The Rules Way of Life

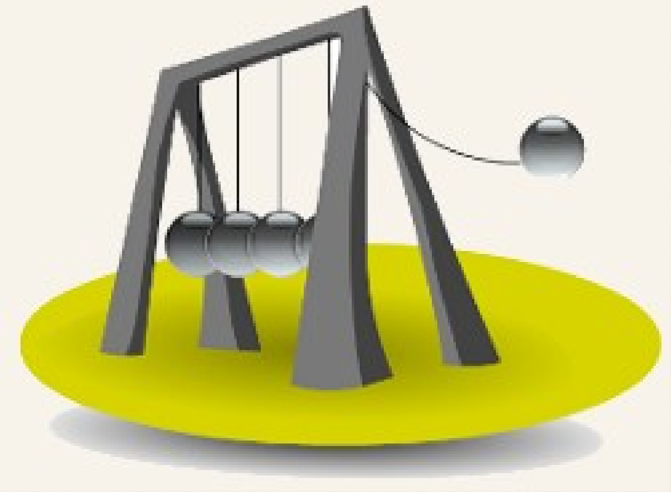
Presented by Wolfgang Ziegler, Johan Falk

Wolfgang Ziegler // d.o. fago
wolfgangziegler.net
twitter.com/the_real_fago

Johan Falk // d.o. Itangalo
nodeone.se/johan-falk
twitter.com/itangalo

Rules?

