

Combo - Development #15043

Permettre à l'utilisateur/agent de gérer en frontoffice le contenu / la position de certaines cellules (tableau de bord)

17 février 2017 13:56 - Frédéric Péters

Statut: Fermé	Début: 17 février 2017
Priorité: Normal	Echéance:
Assigné à:	% réalisé: 0%
Catégorie:	Temps estimé: 0:00 heure
Version cible:	Planning:
Patch proposed: Oui	
Description	
L'idée c'est la production d'un tableau de bord que l'utilisateur composerait en agencant différentes cellules.	
<ul style="list-style-type: none">• pouvoir marquer ce qui servira de zone d'accueil de ces cellules (on ne veut pas tout laisser être modifié), cellule de type "dashboard" (?)• pouvoir associer une cellule à un usager (foreign key user à ajouter au modèle CellBase)• une "API" pour permettre l'ajout de cellules (/dashboard/<dashboard_cell_id>/add?type=linkcell&title=plop&url=http://... pour l'ajout d'un signet, par exemple)• pareil pour la suppression• de l'UI à creuser au-dessus de tout ça	
Demandes liées:	
Lié à Combo - Development #16300: cellule json : rendu direct quand le json e...	Fermé 12 mai 2017
Lié à Combo - Development #16301: passer par le templatetag render_cell pour ...	Fermé 12 mai 2017

Révisions associées

Révision cc84f3c5 - 23 mai 2017 14:38 - Frédéric Péters

general: add initial support for custom user dashboards (#15043)

Historique

#1 - 08 mars 2017 10:48 - Frédéric Péters

"Plan".

Avoir DashboardCell(CellBase) pour faire "contenant", et à côté TileCell(CellBase), pour les tuiles/widgets/etc. qu'on peut ajouter au tableau de bord. C'est dans TileCell qu'on a foreign key dashboard et foreign key user (i.e. on n'ajoute pas ça à toutes les cellules), null=True, pour se permettre d'ajouter ces cellules "tuiles" à d'autres endroits.

Pur déclaratif, et comme on a aujourd'hui dans BlurpCell, une méthode get_cell_types() pour donner l'impression de différents types de cellule.

Dans le settings.TILES déclarant les tuiles :

- key (qui sert aussi à déterminer le template à prendre),
- label,
- config (ou url de config), qui serait "en gros" un formdef.to_json(), ça sert après à ajouter ces infos au contexte de rendu,
- data url, qui serait une url récupérée pour le rendu

(TileCell, ça peut aussi être TileCell(JsonCell), à voir)

#2 - 11 mai 2017 12:35 - Frédéric Péters

- Fichier 0001-add-initial-support-for-custom-user-dashboards-15043.patch ajouté

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

Plan plusieurs fois révisé depuis pour arriver à la proposition attachée, qui permet d'être opérationnel sur le guichet grand lyon, i.e. ça dépend de trucs spécifiques grand-lyon-gnm dans publik-base-theme, et du coup la cellule est désactivée par défaut.

Ça ajoute DashboardCell, qui est une cellule normale, dont le job est de servir de conteneur pour des tuiles, basiquement ça fait que son render() est une boucle affichant chacune des tuiles.

Les tuiles, c'est l'objet Tile, c'est une référence à une cellule et à un utilisateur. (également à un dashboard mais dans la pratique on considère pour le moment un seul dashboard par site).

Il n'y a pas de création de tuile ex nihilo, ça fonctionne pour le moment plus sur le modèle du "favori", i.e. on ajoute une cellule existante comme tuile à son tableau de bord. Dans la pratique, une cellule qui veut pouvoir être ajoutée au tableau de bord fait un include de `combo/dashboard_cell_icons.html`, qui ajoutera une icône "mettre en favori".

Pour la suppression, quand il y a rendu d'une cellule on voit que c'est en fait une tuile (ça oblige à mettre un peu de la logique du dashboard dans le `combo/public/views.py`) et dans ce cas on ajoute "in_dashboard" au contexte, ce qui permet à `combo/dashboard_cell_icons.html` de mettre un menu de suppression plutôt qu'une icône "mettre en favori".

