w.c.s. - Development #23413

amélioration du widget Liste à choix multiples pour permettre la personnalisation des templates

25 avril 2018 15:54 - Anonyme

Statut:	Rejeté	**Début:**	25 avril 2018
Priorité:	Normal	**Echéance:**	
Assigné à:	Frédéric Péters	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

refonte de CheckboxesWidget avec fichier template checkboxes.html surchargeable
tout en rendant disponible aux templates qui surchargeraient l'original d'accéder à des options en plus passées éventuellement à l'instanciation (par ex. données agenda)

Demandes liées:

Lié à w.c.s. - Development #23403: ajouter get_options pour CheckboxesWidget	**Rejeté**	**25 avril 2018**
Lié à Intégrations graphiques Publik - Development #23230: widget checkboxes-...	**Rejeté**	**17 avril 2018**

Historique

#1 - 25 avril 2018 15:56 - Anonyme

- Fichier 0001-create-template-for-CheckboxesWidget-23413.patch ajouté

- Patch proposed changé de Non à Oui

#2 - 25 avril 2018 15:57 - Anonyme

- Lié à Development #23403: ajouter get_options pour CheckboxesWidget ajouté

#3 - 25 avril 2018 16:09 - Frédéric Péters

C'est exactement mon code, là. (l'attribuer correctement me permettra de pester sur moi-même plus tard)

```
<input type="hidden" name="{{ w.name }}" value="{{ yes }}">
```

Ça n'a pas été suffisamment testé, {{ yes }} ça va pas le faire.

#4 - 25 avril 2018 16:10 - Anonyme

- Lié à Development #23403: ajouter get_options pour CheckboxesWidget supprimé

#5 - 25 avril 2018 18:01 - Anonyme

- Fichier 0001-create-template-for-CheckboxesWidget-23413.patch ajouté

corrigé, équivalent à la fonction render remplacée

#6 - 25 avril 2018 18:04 - Anonyme

- Fichier 0001-create-template-for-CheckboxesWidget-23413.patch supprimé

#7 - 25 avril 2018 18:14 - Anonyme

- Fichier 0001-create-template-for-CheckboxesWidget-23413.patch ajouté

#8 - 26 avril 2018 09:00 - Frédéric Péters

- Lié à Development #23403: ajouter get_options pour CheckboxesWidget ajouté

#9 - 26 avril 2018 09:04 - Frédéric Péters

La logique d'ajouter un template puis d'ajouter une méthode get_options dans un autre ticket mais ne pas l'utilisern alors qu'elle est ajoutée pour simplifier l'écriture de template, ne tenait pas.

Ici, le template est encore totalement basé sur la structure interne (via CompositeWidget), il me semble que le template devrait dépasser ça et exploiter une méthode get_options.

#10 - 26 avril 2018 10:40 - Anonyme

- Fichier 0001-created-Checkboxeswidget.get_options-and-tests-23413.patch ajouté

- Sujet changé de créer un template qommon/forms/checkboxes.html à créer un template qommon/forms/checkboxes.html et CheckboxesWidget.get_options

Frédéric Péters a écrit :

> La logique d'ajouter un template puis d'ajouter une méthode get_options dans un autre ticket mais ne pas l'utilisern alors qu'elle est ajoutée pour simplifier l'écriture de template, ne tenait pas.
>
> Ici, le template est encore totalement basé sur la structure interne (via CompositeWidget), il me semble que le template devrait dépasser ça et exploiter une méthode get_options.

OK, comme 1ère étape à rejoindre les deux tickets, je rétablis d'abord get_options et ses tests, sans pour l'instant remplacer CheckboxesWidget.render_content() par un template checkboxes.html (je terminerai ce ticket avec un 2e commit dans les prochains jours)

#11 - 26 avril 2018 11:10 - Frédéric Péters

Que ça soit clair :

```
 def test_checkboxes_widget():
     widget = CheckboxesWidget('test',
-            options=[('apple', 'Apple', 'apple'), ('pear', 'Pear', 'pear'), ('peach', 'Peach', 'peach')])
+            options=[('apple', 'Apple', 'apple', {'datetime': datetime.datetime.now()}), ('pear', 'Pear', 'pe
ar'), ('peach', 'Peach',
'peach')])
```

ça s'appelle modifier un test existant et j'ai précisé qu'il ne fallait pas faire ça.