http://drupal.org/project/rules



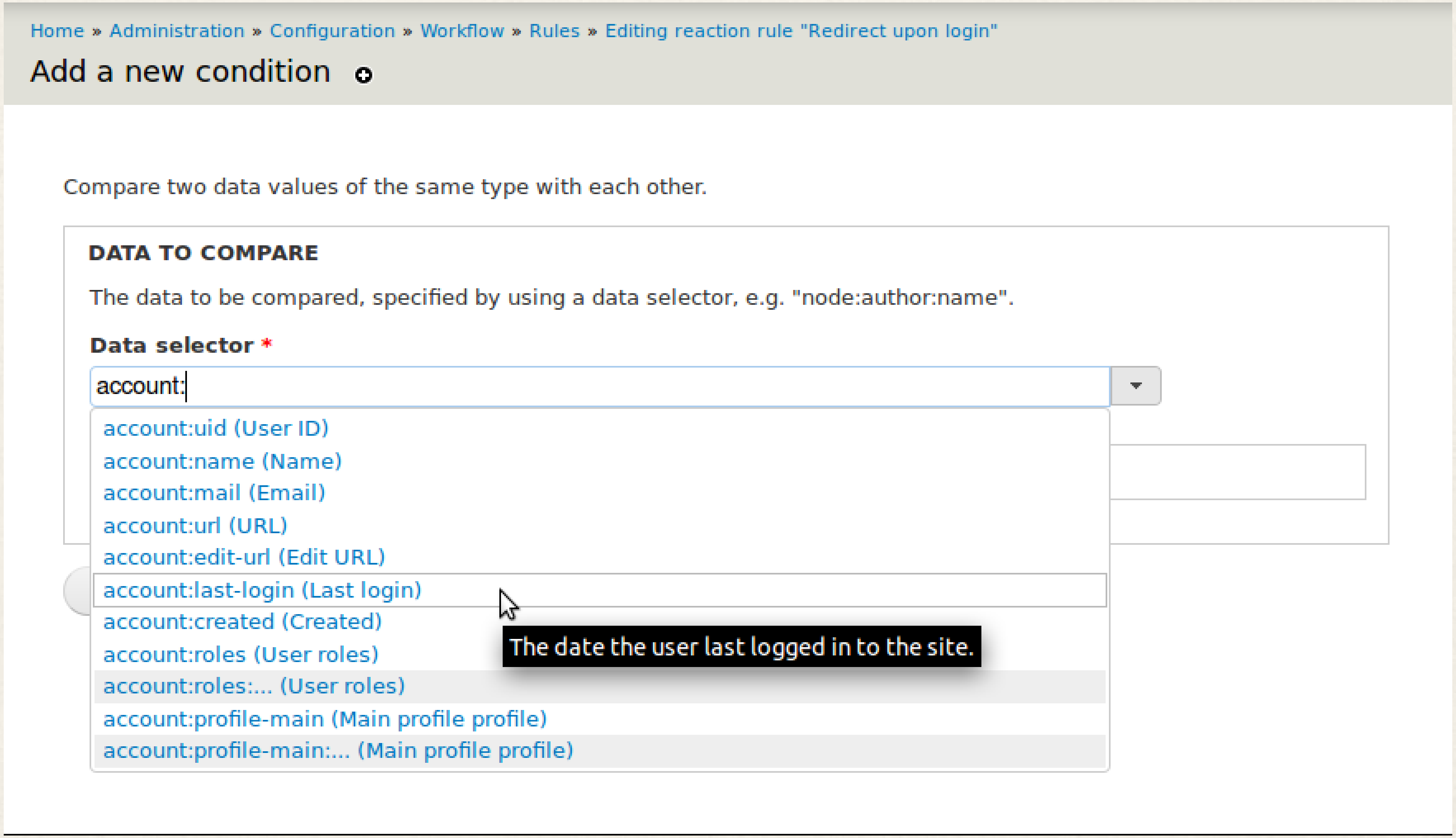
Reactions

Event-Condition-Action

Rules 2.0-rc1 for Drupal 7

What's new?

Data selection



Home » Administration » Configuration » Workflow » Rules » Editing reaction rule "Redirect upon login"

Add a new condition

Compare two data values of the same type with each other.

DATA TO COMPARE

The data to be compared, specified by using a data selector, e.g. "node:author:name".

