

3954

# Ignore "system" users (TM, sync, GT) in notifications #3954

Open Task



flodolo opened on Feb 8

Collaborator ...

My local install only has 1 user. While testing notifications, I noticed it would send it to 5 users.

In Django admin there is one user that is completely empty (can't even click it). Why? Is it created at the start?

Then there are 3 "system" users (pontoon-gt, pontoon-sync, pontoon-tm). We shouldn't send notifications to these.

Create sub-issue

in-app notification system, built on django-notifications-hq package

## django-notifications-hq

GitHub notification alike app for Django, derived from django-activity-stream

The major difference between django-notifications and django-activity-stream:

- `django-notifications` is for building something like Github "Notifications"
- While `django-activity-stream` is for building Github "News Feed"

for global/social activity feed (news feeds / timelines / recent activity)

Notifications are actually actions events, which are categorized by four main components.

- `Actor`. The object that performed the activity.
- `Verb`. The verb phrase that identifies the action of the activity.
- `Action Object`. (Optional) The object linked to the action itself.
- `Target`. (Optional) The object to which the activity was performed.

`Actor`, `Action Object` and `Target` are `GenericForeignKeys` to any arbitrary Django object. An action is a description of an action that was performed (`Verb`) at some instant in time by some `Actor` on some optional `Target` that results in an `Action Object` getting created/updated/deleted.

For example: `justquick` (`actor`) `closed` (`verb`) `issue 2` (`action_object`) on `activity-stream` (`target`) 12 hours ago

Nomenclature of this specification is based on the Activity Streams Spec: <http://activitystrea.ms/specs/atom/1.0/>

notifications-hq

designed for personal notifications directed at specific user

(Notifications) for me

(Activity stream) what happened

## Events for Notifications Firing

Event	When	Recipients
New strings	A sync run added new source strings to a project	Users who have contributed translations to that project
Approaching deadline	A project deadline is 7 or 2 days away	Project contributors in incomplete locales
Review recap	Daily summary of approved/rejected suggestions	The author of each reviewed suggestion
Unreviewed suggestions	Weekly nudge	Locale reviewers + previous translators + comment authors
New contributor	A user makes their first translation in a locale	The locale's managers
Comment added	Someone comments on an entity	Project translators, managers, mentioned users
Pinned comment	Someone pins a comment	All users who've translated that entity in any locale
Badge earned	A user crosses a translation/review milestone	The user themselves
Admin direct message	An admin uses the messaging form	Whoever the admin selects

- each event calls `notify.send` (`sender = ...`, `verb = ...`, ...)

- 9 distinct calls across pontoon/sync messaging translations base } each builds own recipient list with diff ORM query, no central helper

way to interact with database using python code instead of raw SQL (application) (database)

- Object Relational Mapping) translation bw Python objects & Relational tables (application) (database)

## System Users

- Translations stored as Translation rows
  - Each row has non-null user foreign key pointing at auth\_user
  - revert flows are structured (both UI & backend) that handle rolling back translation to previous version/state
- ↳ gives info on who produced given translation