

Custom query

2022/05/09 19:23 - Admin Redmine

| | | | |
|-----------------|--------------------------------------|----------------------|------------|
| ステータス: | New | 開始日: | 2022/05/09 |
| 優先度: | 通常 | 期日: | |
| 担当者: | | 進捗率: | 0% |
| カテゴリ: | Search engine_16 | 予定工数: | 0.00時間 |
| 対象バージョン: | | 作業時間: | 0.00時間 |
| Redmineorg_URL: | https://www.redmine.org/issues/37008 | status_id: | 1 |
| category_id: | 16 | tracker_id: | 1 |
| version_id: | 0 | plus1: | 0 |
| issue_org_id: | 37008 | affected_version: | |
| author_id: | 589219 | closed_on: | |
| assigned_to_id: | 0 | affected_version_id: | |
| comments: | 0 | | |

説明

We have made a query for our entities, inheriting the basic redmine Query.
But faced a problem with using custom methods `sql_for_['field_name']_field`

This is how the filters are defined

```
def sql_for_something_field(field, operator, values)
  sql_for_field(field, operator, values, ::StateContract.table_name, :identifier)
end
```

And during the exploitation found that filters with the operator "!=" and string data type do not always work correctly.
All next filters stop working.

Here is the resulting SQL

```
WHERE ( table1.field1 IS NULL OR table1.field1 = " AND (table2.id IN (1000)))
```

The problem is the missing brackets around

```
table1.field1 IS NULL OR table1.field1 = "
```

After some digging, I found (`app/models/query.rb:989`)

```
filters_clauses << send(method, field, operator, v)
```

Which did not expose brackets for the custom filter.

I believe that filters, even custom ones, should be isolated.

Please replace

```
filters_clauses << send(method, field, operator, v)
```

on the

```
filters_clauses << '(' + send(method, field, operator, v) + ')'
```

Note

For most filters, your implementation is not a problem

But for the operator "!=" with the string data type in the query, an OR is set, which breaks the query.

履歴

#1 - 2022/05/10 16:57 - Admin Redmine

- カテゴリ を Search engine_16 にセット