The Tiny Book of Rules

Tiny Drupal Books #1

The Tiny Book of Rules

This is Rules

Rules is a module usually used to automate actions on your Drupal site. Rules can react on events occurring on your site – such as a user logging in or a node being created – and perform customized follow-up actions such as redirecting to a certain page or setting field values. Optionally, Rules can evaluate conditions before executing any actions.

Rules is also a framework to use by other modules, allowing them to evaluate user-configured Rules components, such as action and condition sets, or expose data about variables and entities to other parts of Drupal.

Rules in site building

Using actions

Rules actions are reactions that Rules may perform. Some actions return new data that Rules can work with – such as loading the most recent comment written by a user – but most actions only deal with the data already available to Rules.

Some common actions are:

- Add a variable: This adds a new variable into your configuration, which then can be manipulated and used.
- Calculate a value: This allows you to perform basic arithmetics with numbers.
- Create a new entity: This asks for required data for an entity, and then creates it.
- Fetch entity by id: This allows for fetching nodes and other entities by id.
- Fetch entity by property: This allows for fetching all entities matching selected criteria, for example a selected field value.

- Show a message on the site: This displays an on-site message.
- Publish/unpublish content.
- Create or delete any URL alias.
- Send an e-mail.
- Set a data value (described below).

Actions can be combined in chains, which allows for rules loading new objects and then acting on them. Contributed modules may provide more actions to Rules.

Data in Rules

Actions, and other plugins in Rules, act on specified data types. These data can be simple, such as text, integers and dates; or complex like nodes – which in turn have properties represented by data, such as node id (integer), title (text), or creation date (date). Complex data types may have properties which are complex, too – each node has, for example, an author represented by a user account (which in turn has a number of properties).

Some properties are writable, which means that you can alter these with Rules actions. The action *Set a data value* is particularly useful, and can be used to set any writable data.

For most action parameters, it is possible to switch between *direct input*, meaning manually written input, and *data selection*, a tool for drilling down into the nested properties of complex data types.

The data object site is always available in Rules, containing global site data such as the account for the acting user, the site name, and some other information.

Finally, it should be noted that field values are read- and writable *only if Rules can be certain that the relevant entity has that field*. If working with a node, for example, a field will only appear if conditions have been set that limits the rule to selected content types. (See *using conditions* for more information.) If the entity doesn't have separate bundles – such as users – Rules can access the fields right away.

Using events

Rules provides a number of *events* – places in a Drupal page load that can be used to fire off actions. Rules that trigger on events are called *reaction rules* (in contrast to stand-alone rule components that must be triggered by other means – see the section about components).

Each event may provide some associated variables, which can be used in the rule configuration. For example, the event *After saving new content* provides the saved content as a node variable while *After updating existing content* provides both the updated and original content as variables – thereby allowing comparison between the two.

A reaction rule may have more than one triggering event. If they provide different variables, only the variables provided by *all* events may be used in the configuration, assuring that the rule may be evaluated properly regardless of which of the events is acting as trigger.

Using conditions

When a reaction rule is fired, or in other occasions where rules are used, you have the options to check a number of *conditions* and abort any actions if the conditions aren't met. Conditions can be pretty specific, such as *Content is promoted to front page, Data value is empty* or the comparison between two text strings – but the most used condition is called *Data comparison*. This one is used to compare two variables available to Rules, or to compare a data variable to a manually entered value – for example to check if a field has a certain value or if the node author is the same as the acting user.

Conditions can be logically grouped to create *and* and *or* groups. These groups may in turn contain other logical groups.

There are two conditions that are particularly important when working with fields: *Content is of type* and *Entity has field*. These conditions allows Rules to know that an entity has a particular field, and subsequently makes the field data available in configuration. The condition *Entity is of type* has a similar role to provide entity-specific data, but is used much less frequently. (You can also use the *Data comparison* condition to check the entity bundle.)

Lists and loops

Each data type declared to Rules automatically gets a sibling data type, being a list of this particular data. Thus, Rules cannot only handle nodes, text strings and dates, but also lists of nodes, strings and dates. This is very useful when working with multiple-value fields, but also for a few multiple-value entity properties, such as the roles of a user.

Rules provides a few actions and conditions that are handy when working with lists: *List contains item, Add a value to a list* and *Remove a value from a list*. The latter two can for example be used to add/remove a user from a multiplevalue user reference field.

The most useful thing with lists, though, is the ability to *loop* through them. By adding a loop to your set of actions, you have each action inside the loop being evaluated once for every item in the list you select. This can for example be used to send an e-mail to all users in a multiple-value user reference field.

It is possible to use loops within loops, though you should be aware that evaluation may become performance heavy if there are many items to loop over

Components

Reaction rules are the simplest and usually most common Rules configurations. But you can also create Rules *components* – re-usable condition sets, action sets, single rules or sets of rules. These components can then be called by reaction rules, saving you the work of repeating the same conditions in three different reaction rules.

Apart from avoiding repetitive work, components may also be invoked by other modules. Views Bulk Operations, for example, can make use of Rules components as actions; Rules

Scheduler can schedule the evaluation of any component for a future point in time; and with the help of Rules Bonus Pack, CTools can use condition components as access plugins in modules like Page manager and Panels.

Components are mainly created and edited like reaction rules – but they have no triggers. They also have configurable variables that are either *parameters* or *provided* (or both). Parameters are required input data for the component – variables that have to be specified when calling the component. Provided variables are returned to the caller after evaluation.

Components are very useful both in site building and coding.

Useful additional modules

There are a number of modules taking advantage of Rules as a framework, or providing new actions, conditions and plugins. The most notable are:

- Rules Scheduler. This module is included in the Rules project, and allows scheduling components to be evaluated at a future point in time. The module can for example be used to schedule content deletion, upgrading of user accounts, or managing recurring e-mails.

- Views Bulk Operations (VBO). This module extends Views and allows users to perform actions on selected rows. Apart from some base-level actions, Rules components may be used as actions by VBO. An equally useful feature of VBO is that it allows calling Views from Rules configuration, to load a list of entities listed by Views.

- Rules Bonus Pack. This module serves as an experimental lab for new Rules functionality, and provides a number of new actions and conditions. It also allows using condition components as CTools access plugins, as well as providing a new component type used for validating and modifying arguments (contextual filter values) in Views.

- Rules Link. This module can be used to attach links to entities, and firing rule sets when these links are clicked. Links can also be rendered using Views, and link visibility can be controlled by condition sets.

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- ➢ HTML (Free /Available to everyone)

- ➢ PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)

- ➢ Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below