w.c.s. - Bug #66032

la recherche dans les fiches ne fonctionne pas avec les accents : crash 500

07 juin 2022 16:46 - Thomas Noël

Statut:	Fermé	**Début:**	07 juin 2022
Priorité:	Normal	**Echéance:**	
Assigné à:	Thomas Noël	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Si on a une fiche qui contient le mot "métier", elle est remontée par la recherche quand on indique le mot "metier", mais pas avec le mot "métier".

C'est lié à un crash 500 (invisible en frontoffice parce que caché par ajax)

```
[2022-06-07 16:38:15] exception caught
Exception:
  type = '<class 'KeyError'>', value = ''c140516717812272''
Stack trace (most recent call first):
  File "/usr/lib/python3/dist-packages/wcs/sql.py", line 2377, in get_sorted_ids
  2375         else:
  2376             sql_statement += cls.get_order_by_clause(order_by)
> 2377         cur.execute(sql_statement, parameters)
  2378         ids = [x[0] for x in cur.fetchall()]
  2379         conn.commit()
  locals:
     clause = <ERROR WHILE PRINTING VALUE>
     cls = <class 'wcs.carddef._wcs_Kb_Fiches'>
     conn = <connection object at 0x7fcc993a1890; dsn: 'dbname=wcs_demarches_test_grenoble_iziici_
fr', closed: 0>
     cur = <cursor object at 0x7fcc9948d8b0; closed: 0>
     fts = <ERROR WHILE PRINTING VALUE>
     func_clause = None
     order_by = 'rank'
     parameters = {'c140516913656576': True, 'c140516719785200': 'GRE', 'c140516803034160': 'draft
', 'c140516704870704': 'metier'}
     sql_statement = 'SELECT id FROM carddata_16_kb_fiches WHERE fbobe479389_3732_4b94_bb8d_3ffe6e
738cea = %(c140516913656576)s AND fbob40dab7e_e7a0_4d51_b6f5_34c1fcfb1c70 = %(c140516719785200)s A
ND anonymised IS NULL AND status != %(c140516803034160)s AND fts @@ plainto_tsquery(%(c14051670487
0704)s) ORDER BY ts_rank(fts, plainto_tsquery(%(c140516717812272)s)) DESC'
     where_clauses = ['fbobe479389_3732_4b94_bb8d_3ffe6e738cea = %(c140516913656576)s', 'fbob40dab
7e_e7a0_4d51_b6f5_34c1fcfb1c70 = %(c140516719785200)s', 'anonymised IS NULL', 'status != %(c140516
803034160)s', 'fts @@ plainto_tsquery(%(c140516704870704)s)']
  File "/usr/lib/python3/dist-packages/wcs/sql.py", line 619, in f
   617         except psycopg2.Error:
   618             get_connection().rollback()
>  619             raise
   620
   621     return f
  locals:
     args = <ERROR WHILE PRINTING VALUE>
     func = <function SqlMixin.get_sorted_ids at 0x7fcc99f9f040>
     kwargs = {}
  File "/usr/lib/python3/dist-packages/wcs/forms/backoffice.py", line 325, in get_listing_item_ids
_sql
   323             order_by = '-id'
   324
>  325         return list(formdata_class.get_sorted_ids(order_by, criterias))
   326
   327     def get_listing_items(
  locals:
```

```
     anonymise = False
     criterias = <ERROR WHILE PRINTING VALUE>
