Valentin Deniaud

testdef: test field conditions (#71296)

Historique

#1 - 17 novembre 2022 15:59 - Valentin Deniaud

- Fichier `0002-testdata-add-views-to-run-tests-71296.patch` ajouté
- Fichier `0001-testdata-add-new-table-to-hold-form-tests-71296.patch` ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Total WIP mais voilà du code qui fonctionne pour :

- Ajouter un « test » à partir d'une demande déjà soumise
 - Ce qui est testé c'est simplement qu'on arrive à soumettre la demande
 - Ça ne prend en compte que conditions de sortie de page et (bonus par rapport à la description) champs conditionnels
- Jouer manuellement les tests
- Visualiser les résultats

Il y a plein de trucs niveau parcours/vues/stockage qui sont nulles pour le moment mais je suis surtout preneur de remarque sur la manière de jouer le déroulé de la demande : c'est un simple parcours des champs du formulaire avec remplissage au fur et à mesure d'un `formdata`. Est-ce que c'est une approche viable ou est-ce que ça va droit dans le mur ?

Un chemin tout à fait différent aurait été de vouloir faire au plus proche de la soumission réelle, en réutilisant `wcs.forms.root.FormPage`, mais ça m'a semblé très complexe car une bonne partie du code gère des histoires de `magictoken/brouillons`, avec des `get_request()/get_session()` à tous les coins de rue, et les méthodes renvoient du html.

#2 - 17 novembre 2022 16:11 - Frédéric Péters

Est-ce que c'est une approche viable

Yes c'est ok (même avis il faut éviter FormPage et cie).

#3 - 17 novembre 2022 16:16 - Valentin Deniaud

- Statut changé de *Solution proposée* à *En cours*

OK je continue avec ça, merci !

#4 - 29 novembre 2022 17:23 - Valentin Deniaud

- Fichier *0003-testdef-test-field-conditions-71296.patch* ajouté

- Fichier *0002-testdef-add-views-to-run-tests-71296.patch* ajouté

- Fichier *0001-testdef-add-new-table-to-hold-form-tests-71296.patch* ajouté

- Statut changé de *En cours* à *Solution proposée*

Voilà pour un socle possible, ce n'est pas très utilisable mais c'est déjà assez gros comme ça.

De la description du ticket j'ai laissé tomber la partie « une modif cassante déclenche une alerte », pour le moment les tests doivent être lancés manuellement.

Il n'y a ni vue d'édition ni vue de visualisation d'un test, mais ces deux actions sont possibles en exportant le test en json puis en important le json modifié.

Une question, est-ce que j'ajoute un index d'unicité sur (slug, object_type, object_id) à la table TestDef ?

Suite du plan/tickets à ouvrir ensuite :

- Limiter les demandes à partir desquelles on peut créer un test
 - Car si on permet ça à partir d'une vraie demande, adieu l'anonymisation
- Pouvoir créer un test « cas non passant »
 - Ça risque d'amener le travail sur vue de visualisation/d'édition
- Améliorer la reconstruction de l'objet FormData (actuellement on perd plein d'infos et certainement des choses utilisées dans des conditions)
- Tester plus de choses
 - Pages conditionnelles
 - Validation regex et compagnie
 - J'en trouverai d'autres au fur et à mesure

Et moins prioritaire pour le moment :

- Rapporter automatiquement les erreurs/les rendre plus visible, afterjob sûrement
- S'intéresser à comment on gère les sources de données
- Avoir un export xml des tests inclus dans l'export du formulaire
- Tester le workflow

#5 - 14 décembre 2022 14:43 - Valentin Deniaud

Valentin Deniaud a écrit :

Une question, est-ce que j'ajoute un index d'unicité sur (slug, object_type, object_id) à la table TestDef ?

Et en regardant pour ça je me rends compte que malgré une honorable volonté de créer une table dans 0001, elle n'est par la suite pas utilisée. Je découvre que j'ai raté un détour nécessaire par SqlMixin et compagnie.

Juste pour être sûr, on supporte toujours des déploiements sans base de donnée ? C'est à dire dans mon patch je dois remplacer tous les usages directs de la classe TestDef par des `get_publisher().testdef_class`, `testdef_class` étant sélectionnée dynamiquement en fonction de la présence ou non d'une base ? (en greppant un peu l'historique je n'ai pas réussi à savoir si ce mécanisme existait parce qu'il y a eu migration des objets sérialisés en base ou si il était également nécessaire dans le cas de l'introduction d'un nouvel objet qui lui existera direct en base)

#6 - 14 décembre 2022 14:52 - Frédéric Péters

Juste pour être sûr, on supporte toujours des déploiements sans base de donnée ?

