Authentic 2 - Development #32833

interface de création d'utilisateurs en masse via un CSV

06 mai 2019 17:32 - Frédéric Péters

Statut:	Fermé	**Début:**	06 mai 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	100%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

```
   Fichier
    [ Parcourir... ]
    Fichier CSV avec comme colonnes identifiant applicatif, prénom,
    nom, etc.  <Télécharger un fichier d'exemple>

   Collectivité
    [ Collectivité par défaut |v]

                                                [Annuler] [Importer]
```

La liste des colonnes est dynamique, fonction des attributs configurés. Si le site est configuré pour ne pas afficher d'identifiant, on n'inclut pas d'identifiant.

L'"identifiant applicatif" est autre chose que l'identifiant, il permet de fournir une clé d'unicité différente de l'email, utile en cas d'imports successifs. (ça porrait aller dans UserExternalId avec source = "import")

Demandes liées:

Lié à Authentic 2 - Bug #35800: import csv et champ date	**Fermé**	**04 septembre 2019**

Révisions associées

Révision 853140e8 - 21 juin 2019 13:31 - Benjamin Dauvergne

add unique constraint to UserExternalId (#32833)

Révision 8c06edd1 - 21 juin 2019 13:31 - Benjamin Dauvergne

custom_user: rename clean_fields to validate_unique (#32833)

Révision fe0895da - 21 juin 2019 18:44 - Benjamin Dauvergne

add csv import framework (#32833)

Révision dc3582ed - 22 juin 2019 13:06 - Benjamin Dauvergne

manager: add user import views (fixes #32833)

Historique

#1 - 07 mai 2019 09:16 - Paul Marillonnet

Je ne suis pas bien au courant du besoin derrière tout ça. Est-ce qu'il y a quelque chose qui justifie la nécessité de mentionner la collectivité dans l'UI, plutôt que de la déclarer dans le CSV ?
Est-ce qu'on aurait pas des cas où on souhaite importer en masse des usagers appartenant à X collectivités différentes, avec X suffisamment grand pour que répéter l'import X fois soit pénible pour l'admin ?
(Auquel cas on pourrait imaginer supporter une colonne "Uuid de la collectivité d'appartenance", ou bien deux colonnes "Nom de la collectivité d'appartenance" et "Slug de la collectivité d'appartenance".)

#2 - 07 mai 2019 09:29 - Frédéric Péters

> Est-ce qu'on aurait pas des cas où on souhaite importer en masse des usagers appartenant à X collectivités différentes, avec X suffisamment grand pour que répéter l'import X fois soit pénible pour l'admin ?

Dans l'immédiat, non. Et je dirais que sur un authentic où il y aurait un grand nombre de collectivités, la gestion des comptes sera déléguée à celles-ci, et qu'elles se ramènent au cas où l'import d'un fichier envoie tout dans la même collectivité.

#3 - 07 mai 2019 10:04 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Je ne suis pas bien au courant du besoin derrière tout ça. Est-ce qu'il y a quelque chose qui justifie la nécessité de mentionner la collectivité dans l'UI, plutôt que de la déclarer dans le CSV ?
> Est-ce qu'on aurait pas des cas où on souhaite importer en masse des usagers appartenant à X collectivités différentes, avec X suffisamment grand pour que répéter l'import X fois soit pénible pour l'admin ?
> (Auquel cas on pourrait imaginer supporter une colonne "Uuid de la collectivité d'appartenance", ou bien deux colonnes "Nom de la collectivité d'appartenance" et "Slug de la collectivité d'appartenance".)

Ça m'irait qu'on renomme le champ de formulaire "Collectivité" par "Collectivité par défaut", dans le CSV on supporte ou__slug, ou__uuid ou ou__name en plus si pas de colonne ou (ou colonne OU ambiguë) et pas de collectivité par défaut, on lève une erreur d'import.

En parallèle il faudra abandonner Collectivité et revenir à unité d'organisation, ça a assez durer cette histoire.

#4 - 07 mai 2019 11:04 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Ça m'irait qu'on renomme le champ de formulaire "Collectivité" par "Collectivité par défaut", dans le CSV on supporte ou__slug, ou__uuid ou ou__name en plus si pas de colonne ou (ou colonne OU ambiguë) et pas de collectivité par défaut, on lève une erreur d'import.

Ok moi je veux bien, mais je ne sais pas comment gérer ça avec le champ OU du formulaire.
On peut décider par exemple qu'en cas d'absence de ces colonnes ou_…, ou de en cas d'absence de valeurs pour ces colonnes ou_…, on place l'usager dans l'OU qui a été choisie directement via le formulaire.
Dans le cas contraire (présence des colonnes et valeurs renseignées), alors la valeur choisie dans le formulaire est laissée de côté au profit des informations contenues dans le CSV.

Aussi, il faudrait, sur la validation du formulaire, effectuer un dry-run en premier, pour vérifier que pas d'erreur d'import, avant de commencer à créer séquentiellement les usagers.

Une autre chose : je ne vois pas encore une façon propre de générer un CSV d'exemple.
Je pensais avoir un jeu de données d'exemples pour chaque type d'attribut (attribute_kind) par défaut (et donc en dur dans le code), et pour les types d'attributs définis par la suite, essayer de générer des données d'exemple en fonction du field_class.
Mais ça ne couvre pas tous les cas, et on pourrait se retrouver avec des attributs pour lesquels on ne sait pas générer des données d'exemple.

Une dernière difficulté encore (et en lien avec ce qui est écrit plus haut à propos du fichier d'exemple) : est-ce qu'on cherche à gérer ici les attributs multiples ?

> En parallèle il faudra abandonner Collectivité et revenir à unité d'organisation, ça a assez durer cette histoire.

Là c'est plutôt un ticket de traduction, non ?

#5 - 07 mai 2019 11:27 - Pierre Cros

Benjamin Dauvergne a écrit :

> En parallèle il faudra abandonner Collectivité et revenir à unité d'organisation, ça a assez durer cette histoire.

Au moment ou A2 ne sert plus qu'à Publik, abandonner un libellé adopté par toutes les instances de Publik ? Vraiment pas.

#6 - 07 mai 2019 11:49 - Paul Marillonnet

Autre chose qui m'irait bien (et qui est présente dans pas mal d'UI d'import de csv), c'est la possibilité de mentionner que la première ligne contient la première entrée, et non pas l'intitulé des colonnes.

La possibilité de choisir le délimiteur me plairait bien aussi.

En modifiant l'idée de formulaire proposée par Frédéric dans la description du ticket, peut-être quelque chose comme ça ?

```
Fichier
 [ Parcourir... ]
 Fichier CSV avec comme colonnes identifiant applicatif, prénom,
 nom, etc.  <Télécharger un fichier d'exemple>

 [x] La première ligne contient la première entrée, et non pas l'intitulé des colonnes.
```

```
    Délimiteur de colonne:
    [ ',' (virgule) |v]

   Collectivité
    [ Collectivité par défaut |v]

                                              [Annuler] [Importer]
```

#7 - 07 mai 2019 11:52 - Benjamin Dauvergne

Pierre Cros a écrit :

> Benjamin Dauvergne a écrit :
>
> > En parallèle il faudra abandonner Collectivité et revenir à unité d'organisation, ça a assez durer cette histoire.
>
> Au moment ou A2 ne sert plus qu'à Publik, abandonner un libellé adopté par toutes les instances de Publik ? Vraiment pas.

