

Consistent interface for Principal.member_of and not_member_of

2022/05/09 18:29 - 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: | | 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/28243 | status_id: | 1 |
| category_id: | 0 | tracker_id: | 3 |
| version_id: | 0 | plus1: | 0 |
| issue_org_id: | 28243 | affected_version: | |
| author_id: | 4 | closed_on: | |
| assigned_to_id: | 0 | affected_version_id: | |
| comments: | 0 | | |

説明

Principal model has the following two scopes: member_of and not_member_of, both of them expect project or collection of projects as an argument. The problem is that given

```
projects = Project.where(:name => 'My Project')
```

you'll get correct results from

```
Principal.member_of(projects)
```

, but an exception will be thrown if you'll try to run

```
Principal.not_member_of(projects)
```

. That's because not_member_of puts its argument into an array if it's not a subclass of Array (which Project::ActiveRecord_Relation is not, though it knows how to #map).

I think one should be able to call both member_of and not_member_of with any kind of project collection, not just Array, and both of them should process arguments in the same way. So I provide a patch for that.