

Notifications Revamp

With the new social features in Tiki 12

- [Activity Streams](#)
- Friendship Network revamp

The goal of the Notifications revamp is to standardize the current notifications feature and extend new functionality to make things more social and to continue to increase ideas found in the [Stickiness Project](#)

What exactly is a notification?

In our case, a notification is not the end result. In other words, its not the email, or SMS or popup that gets sent. The notification should be an object type that contains the basic series of information that could then be distributed in various formats to the user at whatever time the user wants.

What information should be stored in a Notification

- Name of notification
- Related object
- Trigger
- Associated User
- ??

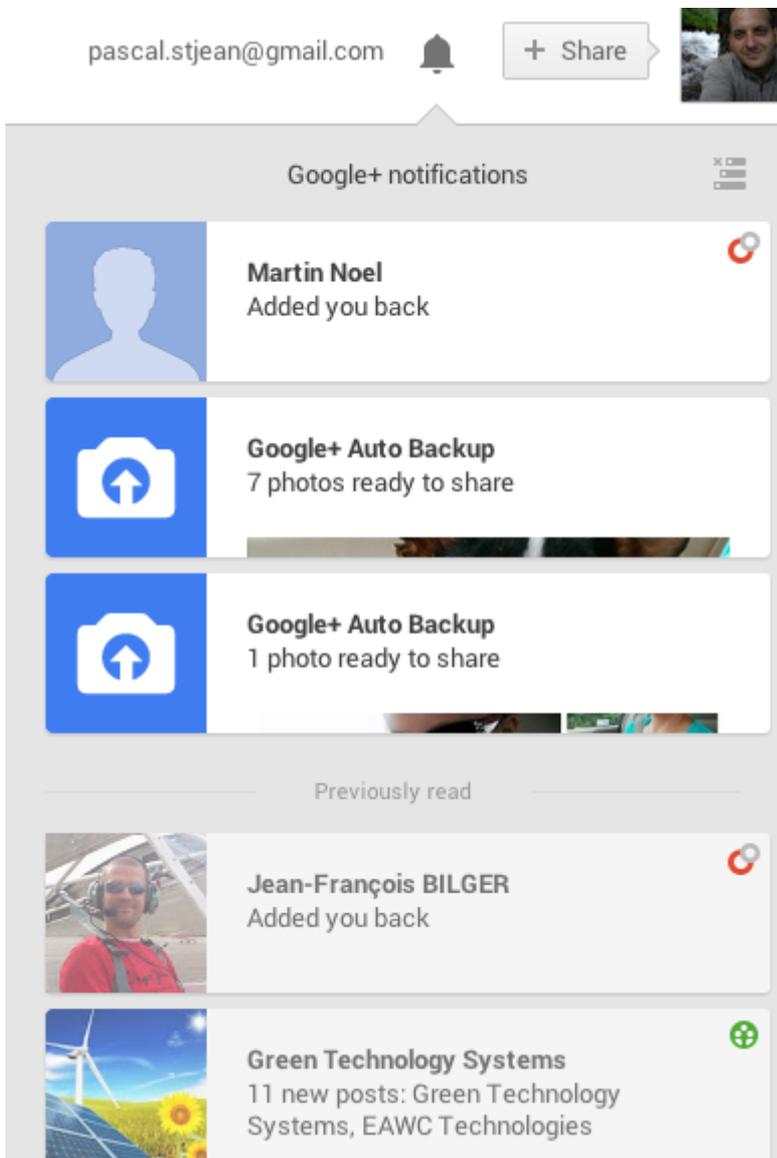
What triggers a notification?

- If an activity is triggered
 - You are watching an object
 - You are watching a friend
 - You are watching general activity stream
- If your name or group is mentioned in any content block
 - @username
 - @@groupname
- If a content gets categorized in a category you are watching
 - Hashtag categoryName
- An event in an Activity Stream (This will allow for more custom types of notifications rather than basic watches for tiki objects.)
 - <https://dev.tiki.org/Activity+Streams> [↗](#)
 - <https://doc.tiki.org/Activity+Stream> [↗](#)
- A goal or milestone was reached - <http://dev.tiki.org/Goals+-+Recognitions+-+Rewards> [↗](#)
- A reminder (ideally this would be a new feature in Trackers)

Notifications should be central to Tiki's interface

Notifications are becoming core to most operating systems (Desktop or Mobile). this is also something that is being added more and more to web applications. Tiki falls somewhere in between a Web OS and Web Application, we should therefore make Notifications central to our interface.

- See Google Plus bell or Facebook Notification window
- Items should be able to be marked as read



Allow any broadcast system to send out notifications

Once a notification is triggered, it should be able to be sent via

- E-mail
- Web Notifications
 - <http://www.html5rocks.com/en/tutorials/notifications/quick/>
 - <http://blog.brunoscopelliti.com/start-using-web-notification>
 - <http://www.dangercove.com/html5-notifications/>
 -
- XMPP
- XMPP to Push Notification
 - <http://stackoverflow.com/questions/6627415/how-to-implement-xmpp-to-send-push-notifications>
 - <http://stackoverflow.com/questions/16873115/xmpp-openfire-push-notifications-android>
- SMS

When can notifications be sent

The user should be able to choose how and when to receive notifications.