Me semblait qu'on l'avait acté au dernier eocamp, mais ça sort de ce ticket, reparlons en la semaine prochaine.

#8 - 07 mai 2019 11:53 - Thomas Noël

Faut faire un truc compatible avec l'export, sinon ça sera incompréhensible, àmha.

#9 - 07 mai 2019 11:56 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Benjamin Dauvergne a écrit :
>
> > Ça m'irait qu'on renomme le champ de formulaire "Collectivité" par "Collectivité par défaut", dans le CSV on supporte ou__slug, ou__uuid ou ou__name en plus si pas de colonne ou (ou colonne OU ambiguë) et pas de collectivité par défaut, on lève une erreur d'import.
>
> Ok moi je veux bien, mais je ne sais pas comment gérer ça avec le champ OU du formulaire.
> On peut décider par exemple qu'en cas d'absence de ces colonnes ou_…, ou de en cas d'absence de valeurs pour ces colonnes ou_…, on place l'usager dans l'OU qui a été choisie directement via le formulaire.
> Dans le cas contraire (présence des colonnes et valeurs renseignées), alors la valeur choisie dans le formulaire est laissée de côté au profit des informations contenues dans le CSV.

Oui, ce n'est pas ce que j'ai écrit ?

> Aussi, il faudrait, sur la validation du formulaire, effectuer un dry-run en premier, pour vérifier que pas d'erreur d'import, avant de commencer à créer séquentiellement les usagers.

with transaction.atomic() suffit pour cela.

> Une autre chose : je ne vois pas encore une façon propre de générer un CSV d'exemple.

Tu peux simplement générer la ligne d'entête.

> Je pensais avoir un jeu de données d'exemples pour chaque type d'attribut (attribute_kind) par défaut (et donc en dur dans le code), et pour les types d'attributs définis par la suite, essayer de générer des données d'exemple en fonction du field_class.

Oui ça concerne les 3/4 types de champs qui attendent un format spécifique on peut attendre attribute_kind pour cela.

> Mais ça ne couvre pas tous les cas, et on pourrait se retrouver avec des attributs pour lesquels on ne sait pas générer des données d'exemple.

Pour string, qui représente souvent plus de 50% des champs, c'est libre donc pas d'exemple.

> Une dernière difficulté encore (et en lien avec ce qui est écrit plus haut à propos du fichier d'exemple) : est-ce qu'on cherche à gérer ici les attributs multiples ?

Non ils ne fonctionnent pas dans authentic de toute façon (je l'ai déjà dit sur les derniers tickets qui concernaient les attributs mais ça n'a pas sembler porter ses fruits :)).

> En parallèle il faudra abandonner Collectivité et revenir à unité d'organisation, ça a assez durer cette histoire.

> Là c'est plutôt un ticket de traduction, non ?

C'est du troll de ma part sur un ticket qui n'a aucun rapport, continuons sur le sujet.

#10 - 07 mai 2019 11:57 - Benjamin Dauvergne

Thomas Noël a écrit :

> Faut faire un truc compatible avec l'export, sinon ça sera incompréhensible, àmha.

On peut au passage corriger l'export pour qu'il corresponde à notre import bien sûr, ne nous sentons pas limiter par l'export actuel s'il nous gêne.

#11 - 07 mai 2019 12:06 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Autre chose qui m'irait bien (et qui est présente dans pas mal d'UI d'import de csv), c'est la possibilité de mentionner que la première ligne contient la première entrée, et non pas l'intitulé des colonnes.

Non non il faut absolument une ligne d'entête et valider que toutes les colonnes sont comprises, sinon c'est la chienlit.

> La possibilité de choisir le délimiteur me plairait bien aussi.

Ça m'irait qu'on aille pas plus loin que ce qu'on fait dans csvdatasource voir moins loin sachant qu'on a qu'un seul usager pour l'instant pour cet import (le Nord); on pourra toujours proposer ODT, XLS, etc.. plus tard voir factoriser tout ça entre nos différents logiciels à un moment.

#12 - 07 mai 2019 13:44 - Thomas Noël

Je note aussi un truc ici : on parle d'une interface, mais ce qu'il faut c'est d'abord le code (la fonction) qui gère l'import, et puisse être appelée via cette interface mais aussi en ligne de commande

#13 - 07 mai 2019 14:43 - Benjamin Dauvergne

Thomas Noël a écrit :

> Je note aussi un truc ici : on parle d'une interface, mais ce qu'il faut c'est d'abord le code (la fonction) qui gère l'import, et puisse être appelée via cette interface mais aussi en ligne de commande

Oui comme pour l'import de site c'est peut-être plus simple de bosser sur/tester le code d'import lui même (utilisable en ligne de commande donc) avant de créer l'interface, c'est pas pour repousser la livraison juste que ça me semble plus facile d'aller dans ce sens; l'interface suivra facilement ensuite.

#14 - 07 mai 2019 17:02 - Paul Marillonnet

- Fichier 0001-WIP-import-users-from-csv-file-32833.patch ajouté

- Statut changé de Nouveau à En cours

J'ai écrit un début de fonction d'import dans authentic2.data_transfer, je vais écrire la partie CLI correspondante.

Un patch wip pour donner une idée de ce que j'ai commencé à faire.
J'avais déjà fait un petit bout d'UI ce matin.

Je vais écrire des tests.

#15 - 09 mai 2019 16:34 - Paul Marillonnet

- Fichier 0001-import-users-from-csv-file-32833.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

Voilà une première version à proposer, sans la génération d'un csv d'exemple pour l'instant.
Si à la relecture on me demande aussi des tests sur la vue, je le ferai.

Je propose de faire la version CLI dans un autre ticket, car ça ne correspond pas au besoin exprimé ici.

#16 - 09 mai 2019 16:41 - Frédéric Péters

> Si à la relecture on me demande aussi des tests sur la vue, je le ferai.

Il faut des tests sur tout.

#17 - 09 mai 2019 16:42 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

Au revoir paresse.

#18 - 10 mai 2019 15:21 - Paul Marillonnet

- Fichier 0001-import-users-from-csv-file-32833.patch ajouté

- Statut changé de En cours à Solution proposée

Voilà, en ajoutant un test de la vue associée à la fonction d'import.

#19 - 10 mai 2019 15:27 - Paul Marillonnet

- Fichier 0001-import-users-from-csv-file-32833.patch ajouté

Flagrant délit de copier-coller. Corrigé ici quelques lignes superflues dans le gabarit de la vue.

#20 - 10 mai 2019 16:54 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

(Après une discussion de vive voix avec Thomas sur le ticket) Je relis maintenant la description et comprends que Frédéric attend que le code fasse de la synchro, en plus de l'import.
On est reparti pour un tour.

#21 - 10 mai 2019 17:26 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> (Après une discussion de vive voix avec Thomas sur le ticket) Je relis maintenant la description et comprends que Frédéric attend que le code fasse de la synchro, en plus de l'import.
> On est reparti pour un tour.

Oui en général c'est bien quand les actions d'import sont idempotentes (de la synchro donc).

