

Rules: Duct Tape for Drupal

Ryan Price
Blink Reaction
@liberatr

slides.com/ryanprice

@liberatr

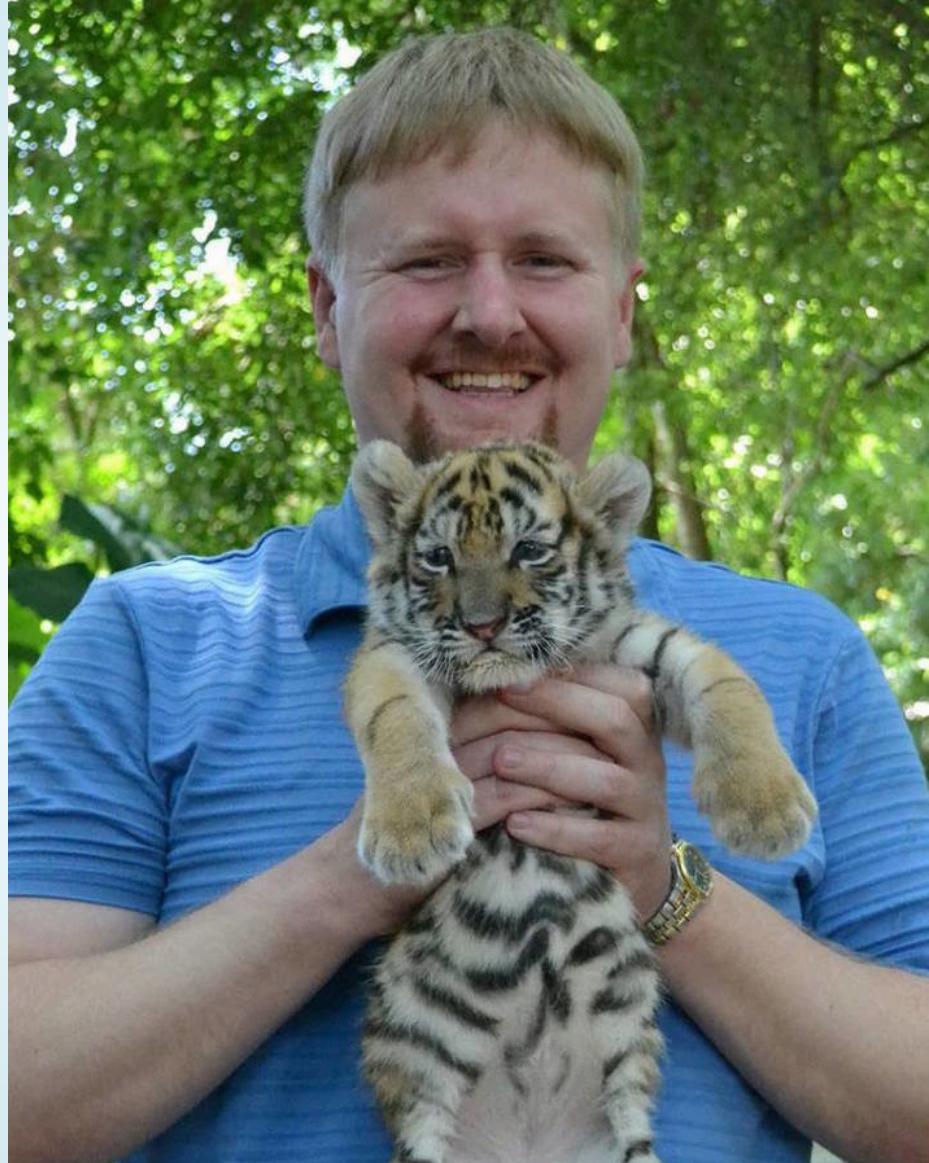
9 years in Drupal, 15 in web

Orlando, FL

FLDrupalCamp.org

April 11-12, 2015

Faves: Theatre, Pottery,
Photography, Podcasts, Beer,
Coworking, Mentoring, and
Open Source



blink reaction



WE'RE
HIRING

WE
PROMISE
OFFER
NEED

exciting challenges
rapid growth
expert drupalists



blink reaction

Digital Experiences That Deliver Results

DrupalEasy Podcast

DrupalEasy.com/podcast

@DrupalEasy

Facebook

iTunes, Stitcher, Miro or
your favorite podcast app.