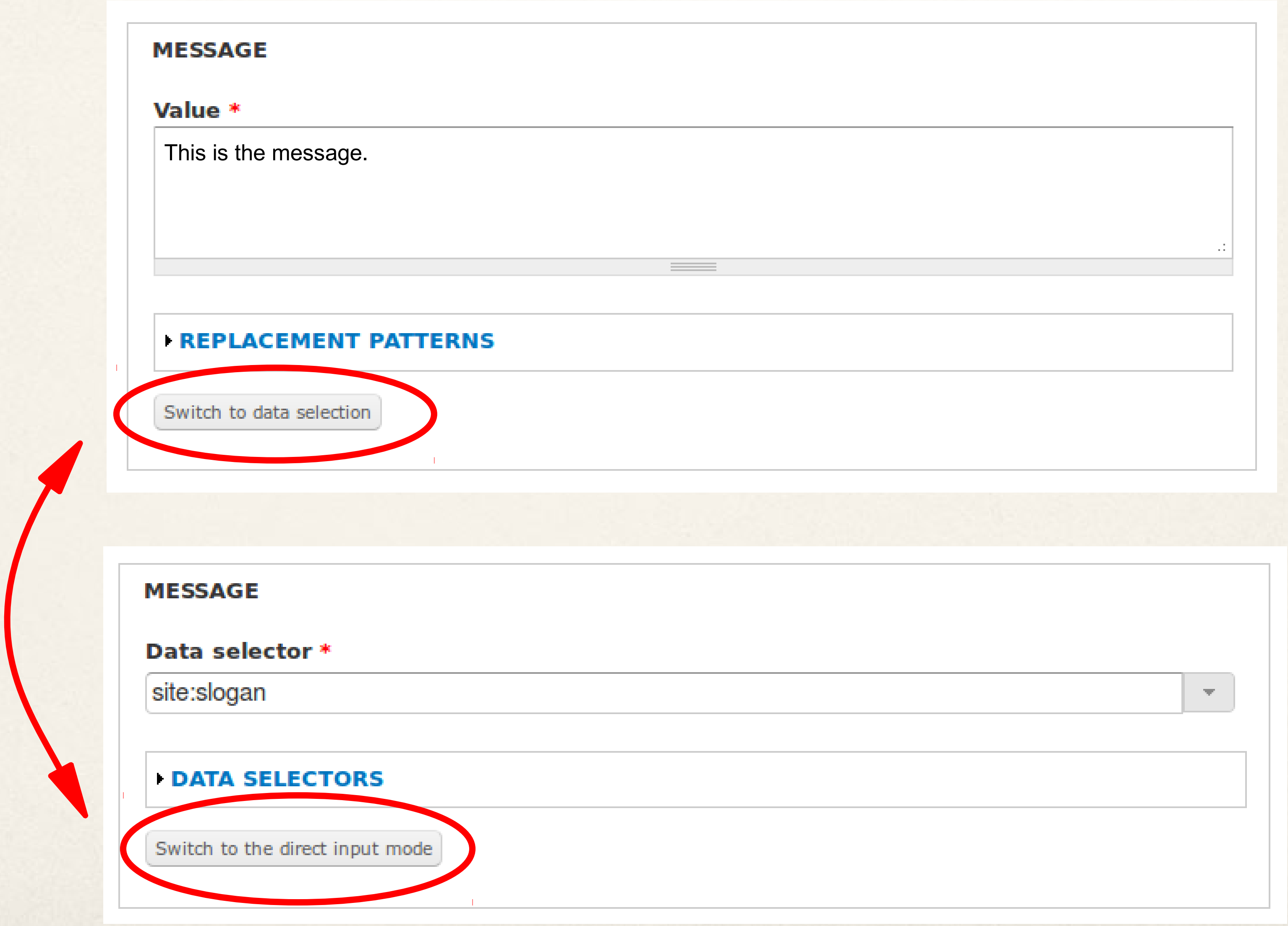
Data selector *

account:

- account:uid (User ID)
- account:name (Name)
- account:mail (Email)
- account:url (URL)
- account:edit-url (Edit URL)
- account:last-login (Last login)
- account:created (Created)
- account:roles (User roles)
- account:roles:... (User roles)
- account:profile-main (Main profile profile)
- account:profile-main:... (Main profile profile)

The date the user last logged in to the site.

Switch parameter input modes



MESSAGE

Value *

This is the message.

