

## Hobo - Development #60699

### écrans d'assemblage et d'installation d'une application

16 janvier 2022 19:21 - Frédéric Péters

<b>Statut:</b>	Fermé	<b>Début:</b>	16 janvier 2022
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Frédéric Péters	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Oui		
<b>Description</b>			
Depuis un écran d'hobo on doit pouvoir assembler des éléments (formulaires, modèles de fiche, etc.) pour créer une application, et à partir de cet assemblage pouvoir déployer l'application constituée.			
<b>Demandes liées:</b>			
Lié à Publik - Development #49204: "Applification" des modules (applications ...		<b>En cours</b>	<b>08 décembre 2020</b>

#### Révisions associées

##### Révision 62220edc - 17 février 2022 16:05 - Frédéric Péters

add section to create/deploy applications (#60699)

#### Historique

##### #1 - 16 janvier 2022 19:23 - Frédéric Péters

- Lié à Development #49204: "Applification" des modules (applications métiers) Publik ajouté

##### #2 - 15 février 2022 10:21 - Frédéric Péters

- Fichier 0001-add-section-to-create-deploy-applications-60699.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Voilà version beta, il n'y a rien de fabuleux dedans,

- pas de lien "applications" en barre latérale de la page d'accueil actuellement, feature "cachée" / en développement pour le moment,
- dans hobo/applications/utlis.py avec une classe Requests similaire à celle dans Combo, pour signer automatiquement les appels,
- pour le déploiement effectif, pour le moment ça envoie l'application aux différents modules, ça ne gère pas d'éventuels retours, pour une barre de progression et ce genre de choses, j'ai posé des TODO à ce sujet.
- etc.

##### #3 - 17 février 2022 09:19 - Lauréline Guérin

Dans models: if service\_id != 'wcs':

Alors que dans views: if service\_id not in ('combo', 'wcs'):

Il manque le template hobo/applications/edit-metadata.html, du coup il doit aussi manquer un test :)

Petite remarque, j'ai créé une app en local, et pour voir je l'ai réimportée au même endroit; ça m'a mis à jour l'app, qui n'est plus éditable. Peut-être ne pas flager `editable=False` si l'app existe avec `editable=True` ?

##### #4 - 17 février 2022 09:29 - Frédéric Péters

Dans models: if service\_id != 'wcs':

Alors que dans views: if service\_id not in ('combo', 'wcs'):

Ah oui j'ai coupé en court de route la partie combo, à moitié.

Il manque le template hobo/applications/edit-metadata.html, du coup il doit aussi manquer un test :)

Et oui, quand on se dit que la bête vue avec le formulaire élémentaire ne mérite pas de test...

Petite remarque, j'ai créé une app en local, et pour voir je l'ai réimportée au même endroit; ça m'a mis à jour l'app, qui n'est plus éditable. Peut-être ne pas flager `editable=False` si l'app existe avec `editable=True` ?

Yes ça facilitera, je change ça.

#### #5 - 17 février 2022 10:52 - Frédéric Péters

- Fichier `0001-add-section-to-create-deploy-applications-60699.patch` ajouté

#### #6 - 17 février 2022 11:07 - Lauréline Guérin

```
element, created = Element.objects.get_or_create(
    type=element_dict['type'], slug=element_dict['slug'], defaults={'name': element_dict['name']}
)
```

Peut-être que tu pourrais ajouter un champ `service` au modèle `Element`, pour éviter les collisions de type ? Je pense au type `categories` qu'on pourrait peut-être retrouver dans plusieurs briques, genre `wcs` et `chrono` ?

#### #7 - 17 février 2022 11:41 - Frédéric Péters

Je préfère rester ainsi; surtout pour les dépendances j'avais idée qu'un formulaire pourrait déclarer une dépendance ('role', 'agent-traitant'), et ça voudrait dire alors mettre ici ('authentic', 'role', 'agent-traitant'), et ça serait alors systématiser au niveau des dépendances le tuple avec les trois éléments, et je trouvais que ça faisait redondant un peu moche, soit gérer des tuples parfois avec deux éléments parfois trois, et ça compliquait pour pas grand chose.

Et donc ma préférence à juste avoir un `slug`, et il y a "juste" à faire attention aux collisions quand un module se met à publier un nouveau type d'objet, ce qui devrait être peu commun. (pour les catégories on a déjà `prefix-categories` de manière systématique, même pour les démarches, `forms-categories`).

#### #8 - 17 février 2022 11:49 - Lauréline Guérin

- Statut changé de `Solution proposée` à `Solution validée`

Ok comme ça

#### #9 - 17 février 2022 16:06 - Frédéric Péters

- Statut changé de `Solution validée` à `Résolu (à déployer)`

```
commit 62220edc86bffe68763f8a53f01c0370640c19ee
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Sat Jan 8 16:07:08 2022 +0100
```

```
add section to create/deploy applications (#60699)
```

#### #10 - 17 février 2022 22:16 - Transition automatique

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

#### #11 - 24 avril 2022 04:42 - Transition automatique

Automatic expiration

#### Fichiers

<code>0001-add-section-to-create-deploy-applications-60699.patch</code>	70,8 ko	15 février 2022	Frédéric Péters
<code>0001-add-section-to-create-deploy-applications-60699.patch</code>	71,9 ko	17 février 2022	Frédéric Péters