#12 - 02 mai 2018 11:11 - Anonyme

- Fichier 0001-add-Checkboxeswidget.get_options-and-tests-23413.patch ajouté

Frédéric Péters a écrit :

> Que ça soit clair :
>
> [...]
>
> ça s'appelle modifier un test existant et j'ai précisé qu'il ne fallait pas faire ça.

ça s'appelle un oubli de ma part, c'est corrigé

#13 - 03 mai 2018 11:32 - Anonyme

- Fichier 0001-add-CheckboxesWidget.get_options-and-checkboxes.html.patch ajouté

#14 - 03 mai 2018 12:07 - Thomas Noël

Ton test fait le job pour vérifier le get_options, mais il faut aussi vérifier (un peu) que le rendu fonctionne, et donc ajouter le mock_form_submission comme dans le test test_checkboxes_widget.

#15 - 03 mai 2018 13:52 - Anonyme

Thomas Noël a écrit :

> Ton test fait le job pour vérifier le get_options, mais il faut aussi vérifier (un peu) que le rendu fonctionne, et donc ajouter le mock_form_submission comme dans le test test_checkboxes_widget.

Il y avait déjà un test pour ça, il est au-dessus test_checkboxes_widget

#16 - 03 mai 2018 14:05 - Thomas Noël

Oui mais pour vérifier que ça passe toujours y compris avec des options "{'datetime': test_dt}", non ?

#17 - 03 mai 2018 15:12 - Anonyme

- Fichier 0001-add-CheckboxesWidget.get_options-and-checkboxes.html.patch ajouté

#18 - 03 mai 2018 15:42 - Anonyme

- Lié à Development #23230: widget checkboxes-meetings.html ajouté

#19 - 04 mai 2018 10:58 - Thomas Noël

J'ai pas assez suivi l'affaire, mais pour moi, un get_options qui renvoie des widgets, ça va pas. C'est pas plutôt get_widgets qu'il fallait surcharger ?

#20 - 04 mai 2018 11:17 - Anonyme

Thomas Noël a écrit :

> J'ai pas assez suivi l'affaire, mais pour moi, un get_options qui renvoie des widgets, ça va pas. C'est pas plutôt get_widgets qu'il fallait surcharger ?

Le nom est peut-être mal donné: il s'agit d'un get_widgets où on ajoute l'attribut extra_options à chaque widget.
Peut-être ne pas surcharger get_widgets, mais sinon appeler la nouvelle méthode get_widget_with_options ?

#21 - 04 mai 2018 11:25 - Thomas Noël

Elias Showk a écrit :

> Thomas Noël a écrit :
>
> > J'ai pas assez suivi l'affaire, mais pour moi, un get_options qui renvoie des widgets, ça va pas. C'est pas plutôt get_widgets qu'il fallait surcharger ?
>
> Le nom est peut-être mal donné: il s'agit d'un get_widgets où on ajoute l'attribut extra_options à chaque widget.
> Peut-être ne pas surcharger get_widgets, mais sinon appeler la nouvelle méthode get_widget_with_options ?

Je serais tenté par voir ce que donne une surcharge de get_widgets.

Disclaimer : je relis en l'absence de Frédéric et j'avoue que je serais beaucoup plus à l'aise s'il lisait par dessus mon épaule... ça va être un peu chaud pour moi pour acker l'affaire :/ (qui semble si simple... et c'est justement ce qui me tracasse, je suis de nature inquiète)

#22 - 04 mai 2018 12:10 - Anonyme

- Fichier 0001-add-CheckboxesWidget.get_options-and-checkboxes.html.patch ajouté

Thomas Noël a écrit :

> Je serais tenté par voir ce que donne une surcharge de get_widgets.

Voilà renommé
Mon idée était qu'avec get_options, on ne touche pas à l'original get_widgets et on a une version spéciale de get_widgets pour construire

#23 - 04 mai 2018 14:17 - Thomas Noël

Allez, pour la forme et que les choses soient plus explicites, je préférerais que la fonction commence par un habituel widgets = super(CheckboxesWidget, self).get_widgets() qui "montre" la surcharge.