BATTideas.com

art & technology

OurYellowHouse.com

community & work

**bringing
(the) arts +
technology
together
batt[south]**



Lashing

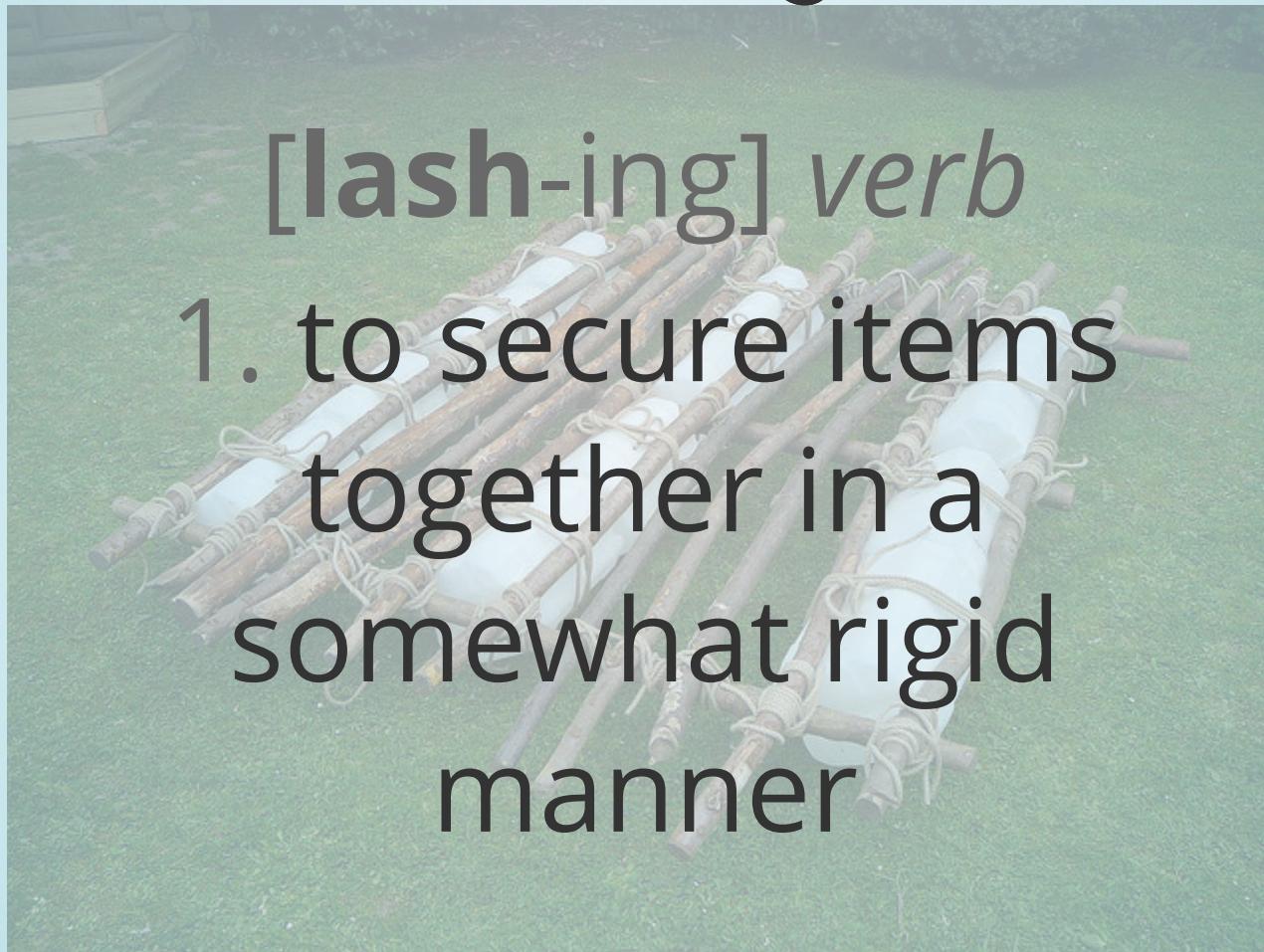


flickr.com/photos/30915596@N00/14252674920/

Lashing

[lash-ing] *verb*

1. to secure items
together in a
somewhat rigid
manner



flickr.com/photos/30915596@N00/14252674920/

Lashing

- non-permanent
- utilitarian
- improvised
- re-usable materials
- methodical



flickr.com/photos/30915596@N00/14252674920/

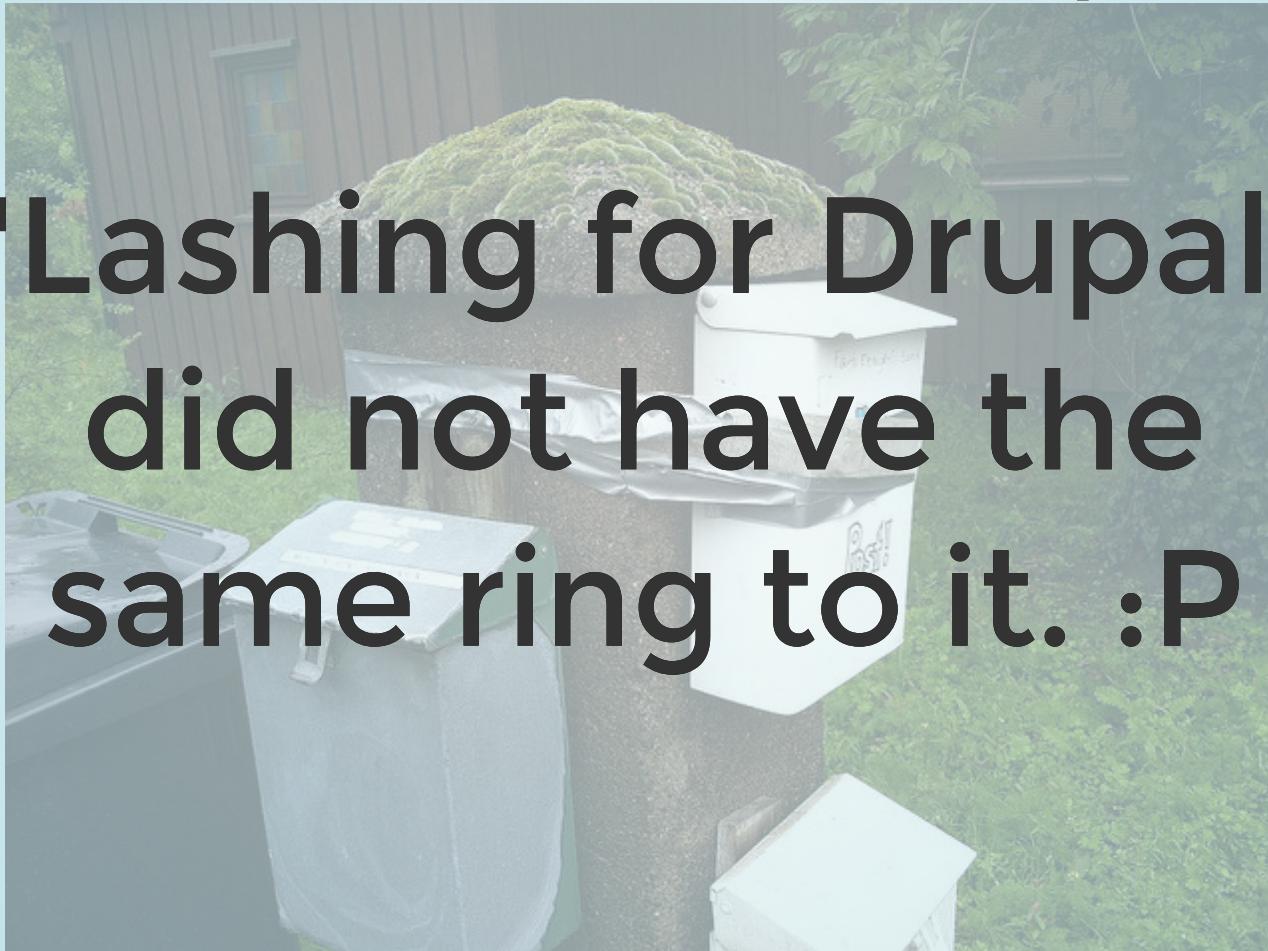
What about Duct Tape?



flickr.com/photos/fontourist/8652036894/

What about Duct Tape?

"Lashing for Drupal"
did not have the
same ring to it. :P



flickr.com/photos/fontourist/8652036894/

Duct Tape



- non-permanent
- utilitarian
- improvised
- *~~re-usable materials~~*
- *~~methodical~~*

flickr.com/photos/fontourist/8652036894/

Duct Tape!



flickr.com/photos/joshey/190337619/

Rules

**Drupal is not about
writing code**

**Drupal is all about
features**

(not those features)

Dries Buytaert

*"using the internet
to eliminate
middlemen"*

the programmer is a
middleman between
your data and your
business

not knowing how to
code can be as bad
as having as having a
middle man

How to: Rules

1. Deconstruct
2. Modify
3. Construct
4. Compose
5. Trigger

Events

EVENT	OPERATIONS
After saving a new comment	delete
+ Add event	

Conditions

ELEMENTS	WEIGHT	OPERATIONS
None	if this is true	
+ Add condition + Add or + Add and		

Actions

[Show row weights](#)

ELEMENTS	OPERATIONS
 Send mail Parameter: <i>To: mail@urbanrethink.com, Subject: New comment on [comment:node], Message: http://urbanrethink.com..., From: mail@urbanrethink.com</i>	edit delete
