



NOTIFICATIONS

<http://drupal.org/project/notifications>

The notifications framework provides a mechanism for informing users about events on a drupal site.

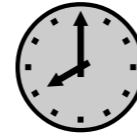
Every few hours, days or weeks, users are sent information about site content changes. Intervals and delivery formats are all configurable and extensible.

Plug-in modules can add all sorts of extra goodies.

PROCESS OVERVIEW

- A node is updated.
- A notifications module triggers an event.
- Updates are queued for all interested users.
- After a specified interval, updates are collected and a message is sent with digested data.

MESSAGE PROCESSING



Cron hook

All subscriptions are managed by interval. Users are able to specify that they only want to receive notifications every x hours/days/weeks.

A user will receive messages if they have entries in the queue, and they have not had any updates within their chosen timespan.

All pending events are then compiled and sent as one update. The exact form of this update is defined by the messaging module.

QUEUE STORE

table: notifications_queue

The notifications queue stores data on all pending messages. Entries are linked to the events and notifications tables for extra detail.

This queue is then processed on cron, and messages compiled into digests before being sent.

notifications_queue

Since users can manage their own notification intervals and delivery methods, we store one entry per event per user in the queue. These are then aggregated at each send.

SENDING

When a user is due a notification, all pending events are compiled into a digest..

The message content is themed, with all related events grouped together and duplicates removed.

Once the content is generated, control is passed to the messaging framework, which then handles how the content is delivered. (Email, private message, sms or any other available plugin.)



EVENT TRIGGER

function: notifications_event

Notifications are triggered via a call to notifications_event. This is triggered from some location in the code, normally, via an insert or update event in nodeapi.

Event data is immediately written to the event table, and then the subscriptions are queried and notifications are queued for any subscribed users.

EVENT STORE

table: notifications_event

The notifications event table keeps track of triggered events. This will assign an event an id, and store some of the basic properties of the event: who triggered it, was it an update or insert, and allow any other properties to be logged.

This table is cleared once all users waiting to be notified about it have been informed.

notifications_event



"Any" kind of event can be catered for. Events can be node related, user related, or anything else. Modules are able to define and respond to events of their own design.



QUEUE GENERATION

function: notifications_queue

When an event is triggered, the main notifications table is checked to see who should be informed. An entry is added to the queue for every subscribed user.

Notifications plugins implement the a hook to feed data to this queue process. The update is handled through one great big select...insert statement, so almost all the work happens on the database.

notifications

notifications_fields

SUBSCRIPTION DATA

tables: notifications & notifications_fields

Information about which users are subscribed to which events is stored in these two tables. The main notifications table keeps track of various "subscriptions", user, event, interval, message type etc, while the notifications_fields stores the specific details.

Multiple fields can be associated with each subscription, allowing subscriptions on a few different nodes, or multiple taxonomy terms.

notifications_queue



NOTIFICATIONS_UI

Most of the user interface for managing subscriptions is managed view the notifications_ui module.

Notifications modules can define what kind of information they offer, and the UI provides the tools for users to add and remove their subscriptions.

NODE TAXONOMY NOTIFICATIONS

This document has been produced as a result of our work to build nodetaxonomy_notifications, a module which allows subscriptions to be based on taxonomy terms assigned to a node.

The nodetaxonomy_notifications module has been designed to not use the normal notifications_ui, and uses taxonomy management in stead.

The Notifications and Messaging frameworks were created by Jose A. Rejero: <http://www.rejero.net>

The modules can be downloaded from: <http://drupal.org/project/notifications> <http://drupal.org/project/messaging>

Nodetaxonomy_notifications is available at: http://drupal.org/project/nodetaxonomy_notifications

This document was produced by School of Everything: <http://www.schoolofeverything.com>

school of everything