#22 - 10 mai 2019 17:27 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

> Paul Marillonnet a écrit :
>
> > (Après une discussion de vive voix avec Thomas sur le ticket) Je relis maintenant la description et comprends que Frédéric attend que le code fasse de la synchro, en plus de l'import.
> > On est reparti pour un tour.
>
> Oui en général c'est bien quand les actions d'import sont idempotentes (de la synchro donc).

La seule décision à prendre c'est de savoir si on met à jour une fiche existante ou si on l'ignore (je trouve plus propre d'ignorer, en général ça apporte moins de surprises, ou en tout cas des surprises moins graves que d'écraser des données modifiées laborieusement à la main).

#23 - 11 mai 2019 10:38 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> La seule décision à prendre c'est de savoir si on met à jour une fiche existante ou si on l'ignore (je trouve plus propre d'ignorer, en général ça apporte moins de surprises, ou en tout cas des surprises moins graves que d'écraser des données modifiées laborieusement à la main).

Ok oui faut voir ce qu'attend Frédéric.

Bon, en plus, mon code ne semble pas compatible django1.8, version dans laquelle la vue de formulaire ne s'affiche carrément pas. Je regarde d'où ça vient.

#24 - 11 mai 2019 11:20 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Benjamin Dauvergne a écrit :
>
> > La seule décision à prendre c'est de savoir si on met à jour une fiche existante ou si on l'ignore (je trouve plus propre d'ignorer, en général ça apporte moins de surprises, ou en tout cas des surprises moins graves que d'écraser des données modifiées laborieusement à la main).
>
> Ok oui faut voir ce qu'attend Frédéric.
>
> Bon, en plus, mon code ne semble pas compatible django1.8, version dans laquelle la vue de formulaire ne s'affiche carrément pas. Je regarde d'où ça vient.

Ce n'est pas grave, ça me va si tu marques le test django1.11 only, on peut commencer à écrire du code de ce genre pour de nouvelles fonctionnalités (juste si ça peut éviter de faire des traces pour rien sous Django 1.8 c'est mieux), la deadline étant proche pour l'abandon de Django 1.8, faudra juste garder des versions "legacy" dans un coin pour les migrations des clients restants dans des coins (formiris, les echos, vinci).

#25 - 11 mai 2019 11:41 - Frédéric Péters

> La seule décision à prendre c'est de savoir si on met à jour une fiche existante ou si on l'ignore (je trouve plus propre d'ignorer, en général ça apporte moins de surprises, ou en tout cas des surprises moins graves que d'écraser des données modifiées laborieusement à la main).

Ok oui faut voir ce qu'attend Frédéric.

(à noter que je suis surtout le messager).

Je serais pour une option, parce que les deux situations doivent pouvoir se rencontrer, genre une case [ ] écraser les utilisateurs existants (sans être fan du libellé, qui peut laisser penser qu'il y aurait suppression de tous les utilisateurs, pas uniquement ceux du fichier), mais à défaut, comme dit Benjamin, mieux vaut ne pas écraser, et si en fin d'opération on peut annoncer à l'admin que n lignes ont été ignorées, c'est très bien.

#26 - 14 mai 2019 16:41 - Paul Marillonnet

- Fichier 0001-import-users-from-csv-file-32833.patch ajouté

- Statut changé de En cours à Solution proposée

Voilà, une version qui tient compte de vos remarques.
Thomas me fait remarquer qu'un setting pour le nom de la colonne portant l'identifiant applicatif n'est pas forcément une bonne idée. Avec des colonnes nommées et non nécessairement ordonnées je ne vois pas trop comment faire autrement.

#27 - 19 mai 2019 21:18 - Thomas Noël

Chose à étudier qui va être demandée (l'est déjà en filigrane d'un autre projet) : que l'import génère des comptes qui seront fédérés lors d'un SSO à travers OIDC, SAML voire CAS (c-à-d quand Authentic est proxy d'un autre IdP). Je ne sais pas trop les pré-requis pour cette situation, mais ça serait pertinent de vérifier que l'import CSV est ok dans cette situation. Je pense que le cas a déjà été étudié pour la gestion proxy LemonLDAP avec import LDAP préalable.

#28 - 19 mai 2019 23:34 - Benjamin Dauvergne

Thomas Noël a écrit :

> Chose à étudier qui va être demandée (l'est déjà en filigrane d'un autre projet) : que l'import génère des comptes qui seront fédérés lors d'un SSO à travers OIDC, SAML voire CAS (c-à-d quand Authentic est proxy d'un autre IdP). Je ne sais pas trop les pré-requis pour cette situation, mais ça serait pertinent de vérifier que l'import CSV est ok dans cette situation. Je pense que le cas a déjà été étudié pour la gestion proxy LemonLDAP avec import LDAP préalable.

On déjà eu ce cas ?

#32 - 24 mai 2019 10:33 - Paul Marillonnet

Jusque là je pensais implémenter une fonction d'import qui rende l'identifiant applicatif obligatoire, c'est-à-dire qu'à chaque usager importé soit associé un UserExternalId qui rende compte de cet import et qui mentionne l'identifiant applicatif présent dans le csv.
Après réflexion je me dis que c'est peut-être une contrainte inutile ou injustifiée, et qu'on ne peut pas demander aux agents de fournir un csv dont nécessairement une colonne mentionne des identifiants applicatifs -- et qu'on peut aussi avec des cas d'usage de cette fonction d'import qui ne

nécessite pas d'identification par clé unique alternative sur les usagers.

TL;DR: Par rapport aux patchs présentés jusque là dans ce ticket, je veux rendre la colonne d'identifiant externe optionnelle, et non plus obligatoire.

#33 - 24 mai 2019 11:50 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Jusque là je pensais implémenter une fonction d'import qui rende l'identifiant applicatif obligatoire, c'est-à-dire qu'à chaque usager importé soit associé un UserExternalId qui rende compte de cet import et qui mentionne l'identifiant applicatif présent dans le csv.

Moi ça me va d'ajouter un source_domain et source_id text, nullable et unique dans un coin du modèle User pour abandonner ça, et prévoir au niveau de l'import csv deux colonnes obligatoires, source_domain/app_id et external_id. C'est la même idée que coté chrono pour l'import des exceptions que je pousse depuis un bail.

> Après réflexion je me dis que c'est peut-être une contrainte inutile ou injustifiée, et qu'on ne peut pas demander aux agents de fournir un csv dont nécessairement une colonne mentionne des identifiants applicatifs -- et qu'on peut aussi avec des cas d'usage de cette fonction d'import qui ne nécessite pas d'identification par clé unique alternative sur les usagers.

Sisi c'est très bien, ça les oblige à réfléchir, ça peut aussi être un bête uid ldap, l'important c'est le source_domain, le source_id osef ça doit juste être unique pour le domaine.

> TL;DR: Par rapport aux patchs présentés jusque là dans ce ticket, je veux rendre la colonne d'identifiant externe optionnelle, et non plus obligatoire.

Non, c'est important, sinon aucun import incrémental n'est possible (il faut penser suppression/mise-à-jour aussi sinon ça nous reviendra dans la figure comme chaque fois qu'on propose des imports), ça me pousserait presque à écrire un HowDoWeDoImport.