+ Add action + Add loop	

Event:

before I leave for work

Condition:

if there are dark clouds in the sky

Action:

bring the umbrella from the hall

Event:

after I leave for work

Condition:

if the day is not Friday

if the time is before 8:45am

Action:

get credit card from wallet

purchase coffee with credit card

Events

EVENT	OPERATIONS
User has become a group member	delete
+ Add event	

when this happens

Conditions

[Show row weights](#)

ELEMENTS	OPERATIONS
+ Data comparison Parameter: <i>Data to compare</i> : [og-membership:state], <i>Data value</i> : Active	edit delete
+ Entity is of type Parameter: <i>Entity</i> : [og-membership:group], <i>Entity type</i> : Node	edit delete
+ Add condition + Add or + Add and	

if both of these are true

Actions

[Show row weights](#)

ELEMENTS	OPERATIONS
+ Add a variable Parameter: <i>Type</i> : Node, <i>Value</i> : [og-membership:group] Provides variables: Added variable (group)	edit delete
+ Send mail Parameter: <i>To</i> : [account:mail], <i>Subject</i> : Your membership request was..., <i>Message</i> : [account:name], ...	edit delete
+ Send mail Parameter: <i>To</i> : [group:author:mail], <i>Subject</i> : [account:name] joined ..., <i>Message</i> : [group:author:name], ...	edit delete
+ Add action + Add loop	

set up this data

perform this action with data

Modify

(demo)

Events

EVENT	OPERATIONS
A resource conflict node form is validated	delete
+ Add event	

Conditions

[Show row weights](#)

ELEMENTS	OPERATIONS
⊕ Contains a resource conflict Parameter: <i>Node</i> : [node]	edit delete
+ Add condition + Add or + Add and	

Woah, not sure what this rule does...