Non.

C'est à dire dans mon patch je dois remplacer tous les usages directs de la classe TestDef par des get_publisher().testdef_class (...)

Tu peux faire sans cette indirection, tu peux regarder ce que j'ai fait pour le multilinguisme, j'ai wcs/i18n.py avec class TranslatableMessage(sql.TranslatableMessage): (et très peu de code dans ce cas précis), et le code spécifique SQL laissé dans sql.py.

#7 - 14 décembre 2022 18:09 - Valentin Deniaud

- Fichier 0003-testdef-test-field-conditions-71296.patch ajouté

- Fichier 0002-testdef-add-views-to-run-tests-71296.patch ajouté

- Fichier 0001-testdef-add-new-table-to-hold-form-tests-71296.patch ajouté

Merci pour le pointeur sur TranslatableMessage c'est ce qu'il me fallait, revoici avec 0001 corrigé et des petits ajustements qui ont ruisselés dans la suite.

#8 - 23 décembre 2022 09:49 - Frédéric Péters

Dans 0002,

```
raise TestError(_('Failed to evaluate page %s post condition.') % page_no)
```

Si je compte bien ça va affiche page 0 pour la première page et ainsi de suite.

(il y a aussi un page_no dans le 0003, peut-être concerné aussi.)

#9 - 23 décembre 2022 16:30 - Valentin Deniaud

page_no est incrémenté dès qu'on rentre sur une page, puis à la sortie la condition est évaluée et page_no vaut bien 1. Là où il y a un bug c'est bien dans 0003, si le formulaire n'a aucune page ça affiche 0 (tout ça est visible dans les tests).

Bon en relisant ce bout de code je l'ai trouvé pas très beau alors je l'ai retouché en virant l'incrément et en se basant sur la vraie position de la page, ça permettra de ne pas toucher le code à l'ajout du support des pages conditionnelles (sur la branche ça donne quelques modif dans 0002 et un 0 qui se transforme en 1 dans les tests de 0003).

#10 - 27 décembre 2022 10:32 - Frédéric Péters

- Statut changé de Solution proposée à Résolu (à déployer)

J'ai rebasé pour actualiser le numéro de migration + déplacer les TestDef.do_table() vers tests/utilities.py, que la table soit disponible créée pour tous les tests.

```
commit a84b16f42619d0fef34b502380fdd2def4898220
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Tue Nov 29 16:16:26 2022 +0100
```

```
testdef: test field conditions (#71296)
```

```
commit 3916f88586c18f0931f7e2d6bd1d298b19b9423a
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Wed Dec 14 16:38:10 2022 +0100
```

```
testdef: add views to run tests (#71296)
```

```
commit a9f63c57f5e37cad681b00a235548aec0defc706
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Nov 17 15:22:15 2022 +0100
```

```
testdef: add new table to hold form tests (#71296)
```

#11 - 27 décembre 2022 12:14 - Transition automatique

- Statut changé de Résolu (à déployer) à Solution déployée

#12 - 26 février 2023 04:42 - Transition automatique

Automatic expiration

Fichiers

0002-testdata-add-views-to-run-tests-71296.patch	11,8 ko	17 novembre 2022	Valentin Deniaud
0001-testdata-add-new-table-to-hold-form-tests-71296.patch	3,93 ko	17 novembre 2022	Valentin Deniaud
0003-testdef-test-field-conditions-71296.patch	2,73 ko	29 novembre 2022	Valentin Deniaud

0002-testdef-add-views-to-run-tests-71296.patch	26 ko	29 novembre 2022	Valentin Deniaud
0001-testdef-add-new-table-to-hold-form-tests-71296.patch	3,86 ko	29 novembre 2022	Valentin Deniaud
0003-testdef-test-field-conditions-71296.patch	2,73 ko	14 décembre 2022	Valentin Deniaud
0002-testdef-add-views-to-run-tests-71296.patch	26,3 ko	14 décembre 2022	Valentin Deniaud
0001-testdef-add-new-table-to-hold-form-tests-71296.patch	6,29 ko	14 décembre 2022	Valentin Deniaud