À noter qu'on peut aussi prévoir que le source_domain soit un paramètre de l'import, comme l'ou par défaut.

Mon interface idéale ce serait ça :

1er écran :

```
Votre fichier d'import [ Choisir un fichier ]

<a>Fichier d'exemple en CSV</a>
<a>Fichier d'exemple en ODT</a>
```

2ème écran , configuration :

```
145 lignes dans le fichier d'import

Collectivité par défaut [-----]
Colonne pour collectivié [ ou_slug ] <- rempli si colonne correspondante

Source par défaut [.........]
Colonne pour source [ source_domain ] <- rempli si colonne correspondante
Colonne pour identifiant external par défaut [ external_id ] <- idem

Colonnes:
* Prénom [----]
* Nom [----]
* Email
etc...

<Button> Simuler l'import </Button>
```

3ème écran, simulation :

```
100 utilisateurs mis à jour
45 utilisateurs créés

Répartition par collectivité :
* Col1 : 1 crée 10 mis à jour
...

Répartition par source :
* source_domain1 : 10 créés, 5 mis à jour

Attributs mis à jour :
```

```
* email
* prénom
etc...

<button>Finaliser l'import</button>
```

4ème écran, c'est fini :

```
Import réussi :

<même résumé qu'en simu>

[ Retour ]
```

#34 - 24 mai 2019 12:03 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Moi ça me va d'ajouter un source_domain et source_id text, nullable et unique dans un coin du modèle User pour abandonner ça,

Dans ce cas il faut créer un ticket pour déprécier le modèle UserExternalId, parce que la coexistence de ce dernier et de ce que tu proposes ci-dessus va poser des problèmes.

> et prévoir au niveau de l'import csv deux colonnes obligatoires, source_domain/app_id et external_id. C'est la même idée que coté chrono pour l'import des exceptions que je pousse depuis un bail.

Ok. Il y a du code/des tickets côté chrono qui illustrent/relatent ce dont tu parles ici ?

> Sisi c'est très bien, ça les oblige à réfléchir, ça peut aussi être un bête uid ldap, l'important c'est le source_domain, le source_id osef ça doit juste être unique pour le domaine.

Ok, très bien.

> À noter qu'on peut aussi prévoir que le source_domain soit un paramètre de l'import, comme l'ou par défaut.

Ok.

> Mon interface idéale ce serait ça :

D'ac, c'est chouette, je vais partir là-dessus.

#35 - 24 mai 2019 13:00 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Benjamin Dauvergne a écrit :
>
> > Moi ça me va d'ajouter un source_domain et source_id text, nullable et unique dans un coin du modèle User pour abandonner ça,
>
> Dans ce cas il faut créer un ticket pour déprécier le modèle UserExternalId, parce que la coexistence de ce dernier et de ce que tu proposes ci-dessus va poser des problèmes.

Lesquels ? Ce n'est utilisé que par LDAP, et l'import CSV n'est pas prévu pour inter-opérer avec LDAP.

> > et prévoir au niveau de l'import csv deux colonnes obligatoires, source_domain/app_id et external_id. C'est la même idée que coté chrono pour l'import des exceptions que je pousse depuis un bail.
>
> Ok. Il y a du code/des tickets côté chrono qui illustrent/relatent ce dont tu parles ici ?

Et moi qui pensait que tout le monde connaissait tous les tickets, #29209 et tous ceux liés. C'est surtout dans l'esprit que c'est la même chose, l'implémentation est particulière parce qu'on parle d'.ics et pas de CSV.

#36 - 27 mai 2019 08:58 - Paul Marillonnet

Benjamin Dauvergne a écrit :

> Lesquels ? Ce n'est utilisé que par LDAP, et l'import CSV n'est pas prévu pour inter-opérer avec LDAP.

Oui je comprends ce que tu veux dire, mais c'est l'idée d'introduire de la redondance dans le code, pour des usages similaires, qui me gêne un peu. Et aussi que, le modèle UserExternalId n'étant pas défini dans le backend LDAP mais dans le fichier principal de modèles, on peut y voir à la lecture du code un souhait de généricisation/factorisation des usages pour ce modèle.

> Et moi qui pensait que tout le monde connaissait tous les tickets, #29209 et tous ceux liés. C'est surtout dans l'esprit que c'est la même chose, l'implémentation est particulière parce qu'on parle d'.ics et pas de CSV.

Merci.

#37 - 27 mai 2019 08:58 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

#38 - 27 mai 2019 10:54 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Oui je comprends ce que tu veux dire, mais c'est l'idée d'introduire de la redondance dans le code, pour des usages similaires, qui me gêne un peu. Et aussi que, le modèle UserExternalId n'étant pas défini dans le backend LDAP mais dans le fichier principal de modèles, on peut y voir à la lecture du code un souhait de généricisation/factorisation des usages pour ce modèle.

Le souci c'est l'index d'unicité sur UserExternalId, il faudrait en ajouter deux :

- (user, source) : un utilisateur ne peut avoir qu'un identifiant externe par source
- (source, external_id) : un identifiant externe d'une même source ne peut correspondre qu'à un seul utilisateur

Si on fait ça ce serait ok pour repartir du même modèle et transférer d'autres utilisateurs dessus (auth_oidc, auth_fc, etc..).

Et dans ce cas on pourra rendre l'import CSV compatible avec la connexion LDAP, il suffit d'avoir le même nom de source et les mêmes identifiants externes.

#39 - 06 juin 2019 14:28 - Paul Marillonnet

- Fichier 0001-manager-users-csv-import.patch ajouté

- Statut changé de En cours à Solution proposée

Voilà ce que je souhaite proposer à la relecture suite à nos discussions.

Quelques remarques :

- L'algo d'import, dans data_transfer, reste à peu de choses près le même que dans les précédents patches.
- J'ai fait une UI en 5 pages et non en 4 telle que le proposait Benj, car il me semble important de séparer la config des colonnes relatives aux informations d'organisation (ou, source), de celles des attributs * du profil utilisateur (attributs de base et attributs de profil étendu).
- Le formulaire de config des colonnes d'attributs se dote de nouveau champs en fonction des attributs de profil étendu.

#40 - 06 juin 2019 14:32 - Benjamin Dauvergne

Supprime ta branche wip/* inutile (qu'on ait pas deux résultats de build ici).

#41 - 06 juin 2019 14:34 - Paul Marillonnet

Oups oui, pardon, c'est fait.

#42 - 06 juin 2019 14:49 - Paul Marillonnet

(Les contraintes nouvellement introduites sur l'unicité des champs dans le modèle d'identifiant externe (UserExternalId) font exploser les tests LDAP. Je regarde ça.)

#43 - 06 juin 2019 15:04 - Benjamin Dauvergne

Je commence à relire en vrac.

- get_ou : ne pas faire de logger.error dans un truc qui sera appelé depuis une UI, si erreur elle doit être rapporté à l'écran pas nous spamer de mails (chaque logger.error ou exception, c'est un mail qui part), c'est vrai pour tous les logger.error()

- get_ou : on ne peut pas ignorer les lignes où on ne trouve la colonne ou, ça doit être bloquant (en renvoyant None par exemple géré plus haut), le fallback c'est uniquement si rien n'est indiqué, i.e. pas de colonne ou colonne vide (utiliser .strip() sur le contenu de la colonne avant son usage)

Lors de la création d'un utiliser hasattr(user, field) n'est pas suffisant, il faut faire un User._meta.get_field(field) pour vérifier que le champs existe vraiment sur le modèle (ou alors mettre une liste explicite, 'first_name, last_name, username, email, is_staff, is_superuser, etc..').