(Dans une phase ultérieure, pour d'autres besoins, il y a aura vraisemblablement la création/configuration, mais je sors ça de ce ticket)

Plus tard aussi, le dragndrop pour réagencer les tuiles. Dans ce cadre, peut-être, plutôt qu'un simple "order", ça divisera ça en `order_x/order_y`, pour imaginer un tableau de bord comme on avait univcloud.

#3 - 12 mai 2017 14:09 - Frédéric Péters

- Fichier `0001-general-add-initial-support-for-custom-user-dashboar.patch` ajouté

Réagencé pour venir après [#16300](#) et [#16301](#). Le truc étant que le dashboard profite aussi de ces tickets, pour faire moins de requêtes sur du contenu en cache.

#4 - 12 mai 2017 17:10 - Frédéric Péters

- Fichier `0001-general-add-initial-support-for-custom-user-dashboar.patch` ajouté

(j'avais oublié d'inclure cinq lignes de js)

#5 - 12 mai 2017 17:11 - Frédéric Péters

- Fichier `0001-grandlyon-gnm-add-styles-for-the-dashboard-15043.patch` ajouté

#6 - 12 mai 2017 17:13 - Frédéric Péters

- Fichier `Capture d'écran 2017-05-12 à 17.12.40.png` ajouté

#7 - 12 mai 2017 17:15 - Frédéric Péters

- Lié à Development [#16300](#): cellule json : rendu direct quand le json est dans le cache ajouté

#8 - 12 mai 2017 17:15 - Frédéric Péters

- Lié à Development [#16301](#): passer par le `templatetag render_cell` pour le rendu ajax ajouté

#10 - 15 mai 2017 16:17 - Frédéric Péters

- Fichier `0001-general-add-initial-support-for-custom-user-dashboar.patch` ajouté

Actualisé avec des modifs sur le MANIFEST.in pour distribuer les templates et static.

#11 - 15 mai 2017 16:17 - Frédéric Péters

Aussi, c'est déployé sur <https://portail-citoyen.quichet-recette.grandlyon.com/> pour tester.

#12 - 23 mai 2017 14:33 - Thomas Noël

Ack

Je le dis dans ce ticket :

- un seul dashboard par site (parce que sinon, de toute façon, les UIs sont pas simples)
- exemple d'utilisation à GNM, qui ajoute ce qu'il faut en html/js/css pour que ça marche (ça ne marche pas tout seul)

#13 - 23 mai 2017 14:43 - Frédéric Péters

- Statut changé de *En cours* à *Résolu* (à déployer)

Merci,

un seul dashboard par site (parce que sinon, de toute façon, les UIs sont pas simples)

Yep, à noter que dans des évolutions futures, ça pourrait être plus fin que ça, par exemple le dashboard "liens favoris" pourrait être différent du dashboard "événements favoris", ou autre exemple en rapport avec le portail agent. Mais ça demande une évolution qui serait l'ajout de tuile depuis le dashboard plutôt que depuis la tuile.

exemple d'utilisation à GNM, qui ajoute ce qu'il faut en html/js/css pour que ça marche (ça ne marche pas tout seul)

Oui et je viens de pousser ces styles; c'est particulier GNM parce qu'ils sont en mode "progressive web app" et que du coup je n'ai pas réfléchi pour un publik normal, je verrai rapidement pour avoir des styles convenables.

```
commit cc84f3c5be708c04b7f6f2824841457ebca717bd
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Fri May 12 14:06:55 2017 +0200
```

```
general: add initial support for custom user dashboards (#15043)
```

#14 - 23 décembre 2018 15:21 - Frédéric Péters

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

Fichiers

0001-add-initial-support-for-custom-user-dashboards-15043.patch	19,5 ko	11 mai 2017	Frédéric Péters
0001-general-add-initial-support-for-custom-user-dashboar.patch	18,8 ko	12 mai 2017	Frédéric Péters
0001-general-add-initial-support-for-custom-user-dashboar.patch	19,4 ko	12 mai 2017	Frédéric Péters
0001-grandlyon-gnm-add-styles-for-the-dashboard-15043.patch	2,97 ko	12 mai 2017	Frédéric Péters
Capture d'écran 2017-05-12 à 17.12.40.png	27,9 ko	12 mai 2017	Frédéric Péters
0001-general-add-initial-support-for-custom-user-dashboar.patch	20,4 ko	15 mai 2017	Frédéric Péters