MESSAGE

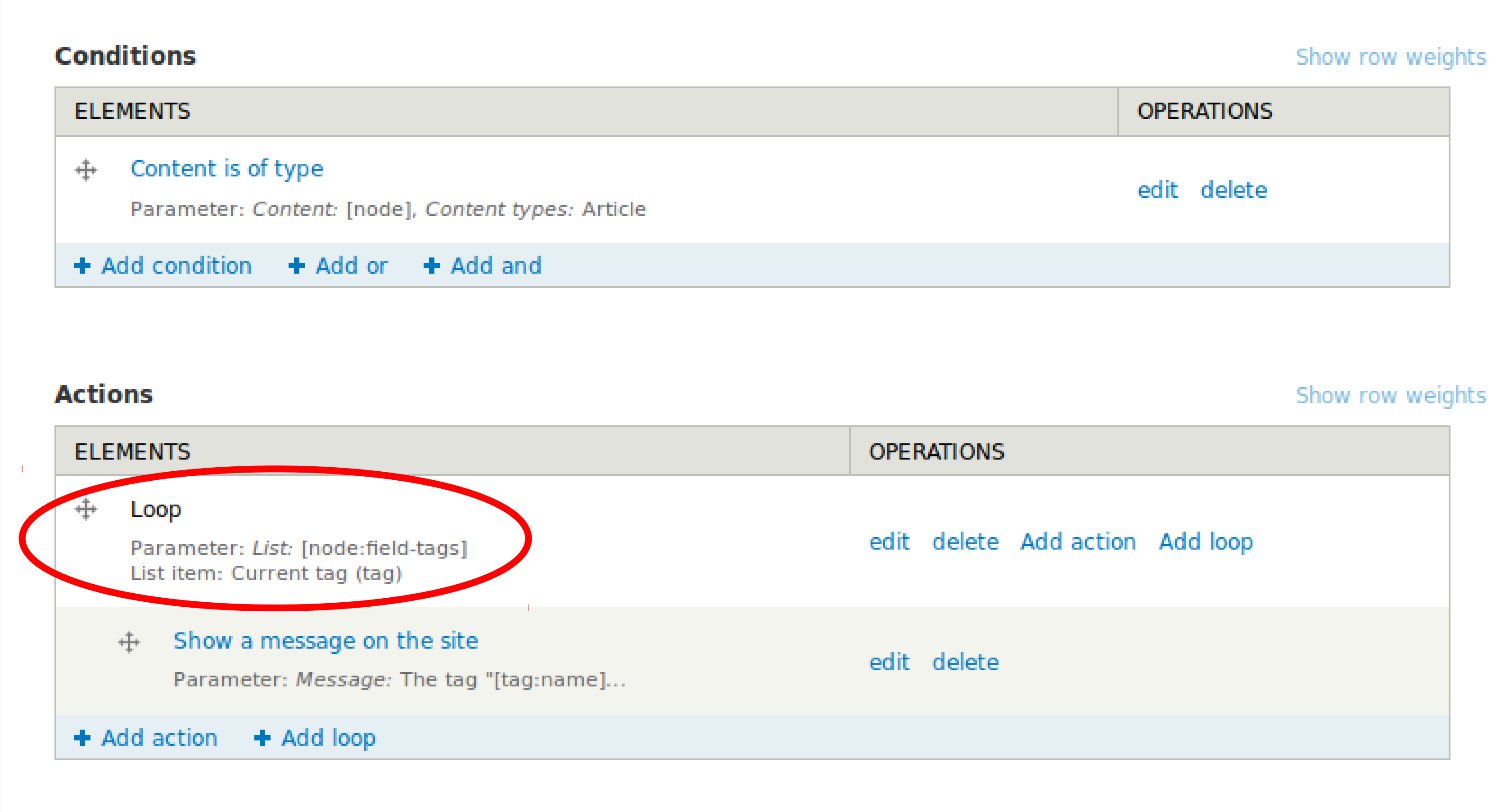
Data selector *

site:slogan

Create components!

- Reusable configurations.
- Rule sets, Action sets, Condition sets
- Use them in reaction rules or other modules.

Loops & Lists



Conditions

Show row weights

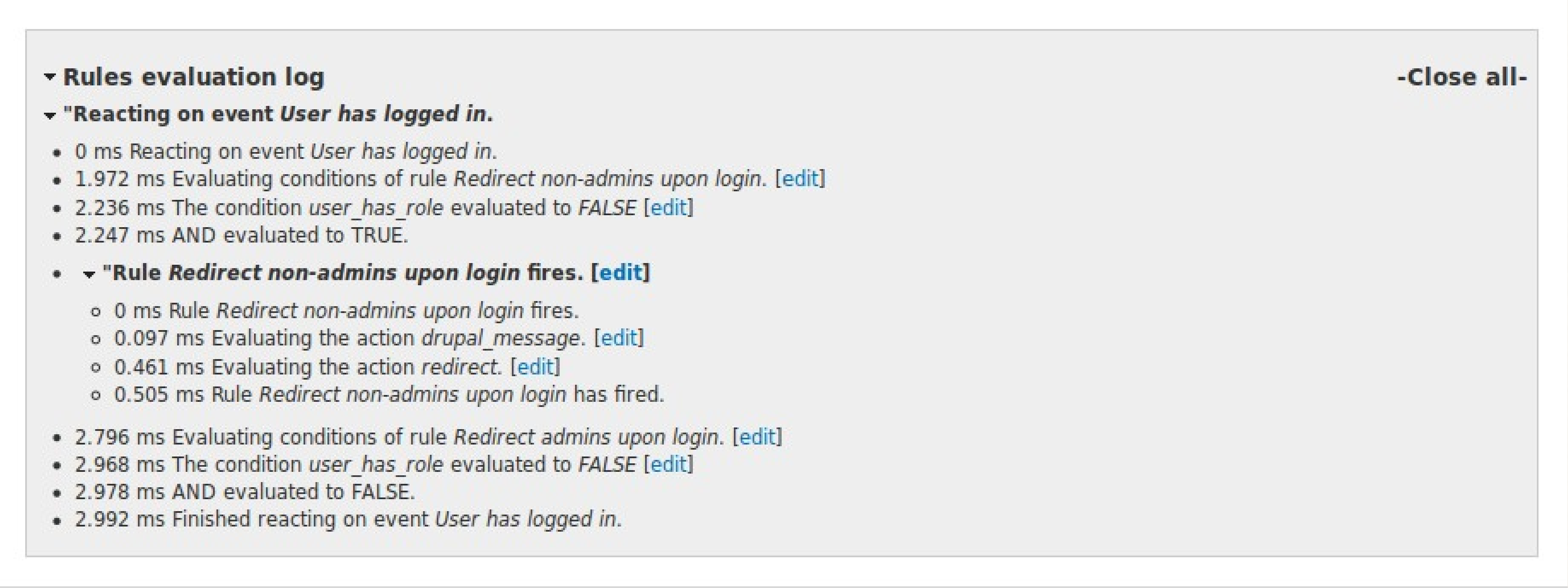
ELEMENTS	OPERATIONS
Content is of type Parameter: *Content:* [node], *Content types:* Article	edit delete
+ Add condition + Add or + Add and	