Actions

[Show row weights](#)

ELEMENTS	OPERATIONS
⊕ Show a message on the site Parameter: <i>Message</i> : Test	edit delete
⊕ Load a list of conflicting nodes Parameter: <i>node to check for conflicts</i> : [node] Provides variables: Conflict List (conflict_list)	edit delete
⊕ Loop Parameter: <i>List</i> : [conflict-list] List item: Current list item (list_item)	edit delete Add action Add loop
⊕ rule: Check venue Parameter: <i>Conflicted Node</i> : [list-item], <i>Unsaved Node</i> : [node], <i>Form</i> : [form]	edit delete edit component
+ Add action + Add loop	

I can edit this "Test" message though.



Construct

(demo)

**If I unpublish a node,
delete all the comments**

What Event?

Saving a node

What Condition?

If it's unpublished

What Action?

Delete the comments

Until we learn to use "Loop", we have to
delete comments one-by-one

/admin/config/workflow/rules/reaction/add

Name *
If I unpublish a node, delete all the comments Machine name: if_i_unpublish_a_node_delete_all_the_comments [\[Edit\]](#)

Tags
demo
Tags associated with this configuration, used for filtering in the admin interface. Separate multiple tags with commas.

React on event *

- Select -

- Select -

Comment

- A comment is viewed
- After deleting a comment
- After saving a new comment
- After updating an existing comment
- Before saving a comment

Node

- After deleting content
- After saving new content
- After updating existing content** is triggered. 
- Before saving content
- Content is viewed

OG membership

- After deleting a og membership
- After saving a new og membership
- After updating an existing og membership
- Before saving a og membership
- OG role granted to user
- OG role revoked from user

When this happens...

Events

EVENT	OPERATIONS
After updating existing content	I have an event delete
+ Add event	

Conditions

ELEMENTS	WEIGHT	OPERATIONS
None		
+ Add condition + Add or + Add and		

Actions

ELEMENTS	WEIGHT	OPERATIONS
None		
+ Add action + Add loop		



Add a Condition for Unpublished

Select the *condition* to add

Data comparison

Data

- Data comparison
- Data value is empty
- List contains item
- List count comparison
- Text comparison

Entities

- Entity has field
- Entity is new
- Entity is of bundle
- Entity is of type
- User has field access

Node

- Content is of type
- Content is promoted to frontpage

Content is published

- Content is sticky

Organic groups

- Entity is group
- Entity is group content

What about
"unpublished"?

CONTENT

Data selector *

node



The data selector helps you drill down into the data available to Rules. To make entity fields appear in the data selector, you may have to use the condition 'entity has field' (or 'content is of type'). More useful tips about data selection is available in [the online documentation](#).

Data types: Select data of the type *Node*.

► DATA SELECTORS

[Switch to the direct input mode](#)

Negate



I want the opposite of published

If checked, the condition result is negated such that it returns TRUE if it evaluates to FALSE.

[Save](#)

Events	
EVENT	OPERATIONS
After updating existing content	delete
+ Add event	

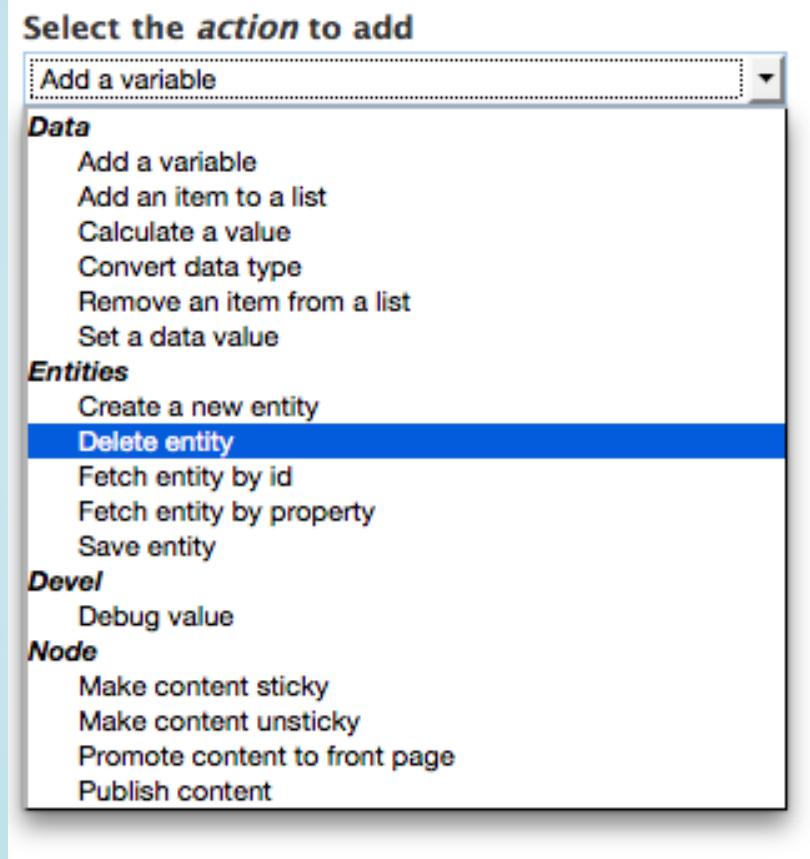
Event and Condition

Conditions	
ELEMENTS	OPERATIONS
⊕ NOT Content is published Parameter: <i>Content: [node]</i>	edit delete
+ Add condition + Add or + Add and	