     direction = ''
     field = <ItemField bob40dab7e-e7a0-4d51-b6f5-34c1fcfb1c70 'Dt Public concerné'>
     formdata_class = <class 'wcs.carddef._wcs_Kb_Fiches'>
     order_by = 'rank'
     query = 'métier'
     selected_filter = 'all'
     selected_filter_operator = 'eq'
     self = <wcs.forms.backoffice.FormDefUI object at 0x7fcc98153370>
     user = <SqlUser 'Thomas Noël' id:27>
  File "/usr/lib/python3/dist-packages/wcs/forms/backoffice.py", line 164, in get_listing_item_ids
   162     ):
   163         if get_publisher().is_using_postgresql():
>  164             return self.get_listing_item_ids_sql(
   165                 selected_filter, selected_filter_operator, query, order_by, user, criterias
, anonymise
   166             )
  locals:
     anonymise = False
     criterias = [<Equal (attribute: 'fbobe479389-3732-4b94-bb8d-3ffe6e738cea', value: True)>, <Eq
ual (attribute: 'fbob40dab7e-e7a0-4d51-b6f5-34c1fcfb1c70', value: 'GRE')>, <Null (attribute: 'anon
ymised')>]
     order_by = 'rank'
     query = 'métier'
     selected_filter = 'all'
     selected_filter_operator = 'eq'
     self = <wcs.forms.backoffice.FormDefUI object at 0x7fcc98153370>
     user = <SqlUser 'Thomas Noël' id:27>
  File "/usr/lib/python3/dist-packages/wcs/forms/backoffice.py", line 348, in get_listing_items
   346             order_by = None
   347
>  348         item_ids = self.get_listing_item_ids(
   349             selected_filter=selected_filter,
   350             selected_filter_operator=selected_filter_operator,
  locals:
     anonymise = False
     criterias = [<Equal (attribute: 'fbobe479389-3732-4b94-bb8d-3ffe6e738cea', value: True)>, <Eq
ual (attribute: 'fbob40dab7e-e7a0-4d51-b6f5-34c1fcfb1c70', value: 'GRE')>, <Null (attribute: 'anon
ymised')>]
     fields = [<wcs.backoffice.management.FakeField object at 0x7fcc98153340>, <StringField 1 'ide
ntifiant'>, <StringField 2 'Titre'>, <ItemField bo0d7ab966-cbe1-41ca-8a89-431e9e866c95 'Niveau de
fiche'>, <StringField 10 'code thématique'>, <wcs.backoffice.management.FakeField object at 0x7fcc
991827c0>]
     formdata_class = <class 'wcs.carddef._wcs_Kb_Fiches'>
     limit = 20
     offset = 0
     order_by = 'rank'
     query = 'métier'
     selected_filter = 'all'
     selected_filter_operator = 'eq'
     self = <wcs.forms.backoffice.FormDefUI object at 0x7fcc98153370>
     user = <SqlUser 'Thomas Noël' id:27>
  File "/usr/lib/python3/dist-packages/wcs/forms/backoffice.py", line 58, in listing
    56             if using_postgresql:
    57                 criterias.append(Null('anonymised'))
>   58             items, total_count = self.get_listing_items(
    59                 fields,
    60                 selected_filter,
  locals:
     criterias = [<Equal (attribute: 'fbobe479389-3732-4b94-bb8d-3ffe6e738cea', value: True)>, <Eq
