w.c.s. - Development #4507

backoffice : mémorisation de listing

19 mars 2014 15:06 - Frédéric Péters

Statut:	Fermé	**Début:**	19 mars 2014
Priorité:	Normal	**Echéance:**	30 avril 2020
Assigné à:	Frédéric Péters	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Avec #4505, et déjà avec la sélection de colonnes, il devient intéressant de permettre à un agent d'enregistrer son paramétrage de listing.

Demandes liées:

Lié à w.c.s. - Development #1782: listing du backoffice: chaque utilisateur d...	**Fermé**	**15 octobre 2012**
Lié à w.c.s. - Development #15165: Sauvegarde des filtres sur vues globale et...	**Rejeté**	**27 février 2017**
Lié à w.c.s. - Development #15003: Elargir possibilités du filtres statuts da...	**Fermé**	**15 février 2017 17 mars 2017**
Lié à Publik - Development #19754: Possibilité pour les agents de sauvegarder...	**Fermé**	**29 octobre 2017**
Lié à Publik - Development #37532: Vues personnalisées sur les demandes et su...	**Fermé**	**07 novembre 2019**
Lié à w.c.s. - Bug #41768: Enregistrement d'une vue listing : le bouton "supp...	**Fermé**	

Révisions associées

Révision 5c04a440 - 15 avril 2020 17:33 - Frédéric Péters

backoffice: add support for custom views (#4507)

Historique

#1 - 05 avril 2014 12:21 - Pierre Cros

Et même d'en sauvegarder plusieurs, façon Redmine (enfin si ça rajoute pas trop en coûts de dev)

#2 - 08 avril 2014 14:46 - Frédéric Péters

Lien posé vers #1782, qui reprend aussi quelques idées.

#3 - 11 septembre 2014 17:22 - Victor Claudet

- Patch proposed mis à Non

Meyzieu me relance sur cette fonctionnalité, alors je le dis ici.

#4 - 11 septembre 2014 17:24 - Frédéric Péters

On leur a vendu ?

#5 - 07 août 2015 07:45 - Victor Claudet

On l'a vendu a personne et tout le monde la demande (Vincennes aussi par exemple depuis longtemps), Alfortville maintenant.

#6 - 29 novembre 2016 18:30 - Pierre Cros

~~Important pour afficher des cartes des demandes dans Combo~~ #8454

#7 - 29 novembre 2016 18:35 - Frédéric Péters

~~On limite ce ticket à la possibilité pour l'admin (de la fabrique de formulaire) d'enregistrer formdef par formdef sa sélection de filtre/colonnes.~~

#8 - 19 janvier 2017 16:08 - Frédéric Péters

- Priorité changé de Normal à Haut

#9 - 07 mars 2017 09:20 - Brice Mallet

- Lié à Development #15165: Sauvegarde des filtres sur vues globale et par formulaires ajouté

#10 - 08 mars 2017 13:39 - Frédéric Péters

À côté du bouton "Valider" / "Rafraîchir" (besoin d'uniformiser entre vue globale et vue d'un formdef, #15347), quand il y a eu un filtre appliqué (techniquement, quand la query string n'est pas vide), avoir un bouton "Enregistrer". Popup alors pour entrer un nom et si l'agent est admin (de la fabrique de formulaires), case à cocher "visible par tout le monde", case à cocher "par défaut". On enregistre la query string + formdef.id (et None si c'est une requête "globale").

Sous les boutons "Valider" (et maintenant "Enregistrer"), la liste des requêtes enregistrées, sous un titre "Requêtes personnalisées" (?). Ou au-dessus des critères plutôt ? (dans la vue "statistiques globales", on a "Raccourcis", sous les critères). Quand on est sur le listing d'un formulaire précis, afficher à la fois les requêtes "globales" et les requêtes attachées au formulaire en question.

Quand on clique sur un des liens "requête", poser la query string associée + un paramètre identifiant la requête (query_id). En présence de cet identifiant de requête dans l'URL, la popup "enregistrer" propose alors d'écraser ou de créer une nouvelle. Aussi, un bouton "Supprimer la requête".

#11 - 08 mars 2017 17:06 - Pierre Cros

Frédéric Péters a écrit :

> Sous les boutons "Valider" (et maintenant "Enregistrer"), la liste des requêtes enregistrées, sous un titre "Requêtes personnalisées" (?). Ou au-dessus des critères plutôt ?

Sous les critères, mais plutôt "Vues personnalisées".

Pour le reste, tout semble avoir été prévu, top.

