django-simple-friends Documentation

Release 0.5

Atamert Ölçgen

March 10, 2013

CONTENTS

Contents:

CHAPTER

ONE

ABOUT DJANGO-SIMPLE-FRIENDS

I started developing *django-simple-friends* when I needed a simple app that handles friendships for my project. django-friends of Pinax was mature and well written but it was also handling contacts and invitations. So I decided to create my own app and steal some ideas from *django-friends*.

1.1 Highlights

- Only the relationships between registered users are managed.
- Friending with double confirmation. When a user adds another user as a friend, only when the other user accepts or tries to add the first user as a friend, the friendship relationship is created.
- Blocking is possible.

CHAPTER

TWO

SETTING UP DJANGO-SIMPLE-FRIENDS

2.1 Installation

1. Install *django-simple-friends* package:

   ```
   pip install django-simple-friends
   ```

2. Add *friends* to your *INSTALLED_APPS* setting:

   ```
   INSTALLED_APPS = (
       # Other apps
       'friends',
   )
   ```

3. Run *syncdb* to create tables and seed friendship data for existing users:

   ```
   python manage.py syncdb
   ```

4. Run tests to make sure the app is installed correctly:

   ```
   python manage.py test friends
   ```

5. Optionally include *friends.urls* to your URLconf:

   ```
   urlpatterns = patterns('',
       # Other entries
       (r'^friends/', include('friends.urls')),
   )
   ```

2.2 Development Setup

If you want to develop *django-simple-friends* you can follow the steps below to set up a development environment.

1. Log-in to your GitHub account and fork the project.
2. Create a virtual environment:

   ```
   virtualenv --no-site-packages django-simple-friends
   ```

3. Create a local repository:

```
cd django-simple-friends
. bin/activate
git clone git@github.com:muhuk/django-simple-friends.git src
```

Warning: You need to replace *muhuk* with your GitHub username in the command above.

4. Run the tests to make sure everyting is set up correctly:

```
cd src
python example/manage.py test friends
```

5. Pick an issue to work on.

CHAPTER

THREE

CHANGES

3.1 Version 0.5 - Oct 7, 2012

Tested with Django versions **1.3** & **1.4**.

- `friend_list` view is removed. See `friends` template filter.
- View functions are rewritten as class based views. But they still work as aliases.
- `post_syncdb` signals to fix the issue of `Users` without `Friendships`.
- Proper Sphinx powered documentation.
- German & Spanish translations.

3.2 Version 0.4 - Feb 4, 2010

- Initial release.

CHAPTER

FOUR

MODULE DOCUMENTATION

4.1 Views

4.1.1 Class Based Views

All the views are implemented as classes but *view functions* are also provided.

class friends.views.**BaseFriendshipActionView**(***kwargs*)

class friends.views.**FriendshipAcceptView**(***kwargs*)

class friends.views.**FriendshipBlockView**(***kwargs*)

class friends.views.**FriendshipCancelView**(***kwargs*)

class friends.views.**FriendshipDeclineView**(***kwargs*)

class friends.views.**FriendshipDeleteView**(***kwargs*)

class friends.views.**FriendshipRequestView**(***kwargs*)

class friends.views.**FriendshipUnblockView**(***kwargs*)

4.1.2 View Functions

Tip: If you want to customize the views provided, check out *Class Based Views* first.

friends.views.**friendship_request**(*request*, **args*, ***kwargs*)

friends.views.**friendship_accept**(*request*, **args*, ***kwargs*)

friends.views.**friendship_decline**(*request*, **args*, ***kwargs*)

friends.views.**friendship_cancel**(*request*, **args*, ***kwargs*)

friends.views.**friendship_delete**(*request*, **args*, ***kwargs*)

friends.views.**user_block**(*request*, **args*, ***kwargs*)

friends.views.**user_unblock**(*request*, **args*, ***kwargs*)

4.2 Models

class friends.models.**FriendshipRequest**(**args*, ***kwargs*)
FriendshipRequest(id, from_user_id, to_user_id, message, created, accepted)

class friends.models.**FriendshipManager**

class friends.models.**Friendship**(**args*, ***kwargs*)
Friendship(id, user_id)

class friends.models.**UserBlocks**(**args*, ***kwargs*)
UserBlocks(id, user_id)

CHAPTER

FIVE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

f