ual (attribute: 'fbob40dab7e-e7a0-4d51-b6f5-34c1fcfb1c70', value: 'GRE')>, <Null (attribute: 'anon
ymised')>]
     fields = [<wcs.backoffice.management.FakeField object at 0x7fcc98153340>, <StringField 1 'ide
ntifiant'>, <StringField 2 'Titre'>, <ItemField bo0d7ab966-cbe1-41ca-8a89-431e9e866c95 'Niveau de
fiche'>, <StringField 10 'code thématique'>, <wcs.backoffice.management.FakeField object at 0x7fcc
991827c0>]
```

```
     include_checkboxes = True
     items = None
     limit = 20
     offset = 0
     order_by = 'rank'
     query = 'métier'
     selected_filter = 'all'
     selected_filter_operator = 'eq'
     self = <wcs.forms.backoffice.FormDefUI object at 0x7fcc98153370>
     url_action = None
     using_postgresql = True
  File "/usr/lib/python3/dist-packages/wcs/backoffice/management.py", line 2193, in _q_index
  2191                           )
  2192
> 2193         table = FormDefUI(self.formdef).listing(
  2194             fields=fields,
  2195             selected_filter=selected_filter,
  locals:
     action = {'action': <wcs.workflows.WorkflowGlobalAction object at 0x7fcc9843aac0>, 'roles': [
], 'functions': ['_editor']}
     attrs = {'data-visible_for__editor': 'true'}
     criterias = [<Equal (attribute: 'fbobe479389-3732-4b94-bb8d-3ffe6e738cea', value: True)>, <Eq
ual (attribute: 'fbob40dab7e-e7a0-4d51-b6f5-34c1fcfb1c70', value: 'GRE')>, <Null (attribute: 'anon
ymised')>]
     fields = [<wcs.backoffice.management.FakeField object at 0x7fcc98153340>, <StringField 1 'ide
ntifiant'>, <StringField 2 'Titre'>, <ItemField bo0d7ab966-cbe1-41ca-8a89-431e9e866c95 'Niveau de
fiche'>, <StringField 10 'code thématique'>, <wcs.backoffice.management.FakeField object at 0x7fcc
991827c0>]
     function = '_editor'
     limit = 20
     multi_actions = [{'action': <wcs.workflows.WorkflowGlobalAction object at 0x7fcc9843aac0>, 'r
oles': [], 'functions': ['_editor']}]
     multi_form = <wcs.qommon.form.Form object at 0x7fcc981535b0>
     offset = 0
     order_by = 'rank'
     qs = '?ajax=true&offset=0&limit=20&order_by=rank&q=m%C3%A9tier&filter-bobe479389-3732-4b94-bb
8d-3ffe6e738cea-operator=eq&filter-bobe479389-3732-4b94-bb8d-3ffe6e738cea-value=true&filter-bob40d
ab7e-e7a0-4d51-b6f5-34c1fcfb1c70-operator=eq&filter-bob40dab7e-e7a0-4d51-b6f5-34c1fcfb1c70-value=G
RE&filter-bobe479389-3732-4b94-bb8d-3ffe6e738cea=on&filter-bob40dab7e-e7a0-4d51-b6f5-34c1fcfb1c70=
on&id=on&1=on&2=on&bo0d7ab966-cbe1-41ca-8a89-431e9e866c95=on&10=on&status=on&columns-order=id%2C1%
2C2%2Cbo0d7ab966-cbe1-41ca-8a89-431e9e866c95%2C10%2Cstatus'
     query = 'métier'
     selected_filter = 'all'
     selected_filter_operator = 'eq'
     self = <wcs.backoffice.data_management.CardPage object at 0x7fcc981534f0>
```