Il y aura des considérations de permissions à avoir, par exemple je n'ai pas l'impression que ça prenne en compte le droit de créer des utilisateurs dans une seule OU. Si tu veux simplifier tu peux n'autoriser cette vue qu'aux superadmin(s) pour l'instant.

Globalement pour les erreurs tu dois les accumuler durant le chargement dans une liste avec une colonne indiquant la ligne du CSV et pouvoir restituer un résultat, le mode par défaut c'est que toute erreur provoque l'annulation de l'import comme un dry_run. On doit pouvoir restituer un écran du genre :

```
ligne 1: identifiant ou inconnu : xxx
ligne 1: email mal formaté : yyyy
etc..
```

Le templatetag "get_item" : pour prendre un truc dans un dictionnaire dans un template c'est dict.truc.

La communication via la session n'est pas une bonne idée, il vaudrait mieux un modèle et stocker tout le CSV en session, comment dire...

Il faut que tu crée un modèle pour matérialiser tes imports avec différent statuts, "uploaded", "base_configured", "attributes_configured", "dry_run", "runned", idéalement ça créerait un répertoire media/csv-imports/<id-du-modèle>/ pour y stocker le CSV, les résultats des dry_run, etc.. dans un champ JSON tu devrais stocker la configuration, des informations sur les dry_run ou les imports effectués.

On devrait pouvoir reprendre la configuration d'un import avec un nouveau CSV.

Il te faudra une page de garde qui liste les imports déjà effectués, quand on clique sur un import on va sur la vue concernant ce statut, soit un formulaire de configuration si ça n'a pas déjà eu lieu, soit une page récapitulant la configuration, le réstulats des derniers dry_run, etc..

Une fois qu'un import a été effectué on ne devrait plus pouvoir l'exécuter mais juste le cloner.

Le modèle doit avoir une date de création et de modification pour nettoyer au bout d'un certain temps.

Voilà c'est un peu fouilli mais comme tu as avancé sur tout en même temps, j'aurai pensé que tu ferais un truc en ligne de commande avant de faire l'interface.

#44 - 06 juin 2019 15:05 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> (Les contraintes nouvellement introduites sur l'unicité des champs dans le modèle d'identifiant externe (UserExternalId) font exploser les tests LDAP. Je regarde ça.)

Igore ça pour l'instant je m'en occuperai ça ne change rien à ton code, ça le rend juste plus fragile, mais ça ne se verra pas dans des tests.

#45 - 15 juin 2019 17:28 - Paul Marillonnet

Créé #34022 pour ce qui est du clonage des objets d'import.

#46 - 17 juin 2019 01:13 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#47 - 17 juin 2019 01:13 - Benjamin Dauvergne

- Fichier 0003-manager-add-user-import-views-32833.patch ajouté

- Fichier 0001-custom_user-rename-clean_fields-to-validate_unique-3.patch ajouté

- Fichier 0002-add-CSV-import-framework-32833.patch ajouté

Voilà en simplifié, la configuration se fait entièrement au niveau de l'entête du CSV on met, par exemple :

```
email key,first_name,last_name,username unique
tnoel@entrouvert.com,Thomas,Nono,tnoel
fpeter@entrouvert.com,Frédéric,Péters,tnoel
jojo,John,Doe,john.doe
```

Dans chaque entête le premier mot est le nom de l'attribut, ensuite des modificateurs pour indiquer comme gérer les valeurs, est-ce une clé ? doit-elle être unique ? mise à jour ? uniquement utilisée pour une création ?

Par défaut toutes les colonnes sont utilisés en création et en mise à jour.

L'ou est par défaut à sélectionner dans l'interface, pas d'importe multi-ou.

Les imports sont effectués en tâche de fond dans des threads, l'état est maintenu dans des fichiers pickle dans le répertoire media.

Il me reste à proposer un formulaire pour créer un fichier d'exemple en choisissant quelques colonnes.

Les valeurs sont validés via le formulaire d'édition d'un utilisateur classique, les simulations sont possibles et un rapport HTML est généré dans tous les cas.

#48 - 17 juin 2019 01:15 - Benjamin Dauvergne

- Fichier screencast.mp4 ajouté

Et un screencast.

#49 - 17 juin 2019 01:21 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

> Et un screencast.

Ça ne gère pas userexternalid mais on peut l'ajouter sans trop de soucis.

Dans les autres goodies il y aussi avoir des contraintes d'unicité sur plusieurs colonnes (typiquement nom/prénom), juste pour chercher des doublons

#50 - 17 juin 2019 07:59 - Pierre Cros

Merci pour le boulot et le screencast.

L'import se fait nécessairement sur une OU unique (on aurait pu imaginer avec une OU sur chaque ligne dans le CSV) mais pour mon cas d'usage c'est pas un soucis.

En revanche l'histoire de la simulation, les différentes couleurs et les légendes, c'est peu adapté à mon public au Nord qui est déjà en difficulté pour encoder un fichier en UTF-8. Bref on fera avec mais ce qui les intéresse c'est que "ça marche" avec un format donné et c'est tout. Si ça ne marche pas je pense qu'ils nous demanderont pourquoi, quels que soient les messages qu'on leur affiche.

#51 - 17 juin 2019 09:23 - Benjamin Dauvergne

Pierre Cros a écrit :

> En revanche l'histoire de la simulation, les différentes couleurs et les légendes, c'est peu adapté à mon public au Nord qui est déjà en difficulté pour encoder un fichier en UTF-8. Bref on fera avec mais ce qui les intéresse c'est que "ça marche" avec un format donné et c'est tout. Si ça ne marche pas je pense qu'ils nous demanderont pourquoi, quels que soient les messages qu'on leur affiche.

Ça ne marchera jamais avec un format donné, il y a des contraintes forcément sur les données elles mêmes ne serait-ce que pour éviter les doublons. Ce que je fais justement plutôt que de tout bloquer c'est de signaler les lignes qui posent souci.

L'écran de simulation c'est justement pour nous, si ça ne marche pas et qu'on a pas ça on ne saura pas non plus. Concernant l'UTF-8 le plus simple c'est de ne pas faire de CSV de viser directement .ODS qui évitera tout souci de ce genre, je peux le faire tout de suite.

#52 - 17 juin 2019 09:41 - Benjamin Dauvergne

Petit souci sur le module csv, il a apparemment changé entre python 2.7.13 et 2.7.15, en 2.7.13 le sniffer marche moins bien.

#53 - 17 juin 2019 09:50 - Benjamin Dauvergne

- Fichier 0003-manager-add-user-import-views-32833.patch ajouté

- Fichier 0001-custom_user-rename-clean_fields-to-validate_unique-3.patch ajouté

- Fichier 0002-add-CSV-import-framework-32833.patch ajouté

#54 - 17 juin 2019 10:04 - Pierre Cros

Quand je parlais d'un format donné, je parlais d'un format donné par nous. C'est ce que j'ai expliqué au Nord. Si Iodas produit pas le bon format c'est à eux de faire le boulot d'adaptation.

