

Redmine - Defect #37165

Can not get anonymous user using User.find

2022-05-27 04:15 - Yuichi HARADA

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:			
Description			
<p>When I tried to get an anonymous user(id: 6) with User.find, the behavior of User.find is different between 4.2-stable and trunk(r21605).</p> <p>4.2-stable</p> <pre>% RAILS_ENV=development bundle exec rails console Loading development environment (Rails 5.2.8) >> User.find(6) Creating scope :sorted. Overwriting existing method Group.sorted. Creating scope :sorted. Overwriting existing method User.sorted. User Load (0.5ms) SELECT "users".* FROM "users" WHERE "users"."type" IN ('User', 'AnonymousUser') AND "users"."id" = ? LIMIT ? [["id", 6], ["LIMIT", 1]] => #<AnonymousUser id: 6, login: "", hashed_password: "1", firstname: "", lastname: "Anonymous", admin: false, status: 0, last_login_on: nil, language: "", auth_source_id: nil, created_on: "2006-07-19 17:33:19", updated_on: "2006-07-19 17:33:19", type: "AnonymousUser", identity_url: nil, mail_notification: "only_my_events", salt: nil, must_change_passwd: false, passwd_changed_on: nil, twofa_scheme: nil, twofa_totp_key: nil, twofa_totp_last_used_at: nil> >> exit</pre> <p>trunk</p> <pre>% RAILS_ENV=development bundle exec rails console Loading development environment (Rails 6.1.6) [1] pry(main)> User.find(6) Creating scope :sorted. Overwriting existing method User.sorted. (3.5ms) SELECT sqlite_version(*) User Load (0.5ms) SELECT "users".* FROM "users" WHERE "users"."type" = ? AND "users"."id" = ? LIMIT ? [["type", "User"], ["id", 6], ["LIMIT", 1]] ActiveRecord::RecordNotFound: Couldn't find User with 'id'=6 from vendor/bundle/ruby/3.1.0/gems/activerecord-6.1.6/lib/active_record/core.rb:338:in `find' [2] pry(main)> exit</pre>			

History

#1 - 2023-01-05 12:34 - Thomas Löber

In Redmine 5 the classes are lazily loaded, that means Rails does not know about the AnonymousUser class before it is referenced, and so the SELECT does not take into account that AnonymousUser is part of the STI hierarchy.

Once the AnonymousUser class is loaded, the SELECT works fine.

```
>> AnonymousUser
=> AnonymousUser(id: integer, login: string, hashed_password: string, firstname: string, lastname: string, admin: boolean, status: integer, last_login_on: datetime, language: string, ...)
>> User.find(2)
  User Load (0.7ms)  SELECT `users`.* FROM `users` WHERE `users`.`type` IN ('User', 'AnonymousUser') AND `users`.`id` = 2 LIMIT 1
=> #<AnonymousUser id: 2, login: "", hashed_password: [FILTERED], firstname: "", lastname: "Anonymous", admin: false, status: 0, last_login_on: nil, language: "", auth_source_id: nil, ...>
```