Révisions associées

Révision 1cc1b76d - 08 juin 2022 00:15 - Thomas Noël

storage: apply full text search normalization (#66032)

Historique

#1 - 07 juin 2022 16:56 - Anaïs Ecuvillon → en congés, retour le 30/04

- Lié à Development #60875: Autocomplétion source de données issue d'une fiche - recherche sans tenir compte des accents ni des tirets ajouté

#2 - 07 juin 2022 17:02 - Thomas Noël

Au moins facile à reproduire en envoyant un "q=fée" à l'api de recherche :

```
--- a/tests/api/test_formdata.py
+++ b/tests/api/test_formdata.py
@@ -864,6 +864,10 @@ def test_api_list_formdata_order_by_rank(pub, local_user):
     assert len(resp.json) == 2
     assert [int(x['id']) for x in resp.json] == [formdata3.id, formdata1.id]

+    resp = get_app(pub).get(sign_uri('/api/forms/test/list?full=on&q=fée', user=local_user))
+    assert len(resp.json) == 2
```

```
+    assert [int(x['id']) for x in resp.json] == [formdata3.id, formdata1.id]
+

 def test_api_list_formdata_unknown_filter(pub, local_user):
     pub.role_class.wipe()
```

#3 - 07 juin 2022 17:26 - Pierre Ducroquet

La backtrace montre bien le soucis:
type = '<class 'KeyError'>', value = "c140516717812272"

```
parameters = {'c140516913656576': True, 'c140516719785200': 'GRE', 'c140516803034160': 'draft', 'c140516704870
704': 'metier'}
     sql_statement = 'SELECT id FROM carddata_16_kb_fiches WHERE fbobe479389_3732_4b94_bb8d_3ffe6e738cea =
s AND fbob40dab7e_e7a0_4d51_b6f5_34c1fcfb1c70 = %(c140516719785200)s AND anonymised IS NULL AND status != %(c1
40516803034160)s AND fts @ plainto_tsquery(%(c140516704870704)s) ORDER BY ts_rank(fts, plainto_tsquery(%(c1405
16717812272)s)) DESC'
     where_clauses = ['fbobe479389_3732_4b94_bb8d_3ffe6e738cea = %(c140516913656576)s', 'fbob40dab7e_e7a0_4d51
_b6f5_34c1fcfb1c70 = %(c140516719785200)s', 'anonymised IS NULL', 'status != %(c140516803034160)s', 'fts @ pla
into_tsquery((c140516704870704)s)']
```

Et effectivement, le paramètre plainto_tsquery est différent entre le fts @@ et le ts_rank.

Je vais creuser.

#5 - 07 juin 2022 18:07 - Pierre Ducroquet

- Fichier 0001-sql-work-around-crash-with-rank-ordering-66032.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Trouvé.

```
    def get_sorted_ids(cls, order_by, clause=None):
        conn, cur = get_connection_and_cursor()
        sql_statement = 'SELECT id FROM %s' % cls._table_name
        where_clauses, parameters, func_clause = parse_clause(clause)
        assert not func_clause
        if where_clauses:
            sql_statement += ' WHERE ' + ' AND '.join(where_clauses)
        if order_by == 'rank':
            try:
                fts = [x for x in clause if not callable(x) and x.__class__.__name__ == 'FtsMatch'][0]
            except IndexError:
                pass
            else:
                sql_statement += ' ORDER BY ts_rank(fts, plainto_tsquery(%%(c%s)s)) DESC' % id(fts.value)
        else:
            sql_statement += cls.get_order_by_clause(order_by)
        cur.execute(sql_statement, parameters)
```

Entre l'appel à parse_clause et l'appel à id(fts.value), l'objet value du FtsMatch aurait changé d'id. Je note que c'est le seul chemin de code à tenter cette manipulation.
FtsMatch fait un appel à unidecode.unidecode. Or, si il n'y a pas d'accent, l'id de la chaine retournée est le même que celui de la chaîne en entrée, alors qu'avec un accent l'id change à chaque appel.

Maintenant, ce qu'il se passe dans parse_clause :

```
def parse_clause(clause):
    # returns a three-elements tuple with:
    #  - a list of SQL 'WHERE' clauses
    #  - a dict for query parameters
    #  - a callable, or None if all clauses have been successfully translated

    if clause is None:
        return ([], {}, None)

    if callable(clause):  # already a callable
        return ([], {}, clause)

    # create 'WHERE' clauses
    func_clauses = []
    where_clauses = []
```

```
    parameters = {}
    for element in clause:
        if callable(element):
            func_clauses.append(element)
        else:
            sql_class = globals().get(element.__class__.__name__)
            if sql_class:
                sql_element = sql_class(**element.__dict__)
                where_clauses.append(sql_element.as_sql())
                parameters.update(sql_element.as_sql_param())
            else:
                func_clauses.append(element.build_lambda())
```

```
    if func_clauses:
        return (where_clauses, parameters, parse_storage_clause(func_clauses))
    else:
        return (where_clauses, parameters, None)
```

```
sql_element = sql_class(**element.__dict__)
```

=> donc on va reconstruire l'objet à chaque fois, et si il y a un accent, unidecode va générer des ids différents systématiquement.
Et comme l'objet sql_element n'est pas renvoyé, il est impossible de réutiliser y-celui.