#55 - 17 juin 2019 10:07 - Paul Marillonnet

Je tiens quand même à montrer ma copie, dont la conclusion m'a pris une bonne partie du weekend :

https://perso.entrouvert.org/~pmarillonnet/32833/screencast.ogv

Les seuls tests en échec sont une bête régression : un ancien test faisait un clic sur le lien d'export csv, et le clic se fait maintenant sur le lien d'import car les deux libellés portent un nom proche.

#56 - 17 juin 2019 10:17 - Benjamin Dauvergne

Pierre Cros a écrit :

> Quand je parlais d'un format donné, je parlais d'un format donné par nous. C'est ce que j'ai expliqué au Nord. Si Iodas produit pas le bon format c'est à eux de faire le boulot d'adaptation.

Tu joues sur les mots, le format il est bien donné par nous, mais c'est mieux si on peut leur dire où ils ne l'ont pas respecté quand même.

#57 - 17 juin 2019 10:26 - Pierre Cros

Je joue pas sur les mots, il me semblait que tu n'avais pas compris et je précisais.

Benj à écrit :

> Ça ne marchera jamais avec un format donné

Je voulais pointer que ça marcherait toujours avec le format qu'on leur demande de respecter.

@Paul
Ce que je vois sur ta vidéo semble faire le job. Je suis malheureusement pas capable de discuter technique avec vous deux mais ça me semble utile que Benj te dise pourquoi il a refait le truc.

#58 - 17 juin 2019 10:48 - Benjamin Dauvergne

Pierre Cros a écrit :

> Je voulais pointer que ça marcherait toujours avec le format qu'on leur demande de respecter.

Mais si ils nous filent deux fois le même email on fait quoi ? Ou tu inclues les contraintes entre les lignes et entre les lignes et la base existante dans ta définition du mot "format" ?

J'ai aucun souci à bloquer complètement un import si il y a la moindre ligne en erreur.

#59 - 17 juin 2019 10:50 - Benjamin Dauvergne

- Fichier Firefox_Screenshot_2019-06-17T08-45-28.285Z.png ajouté

Avec un petit résumé au début des rapports.

#60 - 17 juin 2019 12:27 - Benjamin Dauvergne

- Fichier 0002-custom_user-rename-clean_fields-to-validate_unique-3.patch ajouté

- Fichier 0001-add-unique-constraint-to-UserExternalId.patch ajouté

- Fichier 0003-add-csv-import-framework-32833.patch ajouté

- Fichier 0004-manager-add-user-import-views-32833.patch ajouté

Avec support optionnel de UserExternalId via deux colonnes _source_name et _source_id.

Il me reste à ajouter des validations d'unicité autoréférentiel (ne pas avoir une ligne qui met à jour une ligne précédente).

#61 - 17 juin 2019 12:56 - Pierre Cros

Benjamin Dauvergne a écrit :

> Mais si ils nous filent deux fois le même email on fait quoi ? Ou tu inclues les contraintes entre les lignes et entre les lignes et la base existante dans ta définition du mot "format" ?

Je ne me suis (presque) pas posé cette question et toutes les réponses me vont (à notre que l'email est pas obligatoire pour mon cas d'usage, il y a bcp de comptes créés sans email, rapport à la cible - les allocataires RSA). Le Nord souhaitait effectivement une détection des doublons mais je n'ai rien vendu en la matière, c'était trop peu spécifié, nous faisons donc comme bon nous semble.

> J'ai aucun souci à bloquer complètement un import si il y a la moindre ligne en erreur.

Laissons ça comme tu as fait, c'est moins rêche, sûr. Et une fois qu'on leur aura appris à comprendre les erreurs, on peut espérer que les clients seront autonomes.

#62 - 18 juin 2019 16:25 - Benjamin Dauvergne

Exemple de fichier pour import depuis un logiciel métier qu'on nommera Tapas, sans chercher à dédoublonner le moins du monde avec des comptes existants en se basant sur l'email ou un téléphone (dans la colonne @source_id on trouvera l'ID en base métier du logiciel Tapas).

```
@source_name,@source_id,first_name,last_name,phone_number,mobile,email
Tapas,1,"Michel Alphonse","RAMUNDO","0344454545","0640343434","michel.ramundo@gmail.com"
Tapas,2,"Josiane","Schmidt","0344454511","0640311434","josiane.schmidt@live.com"
```

#63 - 18 juin 2019 16:37 - Paul Marillonnet

Si la colonne source_id est l'identifiant applicatif de l'usager dans l'appli métier Tapas, alors on va dans le sens de l'unicité du doublet (source_name, source_id), unicité qui n'est pas respectée dans le csv d'exemple que tu fournis. Mais je me trompe peut-être dans mon interprétation.

#64 - 18 juin 2019 16:44 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Si la colonne source_id est l'identifiant applicatif de l'usager dans l'appli métier Tapas, alors on va dans le sens de l'unicité du doublet (source_name, source_id), unicité qui n'est pas respectée dans le csv d'exemple que tu fournis. Mais je me trompe peut-être dans mon interprétation.

J'ai corrigé.

#65 - 18 juin 2019 20:52 - Benjamin Dauvergne

- Fichier 0005-check-self-referene-in-keys-to-rebase.patch ajouté

- Fichier 0001-add-unique-constraint-to-UserExternalId-32833.patch ajouté

- Fichier 0002-custom_user-rename-clean_fields-to-validate_unique-3.patch ajouté

- Fichier 0003-add-csv-import-framework-32833.patch ajouté

- Fichier 0004-manager-add-user-import-views-32833.patch ajouté

Avec check des références à une même clé dans un même import, ça indique la ligne où la clé est apparue la première fois.

#67 - 20 juin 2019 17:45 - Paul Marillonnet

La première moitié de ma relecture :

- Le champ UserExternalId : on avait parlé de l'ajout de l'unicité ('external_id', 'source'), et mais aussi de ('user', 'source'), qui me paraît nécessaire.
- La classe CsvImport : la fonction imbriquée detect_encoding laisse le choix à l'appeler de l'encodage utilisé. Je trouve qu'elle porte mal son nom et qu'on devrait l'appeler set_encoding, toujours avec cette option 'detect' possible.
  Je pense que la signature de la méthode appelante run devrait porter un encoding='detect' par défaut. Même remarque pour la méthode UserCsvImport.run.
- Juste pour info, la doc de python-attr mentionne qu'il est possible d'utiliser les alias attr.attrib et attr.attrs à la place de attr.ib et attr.s.
  Du détail, mais ça peut éviter d'aller chercher dans la doc ce que ces attributs ib et s peuvent bien vouloir dire...
  À toi de voir si tu laisses comme ça ou si tu utilises les alias plus explicites.
- La méthode Error.__eq__ appelle un self.as_error sorti de nulle part. Et le nom de la classe dans l'instruction super de la dernière ligne de cette méthode est erroné. Je pense, en lisant la déclaration de LineError.__eq__ plus bas, que la définition de Error.__eq__ n'est pas nécessaire. Je me plante ?
- Question du débutant, sur le code de classe UserCsvImport : pourquoi les # NOQA: F841 sur des variables assignées mais pas utilisées, alors que pour taire les warnings PEP8 il suffirait de retirer la lvalue (field = ; attributes =) sans que ça change quoi que ce soit à la levée d'erreur dans ces bouts de code?
- Tu définis un SOURCE_COLUMNS = set([SOURCE_NAME, SOURCE_ID]) que tu pourrais utiliser plus bas :

```
            if header.name in (SOURCE_NAME, SOURCE_ID):
```

- Plus loin dans le code de UserCsvImport.parse_header tu instancies une LineError (self.add_error(LineError(...))) sans mentionner la ligne (attribut LineError.line) ni la colonne (attribut LineError.column). C'est dommage (autant utiliser simplement une Error si on ne souhaite pas logguer ces infos).
- Fichier user_import.py : pourquoi mets-tu le préfixe (constant PREFIX) dans la classe Reports et non pas dans la classe Report ?
- Pourquoi l'imbrication exception = repr(repr(e)) dans Report.run ?

