

w.c.s. - Development #31492

évolution de la source de données JSON pour y permettre du select2

17 Mar 2019 09:37 AM - Frédéric Péters

Status:	Solution déployée	Start date:	17 Mar 2019
Priority:	Normal	Due date:	
Assignee:	Frédéric Péters	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Planning:	No
Patch proposed:	Yes		
Description			
<ul style="list-style-type: none">• ajouter aux sources JSON un champ notant que l'URL accepte un paramètre de recherche textuelle (genre ?q=)• et un autre champ notant que l'URL accepte un paramètre de récupération selon identifiant• dans cette situation, sur un champ de type liste configuré avec cette source de données, utiliser select2• et dans le store_structured_value, plutôt que passer sur la liste des options possibles et y chercher l'id qui matche, appeler la source avec ?id=			
Related issues:			
Related to Passerelle - Development #31491: tableur : fournir un filtre ?id= ...		Fermé	17 Mar 2019
Related to w.c.s. - Bug #8845: il n'y a pas de structured enregistré pour les...		Nouveau	02 Nov 2015
Related to w.c.s. - Development #19271: possibilité de champ select2 pour une...		Solution déployée	06 Oct 2017
Related to w.c.s. - Development #31539: champ liste : passer d'un booléen "af...		Solution déployée	16 Mar 2019

Associated revisions

Revision 5351db8a - 04 Apr 2019 07:24 PM - Frédéric Péters

misc: refactor and test jsonp data source (#31492)

Revision 45a374ae - 04 Apr 2019 07:24 PM - Frédéric Péters

api utils: add function to sign an URL if orig is known (#31492)

Revision 065247ae - 04 Apr 2019 07:24 PM - Frédéric Péters

forms: extend json autocomplete support for be asynchronous if possible (#31492)

Revision 3bf61b16 - 04 Apr 2019 07:24 PM - Frédéric Péters

forms: add support for live change of JSON data source (#31492)

History

#1 - 17 Mar 2019 09:37 AM - Frédéric Péters

- Related to Development #31491: tableur : fournir un filtre ?id= automatique aux requêtes added

#2 - 17 Mar 2019 09:38 AM - Frédéric Péters

- Related to Bug #8845: il n'y a pas de structured enregistré pour les champs listes avec datasource en JSONP added

#3 - 17 Mar 2019 09:38 AM - Frédéric Péters

- Related to Development #19271: possibilité de champ select2 pour une liste (peu importe le jsonp) added

#4 - 17 Mar 2019 09:39 AM - Frédéric Péters

- Assignee set to Frédéric Péters

#5 - 27 Mar 2019 04:14 PM - Frédéric Péters

- Related to Development #31539: champ liste : passer d'un booléen "affichage radio" à une option "Affichage" added

#6 - 27 Mar 2019 04:23 PM - Frédéric Péters

- Patch proposed changed from No to Yes
- Status changed from Nouveau to Solution proposée
- File 0001-forms-extend-json-autocomplete-support-for-be-async.patch added

(la branche associée contient également [#31539](#) et [#19271](#) qui viennent avant ce patch)

Ce patch crée un `/api/autocomplete/<token>` qui est fait proxy vers l'URL de la source de données, en signant l'appel si possible. Le `<token>` est associé la session (stockés dans `data_source_query_url_tokens`), ça limite l'accès à la session en cours et ça permet de le faire varier dynamiquement (en cas de source type `https://.../rues?commune={{form_var_commune}}` avec `form_var_commune` sur la même page).

Ensuite, c'est d'une part le code jsonp existant pour le côté javascript (en passant il n'y avait pas de tests sur cette partie jsonp et c'est maintenant le cas) + du nouveau code pour aller chercher les données structurées associées au choix en interrogeant `url?id=xxx`. (le nom des paramètres `?q` et `?id` se paramètrent dans la source de données).

Rien de bien particulier sinon.

#7 - 04 Apr 2019 10:38 AM - Benjamin Dauvergne

- Status changed from Solution proposée to En cours

Pas simple à relire, pas simple à écrire certainement.

Je découperai une première partie avec l'introduction de `get_object()`, `type()`, `get_jsonp_url()`, `can_jsonp()` (qui pourrait être inutile en utilisant juste `get_jsonp_url()` je pense) et des tests jsonp pures et je modifierai à l'occasion dans `fields.py` les bouts jouent sur `real_data_source`, je pense que ça rendrait la lecture des modifications pour la conversion JSON -> JSONP plus claire.

Je suis un peu perturbé par les histoires de `jsonp_display_values` qui sont difficile à suivre ce qui rend les choses compliquées, j'espère qu'on simplifiera plus tard, voir qu'on aura plus besoin du tout du code JSONP pure (j'ai conscience que ça ne concerne que le cas JSONP pure mais comme c'est au milieu du reste).