Voire même :

```
def get_widgets():
    widgets = super(CheckboxesWidget, self).get_widgets()
    if not self.options_with_attributes:
        return widgets
    for idx, widget in enumerate(super(CheckboxesWidget, self).get_widgets()):
        widget.extra_options = self.options_with_attributes[idx][-1]
        yield widget
```

Dans le test je me dis qu'on pourrait se passer du assert isinstance(widget.get_widgets(), collections.Iterator), parce que le get_widgets original renvoie une liste, et d'ailleurs ça m'ennuie de changer cela, on mesure pas forcément l'usage de get_widgets par ailleurs, donc :

```
def get_widgets():
    widgets = super(CheckboxesWidget, self).get_widgets()
    if not self.options_with_attributes:
        return widgets
    widgets_with_extra_options = []
```

```
    for idx, widget in enumerate(super(CheckboxesWidget, self).get_widgets()):
        widget.extra_options = self.options_with_attributes[idx][-1]
        widgets_with_extra_options.append(widget)
    return widgets_with_extra_options
```

Mais encore une fois je suis pas top à l'aise, on n'a jamais remplacé get_widgets...

Et à relire la remarque de Fred «le template est encore totalement basé sur la structure interne (via CompositeWidget), il me semble que le template devrait dépasser ça et exploiter une méthode get_options», je me dis que oui, ça serait plus cohérent avec ce qu'on fait par ailleurs. Renvoyer la liste des options et dans le template l'utilise pour construire la liste des inputs.

Bref : pas ack :/

#24 - 04 mai 2018 15:33 - Anonyme (Notes privées)

Une discussion suite à ça allait dans le sens de la réécriture de get_options() pour se rapprocher de SingleSelectHintWidget.get_options qui renvoit une liste de dict et non pas un "itérable" d'instances de Widgets avec extra_options. J'ai expliqué pourquoi je n'avais pas fait ça, mais n'avais pas la vue globale proposée par Thomas en fin de discussion.

On s'oriente vers le retour à get_options (avant dernier patch) et un itérateur avec des dict pour compiler le template et avoir accès à des données extras dans options. (sans modifier les instances de wigdet avec l'attribut extra_options)

#25 - 07 mai 2018 19:32 - Anonyme

- Fichier 0001-add-CheckboxesWidget.get_options-and-checkboxes.html.patch ajouté

Voilà une réécriture qui reprend l'esprit du get_options existant par ailleurs, sans toucher à get_widgets(), les tests sur jenkins passent pour la branche wip/checkboxes.
Le code de checkboxes--meeting.html est aussi prêt dans le ticket lié.
J'ai ajouté un nouvel assert pour vérifier la présence d'une donnée unique "any_info" passée dans un des 3 widgets de test_checkboxes_get_options

#26 - 07 mai 2018 20:00 - Frédéric Péters

(la docstring parle de yield de widget alors que ce n'est pas le cas) (le message de commit devrait plutôt être de l'ordre de "forms: use a template to render checkboxes widgets (#...)"

```
        {% if widget.readonly or "disabled" in option %}
        <input type="hidden" name="{{option.name}}" value="yes" >
        {% endif %}
        <input type="checkbox" value="yes" name="{{option.name}}"
```

Ça sonne curieux d'avoir une valeur "yes" sur les options désactivées. Et d'avoir deux champs avec le même attribut name.

Sur la forme, plutôt option.disabled (comme c'est le cas pour option.checked).

#27 - 07 mai 2018 20:10 - Anonyme

- Fichier 0001-forms-use-a-template-to-render-checkboxes-widgets-23.patch ajouté

Frédéric Péters a écrit :

> (la docstring parle de yield de widget alors que ce n'est pas le cas) (le message de commit devrait plutôt être de l'ordre de "forms: use a template to render checkboxes widgets (#...)"

Corrigé dans cette version du patch

> Ça sonne curieux d'avoir une valeur "yes" sur les options désactivées. Et d'avoir deux champs avec le même attribut name.
>
> Sur la forme, plutôt option.disabled (comme c'est le cas pour option.checked).

J'aurais aussi aimé remplacer ce "yes" et ce options.disabled, j'ai ramé la semaine dernière pour harmoniser ça, mais finalement ça cassait des choses ailleurs qui sont en dehors de ma compréhension, et j'ai préféré respecter ce que faisait l'ancien render_content() pour limiter les modifications faites dans ce ticket à la logique quixote/wcs.