Possible Options

- As they happen

- Hourly
- Daily
- Weekly

Ideally it would be great to choose how and when independently.

Example

- Web Notification = As they happen
- Email = Daily
- etc...

Agile Development Cycles

phase 1 - creating notification

- common ajax widget that can be used to watch/unwatch/set as priority any object watches (replace all existing eye icon with this new widget)
- activity streams will have this widget as well (watch the stream)
- when goals are hit, this causes notification as well
- tracker status changes etc... (these are just activities right?)

Each notification needs to have a template that can be configured (pretty much same as current watch system). The new notification system can be backward compatible with current watches (old watches can still work).

A notification can be set as priority via the creation widget at time of creation or whenever later on.

Self opt-in.

Set digest email frequency for email containing all notifications that you have not acknowledged (could also include those that are acknowledged in different font/list). This will replace the existing digest system.

In addition, user can set that priority notifications are sent via email to them immediately (in future extensible to other sending mediums e.g. SMS, or audible signal on phone).

phase 2 - receiving notifications

Receiving notification.... instant bell notification of new message... goes into a message stream (if online). Whether online or not, it is stored in a message queue. Email and other notifications are simply rules to send stuff using different medium from the message queue.

Keeping track of read or not state is important.

phase 3 The email that goes out?

This should be similar to the activity stream display where there are microtemplates for each event which are embedded in a master email template. The goals are to be consistent consistent each email (unlike current Tiki). There are just 2 email templates - one for priority (immediate notif) and one for non-priority.

phase 4 - Timeshifted or date triggered notifications

There are some notifications that would need to be timeshifted and or date triggered. For example, tracker field notifications based on date field or other objects (and also user information such as last login) could have date fields that trigger notifications too.

These notifications would have their own UI, for e.g. the date field would have a UI there. But UI aside, the general notifications system needs to have way to keep track of date triggered notifications and send them at the right time (it is unclear if it should keep it's own internal store of the dates or regularly poll for the information from the source object, but whatever the case it needs to keep an up-to-date date info).

There are cases where the date would be timeshifted, for example, reminders would be sent x days before the date, while we might want to send emails x days after user's last login to encourage them to return to the site.

LP's Breakdown

New notifications will tie into the events and activity stream functionality.

There is a need for a flatter, more comprehensive model for the watches. Currently, each feature somehow defines how watches are being handled. It's never too clear what will happen when you watch a category or a structure. It should be possible for a user to watch page updates in a given category, or a specific page.

Internally, finding users needing a notification should be straight-forward and require little changes over time. Adding special notifications should be done by adding events, not modifying the notification handling logic.

Users should choose to watch events given certain conditions. Controlling watches should not be a simple toggle, but rather a modal dialog with options to choose from.

Events can be:

- Update, creation or both for given object types
- Categorization
- Custom events

Conditions can be:

- Direct object
- Part of a category
- Part of a structure
- In a specific forum
- In a specific blog
- A specific tracker
- A specific language

Note: All of the conditions could be implemented using custom events. However, there is a balance to be made with ease of set-up.

Edge cases should likely be implemented as custom events.

Priorities

Having an interest in a page does not mean you want your phone to buzz every time there is a modification to it. Users should be able to select what their level of interest is, and it should not default to annoying.

Proposed priorities:

1. Critical: Send an email right away
2. High: Include in a daily/weekly digest, include in notification tray
3. Low: Include in a recent changes stream

What is done for each priority could potentially be configured by the administrator.

Rendering

The activity stream already provides means to render events using event-specific templates. Currently, these templates typically are small and meant to be in a stream. These same templates are also suitable for a digest email.

However, such a small piece of information may look strange in a dedicated email notification. Perhaps the rendering could have a new standardized option to render extended information, which could include comments listed by default or other related information such as contributions (which are handled by watches for every imaginable case). This same extended view could be used when viewing a single activity (the like target from a digest email).

In a similar fashion, a summary rendering option could be suitable for a notification bar stream, or lower priority items in a digest.

Minor concerns

Notifications have little value long-term. An expiry should likely be added to the activity stream to purge older entries.

The target stream should likely be formalized in Tiki. It's currently possible through ad-hoc properties in custom events.

Notification tray UI is required.

Marking as read can be handled by keeping the last view date initially.

Installations without unified index would be limited to high priorities as everything else is just stream rendering.

Migration tools might be required. It could run on-demand from the admin panel or as a database patch, depending on how fast the watches would be deprecated.

Implementation of mentions is likely to require some thoughts. Including in activity streams should be trivial. Not sure email spam is desirable.

Related links

- <https://whimsical.com/how-slack-decides-to-send-a-notification-P53PjXCBWEi2fdMjP2Hti> ↗
- <https://laravel-notification-channels.com/> ↗