#68 - 21 juin 2019 10:40 - Paul Marillonnet

La suite :

- Je pense qu'il manque un attribut

```
    permissions = ['custom_user.admin_user']
```

  dans la définition de la classe UserImportsForm.

- une typo dans le 5e message de commit (s/referene/reference/)

#69 - 21 juin 2019 11:35 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> La première moitié de ma relecture :
>
> - Le champ UserExternalId : on avait parlé de l'ajout de l'unicité ('external_id', 'source'), et mais aussi de ('user', 'source'), qui me paraît nécessaire.

Oui mais finalement ça ne me parait pas une bonne idée, on pourra très bien avoir un utilisateur issue de deux sources qu'on souhaite réconciliées et ça ne pose pas de problème pour l'import.

> - La classe CsvImport : la fonction imbriquée detect_encoding laisse le choix à l'appeler de l'encodage utilisé. Je trouve qu'elle porte mal son nom et qu'on devrait l'appeler set_encoding, toujours avec cette option 'detect' possible.

En fait detect marche assez mal, en tout cas dans mes tests ça ne marche pas du tout parce que je n'ai pas du contenu réalise (il faut assez de caractères hors ASCII pour qu'il déduise un biais statistique suffisant pour faire un choix).

Renommage effectué.

> Je pense que la signature de la méthode appelante run devrait porter un encoding='detect' par défaut. Même remarque pour la méthode UserCsvImport.run.

Comme ça marche mal, je ne préfère pas.

> - Juste pour info, la doc de python-attr mentionne qu'il est possible d'utiliser les alias attr.attrib et attr.attrs à la place de attr.ib et attr.s.
> Du détail, mais ça peut éviter d'aller chercher dans la doc ce que ces attributs ib et s peuvent bien vouloir dire...
> À toi de voir si tu laisses comme ça ou si tu utilises les alias plus explicites.

Comme toute leur doc utilise les noms cours, le mieux c'est d'utiliser cela.

> - La méthode Error.__eq__ appelle un self.as_error sorti de nulle part. Et le nom de la classe dans l'instruction super de la dernière ligne de cette méthode est erroné. Je pense, en lisant la déclaration de LineError.__eq__ plus bas, que la définition de Error.__eq__ n'est pas nécessaire. Je me plante ?

Vraisemblablement, j'ai viré ça, les tests continuent à passer.

> - Question du débutant, sur le code de classe UserCsvImport : pourquoi les # NOQA: F841 sur des variables assignées mais pas utilisées, alors que pour taire les warnings PEP8 il suffirait de retirer la lvalue (field = ; attributes =) sans que ça change quoi que ce soit à la levée d'erreur dans ces bouts de code?

Reste de code avant j'utilisais field dans une branche d'un if, flake n'est pas capable de déduire que certains chemins sont impossibles d'où NOQA, lvalue virée.

> - Tu définis un SOURCE_COLUMNS = set([SOURCE_NAME, SOURCE_ID]) que tu pourrais utiliser plus bas :
> [...]

Ok.

> - Plus loin dans le code de UserCsvImport.parse_header tu instancies une LineError (self.add_error(LineError(...))) sans mentionner la ligne (attribut LineError.line) ni la colonne (attribut LineError.column). C'est dommage (autant utiliser simplement une Error si on ne souhaite pas logguer ces infos).

Ajouté un line=1, comme ça concerne toute la ligne je garde le column=0 implicite.

> - Fichier user_import.py : pourquoi mets-tu le préfixe (constant PREFIX) dans la classe Reports et non pas dans la classe Report ?

Ça me parait plus logique, c'est Reports qui sait comment trouver un Report.

- Pourquoi l'imbrication exception = repr(repr(e)) dans Report.run ?

C'est au cas où six.text_type foire, on est pas certain que repr(e) renvoie de l'ASCII, j'aurai pu rajouter encore un niveau de try/except mais ça devrait juste être un utils.exception_to_text() comme dans passerelle.

#70 - 21 juin 2019 11:58 - Frédéric Péters

authentic a son propre chargement de jquery et celui venant de gadjo est annulé ({% block gadjo-js }{ endblock %}, #25045); le résultat est que sur cette page d'import, jquery n'est pas rendu disponible, et ainsi échec à afficher le menu Pubkik.

#71 - 21 juin 2019 13:13 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> La suite :
>
> - Je pense qu'il manque un attribut
> [...]
> dans la définition de la classe UserImportsForm.

Ajouté.

- une typo dans le 5e message de commit (s/referene/reference/)

Ce n'est pas un vrai message de commit; quand j'écris "(to rebase)" ça veut dire que ce sera mergé plus tard dans un autre commit.

#72 - 21 juin 2019 13:33 - Benjamin Dauvergne

- Fichier 0005-check-self-referene-in-keys-to-rebase.patch ajouté

- Fichier 0001-add-unique-constraint-to-UserExternalId-32833.patch ajouté

- Fichier 0002-custom_user-rename-clean_fields-to-validate_unique-3.patch ajouté

- Fichier 0003-add-csv-import-framework-32833.patch ajouté

- Fichier 0006-derni-res-remarques-to-rebase.patch ajouté

- Fichier 0004-manager-add-user-import-views-32833.patch ajouté

Voilà:

- jQuery rétabli via l'ajout de MediaMixin
- test de la partie manage ajouté, et des permissions notamment, j'ai ajouté un drapeau pour contrôler l'application des permissions. La plupart des vues se contentent d'une permission possible, ici on veut la permission globale ou rien,

#73 - 21 juin 2019 14:50 - Paul Marillonnet

- Fichier out.ogv ajouté

Je crois voir un bug dans l'UI : l'import sur la page de garde de rapports semble cliquable, mais ne l'est pas (cf capture vidéo).

#74 - 21 juin 2019 14:53 - Paul Marillonnet

Paul Marillonnet a écrit :

> Je crois voir un bug dans l'UI : l'import sur la page de garde de rapports semble cliquable, mais ne l'est pas (cf capture vidéo).

Pardon, je me suis mal exprimé : l'entête de la table semble cliquable.

#75 - 21 juin 2019 15:00 - Paul Marillonnet

Bon, ma faute, après une application des migrations et une collecte des fichiers statiques l'UI est tout de suite plus compréhensible.
J'arrive à reproduire le parcours que tu décris dans le screencast.

