

Obtenir les exceptions lors de l'évaluation d'une condition au format Django

29 mars 2018 11:17 - Benjamin Dauvergne

Statut:	Fermé	Début:	29 mars 2018
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Non		
Description			
Les exceptions levés par l'expression dans un {% if %} sont entièrement cachées par Django via ce code:			
<pre># django/template/smartif.py def infix(bp, func): """ Creates an infix operator, given a binding power and a function that evaluates the node """ class Operator(TokenBase): lbp = bp def led(self, left, parser): self.first = left self.second = parser.expression(bp) return self def eval(self, context): try: return func(context, self.first, self.second) except Exception: # Templates shouldn't throw exceptions when rendering. We are # most likely to get exceptions for things like {% if foo in bar # %} where 'bar' does not support 'in', so default to False return False return Operator</pre>			
On souhaiterait que ce ne soit pas le cas pour permettre le même genre de log qu'avec les conditions Python.			
Demandes liées:			
Lié à w.c.s. - Development #21545: permettre des conditions utilisant le form...		Fermé	31 janvier 2018

Historique

#1 - 29 mars 2018 11:18 - Benjamin Dauvergne

- Lié à Development #21545: permettre des conditions utilisant le format Django ajouté

#2 - 29 mars 2018 11:25 - Benjamin Dauvergne

Ticket Django existant: <https://code.djangoproject.com/ticket/17664>

Le ticket explique encore mieux les choses que je n'ai pu le faire, si a lève une exception alors:

- {% if a %} aussi
- {% if a|filter %} aussi
- mais pas {% if not a %}
- mais pas {% if a != 3 %}

i.e. le problème n'apparaît que sur une condition booléenne.

Par contre si a n'existe pas, aucune exception, VariableDoesNotExist est caché dans tous les cas et la valeur forcée à None (la condition échoue).

```
# django/template/defaulttags.py IfNode

def render(self, context):
    for condition, nodelist in self.conditions_nodelists:

        if condition is not None:          # if / elif clause
            try:
                match = condition.eval(context)
            except VariableDoesNotExist:
                match = None
        else:                               # else clause
            match = True

        if match:
            return nodelist.render(context)

    return ''
```

#3 - 03 janvier 2024 10:45 - Frédéric Péters

- Statut changé de Nouveau à Fermé

- Planning mis à Non

Ticket Django existant: <https://code.djangoproject.com/ticket/17664>

Il a depuis été rejeté sur l'idée que ça changerait trop le comportement, je vais fermer ici aussi.