Je propose qu'on me passe ces bizarreries qui étaient déjà dans le code qui a été remplacé.

#28 - 07 mai 2018 20:15 - Frédéric Péters

> Je propose qu'on me passe ces bizarreries qui étaient déjà dans le code qui a été remplacé.

Dans le code précédent il y avait justement de quoi faire en sorte que le même attribut ne soit pas utilisé (l'ajout du "xx").

#29 - 07 mai 2018 20:34 - Anonyme

- Fichier Capture d'écran de 2018-05-07 20-28-50.png ajouté

- Fichier 0001-forms-use-a-template-to-render-checkboxes-widgets-23.patch ajouté

Frédéric Péters a écrit :

> > Je propose qu'on me passe ces bizarreries qui étaient déjà dans le code qui a été remplacé.
>
> Dans le code précédent il y avait justement de quoi faire en sorte que le même attribut ne soit pas utilisé (l'ajout du "xx").

Oui donc rétablissement de xx
Et un nouveau screenshot pour illustrer que la version read-only est valide.

#30 - 07 mai 2018 20:38 - Frédéric Péters

> Oui donc rétablissement de xx

Non :/ Le xx était un hack de contournement nécessaire par l'organisation précédente, ici, comme on a un contrôle total sur le template, on devrait simplement ne pas avoir d'attribut name.

Je ne comprends toujours pas pourquoi on force une valeur "yes" en cas de {% if widget.readonly or "disabled" in option %}.

#31 - 09 mai 2018 11:48 - Anonyme

Frédéric Péters a écrit :

> > Oui donc rétablissement de xx
>
> Non :/ Le xx était un hack de contournement nécessaire par l'organisation précédente, ici, comme on a un contrôle total sur le template, on devrait simplement ne pas avoir d'attribut name.
>
> Je ne comprends toujours pas pourquoi on force une valeur "yes" en cas de {% if widget.readonly or "disabled" in option %}.

Pour finaliser ce ticket : est-ce que tu pourrais s'il-te-plaît regrouper les besoins pour éviter quelques aller-retours ? (si possible de préciser ce qui est de l'ordre du souhaitable et de l'ordre du nécessaire).

Je reprendrai un par un les points, je vérifierai le fonctionnement et les tests (certains trucs comme "yes", je ne suis pas sûr que ça passe par exemple), et je mettre un patch regroupant tout ça.

#32 - 09 mai 2018 12:52 - Frédéric Péters

Ce n'est pas évident de faire une liste de toutes les erreurs à ne pas commettre.

L'objectif de ce ticket (je note l'erreur de faire un ticket dont l'intitulé est une description technique trop précise) c'est de remplacer le rendu qui est aujourd'hui dans render_content, pour passer en toute situation par une template. Dans la réalisation de cet objectif, il s'agit pour le nouveau template de fournir un fonctionnement correct (bien sûr) mais également d'être et clair et compréhensible, avec en tête les besoins alternatifs (l'utilisation pour la multi-sélection de plages horaires) pour que ceux-ci puissent également être réalisés de manière claire et compréhensible.

Un exemple à ce niveau, dans le template proposé en #23230, {% ifnotequal attr.0 'options_with_attributes' }{{attr.0}}="{{attr.1}}"{ endifnotequal %} me laisse à penser qu'il y a un truc trop peu carré ici.

Clair et compréhensible, ça reprend aussi le besoin d'éviter l'inutile, l'inutilisé, d'une exécution des tests, je n'ai rien trouvé qui bénéficie de ça :

```
{% for key, value in option.attrs.items }{{key}}="{{value}}"{ endfor %}
```

Un autre point d'attention c'est de ne pas (trop) diverger du balisage actuel car des thèmes en dépendent sans doute. (et si divergence il y a il est utile de s'assurer du rendu, ou de noter qu'il y avait précédemment un bug, comme c'est le cas sur l'ajout des aria-required).

D'un point de vue pratique, il me semble que l'erreur originelle est dans for idx, widget in enumerate(self.get_widgets()):, qui crée une liste d'options à partir d'une liste de widgets, ceux-ci ayant été créés un peu plus tôt à partir d'une liste d'options.