Pour le reste, c'est OK pour moi, à pousser une fois que les tests passeront.

#76 - 21 juin 2019 15:01 - Paul Marillonnet

- Statut changé de Solution proposée à Solution validée

#77 - 21 juin 2019 16:03 - Paul Marillonnet

(Il y a juste quelques petites erreurs dans les tests, erreurs dues je crois à tes derniers patches -- ceux "à rebaser/to rebase".)

#78 - 21 juin 2019 18:11 - Benjamin Dauvergne

- Fichier 0001-add-unique-constraint-to-UserExternalId-32833.patch ajouté

- Fichier 0002-custom_user-rename-clean_fields-to-validate_unique-3.patch ajouté

- Fichier 0003-add-csv-import-framework-32833.patch ajouté

- Fichier 0004-manager-add-user-import-views-32833.patch ajouté

- Statut changé de Solution validée à Solution proposée

Les tests passent, je jette un coup d'oeuil aux tickets de Fred puis je pousse.

#79 - 21 juin 2019 18:57 - Benjamin Dauvergne

- Fichier 0001-add-unique-constraint-to-UserExternalId-32833.patch ajouté

- Fichier 0002-custom_user-rename-clean_fields-to-validate_unique-3.patch ajouté

- Fichier 0003-add-csv-import-framework-32833.patch ajouté

- Fichier 0004-manager-add-user-import-views-32833.patch ajouté

Voilà pris en compte les tickets #34225, #34226 et #34230.

#80 - 22 juin 2019 13:07 - Benjamin Dauvergne

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

```
commit dc3582ed452875bab869780dd1a546bcc50d2c64
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Sat Jun 15 15:59:31 2019 +0200

    manager: add user import views (fixes #32833)

commit fe0895da8b4453b3e06d295f8d72450dee0f3562
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Sat Jun 15 15:19:44 2019 +0200

    add csv import framework (#32833)

commit 8c06edd1a6a6c9f5ee5f428283620154e59cb4ad
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Mon Jun 17 00:12:23 2019 +0200

    custom_user: rename clean_fields to validate_unique (#32833)

commit 853140e8be4115644d8de03991633d7ebe52b62b
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Mon Jun 17 11:13:45 2019 +0200

    add unique constraint to UserExternalId (#32833)
```

#81 - 22 juin 2019 13:07 - Benjamin Dauvergne

- % réalisé changé de 0 à 100

Appliqué par commit authentic2|dc3582ed452875bab869780dd1a546bcc50d2c64.

#82 - 22 juin 2019 22:15 - Frédéric Péters

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

#83 - 11 septembre 2019 17:34 - Nicolas Roche

- Lié à Bug #35800: import csv et champ date ajouté

Fichiers

0001-WIP-import-users-from-csv-file-32833.patch	9,11 ko	07 mai 2019	Paul Marillonnet
0001-import-users-from-csv-file-32833.patch	15,7 ko	09 mai 2019	Paul Marillonnet
0001-import-users-from-csv-file-32833.patch	18,2 ko	10 mai 2019	Paul Marillonnet
0001-import-users-from-csv-file-32833.patch	17,9 ko	10 mai 2019	Paul Marillonnet
0001-import-users-from-csv-file-32833.patch	21,5 ko	14 mai 2019	Paul Marillonnet
0001-manager-users-csv-import.patch	54,4 ko	06 juin 2019	Paul Marillonnet
0003-manager-add-user-import-views-32833.patch	35,3 ko	16 juin 2019	Benjamin Dauvergne
0001-custom_user-rename-clean_fields-to-validate_unique-3.patch	985 octets	16 juin 2019	Benjamin Dauvergne
0002-add-CSV-import-framework-32833.patch	31 ko	16 juin 2019	Benjamin Dauvergne
screencast.mp4	2,22 Mo	16 juin 2019	Benjamin Dauvergne
0003-manager-add-user-import-views-32833.patch	35,3 ko	17 juin 2019	Benjamin Dauvergne
0001-custom_user-rename-clean_fields-to-validate_unique-3.patch	985 octets	17 juin 2019	Benjamin Dauvergne
0002-add-CSV-import-framework-32833.patch	31 ko	17 juin 2019	Benjamin Dauvergne
Firefox_Screenshot_2019-06-17T08-45-28.285Z.png	26,8 ko	17 juin 2019	Benjamin Dauvergne
0002-custom_user-rename-clean_fields-to-validate_unique-3.patch	985 octets	17 juin 2019	Benjamin Dauvergne
0001-add-unique-constraint-to-UserExternalId.patch	1,91 ko	17 juin 2019	Benjamin Dauvergne
0003-add-csv-import-framework-32833.patch	35,4 ko	17 juin 2019	Benjamin Dauvergne
0004-manager-add-user-import-views-32833.patch	35,7 ko	17 juin 2019	Benjamin Dauvergne
0005-check-self-referene-in-keys-to-rebase.patch	4,09 ko	18 juin 2019	Benjamin Dauvergne
0001-add-unique-constraint-to-UserExternalId-32833.patch	1,88 ko	18 juin 2019	Benjamin Dauvergne
0002-custom_user-rename-clean_fields-to-validate_unique-3.patch	985 octets	18 juin 2019	Benjamin Dauvergne
0003-add-csv-import-framework-32833.patch	35,4 ko	18 juin 2019	Benjamin Dauvergne
0004-manager-add-user-import-views-32833.patch	35,7 ko	18 juin 2019	Benjamin Dauvergne
0005-check-self-referene-in-keys-to-rebase.patch	4,09 ko	21 juin 2019	Benjamin Dauvergne
0001-add-unique-constraint-to-UserExternalId-32833.patch	1,88 ko	21 juin 2019	Benjamin Dauvergne
0002-custom_user-rename-clean_fields-to-validate_unique-3.patch	985 octets	21 juin 2019	Benjamin Dauvergne
0003-add-csv-import-framework-32833.patch	35,3 ko	21 juin 2019	Benjamin Dauvergne
0006-derni-res-remarques-to-rebase.patch	14,3 ko	21 juin 2019	Benjamin Dauvergne
0004-manager-add-user-import-views-32833.patch	35,7 ko	21 juin 2019	Benjamin Dauvergne
out.ogv	405 ko	21 juin 2019	Paul Marillonnet
0001-add-unique-constraint-to-UserExternalId-32833.patch	1,88 ko	21 juin 2019	Benjamin Dauvergne
0002-custom_user-rename-clean_fields-to-validate_unique-3.patch	985 octets	21 juin 2019	Benjamin Dauvergne
0003-add-csv-import-framework-32833.patch	36,7 ko	21 juin 2019	Benjamin Dauvergne
0004-manager-add-user-import-views-32833.patch	42,5 ko	21 juin 2019	Benjamin Dauvergne
0001-add-unique-constraint-to-UserExternalId-32833.patch	1,88 ko	21 juin 2019	Benjamin Dauvergne
0002-custom_user-rename-clean_fields-to-validate_unique-3.patch	985 octets	21 juin 2019	Benjamin Dauvergne
0003-add-csv-import-framework-32833.patch	36,7 ko	21 juin 2019	Benjamin Dauvergne
0004-manager-add-user-import-views-32833.patch	42,6 ko	21 juin 2019	Benjamin Dauvergne