Actions

Show row weights

ELEMENTS	OPERATIONS
Loop Parameter: *List:* [node:field-tags] List item: Current tag (tag)	edit delete Add action Add loop
Show a message on the site Parameter: *Message:* The tag "[tag:name]...	edit delete
+ Add action + Add loop	

Improved debug log



▾ Rules evaluation log **-Close all-**

▾ "Reacting on event *User has logged in.*

- 0 ms Reacting on event *User has logged in.*
- 1.972 ms Evaluating conditions of rule *Redirect non-admins upon login*. [edit]
- 2.236 ms The condition *user_has_role* evaluated to *FALSE* [edit]
- 2.247 ms AND evaluated to TRUE.
- **▾ "Rule *Redirect non-admins upon login* fires. [edit]**
    - 0 ms Rule *Redirect non-admins upon login* fires.
    - 0.097 ms Evaluating the action *drupal_message*. [edit]
    - 0.461 ms Evaluating the action *redirect*. [edit]
    - 0.505 ms Rule *Redirect non-admins upon login* has fired.
- 2.796 ms Evaluating conditions of rule *Redirect admins upon login*. [edit]
- 2.968 ms The condition *user_has_role* evaluated to *FALSE* [edit]
- 2.978 ms AND evaluated to FALSE.
- 2.992 ms Finished reacting on event *User has logged in.*