#12 - 09 mars 2017 21:11 - Brice Mallet

- Lié à Development #15003: Elargir possibilités du filtres statuts dans la vue globale ajouté

#13 - 12 mars 2017 09:15 - Brice Mallet

> "Ou au-dessus des critères plutôt ? (dans la vue "statistiques globales", on a "Raccourcis", sous les critères)"

Mais dans les stats BiJoe, les visualisations - construites - se placent en premier. Je serais pour présenter les requêtes enregistrées en premier (si les qq indicateurs / listes sont pertinents, plus besoin d'utiliser de nouveaux filtres) et avec affichage graphique plus fort que la construction de nouveaux filtres

#14 - 12 mars 2017 09:42 - Brice Mallet

> Quand on est sur le listing d'un formulaire précis, afficher à la fois les requêtes "globales" et les requêtes attachées au formulaire en question.

Les requêtes "globales" : cela signifie t'il les requêtes construites sur la base de la vue globale ? ou plutôt les requêtes sauvegardées par l'administrateur de la fabrique de formulaires qui viennent alors en plus des requêtes propres à l'agent ?

#15 - 12 mars 2017 10:41 - Frédéric Péters

> cela signifie t'il les requêtes construites sur la base de la vue globale ?

Oui, celles-là.

On a quatre situations :

- requête créée par un agent, sur un formulaire précis : visible uniquement sur celui-ci, pour cet agent
- requête créée par un agent, sur la vue globale : visible de la vue globale et sur tous les formulaires, pour cet agent
- requête créée et partagée par un admin, sur un formulaire précis : visible uniquement sur le formulaire, par tous les agents
- requête créée et partagée par un admin, sur la vue globale : visible de la vue globale et sur tous les formulaires, par tous les agents.

#17 - 29 octobre 2017 06:44 - Pierre Cros

- Lié à Development #19754: Possibilité pour les agents de sauvegarder des vues personnalisées ajouté

#18 - 07 novembre 2019 16:04 - Frédéric Péters

- Lié à Development #37532: Vues personnalisées sur les demandes et sur les fiches ajouté

#19 - 09 mars 2020 17:33 - Frédéric Péters

- Echéance mis à 30 avril 2020

- Assigné à mis à Frédéric Péters

- Priorité changé de Haut à Normal

#20 - 31 mars 2020 09:44 - Frédéric Péters

- Fichier 0001-backoffice-add-support-for-custom-views-4507.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Patch pour les fonctionnalités essentielles, et un peu au-delà,

- possibilité pour l'agent d'enregistrer des vues personnalisées sur les formulaires et les fiches,
- possibilité pour l'admin d'enregistrer et d'exposer ses vues à tous les agents,
- possibilité de réordonner les colonnes,
- visibilité dans l'API (/api/forms/<slug>/list/<custom>/) du résultat.

Une fois cette base en place il y a quantité d'évolutions possibles (permettre d'exploiter ces vues dans les sources de données, pour par exemple avoir une source "postes à pourvoir (ouverts)", questions que ça pose sur l'export/import, questions sur la possibilité de vues partagées par différents formulaires, filtres supplémentaires...).

Le patch en lui-même est je pense assez clair, sur la page de tableau de traitement (FormPage) un attribut supplémentaire (view) et celui-ci est consulté, avant la query string, pour connaitre le paramétrage à appliquer.

À part ça, dans l'objet CustomView, le truc à noter peut-être c'est l'attribut columns, qui n'est pas une liste des identifiants mais une liste de dictionnaires {'id': ...}, c'est dans l'idée de pouvoir faire évoluer ça pour contenir davantages d'informations (et peut-être avoir des colonnes personnalisées, qui seraient définies à partir d'un gabarit django, pour faire comme les "projections" dans le connecteur CSV).

#21 - 15 avril 2020 01:16 - Thomas Noël

Il y a un bogue ici :

```
+        existing_slugs = {
+            x.slug: True
+            for x in self.select(ignore_errors=True)
+            if (x.user_id == self.user_id and x.visibility == 'owner' and self.visibility == 'owner')
+            and x.formdef_type == self.formdef_type
+            and x.formdef_id == self.formdef_id
+        }
```

qui permet la création de vues "any" avec le même slug, voire mix any/owner. En fait pour éviter tout conflit je pense qu'il faut prendre tous les slugs du même formdef_type/formdef_id, sans autre critère.

Lorsque l'usager n'est pas admin, voir comment préfixer le slug avec "user-", pour éviter qu'un usager pollue le travail de l'admin (typiquement, admin qui doit créer un slug spécifique utilisé par un consommateur d'api).