Je ne suis pas sûr de comprendre pourquoi on fait ça :

```
with get_publisher().substitutions.temporary_feed(
    transient_formdata, force_mode='lazy'):
    data = self.formdef.get_data(form)
```

une histoire de calcul d'URL dans `get_structured_value()` ? Ça mériterait commentaire.

Au niveau du nouveau test JSON, il me semble que le champ caché `f0_display` ne sert plus dans ce cas puisque les données sont résolues après soumission depuis la source, alors peut-être vaut-il mieux tester que cela fonctionne sans ? En plus dans le test JSONP il y a aussi un jeu sur l'ordre du champ qui n'est pas présent ici.

Des bouts de code qui pourraient être factorisés :

```
unsigned_url = url
try:
    signature_key, orig = get_secret_and_orig(url)
except MissingSecret:
    pass
else:
    url += '&orig=%s' % orig
    url = sign_url(url, signature_key)
```

Le système de jeton est ok mais pas mal d'implicite dans le fait que l'URL se termine par le paramètre de query incomplet dans `get_jsonp_url()` si c'est du JSON+autocomplete mais une URL simple si c'est du JSONP (ce n'est donc pas du tout le même genre d'URL qui est retourné), pour éviter un stockage dans session supplémentaire on pourrait aussi utiliser `django.core.signing` avec un dico `{'url':..., 'session_id_hash': sha256(session_id)}`.

Globalement le code ne m'inquiète pas mais c'est pas facile à suivre.

#8 - 04 Apr 2019 10:46 AM - Frédéric Péters

Je suis un peu perturbé par les histoires de `jsonp_display_values (...)`

Oui ça reste parce qu'on continue à exploiter le `JsonpSingleSelectWidget`, qu'on fasse du jsonp pur, ou la version locale. C'est ça aussi qui explique qu'on garde le `_display` ajouté au formulaire, etc. (et que le code reste compliqué de manière générale, la simplification devenant possible quand le jsonp pur sera délogé).

#9 - 04 Apr 2019 12:10 PM - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

Le système de jeton est ok mais pas mal d'implicite dans le fait que l'URL se termine par le paramètre de query incomplet dans `get_jsonp_url()` si c'est du JSON+autocomplete mais une URL simple si c'est du JSONP (ce n'est donc pas du tout le même genre d'URL qui est retourné), pour éviter un stockage dans session supplémentaire on pourrait aussi utiliser `django.core.signing` avec un dico `{'url':..., 'session_id_hash': sha256(session_id)}`.

Ce que j'écris est faux; ne considérer que la fin :

pour éviter un stockage dans session supplémentaire on pourrait aussi utiliser `django.core.signing` avec un dico `{'url':..., 'session_id_hash': sha256(session_id)}`.

#10 - 04 Apr 2019 04:15 PM - Frédéric Péters

Je découperais (...)

C'est fait. (sans vraiment être convaincu que ça rende vraiment l'affaire plus lisible)

Je ne suis pas sûr de comprendre pourquoi on fait ça :

C'est désormais tapé dans un commit dédié, nommé "add support for live change of JSON data source" mais c'est les mêmes raisons, multiples et mystérieuses, que celles justifiant les autres appels du même type. (bref, une explication serait-elle même possible, elle ne tiendrait pas dans un commentaire, et l'objectif reste de revoir toute cette partie, je préfère ne pas y passer trop de temps à en documenter les errances).

Des bouts de code qui pourraient être factorisés :

C'est factorisé et séparé dans un commit "api utils: add function to sign an URL if orig is known".

#11 - 04 Apr 2019 04:15 PM - Frédéric Péters

- Status changed from *En cours* to *Solution proposée*

(branche à jour avec tout ça)

#12 - 04 Apr 2019 05:50 PM - Benjamin Dauvergne

- Status changed from *Solution proposée* to *Solution validée*

Ok.

#13 - 04 Apr 2019 07:52 PM - Frédéric Péters

- Status changed from *Solution validée* to *Résolu (à déployer)*

```
commit 3bf61b16b3e0fe260157020d048ae54edd2036a7
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Thu Apr 4 14:45:00 2019 +0200
```

```
forms: add support for live change of JSON data source (#31492)
```

```
commit 065247ae13110980e0080652fceb139b0a38f75c
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Thu Apr 4 15:22:50 2019 +0200
```

forms: extend json autocomplete support for be asynchronous if possible (#31492)

commit 45a374ae50cad7d7f49cdac8d4b248a696351b26
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Thu Apr 4 15:22:24 2019 +0200

api utils: add function to sign an URL if orig is known (#31492)

commit 5351db8a2ed204779b4ca93a1f36bb60c351ef33
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Thu Apr 4 14:43:42 2019 +0200

misc: refactor and test jsonp data source (#31492)

#14 - 04 Apr 2019 09:16 PM - Frédéric Péters

- Status changed from *Résolu (à déployer)* to *Solution déployée*

Files

0001-forms-extend-json-autocomplete-support-for-be-async.patch	29.2 KB	27 Mar 2019	Frédéric Péters
--	---------	-------------	-----------------