The Rules way of life

Drupal is like Lego

Nodes, Fields, Views, …

+

glue code

Drupal is like Lego

Nodes, Fields, Views, …

+

Rules

Drupal is like Lego

Nodes, Fields, Views, …

+

Rules

+

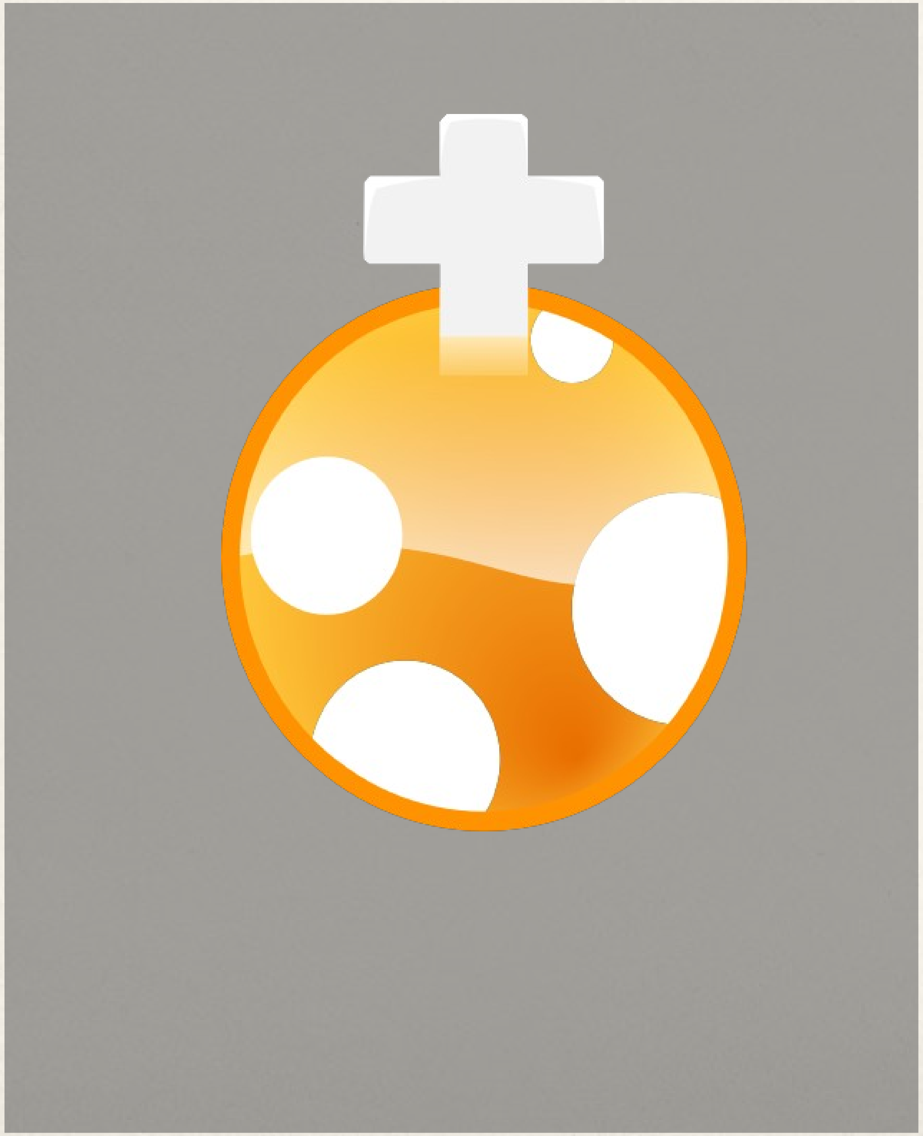
less glue code

Interacting modules.

"The whole is more than the sum of its parts."

– Aristotle

The holy four!



i. Views

ii. Flag

iii. Rules

iv. Message

Modules to be **replaced**...

Scheduler, Workflow, Pathauto,
Notifications, Messaging,
Casetracker, Custom breadcrumbs,

...

your custom module?

Is this a good idea?

Let's do „Feature modules“?

Modules

- have more dependencies
- become harder to understand

→ Just as the sites we build

Feature modules

→ Re-use existing stuff

→ Less work

→ Less code to maintain

→ More flexible solutions

→ Easy to adapt!

Share and remix
custom-tailored solutions!

Examples

Mark as spam

- Flag (Mark as spam / Mark as ham)
- Rules:
    - Unpublish content marked as spam
    - Block spam users
    - Notify moderator about new content
- Messages for spam history

Simple workflow

- Workflow state field
  (draft, needs-review, published)
- Field permissions
- Message + Rules for logging changes
- Views for content listings, message logs.
- Rules for publishing, notification

Scheduled publishing

- Date fields for publishing + unpublishing date
- Rules Components for
  (Un-)Publishing + Logging
- Reaction rules that
  - detect changed (un-)publishing dates
  - schedule component evaluation

Notifications

- Flag „Subscribe to a node“
- Rule „Notify subscribers of changes“
- Rules component „Notify a user“
- User account field „Notification settings“

  e.g. for choosing messaging channel

Demo

Best practices...

- Make use of **tags**!
- Use **components** to organize your rules
- Export rules to feature modules using **Features**
- Fear the devil: **PHP** module.
- Keep your rules simple

Useful modules

Views Bulk Operations

http://drupal.org/project/views_bulk_operations

Execute Rules components on Views-results.

Rules Link

http://drupal.org/project/rules_link

Create custom links which trigger rules.

Content Access

http://drupal.org/project/content_access

Grant role-based access to nodes with rules.

Rules Bonus Pack

http://drupal.org/project/rb

A greenhouse for Rules functionality.

Developing with Rules

Extend Rules

- Provide events, conditions and actions
- Define entity properties using hook_entity_property_info of the entity API
- Provide new entity types

Make use of the Rules API

- Actions API
- Conditions API
- Provide configurable components
- Embed the UI, write tests!

Developments & Outlook

- Usability
- Development ideas
  - Views, ctools, Entity forms, Web hooks
- Drupal 8
  - Entity API with Entity Property API
  - Condition, Action API

Thanks!

Questions?