Dérouler à partir de là, on peut se pencher sur le premier passage d'options à widgets, y trouver que faire un self.add(CheckboxWidget...) pour les champs désactivés n'est plus nécessaire une fois qu'on assure le rendu directement, etc.

Mais faire ce déroulé pratique, expliciter ce qui rendrait les choses plus claires et plus compréhensibles, ça me demande pour ainsi d'écrire le patch.

Peut-être qu'il faut rétropédaler ici, revenir à un premier besoin, qui était de fournir le nécessaire pour permettre l'écriture de la version -meetings.html, et l'œil alors il serait à poser sur la réexploitation future de ces ajouts, au moment où il y aura passage par défaut par un template. (mêler les deux permet de ne pas prendre le risque d'avoir un système marchant dans un cas et l'autre).

#33 - 09 mai 2018 16:07 - Anonyme

Frédéric Péters a écrit :

> Ce n'est pas évident de faire une liste de toutes les erreurs à ne pas commettre.

Ce n'est pas ce que je voulais dire, je parle des besoins de ce ticket, et tu les résumes très bien en dessous, merci, c'est plus clair pour moi.
OK, je vais m'efforcer de nettoyer ça et le rendre plus clair.
Mais sans vouloir insister, j'explique tout de même un peu sur les choix faits jusque là ce-dessous.

> Clair et compréhensible, ça reprend aussi le besoin d'éviter l'inutile, l'inutilisé, d'une exécution des tests, je n'ai rien trouvé qui bénéficie de ça :

Le problème c'est que pour je ne peux pas vraiment savoir ce qui est inutile et inutilisé dans W.C.S. Les tests sont-ils un reflet de l'utilisation ? Une fois l'HTML injecté, qu'en fait-on ? C'est mon premier patch W.C.S. et je ne m'attends pas à tout comprendre d'un coup.

J'évite de m'aventurer dans du nettoyage de code au risque de provoquer des effets de bords (en règle générale depuis des erreurs commises dans d'autres tickets par le passé). Mon niveau de connaissance globale me fait en général me limiter à garder au maximum l'existant, en tout cas je croyais que c'était la démarche préférée et qui faisait consensus parmi nous.

> D'un point de vue pratique, il me semble que l'erreur originelle est dans for idx, widget in enumerate(self.get_widgets()):, qui crée une liste d'options à partir d'une liste de widgets, ceux-ci ayant été créés un peu plus tôt à partir d'une liste d'options.

Idem, je pensais que cette classe se reposait plus logiquement sur son get_widgets et pas sur le dict de liste des options envoyé à l'__init__.
Mes interrogations devant ce code quand je le reprends : pourquoi cet *init* transforme ces options en widgets ? Ça doit vouloir dire que ce self.add() est nécessaire au bon fonctionnement de la machinerie ? Qui me dit si le self.add(CheckboxWidget...) n'est pas nécessaire au bon fonctionnement de quixote/w.c.s./une autre composant dont je ne connais peut-être pas l'existence ? Que sais-je de l'isolation des composants ? Je teste parfois quelques modifications mineures, sans commiter, et quand je vois tout exploser, alors je reviens en arrière et je la joue conservateur.

> Dérouler à partir de là, on peut se pencher sur le premier passage d'options à widgets, y trouver que faire un self.add(CheckboxWidget...) pour les champs désactivés n'est plus nécessaire une fois qu'on assure le rendu directement, etc.

Je n'ai pas touché à *init* pour limiter le champ de mes modifications. Je pensais que c'était la démarche à suivre dans le travail collectif.

> Mais faire ce déroulé pratique, expliciter ce qui rendrait les choses plus claires et plus compréhensibles, ça me demande pour ainsi d'écrire le patch.
> Peut-être qu'il faut rétropédaler ici, revenir à un premier besoin, qui était de fournir le nécessaire pour permettre l'écriture de la version -meetings.html, et l'œil alors il serait à poser sur la réexploitation future de ces ajouts, au moment où il y aura passage par défaut par un template. (mêler les deux permet de ne pas prendre le risque d'avoir un système marchant dans un cas et l'autre).

Je ne comprends pas ce dernier paragraphe. Peux-tu préciser ?

