Building communities using Userpoints

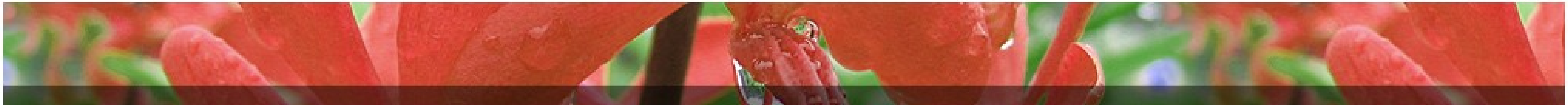
Khalid Baheyeldin

Open Craft Tech Seminar

Cairo, Egypt

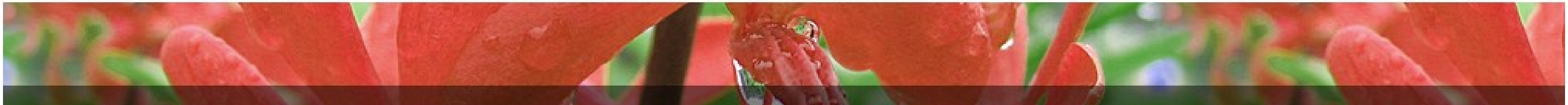
August 25, 2007

http://2bits.com



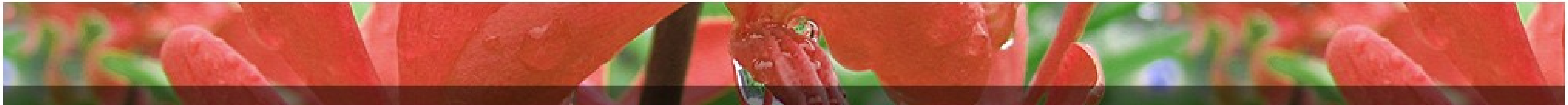
Agenda

- Introduction
- Userpoints Module
  - Features
  - Usage
- Userpoints API
- Hands on
- Conclusion
  - Questions, discussion



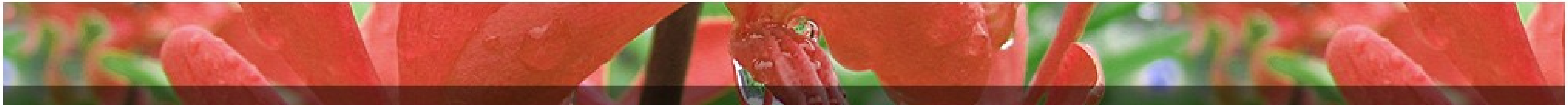
About 2bits

- Based in Waterloo, Ontario
- Active member of the Drupal community since 2003
- Member of security and infrastructure teams
- 25+ modules on drupal.org
- Listed on Drupal.org's service providers section
- Maintain modules that run on drupal.org (donations, feature, lists, ...)
- Google Summer of Code mentoring (2005, 2006, 2007)



2bits Services

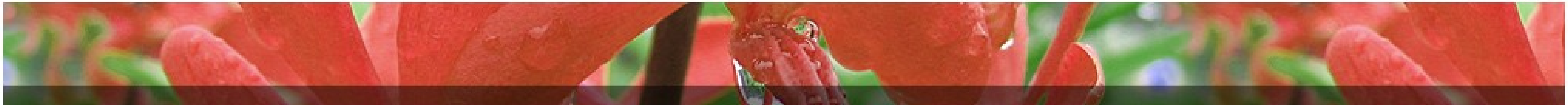
- Clients mainly in the USA and Canada
- Subcontracting development projects
- Customization of existing modules
- Development of new modules
- Installation, upgrades
- Automated backups
- Performance tuning and optimization



About Khalid

- Developing for computers for way too long (22 years), Drupal since 2003
- Core contributions
  - Site maintenance feature
  - Logging and alerts in Drupal 6
  - Many patches
- Member of
  - Drupal security team
  - webmasters team
  - infrastructure team
- Co-founder of 2bits
- Blog at http://baheyeldin.com