Actions		
ELEMENTS	WEIGHT	OPERATIONS
None		
+ Add action + Add loop		



I Need an Action to Perform



There is no Action referring to Comments



Comments are Entities, though

TODO

ENTITY

Specifies the entity, which should be deleted permanently.

Data selector * What is all of this?

The screenshot shows a dropdown menu with the following options:

- site:... (Site information)
- node (updated content)
- node:... (updated content)
- node-unchanged (unchanged content)
- node-unchanged:... (unchanged content)

To the right of the dropdown, there is explanatory text: "es. To make entity fields appear in the node's 'content is of type' field. More useful tips".

DATA SELECTORS

ENTITY

Specifies the entity, which should be deleted permanently.

Data selector * 

The data selector helps you drill down into the data available to Rules. *To make entity fields appear in the data selector, you may have to use the condition 'entity has field' (or 'content is of type'). More useful tips about data selection is available in [the online documentation](#).*

Data types: Select data of the type *Any entity*.

► [DATA SELECTORS](#)

I can drill down into a Node
and one comment at a time

Events

EVENT	OPERATIONS
After updating existing content	delete
+ Add event	

Event, Condition and Action

Conditions

[Show row weights](#)

ELEMENTS	OPERATIONS
⊕ NOT Content is published Parameter: Content: [node]	edit delete
+ Add condition + Add or + Add and	

...but only deletes the first comment

Actions

[Show row weights](#)

ELEMENTS	OPERATIONS
⊕ Delete entity Parameter: Entity: [node:comments:0]	edit delete
+ Add action + Add loop	



I'll add a Loop

LIST

The list to loop over. The loop will step through each item in the list, allowing further actions on them. See [the online handbook](#) for more information on how to use loops.

Data selector *

Drill down to comment list

node:comments

- node:author:... (Author)
- node:source:... (Translation source node)
- node:comments (Comments)
- node:comments:... (Comments)
- node:body:... (The main body text)



es. To make entity fields appear in the
r 'content is of type'). More useful tips

► [DATA SELECTORS](#)

CURRENT LIST ITEM

The variable used for holding each list item in the loop. This variable will be available inside the loop only.

Variable label *

Current Comment

Change "Current List Item" to

something more meaningful to this rule

Variable name *

my_comments

The variable name must contain only lowercase letters, numbers, and underscores and must be unique in the current scope.

Events

EVENT	OPERATIONS
After updating existing content	delete
+ Add event	

Don't forget to Save!

Conditions

[Show row weights](#)

ELEMENTS	OPERATIONS
⊕ NOT Content is published Parameter: Content: [node]	edit delete
+ Add condition + Add or + Add and	

Delete Entity needs to be nested under Loop



* Changes made in this table will not be saved until the form is submitted.

[Show row weights](#)

Actions

ELEMENTS	OPERATIONS
⊕ Loop Parameter: List: [node:comments] List item: Current Comment (my_comment)	edit delete Add action Add loop
⊕ Delete entity Parameter: Entity: [node:comments:0]	edit delete
+ Add action + Add loop	Still one more edit needed

Drill down to comment list

ENTITY
Specifies the entity, which should be deleted permanently.