L'idée de début était bien de fournir des options en plus à des templates optionnels comme checkboxes__meetings.html. En ajoutant les données, on m'a demandé de transformer le render_content en template, tout en ajoutant les options utiles à des templates optionnel qui surchargent le rendu de base. Et maintenant, je ne sais plus ce qu'on attend de moi... et j'aimerais quand même pouvoir tenter de passer de ticket en suivant tes explications, c'est un peu plus clair pour moi maintenant.

#34 - 09 mai 2018 16:45 - Frédéric Péters

> Les tests sont-ils un reflet de l'utilisation ?

Un maximum, et quand ça n'est pas assez, on augmente. L'idée est bien de pouvoir être très sûr de soi quand tous les tests passent.

> Mes interrogations devant ce code quand je le reprends [...]

Le CompositeWidget (quixote) réunit pour un champ différents widgets, donc en gros des widgets sont ajoutés dans le *init*, et pour les deux grandes actions (rendu et inteprétation de réponse), ça fait une boucle sur ceux-ci.

On pourrait garder le même fonctionnement mais il y a quelques trucs moches/incorrects qui pourraient être nettoyés en squeezant la partie rendu, en la prenant en charge totalement. (je pense à l'ajout du xx très moche, ou à la présence de l'aria-required erroné); on pourrait se dire que c'est ok dans une première étape, qu'ensuite ça pourra être nettoyé mais je pense que c'est faux, que faire ainsi ajoute de la complexité générale, qui sera

d'autant plus difficile à défaire par la suite.

> Je n'ai pas touché à init pour limiter le champ de mes modifications. Je pensais que c'était la démarche à suivre dans le travail collectif.

Je ne vais pas faire règle de ça, et c'est si subjectif, mais une vue de la démarche pourrait être de rendre le code dans un meilleur état que celui où on l'a trouvé.

> Je ne comprends pas ce dernier paragraphe. Peux-tu préciser ?

Je ne suis pas sûr du bout pas compris. Pour une partie, il y avait trois options pour faire ce travail, la première c'est faire le -meetings.html et ajouter dans wcs ce qui pourrait manquer, zéro risque, mais faire ça ainsi permet de laisser de côté le travail en cours dans wcs, qui est de faire le rendu des widgets par des templates. Laisser ça de côté c'est dommage voire dangereux, parce qu'à un moment il va bien falloir y arriver et à ce moment-là, avoir du code mal conçu, parce que n'anticipant pas le besoin, sera plutôt un frein. La seconde option, c'est faire le travail sur le rendu générique et faire en sorte que celui-ci permette le nécessaire pour le rendu -meetings.html.

Comme ça peut être difficile d'anticiper les besoins sans y être confronté, que ce soit les besoins génériques dans l'option 1 ou les besoins spécifiques dans l'option 2, la troisième option c'est de réunir au même moment le travail sur les deux, qu'elles se nourrissent.

Et de là mon commentaire, comme quoi peut-être prendre cette troisième option est une erreur, qu'il y aurait bien moins de questions et de risques à partir sur l'option 1. (mais ce serait vrai uniquement en oubliant qu'il faudra faire le reste après).

Après il peut y avoir des entredeux possibles, créer un get_options_with_attributes normalisant bien les choses, y ajoutant le contenu de l'attribut name du widget qui s'y trouvera associé, utiliser cette nouvelle méthode pour le rendu -meetings.html, et déjà de manière générale dans l'__init__, pour reprendre et et raccourcir la boucle enumerate(options), la faire arriver à quelque chose comme :

```
        for _, title, key, attributes in self.get_options_with_attributes():  # NEW!
            element_kwargs = kwargs.copy()
            if attributes.get('disabled'):
                element_kwargs['disabled'] = 'disabled'
                self.disabled_options.append(name)
```

```
            if value and key in value:
                checked = True
            else:
                checked = False
            self.add(CheckboxWidget, name, title=title, value=checked, **element_kwargs)
            self.element_names[name] = key
```

(absolument pas testé, juste la mise en code de l'idée du paragraphe au-dessus)

Je sais bien que tout ça ne donne pas une direction unique, que ça augmente peut-être même les choix à faire.