- Contributed modules
  - Adsense
  - Userpoints
  - Nodevote
  - Job search
  - Favorite nodes
  - Flag content
  - Stock API and module
  - Custom Error
  - Currency
  - Image watermark
  - Site menu
  - Email logging and alerts
  - Second Life
  - Technorati
  - Click thru
  - Referral

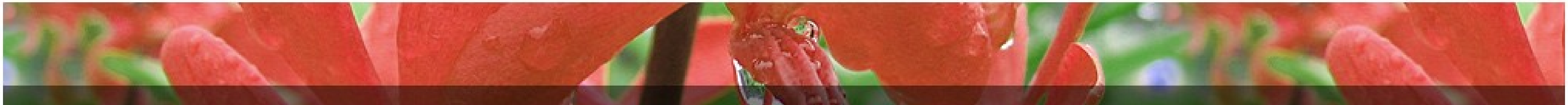


History

- Commissioned by a photography and art site
- As an incentive for promoting participation
- Released to the community

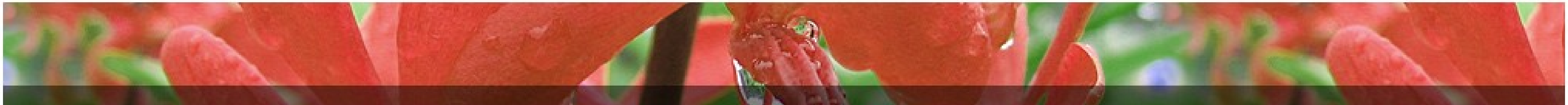
Features and Benefits

- Psychological factor
- Mimics “gaming”
- Provides instant gratification
- Users often strive for more recognition, as well as competition.
- Users can be sort of tiered within the site
- Gifts and awards



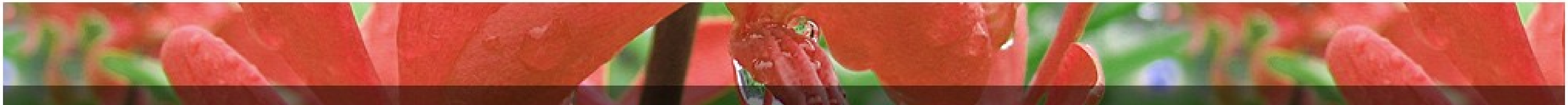
Functionality

- Basic package has two modules
- userpoints
  - Just an API, and management module.
- userpoints_basic
  - Basic core Drupal functionality (posting nodes, comments, ...etc.), as well as invite module



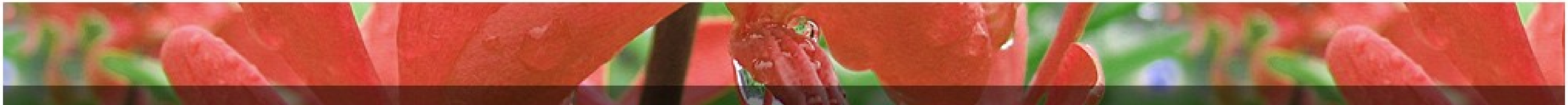
Configuration

- Point moderation
  - Whether points are to be added/subtracted in real time, or delayed until approval.
  - Prevents abuse on sites that use points as a means for getting gifts and such.
  - Under Administer -> User management -> Points
- Allows adding of points manually by the admin



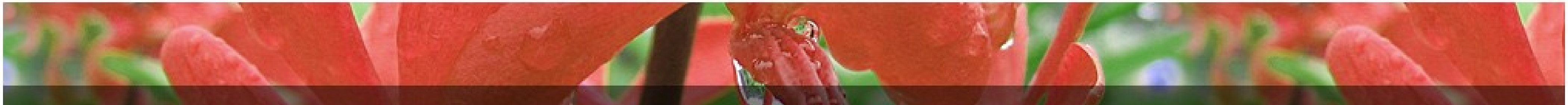
Configuration (cont'd)

- Branding of points
  - Allows calling them something else, for example “brownies”, or some site specific jargon.