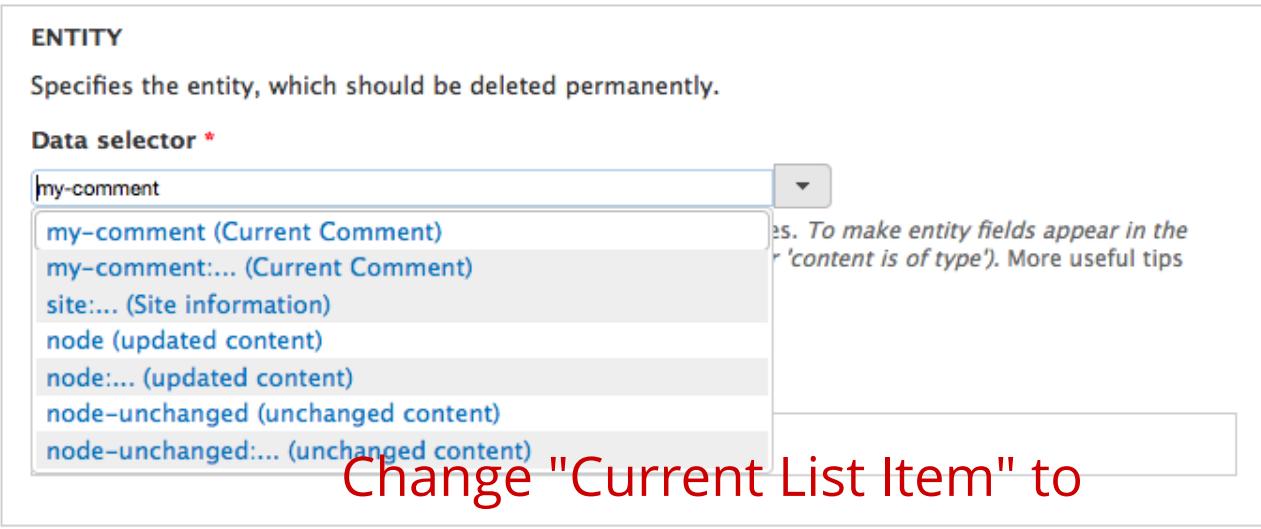
Data selector *

my-comment

- my-comment (Current Comment)
- my-comment:... (Current Comment)
- site:... (Site information)
- node (updated content)
- node:... (updated content)
- node-unchanged (unchanged content)
- node-unchanged:... (unchanged content)

es. To make entity fields appear in the
'content is of type'). More useful tips

Change "Current List Item" to something more meaningful to this rule



Events	
EVENT	OPERATIONS
After updating existing content	delete
+ Add event	

The Finished Rule

Conditions		Show row weights
ELEMENTS	OPERATIONS	
⊕ NOT Content is published Parameter: <i>Content</i> : [node]	edit delete	
+ Add condition + Add or + Add and		

Actions		Show row weights
ELEMENTS	OPERATIONS	
⊕ Loop Parameter: <i>List</i> : [node:comments] List item: Current Comment (my_comment)	edit delete Add action Add loop	
⊕ Delete entity Parameter: <i>Entity</i> : [my-comment]	edit delete	
+ Add action + Add loop		

Compose

Lightbot

The screenshot shows the Lightbot game interface. At the top left are four control buttons: a blue square with a white left arrow, a blue square with a white counter-clockwise arrow, a grey triangle pointing right, and a red speaker icon with a red 'X'. To the right are three more buttons: a blue circle with a white question mark, a red square with a white square inside, and a red button with a white play symbol. In the center is a 3D puzzle platform made of light blue and blue cubes. A small, grey, robot-like character with a yellow glowing head is positioned on a central light blue cube. To the right is a grid of 12 icons labeled "MAIN". The grid contains the following icons in a 3x4 layout:

MAIN			
Stack of cubes	Bulb	Stack of cubes	Counter-clockwise turn
Up arrow	Bulb	Up arrow	Counter-clockwise turn
Up arrow	Up arrow	Up arrow	Bulb

Below the grid is a row of five large numbers: 1, 2, 3, 4, 5. Underneath each number is a corresponding icon from the grid: up arrow, bulb, counter-clockwise turn, counter-clockwise turn, and stack of cubes.

Lightbot

The screenshot shows the Lightbot game interface. At the top, there are navigation buttons: a left arrow, a right arrow, a single step forward, a play button, and a fast-forward button. Below the navigation is a sidebar with icons for volume (muted), a lightbulb, and a speech bubble. The main area features a blue robot with a yellow head on a grey grid. A callout box contains the text: "When I see P1, I simply run all the commands in PROC1!". The robot is positioned above a yellow box labeled "PROC1" containing three commands: two upward arrows and one rightward arrow. To the right of the robot is a grey box labeled "ROC1" containing four commands: a rightward arrow, a downward arrow, an upward arrow, another rightward arrow, and a leftward arrow. A blue arrow points from the "P1" command in "PROC1" to the "P1" command in "ROC1". At the bottom, a row of icons represents available commands: an upward arrow, a lightbulb, a rightward arrow, a leftward arrow, a stack of coins, and a "P1" icon.

MAIN

When I see P1, I simply run
all the commands in
PROC1!

PROC1

ROC1

P1

Lightbot

The screenshot shows the Lightbot game interface. At the top, there are navigation buttons: a left arrow, a right arrow, a double right arrow, and a play button labeled "MAIN". On the left side, there are three more buttons: a right arrow, a speaker icon with a red "X", and a double right arrow. The main area features a robot with a yellow head and blue body on a grey path made of blue rectangular blocks. A callout box contains the text: "You can use the **P1** command inside **PROC1** to make a loop!". Below the path, a yellow rectangular block labeled "PROC1" contains four icons: an upward arrow, a stack of coins, and two "P1" icons. A red dashed box highlights the "PROC1" block. At the bottom, there is a row of six grey icons corresponding to the commands in the PROC1 block: upward arrow, lightbulb, right arrow, curved arrow, stack of coins, and "P1".

You can use the **P1** command inside **PROC1** to make a loop!