Typo sur self.admin_permisison qui se répète heureusement partout, mais bon, à corriger (partout donc).

Dans get_custom_view_form il y a d'ailleurs un is_accessible('forms') à remplacer alors par is_accessible(self.admin_permission) (au niveau de la vérification de la permission de mise à jour de la vue)

J'ai eu un doute sur "Vues personnalisées" qui va s'afficher quand il n'y aura que des vues partagées (créées par un admin). Mais je n'ai pas d'autre proposition alors gardons cela ainsi, c'est simple et très bien. Et après tout, ce sont des vues personnalisées, par une autre personne, mais personnalisées quand même.

J'ai rien vu d'autre ce soir. Ca marche plutôt bien comme tout.

Dans les évolutions il faudra sans doute prévoir (et je pense que le code ici présent le permettra) :

- d'exporter/importer toutes les vues "any" liées à un formdef/carddef lors de l'export/import de celui-ci
- de permettre qu'une vue any soit la vue par défaut au niveau de l'UI (mais pas de l'API), juste par un redirect quand on arrive sur backoffice/management/formslug/

#22 - 15 avril 2020 08:35 - Frédéric Péters

> qui permet la création de vues "any" avec le même slug, voire mix any/owner. En fait pour éviter tout conflit je pense qu'il faut prendre tous les slugs du même formdef_type/formdef_id, sans autre critère.

+

> Lorsque l'usager n'est pas admin, voir comment préfixer le slug avec "user-", pour éviter qu'un usager pollue le travail de l'admin (typiquement, admin qui doit créer un slug spécifique utilisé par un consommateur d'api).

Je préfère du coup garder l'unicité par utilisateur (+ correction pour avoir une unicité aussi pour les vues "any"), certes si un usager crée une vue avec un nom utilisé par l'admin celle-ci prendra le dessus mais je trouve limite ça bien. (par contre il faut être sûr que ça soit toujours le cas, branche modifiée pour que ça soit le cas).

~~

> Dans les évolutions (...)

Oui ça fait partie de bouts envisagés, pas compliqués une fois la base en place.

#23 - 15 avril 2020 10:35 - Thomas Noël

Frédéric Péters a écrit :

> certes si un usager crée une vue avec un nom utilisé par l'admin celle-ci prendra le dessus

Y'a encore du conflit : je suis admin, je créé une vue perso nommée test, slug test. Je créé une autre vue nommée test, cette fois partagée, elle prend également le slug test... oups (de plus je n'y ai pas accès, c'est la vue perso test qui prend le dessus).

C'est une situation qui va arriver et qui sera difficile à déboguer.

> si un usager crée une vue avec un nom utilisé par l'admin celle-ci prendra le dessus mais je trouve limite ça bien.

Pas moi : si l'usager avait déjà créé sa vue, il ne comprendra pas pourquoi il n'a jamais accès à la nouvelle vue proposée par un admin. (et personne ne comprendra, impossible de le voir sans se connecter à la place de l'utilisateur en panne)

#24 - 15 avril 2020 11:35 - Frédéric Péters

J'ai besoin d'un récap de ce qui est proposé, je partais de :

- une vue partagée peut avoir n'importe quel slug (pas déjà utilisé, ce qui manquait dans mon patch initial),
- une vue privée peut avoir n'importe quel slug (pas déjà utilisé, et le patch initial acceptait un slug doublon d'une vue partagée sans définir de comportement, et le nouveau patch accepte aussi mais définit le comportement).

S'ajouterait "pour éviter tout conflit je pense qu'il faut prendre tous les slugs du même formdef_type/formdef_id, sans autre critère" :

- une vue privée peut avoir n'importe quel slug sauf un slug déjà utilisé (peu importe partagée ou pas et l'utilisateur concerné),

+ "voir comment préfixer le slug avec "user-", pour éviter qu'un usager pollue le travail de l'admin"

- une vue privée doit commencer par "user-",
- une vue partagée ne peut pas commencer par "user-".

(pas de patch, besoin de clarifier tout ça d'abord)

#25 - 15 avril 2020 12:10 - Thomas Noël

Frédéric Péters a écrit :

> J'ai besoin d'un récap de ce qui est proposé, je partais de :
>
> - une vue partagée peut avoir n'importe quel slug (pas déjà utilisé, ce qui manquait dans mon patch initial),
> - une vue privée peut avoir n'importe quel slug (pas déjà utilisé, et le patch initial acceptait un slug doublon d'une vue partagée sans définir de comportement, et le nouveau patch accepte aussi mais définit le comportement).
>
> S'ajouterait "pour éviter tout conflit je pense qu'il faut prendre tous les slugs du même formdef_type/formdef_id, sans autre critère" :
>
> - une vue privée peut avoir n'importe quel slug sauf un slug déjà utilisé (peu importe partagée ou pas et l'utilisateur concerné),
>
> + "voir comment préfixer le slug avec "user-", pour éviter qu'un usager pollue le travail de l'admin"

- une vue privée doit commencer par "user-",
- une vue partagée ne peut pas commencer par "user-".