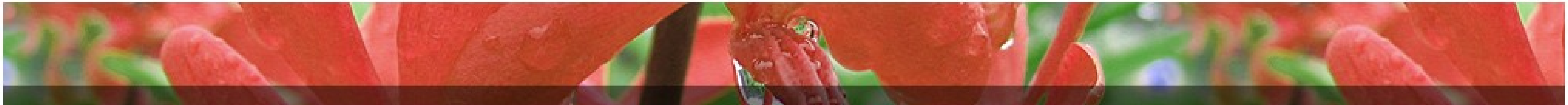
Configuration (cont'd)

- Basic events
  - Posting a node
  - Each type (blog, page, story, cck, ...etc.) can have a different value.
  - Posting a comment
  - Moderating a comment
  - No fractional points please!



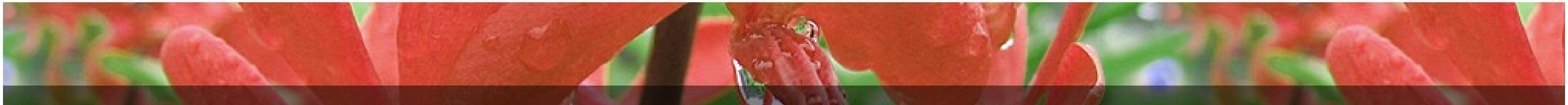
Contributions

- Mini ecosystem, like views
- Written by various people for various things
- Following is an overview of some of them



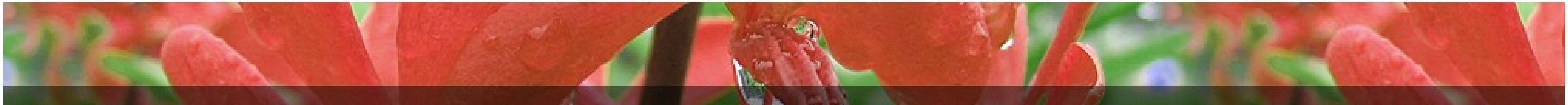
Contributions (cont'd)

- Userpoints Reset
  - Resets all points for all users
  - Destructive
  - One time operation
- Userpoints Retroactive
  - Only for nodes and comments
  - Nothing else



Contributions (cont'd)

- Ecommerce
  - Users earn points for purchasing from the ecommerce store.
  - Points can also be used as a payment form
- Node limit
  - Allows limiting of node creation depending on how many points a user has.
  - Can be used for subscriptions

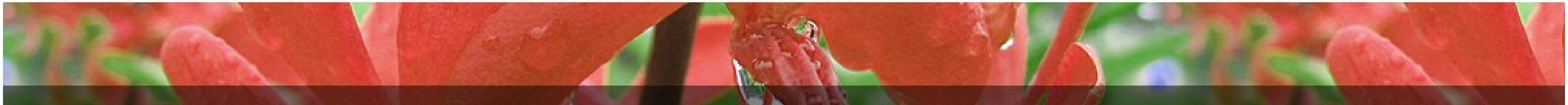


Contributions (cont'd)

- Referral
  - Users earn points for referring a user.
  - Users earn points for actions performed by users they referred.
  - A multiplier is used for the calculation.
- Invite
  - Interface for invite module
  - Points for inviting, and registering

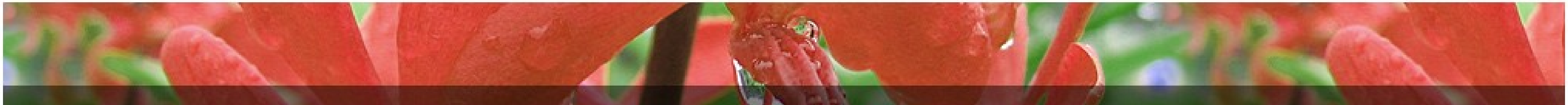
Contributions (cont'd)

- Role
  - Users join/leave roles as they gain/lose points
  - Can be used to allow access to other areas of the site, or more privileges
- Role exempt
  - Certain roles are exempted from points, e.g. admins or moderators, so they do not “compete” with regular users.