PROC1

↑ ⚡ ↗ ↙ ⚡ P1

Loop

Send email to these users

Actions

ELEMENTS

⊕ Add a variable

Parameter: *Type*: List of user items, *Value*: 586
Provides variables: List of users to be notified (list_user)

⊕ Loop

Parameter: *List*: [list-user]
List item: Current list item (list_item)

⊕ Send mail

Parameter: *To*: [list-item:mail], *Subject*: Bus Parking: [node:title]..., *Message*: [site:current-user] has..., *From*: [site:current-user:mail]

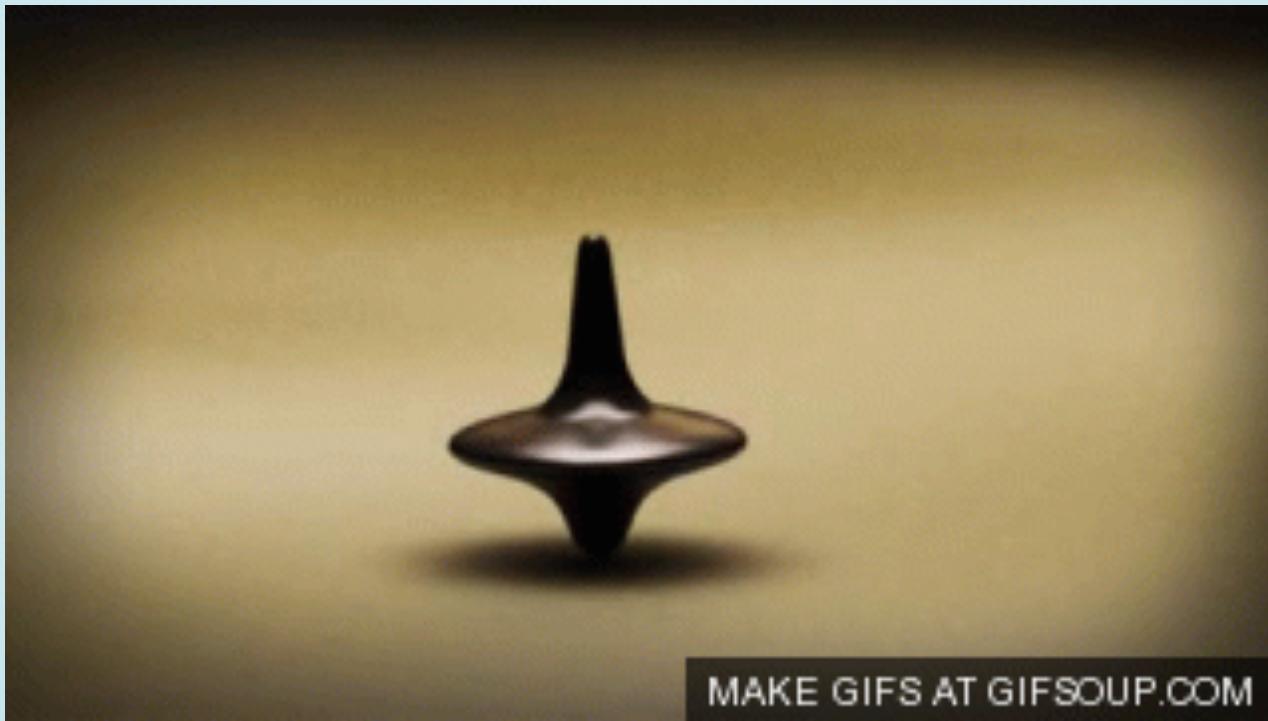
[+ Add action](#) [+ Add loop](#)

Loop

Run another Rule on these Nodes

Actions		Show row weights
ELEMENTS	OPERATIONS	
⊕ Show a message on the site Parameter: <i>Message</i> : Test	edit delete	
⊕ Load a list of conflicting nodes Parameter: <i>node to check for conflicts</i> : [node] Provides variables: Conflict List (conflict_list)	edit delete	
⊕ Loop Parameter: <i>List</i> : [conflict-list] List item: Current list item (list_item)	edit delete Add action Add loop	
⊕ rule: Check venue Parameter: <i>Conflicted Node</i> : [list-item], <i>Unsaved Node</i> : [node], <i>Form</i> : [form]	edit delete edit component	
+ Add action + Add loop		

Rule Inception



Conditions

Rules Components

[Show row weights](#)

ELEMENTS	OPERATIONS
⊕ Entity is of bundle Parameter: Entity: [node], type: Node, Entity bundle: Meeting	edit delete
⊕ Entity is of bundle Parameter: Entity: [venue], type: Taxonomy term, Entity bundle: Venue	edit delete
+ Add condition + Add or + Add and	

Notice, there is no "Event" in a component

Actions[Show row weights](#)

ELEMENTS	OPERATIONS
⊕ Set a data value Parameter: Data: [node:field-venue], Value: [venue]	edit delete
⊕ Show a message on the site Parameter: Message: Added a copy of [node:title...]	edit delete
+ Add action + Add loop	

DATA TYPE	LABEL	MACHINE NAME	USAGE
⊕ Node	▼ Node	node	 Parameter
⊕ Taxonomy term	▼ Venue	venue	 Parameter
⊕ Node	▼ Parent Node	parent_node	Parameter

Compose

(demo)

