

Fargo - Development #22470

paramétrage des colonnes affichées

12 Mar 2018 03:21 PM - Frédéric Péters

Status:	En cours	Start date:	12 Mar 2018
Priority:	Normal	Due date:	
Assignee:	Paul Marillonnet	% Done:	0%
Category:		Estimated time:	0:00 hour
Target version:	0.26	Planning:	
Patch proposed:	Yes		
Description			
Pour le moment on hardcode titre ... vignette actions, avec les ... étant les colonnes arrivant via les attributs de la classe DocumentTable.			
J'aimerais bien avoir un COLUMNS dans les settings, où je pourrais mettre ['title', 'creation_date', 'expiration_date', 'thumbnail'], par exemple.			
(on garderait par défaut ['title', 'size', 'creation_date', 'thumbnail']).			

History

#1 - 14 Mar 2018 04:27 PM - Paul Marillonnet

- Assignee set to Paul Marillonnet

#2 - 14 Mar 2018 05:42 PM - Paul Marillonnet

- File 0001-WIP-configure-displayed-columns-22470.patch added

- Patch proposed changed from No to Yes

Un peu à court d'inspiration ce soir, c'est pas bien beau et je ne vois pas d'autres solutions.

#3 - 14 Mar 2018 07:20 PM - Benjamin Dauvergne

Un jour on enlève django-tables2 le lendemain on continue à l'utiliser, j'ai du mal à suivre ;)

Bon sinon tu ne réponds pas à la demande exacte de Fred (on ne peut pas réorganiser title et thumbnail avec ton code), donc j'attendrai son avis sur ce point, sinon pour le code c'est ack, faudrait juste un petit test que ça marche.

#4 - 14 Mar 2018 07:52 PM - Frédéric Péters

Un jour on enlève django-tables2 le lendemain on continue à l'utiliser, j'ai du mal à suivre ;)

À nouveau, mon propos c'est juste que quelqu'un s'occupe de ce module; que ça se fasse en utilisant django-tables ou pas, peu m'importe. (juste que django-tables avait pu apparaitre à un moment comme ne facilitant pas l'entrée).

Bon sinon tu ne réponds pas à la demande exacte de Fred (on ne peut pas réorganiser title et thumbnail avec ton code),

En vrai on peut s'abstenir là-dessus, mais FARGO_TABLE_COLUMNS = ('size', 'creation_date', 'filename') me laisse penser que ça ajoutera une colonne filename alors que la colonne titre est déjà posée, j'évitais ça.

Pour revenir à l'objectif effectif (stras), c'est avoir titre, date de création, date d'expiration, vignette.

(ici un commentaires sur title(), creation_date vs created et l'ordre des colonnes, zappé, pour l'important qui suit)

Bref, il me semble qu'on doit définir dans tables l'ensemble des colonnes possibles (en pratique, je dirais origin et expiration_date), puis passer par :

```
.. attribute:: sequence
```

```
The sequence/order of columns the columns (from left to right).
```

```
:type: iterable
```

```
Items in the sequence must be :term:`column names <column name>`, or
``"...`` (string containing three periods). ``...`` can be used as a
catch-all for columns that aren't specified.
```

(attribut de la classe TableBase).

#5 - 15 Mar 2018 05:31 PM - Paul Marillonnet

Frédéric Péters a écrit :

(ici un commentaires sur title(), creation_date vs created et l'ordre des colonnes, zappé, pour l'important qui suit)

Bref, il me semble qu'on doit définir dans tables l'ensemble des colonnes possibles (en pratique, je dirais origin et expiration_date), puis passer par :

Tu penses que l'attribut sequence est nécessaire pour faire respecter l'ordre d'apparition des colonnes.