Contributions (cont'd)

- Email notify
  - Users are notified by email when they reach a certain level of points
- Inactive expiry
  - Inactive users lose a certain number of points every preset period (can you think of banks?)

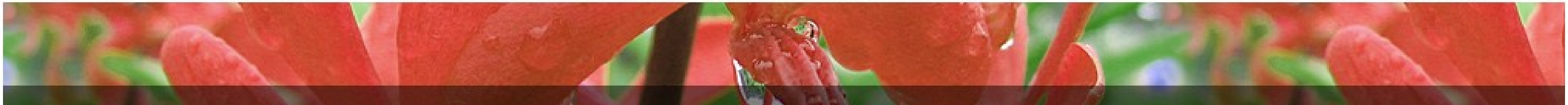


Contributions (cont'd)

- Periodic expiry
  - All points for all users are expired every certain period.
  - Good for contests and the like.
- Autoapprove
  - Automatically approves transactions after a certain period

Contributions (cont'd)

- Userpoints Cap
  - Limit the number of points a user can earn.
- Reserve
  - Sets exchange rate for points depending on cash in Ecommerce

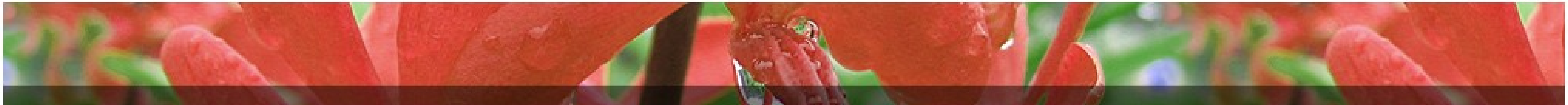


Userpoints API

- Simple API for modules to use
- Documented in the README.txt
- Your module decides to award points at certain events.
- Your module can act on other modules awarding points
- Examples abound in the contrib directory

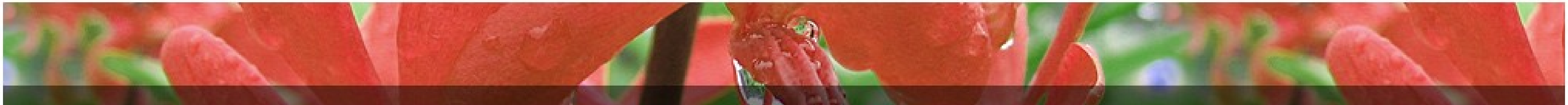
Userpoints hook

- hook_userpoints($op, $points, $uid, $event)
  - $op is operation type
  - $points is number of points to add (or subtract, if negative)
  - $uid is the Drupal user id of the user
  - $event is an event identifier



$op

- $op can be one of:
  - 'setting'
    - Build your configuration form, and return it, and they will automatically appear in userpoints settings.
  - 'points before'
    - Calls your modules (and other modules) before the points are awarded. Returning FALSE prevents points from being awarded.

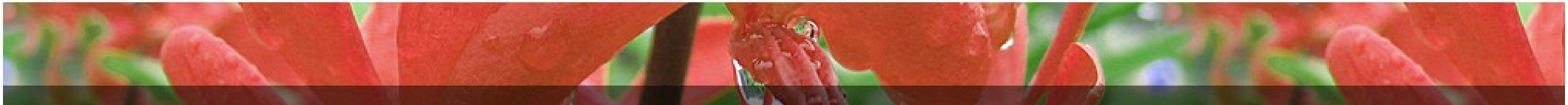


$op (cont'd)

- 'points after'
  - Points have been awarded, your module can take actions on that event

API function call

- userpoints_userpointsapi('points', $points, $uid, $event, $description, $reference)
  - Modules call this function explicitly to add/subtract points to a user.
  - Points are added (if positive) or subtracted (if negative)



API call (cont'd)

- $event
  - An identifier of the event the points is being awarded for, this
  - Gets recorded in the transaction table (allows auditing, recalculating, ...etc).
  - May require events to be registered in the future.