Trigger

Triggering Components

Views Bulk Operations

<input checked="" type="checkbox"/>	eliquid--12-21-22	Tyrian (30ml, Tyrian, 0mg)
<input checked="" type="checkbox"/>	eliquid--12-21-23	Tyrian (30ml, Tyrian, 3mg)
<input checked="" type="checkbox"/>	eliquid--12-21-24	Tyrian (30ml, Tyrian, 6mg)
<input checked="" type="checkbox"/>	eliquid--12-21-25	Tyrian (30ml, Tyrian, 8mg)
<input checked="" type="checkbox"/>	eliquid--12-21-5	Tyrian (30ml, Tyrian, 12mg)

Bulk operations

- [Add to Existing Product Display Node](#)
- [Create Product Display Node](#)
- [Delete item](#)
- [Modify entity values](#)

Bulk operations

Apply

Triggering Components

Drupal Commerce

+ Add a payment method rule

Enabled payment method rules

Name	Event	Status	Operations
Example payment Machine name: commerce_payment_commerce_payment_example, Weight: 0 Tags: Commerce Payment <ul style="list-style-type: none">Available on the checkout formAvailable on the order payment terminal	Select available payment methods for an order	Default	edit disable clone export

Disabled payment method rules

Name	Event	Status	Operations
PayPal WPS Machine name: commerce_payment_paypal_wps, Weight: 0 Tags: Commerce Payment <ul style="list-style-type: none">Available on the checkout formNot available on the order payment terminal	Select available payment methods for an order	Default	edit enable clone export

Triggering Components

Entityform

Enabled Entityform submission Rules

NAME	STATUS
Email Entityform Submitter Machine name: rules_email_entityform_submitter Parameter: Submitted Entityform (entityform), Entityform Type (entityform_type) Tags: entityform submission	Default
Email Entityform Admin Machine name: rules_email_entityform_admin Parameter: Submitted Entityform (entityform), Entityform Type (entityform_type) Tags: entityform submission	Overridde
Email Entityform CC Field Machine name: rules_email_entityform_cc_field Parameter: Submitted Entityform (entityform), Entityform Type (entityform_type) Tags: entityform submission <i>Error: Data selector entityform:field-cc for parameter list is invalid.</i>	Custom
Email Entityform Submitter (status change) Machine name: rules_email_entityform_submitter_status_change_ Parameter: Submitted Entityform (entityform), Entityform Type (entityform_type) Tags: entityform submission	Custom

Trigger

(demo)

Drupal 8?

www.d8rules.org

Site building based on Drupal 8 core integration wins

- Admin UI usability improvements
- Simple Views Bulk operations in core

Reusable components

- Plug-in based architecture & fully object-oriented code
- Rules data selector for tokens, contexts and other use cases
- Typed data widgets & formatters
- Embeddable Rules UI components to integrate actions & conditions

Evolved developer experience

- Unified DX based on Drupal 8 plug-in system
- Symfony 2 event system integration
- Deployable config via CMI

Milestones

1. Drupal 8 core integration &
Rules core engine **\$\$\$**
2. Rules core API completion
3. Rules 8.x release

Rules Recipes

at DropBucket.org, the Drupal Pastebin

dropbucket.org/liberatr

dropbucket.org/snippets/rules

Useful Rules Modules

Rules Scheduler

<https://www.drupal.org/project/rules>

Create Components

The "Event" is Cron

ex. Reminder: You haven't edited your profile in a year

Views Bulk Operations

https://www.drupal.org/project/views_bulk_operations

Create Components

Edit View "Bulk Operations" Field and add your Component

Make sure you have a "Node" Parameter

(or whatever Entity your view pulls)

Maillog

<https://www.drupal.org/project/maillog>

Show me the emails

Log all messages

Reroute instead of sending

Commerce

<https://www.drupal.org/project/commerce>

Enabling Payment Types

Adding Taxes

Price Adjustments

Rules Forms Support

https://www.drupal.org/project/rules_forms

Event: When a form is being validated

Invalidate a form before it gets saved

Validate fields, display custom messages

Don't need to use Form API

Used by Resource Conflict

Resource Conflict

https://www.drupal.org/project/resource_conflict

Check if an event is in the same room at the same time

 Use Taxonomy Terms for "rooms"

Can invalidate the form, so it won't be saved if a conflict exists

Workflow Rules

<https://www.drupal.org/project/workflow>

React to changes in Workflow State

e.g. Draft, In Progress, Needs Review, etc.

Cache Actions

https://www.drupal.org/project/cache_actions

Clear Drupal cache bins

Clear CSS/JS cache

Clear the cache of specific views

Clear the cache of specific panel pages

Clear the cache of specific mini panels

Entityforms

<https://www.drupal.org/project/entityform>

Excellent Rules support built in
Rules drive email Notifications

Logintoboggan Rules

<https://www.drupal.org/project/logintoboggan>

Event when the user account is validated.

Organic Groups

<https://www.drupal.org/project/og>

Events when a user is added to a group,

Joins/Leaves a group

When group content is added

Rules Repeated Events

https://www.drupal.org/project/rules_repeated_events

My preferred means of creating recurring events.

Good example of Provided vs Parameter

Context Rules

https://www.drupal.org/project/context_rules

Event when a Context is activated,
from the Context module

Rules: Lashing for Drupal

Ryan Price
Blink Reaction
@liberatr

slides.com/ryanprice