~~

Par ailleurs évidemment on ne peut pas tout connaitre, je n'aurais pas pu dire que le passage option.attrs.items était inutile, ce que j'ai fait c'est lancer les tests avec une sonde pour voir la variété des usages :

```
             self.add(CheckboxWidget, name, title=title, value=checked, **element_kwargs)
+            if self.get_widget(name).attrs:
+                open('/tmp/kwargs', 'a+').write('%r\n' % self.get_widget(name).attrs)
             self.element_names[name] = key
```

#35 - 17 mai 2018 14:03 - Anonyme

- Sujet changé de créer un template qommon/forms/checkboxes.html et CheckboxesWidget.get_options à refonte de CheckboxesWidget avec fichier template checkboxes.html surchargeable

- Description mis à jour

#36 - 18 mai 2018 12:14 - Anonyme

Dans une tentative de reprendre le code de CheckboxesWidget en suivant les préconisations (code sur la branche **wip/checkboxes**, c'est wip wip wip, ne pas faire gaffe à tout les trucs trainant encore dans le code), je reste bloqué pour le moment sur un problème de mauvais attribut "name" sur les éléments *input checkbox* et qui *semblent* être doublonnés.
La trace d'exécution d'un test (test_checkboxes_widget) montre ici que name a comme valeurs à la fois 'False' et 'elementX':
http://paste.debian.net/1025217/
Autre moyen de voir ce problème : https://jenkins.entrouvert.org/job/wcs-wip-branches/166/
Ce problème d'input introuvable après la soumission par mock_form_submission, je l'avais rencontré au début du ticket, et j'en avais conclu à tort peut-être qu'il valait mieux construire le get_options() à partir du get_widgets() et en reprenant les attributs internes à l'objet CheckboxWidget.

Toute aide est la bienvenue, je continue quand même de mon côté à essayer de comprendre pourquoi l'attribut name n'est pas facilement accessible et unique dans le code. Merci d'avance.

#37 - 07 juin 2018 14:12 - Anonyme

- Patch proposed changé de Oui à Non

#38 - 11 juin 2018 11:49 - Anonyme

Changement de branche remote pour le travail en cours wip/checkboxes-tpl

#39 - 14 juin 2018 10:25 - Anonyme

- Sujet changé de refonte de CheckboxesWidget avec fichier template checkboxes.html surchargeable à amélioration du widget Liste à choix multiples pour permettre la personnalisation des templates

- Description mis à jour

#40 - 26 juin 2018 15:37 - Thomas Noël

- Assigné à changé de Anonyme à Thomas Noël

Je vais regarder ça.

#42 - 08 novembre 2020 09:28 - Frédéric Péters

- Assigné à changé de Thomas Noël à Frédéric Péters

Considérons tous les commentaires passés obsolètes.

#43 - 08 novembre 2020 10:29 - Frédéric Péters

- Statut changé de En cours à Rejeté

J'ai créé #48387 pour faire le taf sans avoir à trainer ces 40 commentaires.

Fichiers

0001-create-template-for-CheckboxesWidget-23413.patch	2,81 ko	25 avril 2018	Anonyme
0001-create-template-for-CheckboxesWidget-23413.patch	2,8 ko	25 avril 2018	Anonyme
0001-created-Checkboxeswidget.get_options-and-tests-23413.patch	3,33 ko	26 avril 2018	Anonyme
0001-add-Checkboxeswidget.get_options-and-tests-23413.patch	2,99 ko	02 mai 2018	Anonyme
0001-add-CheckboxesWidget.get_options-and-checkboxes.html.patch	4,48 ko	03 mai 2018	Anonyme
0001-add-CheckboxesWidget.get_options-and-checkboxes.html.patch	4,99 ko	03 mai 2018	Anonyme
0001-add-CheckboxesWidget.get_options-and-checkboxes.html.patch	4,99 ko	04 mai 2018	Anonyme
0001-add-CheckboxesWidget.get_options-and-checkboxes.html.patch	5,61 ko	07 mai 2018	Anonyme
0001-forms-use-a-template-to-render-checkboxes-widgets-23.patch	5,61 ko	07 mai 2018	Anonyme
Capture d'écran de 2018-05-07 20-28-50.png	156 ko	07 mai 2018	Anonyme
0001-forms-use-a-template-to-render-checkboxes-widgets-23.patch	5,65 ko	07 mai 2018	Anonyme