Un correctif pourrait être de réinjecter, dans le tri par rank, le paramètre.
Sinon, je suis curieux de savoir pourquoi il y a ce sql_class(**element.__dict__). Le code date de 2014, mais je ne vois pas pourquoi faire ce deepcopy du pauvre, et aucun commentaire ne vient l'appuyer.

#6 - 07 juin 2022 18:46 - Thomas Noël

- Statut changé de Solution proposée à En cours

J'avais déjà joué un peu comme ça, mais trop facile, passe pas, parce que :

- c'est c%s qu'il faut ajouter dans parameters (parameters['c%s' % id(fts.value)] = fts.value)
- mais de toute façon fts.value contient "fée" et non pas "fee", faut pas me demander pourquoi c'est pas passé par le init et donc le FtsMatch.get_fts_value... et l'ordre résultant n'est pas le bon et boum

Bref j'ai ce patch qui passe mais ça me plait très peu

```
@@ -2371,7 +2371,9 @@ class SqlMixin:
             except IndexError:
                 pass
             else:
-                sql_statement += ' ORDER BY ts_rank(fts, plainto_tsquery(%%(c%s)s)) DESC' % id(fts.value)
+                value = FtsMatch.get_fts_value(fts.value)  ### XXX c'est ce moment que j'ai ajouté après quel
ques breakpoint ; je ne le comprends pas, normalement fts a déjà joué le get_fts_value sur self.value...
+                sql_statement += ' ORDER BY ts_rank(fts, plainto_tsquery(%%(c%s)s)) DESC' % id(value)
+                parameters['c%s' % id(value)] = value
         else:
             sql_statement += cls.get_order_by_clause(order_by)
         cur.execute(sql_statement, parameters)
```

(disclaimer: je ne suis pas intime avec wcs/sql.py, j'indique juste un patch qui semble faire passer les tests... sans le trouver joli)

#7 - 08 juin 2022 00:19 - Thomas Noël

- Fichier 0001-storage-apply-full-text-search-normalization-66032.patch ajouté

- Statut changé de En cours à Solution proposée

- Assigné à mis à Thomas Noël

Fini par voir que FtsMatch dans les critères c'est celui de qommon/storage.py, or celui-là ne gère pas la normalisation unidecode.unidecode, alors que sql.FtsMatch si, et donc ça disjoncte.

Tout rentre dans l'ordre en ajoutant la même normalisation dans qommon.storage.FtsMatch

#8 - 08 juin 2022 00:56 - Thomas Noël

- Lié à Development #60875: Autocomplétion source de données issue d'une fiche - recherche sans tenir compte des accents ni des tirets supprimé

#9 - 08 juin 2022 08:11 - Pierre Ducroquet

- Statut changé de Solution proposée à Solution validée

- Assigné à Thomas Noël supprimé

Thomas Noël a écrit :

> Fini par voir que FtsMatch dans les critères c'est celui de qommon/storage.py, or celui-là ne gère pas la normalisation unidecode.unidecode, alors que sql.FtsMatch si, et donc ça disjoncte.
>
> Tout rentre dans l'ordre en ajoutant la même normalisation dans qommon.storage.FtsMatch

Ok, donc c'est ça qui explique le code de parse_clause.
Pour ma part je valide, mais je ne sais pas dire si c'est la meilleure solution architecturalement.

#10 - 08 juin 2022 08:12 - Pierre Ducroquet

- Assigné à mis à Thomas Noël

#11 - 08 juin 2022 10:09 - Thomas Noël

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

```
commit 1cc1b76d0667ed4eaf3a1a09fd2c76a277eb3390
Author: Thomas NOËL <tnoel@entrouvert.com>
Date:   Wed Jun 8 00:11:45 2022 +0200

    storage: apply full text search normalization  (#66032)
```

#12 - 08 juin 2022 12:14 - Transition automatique

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

#13 - 14 août 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-sql-work-around-crash-with-rank-ordering-66032.patch	1,55 ko	07 juin 2022	Pierre Ducroquet
0001-storage-apply-full-text-search-normalization-66032.patch	1,67 ko	07 juin 2022	Thomas Noël