How many?

- userpoints_get_current_points($uid)
  - You can call this function to know how much points a use has.
  - Returns the current number of points for that user.
- Use in themes or other modules for displaying points (e.g. below the avatar).
- Some examples on http://2bits.com

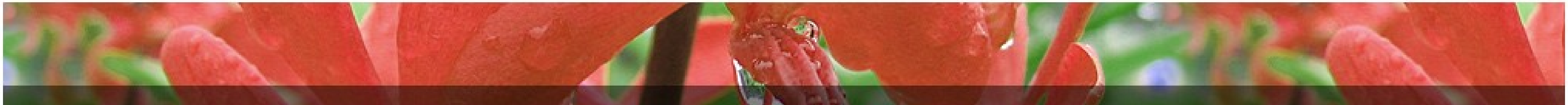
Real Life Example

- 7 am Thursday someone sent me an email asking for an unreleased module, and two other custom ones to be developed.
- Replied around noon.
- Got this reply by 2pm:

Thanks to your API, it turned out much easier than expected. I was able to code all 3 by the time you responded to this email and will post them on drupal.org for you to review. Great module! :)

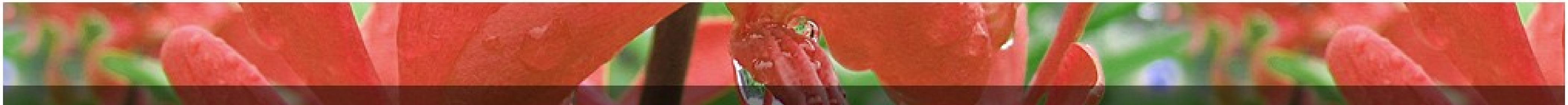
The future?

- Maximum ever points earned
  - field in the database already, but not presented, and may not be calculated correctly now
- ~~Points from one user to another~~
  - ~~Barter or site currency~~
- More than one point “pools”.
- Time dimension
  - How many points for a user in the last month



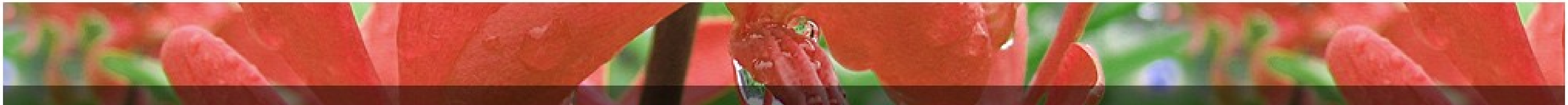
The future (cont'd)

- Integration with votingAPI
  - Vote Up/Down already uses it.
  - Need more, eg. fivestar, ...etc.
- Event registry
  - More structure and discipline for the data
- Better UI for managing points (AJAX?)



API redesign

- API 3.0
  - Arguments to be an array
  - Sensible defaults (e.g. if uid not specified, then current user)
  - More here http://drupal.org/node/153626



Sites

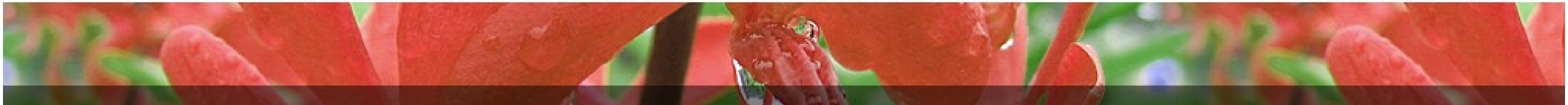
- Performancing – Tools for bloggers

  http://performancing.com
- Altiris users site – Community for Altiris users
  http://juice.altiris.com/points
- MEPIS Linux distro, Debian derived

  http://www.mepis.org/userpoints
- Poop Report – an intellectual scatology site

  http://www.poopreport.com
- Ruby Baboon

  http://www.rubybaboon.com/



Hands On

Let us see it all in action ...

Conclusion

- Questions?
- Comments?
- Discussions?