(pas de patch, besoin de clarifier tout ça d'abord)

Je suis perdu aussi, je pense qu'on cherche à être intelligent et c'est jamais bon signe :) Ce que je vois c'est que les slugs ne peuvent pas changer alors que la visibilité (perso/partagée) peut, ça complexifie l'algo.

À partir de là, ma proposition est bête-et-méchante :

- ne jamais permettre de doublon de slug, dans aucun cas, quelque soit la visibilité ou le créateur. Pas d'intelligence, garantie zéro conflit.
- quand le créateur n'est pas admin, les slug sont préfixés avec « user-<id>-- » ou juste « user-- » ou «u--». L'admin en revanche n'est pas soumis à cette contrainte, même pour ses vues perso.

La raison de cette dernière règle : ne pas polluer les admin. Ils pourront toujours créer des slugs comme il veut (principalement dans le but de créer des slugs «parlants» et «fixés» pour les API/sources-de-données/etc). Et pour toutes ses vues, quelle que soit la visibilité, afin qu'il puisse les créer/tester d'abord en vue perso avant de les partager.

Effectivement ainsi on ne permet pas à un agent d'écraser une vue partagée par une vue perso, parce que c'est source d'incompréhension.

#26 - 15 avril 2020 13:30 - Frédéric Péters

Je continue à trouver pas terrible de "polluer" les URL avec des suffixes tiret-numéro.

J'en viens à ma proposition initiale (slugs unique par utilisateur / globalement quand c'est "any") mais modifier le lookup pour que .../<slug> cherche uniquement les vues "any" et que .../user-<slug> cherche uniquement les vues "owner"; ça me semble éviter tous les problèmes :

- les vues privées et publiques ont toujours des adresses différentes
- les vues publiques ont des slugs uniques
- les vues privées, par utilisateur, ont des slugs uniques, et c'est les seuls qui sont vues

#27 - 15 avril 2020 13:52 - Thomas Noël

Frédéric Péters a écrit :

> Je continue à trouver pas terrible de "polluer" les URL avec des suffixes tiret-numéro.
>
> J'en viens à ma proposition initiale (slugs unique par utilisateur / globalement quand c'est "any") mais modifier le lookup pour que .../<slug> cherche uniquement les vues "any" et que .../user-<slug> cherche uniquement les vues "owner"; ça me semble éviter tous les problèmes :
>
> - les vues privées et publiques ont toujours des adresses différentes
> - les vues publiques ont des slugs uniques
> - les vues privées, par utilisateur, ont des slugs uniques, et c'est les seuls qui sont vues

Yep ça roule pas mal comme ça. (Note : cas certainement rare mais il faudra interdire qu'un slug commence par "user-" pour ne pas qu'il recouvre une vue "owner" si un jour il est slug d'une vue "any")

#28 - 15 avril 2020 14:04 - Frédéric Péters

(branche modifiée/poussée)

> Yep ça roule pas mal comme ça. (Note : cas certainement rare mais il faudra interdire qu'un slug commence par "user-" pour ne pas qu'il recouvre une vue "owner" si un jour il est slug d'une vue "any")

Sur ce cas si jamais ça arrive le slug sera créé comme étant "userx-..." plutôt que "user-...", totalement arbitrairement.

#29 - 15 avril 2020 14:31 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Ça me va comme ça, allons-y !

#30 - 15 avril 2020 17:34 - Frédéric Péters

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

```
commit 5c04a4402a8c65b16ef5d1f604ddc283d2a710be
Author: Frédéric Péters <fpeters@entrouvert.com>
Date:   Mon Mar 9 09:35:09 2020 +0100

    backoffice: add support for custom views (#4507)
```

#31 - 15 avril 2020 21:16 - Frédéric Péters

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

#32 - 15 avril 2020 22:50 - Marie Kuntz

<3

#33 - 15 avril 2020 22:52 - Marie Kuntz

- Lié à Bug #41768: Enregistrement d'une vue listing : le bouton "supprimer la vue" est coupé ajouté

Fichiers

0001-backoffice-add-support-for-custom-views-4507.patch	75,7 ko	31 mars 2020	Frédéric Péters