

## Combo - Development #23408

### clarifier l'usage de is\_relevant()

25 avril 2018 15:16 - Frédéric Péters

<b>Statut:</b> Nouveau	<b>Début:</b> 25 avril 2018
<b>Priorité:</b> Normal	<b>Echéance:</b>
<b>Assigné à:</b>	<b>% réalisé:</b> 0%
<b>Catégorie:</b>	<b>Temps estimé:</b> 0:00 heure
<b>Version cible:</b>	<b>Planning:</b>
<b>Patch proposed:</b> Non	

**Description**

Pour le moment c'est parfois utilisé pour ne pas faire de rendu pour des cellules configurées,

```
def is_relevant(self, context):  
    return bool(self.text)
```

et parfois pour ne pas faire le rendu de cellules demandant à un utilisateur connecté,

```
def is_relevant(self, context):  
    if not (getattr(context['request'], 'user', None) and context['request'].user.is_authenticated()):  
        return False  
    return True
```

et parfois davantage encore :

```
def is_relevant(self, context):  
    if not (getattr(context['request'], 'user', None) and context['request'].user.is_authenticated()):  
        return False  
    return BasketItem.get_items_to_be_paid(context['request'].user).count() > 0
```

et parfois c'est pris comme synonyme de is\_enabled(),

```
def is_relevant(self, context):  
    return self.is_enabled()
```

Je nettoierais pour avoir :

- is\_enabled (existe déjà) → concerne le /manage/, sans le is\_enabled on ne présente même pas la cellule (ça peut être sur un flag ENABLED mais aussi sur d'autres choses, genre on ne présente pas le paiement tant qu'une régie n'a pas été configurée)
- is\_configured (à créer) → pour la situation du bool(self.text) (et autres similaires), pour ne pas tenter un rendu public alors que la cellule n'est pas configurée.
- is\_visible (existe déjà) → pour la vérif par rapport à l'usager connecté
- is\_relevant → pour le reste.

Les deux endroits où c'est utilisé, c'est lors du rendu d'un placeholder pour voir les cellules à mettre dedans, et le rendu d'une cellule pour la recherche (render\_for\_search).