Dans la version de django_tables2 utilisée par Fargo, on dirait que l'attribut fields respecte aussi l'ordre d'apparition (extrait de DeclarativeColumnMetaClass.__new__):

```
# Possibly add some generated columns based on a model
if opts.model:
    extra = OrderedDict()
    # honor Table.Meta.fields, fallback to model._meta.fields
    if opts.fields:
        # Each item in opts.fields is the name of a model field or a
        # normal attribute on the model
        for field_name in opts.fields:
            try:
                field = opts.model._meta.get_field(field_name)
            except FieldDoesNotExist:
                extra[field_name] = columns.Column()
            else:
                extra[field_name] = columns.library.column_for_field(field)

    else:
        for field in opts.model._meta.fields:
            extra[field.name] = columns.library.column_for_field(field)
    attrs["base_columns"].update(extra)
```

Autre chose : ici, je ne vois pas comment éviter l'usage de title(). Tu penses qu'on peut se permettre d'attendre un dictionnaire à la place d'un tuple pour settings.FARGO_TABLE_COLUMNS ? (genre {nom de colonne: titre de colonne} ?

#6 - 15 Mar 2018 05:54 PM - Frédéric Péters

Tu penses que l'attribut sequence est nécessaire pour faire respecter l'ordre d'apparition des colonnes.

Oui.

Autre chose : ici, je ne vois pas comment éviter l'usage de title(). Tu penses qu'on peut se permettre d'attendre un dictionnaire à la place d'un tuple pour settings.FARGO_TABLE_COLUMNS ? (genre {nom de colonne: titre de colonne} ?

Je me réexplique, dans ce paramètre, juste une liste d'identifiants. Dans la classe de la table, toutes les colonnes possibles :

```
class DocumentTable(tables.Table):
    '''Display the list of documents of the user'''
    size = tables.TemplateColumn(template_code='{ record.document.content.size|filesizeformat }',
                                orderable=False,
                                verbose_name=_('size').title())
    created = tables.DateTimeColumn(verbose_name=_('creation date').title())
+   expiration_date = tables.WhateverColumn(verbose_name=_('Expiration'))
+   origin = tables.WhateverColumn(verbose_name=_('Origin'))
```

(on laisse title et description et thumbnail de côté, elles sont particulières).

C'est-à-dire que les colonnes, elles sont ainsi préparées, avec un beau titre, elles ne sont pas issues d'attributs créés à la volée.

#7 - 15 Mar 2018 06:24 PM - Paul Marillonnet

- File 0001-WIP-configure-displayed-columns-22470.patch added

Le patch est plus simple dans ce cas (même si je n'aime pas cette double redéfinition de DocumentTable.Meta.{sequence,fields} à la même valeur).

#8 - 15 Mar 2018 10:49 PM - Frédéric Péters

Je ne sais pas du tout pour fields/sequence dans la class Meta, ma lecture du code m'iimaginait simplement avec une propriété sequence posée sur la classe, définie une seule fois. (mais je connais mal django-tables).

#9 - 16 Mar 2018 12:49 PM - Frédéric Péters

- Target version set to 0.24

#10 - 21 Mar 2018 03:13 PM - Benjamin Dauvergne

- Target version changed from 0.24 to 0.25

#11 - 05 Apr 2018 05:53 PM - Josué Kouka

- Status changed from Nouveau to En cours

#12 - 10 Apr 2018 11:28 AM - Benjamin Dauvergne

sequence ne limite pas la visibilité des colonnes, il définit simplement l'ordre, donc Paul a bien raison de définir aussi field.

Le souci ici c'est qu'il ne sera pas possible de faire varier les colonne par tenant, fields et sequence étant défini statiquement au niveau du module et pas dynamiquement dans un *init*.

#13 - 10 Apr 2018 01:36 PM - Benjamin Dauvergne

- Target version changed from 0.25 to 0.26

#14 - 10 Apr 2018 01:43 PM - Benjamin Dauvergne

Ensuit pour pouvoir gérer la séquence il faut faire d'absolument toutes les colonnes des colonnes de la table (title et thumbnail compris).

Files

0001-WIP-configure-displayed-columns-22470.patch	1.74 KB	14 Mar 2018	Paul Marillonnet
0001-WIP-configure-displayed-columns-22470.patch	1.65 KB	15 Mar 2018	Paul Marillonnet