

プラグイン名

acts_as_commentableプラグイン

このプラグインができること

1. いろんなモデルにコメント(デフォルトだとタイトルと本文)を付けれる
2. ユーザモデルと連動して、

- 特定ユーザがコメントしたコメント一覧
- 特定ユーザがコメントした対象モデル一覧
- 対象モデルへのコメント一覧

を取得可能

対象バージョン

1.2系 2.0系 2.1系

ちょー簡単な使い方

```
script/plugin install http://juixe.com/svn/acts\_as\_commentable
```

でインストールして、マイグレーションしてから

```
class Post < ActiveRecord::Base  
  
  acts_as_commentable  
  
end
```

ARにacts_as_commentable入れて

```
post.comments << Comment.new(:title => titleStr, :comment => commentStr) #コメント追加  
post.comments #全コメント  
comments = Comment.find_comments_by_user(userInstance) #特定ユーザのコメント一覧  
postComments = Post.find_comments_by_user(userInstance) #特定ユーザがコメントしたモデル一覧
```

で幸せに。

公式ページ

[Juixe TechKnow » Acts As Commentable Plugin](#)

いや、公式かどうかはわからない

日本語解説ページ

- ▶ 見つからない

外国語解説ページ

- ▶ 良いところがない

のうほう

- ▶ Commentモデルは、このプラグインに付いてくるので、作らなくて良い
- ▶ このデフォルトのCommentモデルには、Userモデルへの関連が付いているので、ユーザ情報管理モデルはUserが好ましい。
- ▶ プラグインの中のモジュールを読めば解るが、commentsテーブルにはuser_idカラムがあるものの、Userモデルからhas_manyで関連づけする必要はない。特定ユーザが書いたコメントだけ取得したければ、Comment.find_comments_by_user()で絞り込むので、Userモデル側にhas_manyなくても問題無い(User#commentsとかしない)ということ。
- ▶ マイグレーションファイルの構造を載せておく

```
class AddRatingTables < ActiveRecord::Migration
  def self.up
    create_table :comments, :force => true do |t|
      t.column :title, :string, :limit => 50, :default => ""
      t.column :comment, :string, :default => ""
      t.column :created_at, :datetime, :null => false
      t.column :commentable_id, :integer, :default => 0, :null => false
      t.column :commentable_type, :string, :limit => 15,
        :default => "", :null => false
      t.column :user_id, :integer, :default => 0, :null => false
    end
    add_index :comments, ["user_id"], :name => "fk_comments_user"
  end
  def self.down
    drop_table :comments
  end
end
```



[WWW SQL Designerファイル](#)

コメント

- ▶ good morning - k 2011-08-29 14:04:14

名前: