

ffwagency.com

Drupal Console: An Overview of the New Drupal CLI

Thought by Jesus Manuel Olivas, Drupal 8 Solutions Engineer

Drupal Console is the new CLI (Command Line Interface) for Drupal. This tool can help you to generate boilerplate code, as well as interact with, and debug Drupal 8. From the ground up, it is built to utilize the same modern PHP practices that have been adopted in Drupal 8.

Drupal Console takes advantage of the Symfony Console and other well-known third-party components like Twig, Guzzle, and Dependency Injection among others. By embracing those standard components, we're fully participating in the PHP community, building bridges and encouraging the PHP community to join the Drupal project and allow us to reduce the isolation of Drupal.

Why is Drupal Console important?

Drupal is infamous for having a steep learning curve, complete with its own language of "Drupalisms". While Drupal 8 simplifies and standardizes the development process, it is more technically

advanced and complex than its predecessor.

Managing the increasing complexity of Drupal 8 could be a daunting task for anyone. Drupal Console has been designed to help you manage that complexity, facilitating Drupal 8 adoption while making development and interaction more efficient and enjoyable. Drupal Console was created with one goal in mind: to allow individuals and teams to develop smarter and faster on Drupal 8.

Drupal Console features

In this blog post, I will mention some of the most common features and commands of Drupal Console, to serve as a good introduction.

Install Drupal Console

```
# Download the latest Drupal Console release.
```

```
curl https://drupalconsole.com/installer -L -o drupal.phar
```

```
# Move to a global accessible place on the system.
```

```
mv drupal.phar /usr/local/bin/drupal
```

```
# Apply executable permissions on the downloaded file.
```

```
chmod +x /usr/local/bin/drupal
```

Copy configuration files

The *init* command copy application configuration files to the user

home directory. Modifying this configuration files is how the behavior of the application can be modified.

```
3. fish /Users/jmolivas/develop/drupal/sites (fish)
~/d/d/sites ➤ drupal init --override
Copied files
User home path: /Users/jmolivas/.console/
1 - aliases.yml
2 - chain/create-data.yml
3 - chain/form-sample.yml
4 - chain/quick-start-mysql.yml
5 - chain/quick-start.yml
6 - chain/sample.yml
7 - chain/site-drop-restore.yml
8 - chain/site-install.yml
9 - chain/update-gitbook.yml
10 - commands.yml
11 - config.yml
12 - phpcheck.yml
13 - router.php
14 - site.mode.yml
15 - sites/sample.yml

Bash or Zsh: Add this line to your shell configuration file:
source "$HOME/.console/console.rc" 2>/dev/null

Fish: Create a symbolic link
ln -s ~/.console/drupal.fish ~/.config/fish/completions/drupal.fish

~/d/d/sites ➤
```

Validate system requirements

The **check** command will verify the system requirements and throw error messages if any required extension is missing.

```
3. vagrant@local: ~ (vagrant)
vagrant@local:~$ drupal check

[ERROR] The extension gd is missing.
```

```
[ERROR] The extension curl is missing.

The extension mysql is recommended to install.

The extension sqlite3 is recommended to install.

The configuration date.timezone was missing and overwritten with America/Tijuana.

vagrant@local:~$
```

Install Drupal 8

The easiest way to try Drupal 8 on your local machine is by executing the chain command and pass the option `--file=~/.console/chain/quick-start.yml`

The ***chain*** command helps you to automate command execution, allowing you to define an external YAML file containing the definition name, options, and arguments of several commands and execute that list based on the sequence defined in the file.

In this example, the ***chain*** command will download and install Drupal using SQLite, and finally, start the PHP's built-in server. Now you only need to open your browser and point it to `127.0.0.1:8088`.

```
3. drupal /Users/jmolivas/develop/drupal/sites (php)
~/d/d/sites ➤ drupal chain --file=~/.console/chain/quick-start.yml

// site:new
Downloading drupal 8.1.0

[OK] Drupal 8.1.0 was downloaded in directory
/Users/jmolivas/develop/drupal/sites/drupal8.dev

// site:install
Starting Drupal 8 install process

[OK] Your Drupal 8 installation was completed successfully

// server

/usr/bin/php -S 127.0.0.1:8088 /Users/jmolivas/.console/router.php

[OK] Executing php from /usr/bin/php.

PHP 5.5.31 Development Server started at Wed Apr 20 22:56:09 2016
Listening on http://127.0.0.1:8088
Document root is /Users/jmolivas/develop/drupal/sites/drupal8.dev
Press Ctrl-C to quit.
█
```

Generate a module

The ***generate:module*** command helps you to:

- Generate a new module, including a new directory named *hello_world* at *modules/custom* directory.
- Creates a *hello_world.info.yml* file at *modules/custom/hello_world* directory.

```
3. fish /Users/jmolivas/develop/drupal/sites/drupal8.dev (fish)
~/d/d/s/drupal8.dev > drupal generate:module

// Welcome to the Drupal module generator

Enter the new module name:
> Hello World

Enter the module machine name [hello_world]:
>

Enter the module Path [/modules/custom]:
>

Enter module description [My Awesome Module]:
> Hello Wold demo module

Enter package name [Custom]:
>

Enter Drupal Core version [8.x]:
>

Do you want to generate a .module file (yes/no) [yes]:
>

Define module as feature (yes/no) [no]:
>

Do you want to add a composer.json file to your module (yes/no) [yes]:
```

Generate a service

The ***generate:service*** command helps you to:

- Generate a new service class and register it with the *hello_world.services.yml* file.

```
3. fish /Users/jmolivas/develop/drupal/sites/drupal8.dev (fish)
~/d/d/s/drupal8.dev > drupal generate:service

// Welcome to the Drupal service generator
Enter the module name [hello_world]:
>

Enter the service name [hello_world.default]:
> hello_world.sample

Enter the Class name [DefaultService]:
> HelloService

Create an interface (yes/no) [yes]:
>

Do you want to load services from the container (yes/no) [no]:
> y

Type the service name or use keyup or keydown.
This is optional, press enter to continue

Enter your service [ ]:
> entity_type.manager
Enter your service [ ]:
>

Do you confirm generation? (yes/no) [yes]:
>
```

Generate a Controller

The ***generate:controller*** command helps you to:

- Generate a new *HelloController* Class with a *hello* method at *src/Controller* directory.
- Generate a route with a path to */hello/{name}* at *hello_world.routing.yml* file.

```
3. drupal /Users/jmolivas/develop/drupal/sites/drupal8.dev (php)
~/d/d/s/drupal8.dev > drupal generate:controller

// Welcome to the Drupal Controller generator
Enter the module name [hello_world]:
>

Enter the Controller class name [DefaultController]:
> HelloController

Enter the Controller method title (leave empty and press enter when done) [ ]:
> Hello

Enter the action method name [hello]:
>

Enter the route path [/hello_world/hello/{name}]:
> /hello/{name}

Enter the Controller method title (leave empty and press enter when done) [ ]:
>

Do you want to generate a unit test class (yes/no) [yes]:
>

Do you want to load services from the container (yes/no) [no]:
> yes

Type the service name or use keyup or keydown.
This is optional, press enter to continue

Enter your service [ ]:
> hello_world.sample
Enter your service [ ]:
> 
```

Generate a Configuration Form

The ***generate:form:config*** command helps you to:

- Generate a *SettingsForm.php* class at *src/Form* directory,
- Generate a route with path to */admin/config/hello_world*
/settings at *hello_world.routing.yml*

- Register at the *hello_world.links.menu.yml* file the *hello_world.settings_form* route using *system.admin_config_system* as parent.

This command allows you to add a form structure to include form fields based on the field API. Also generates a *buildForm* and *submitForm* methods with the required code to store and retrieve form values from the configuration system.

NOTE: The parent route *system.admin_config_system* for the *menu_link* can be selected from the command interaction.

```
3. drupal /Users/jmolivas/develop/drupal/sites/drupal8.dev (php)
~/d/d/s/drupal8.dev > drupal generate:form:config
Enter the module name [hello_world]:
>

Enter the Form Class name [DefaultForm]:
> SettingsForm

Enter the Form id [settings_form]:
>

Do you want to load services from the container (yes/no) [no]:
>

Do you want to generate a form structure? (yes/no) [yes]:
>

Type [ ]:
> textfield

Input label:
> My text field

Input machine name [my_text_field]:
>

Maximum amount of characters [64]:
>

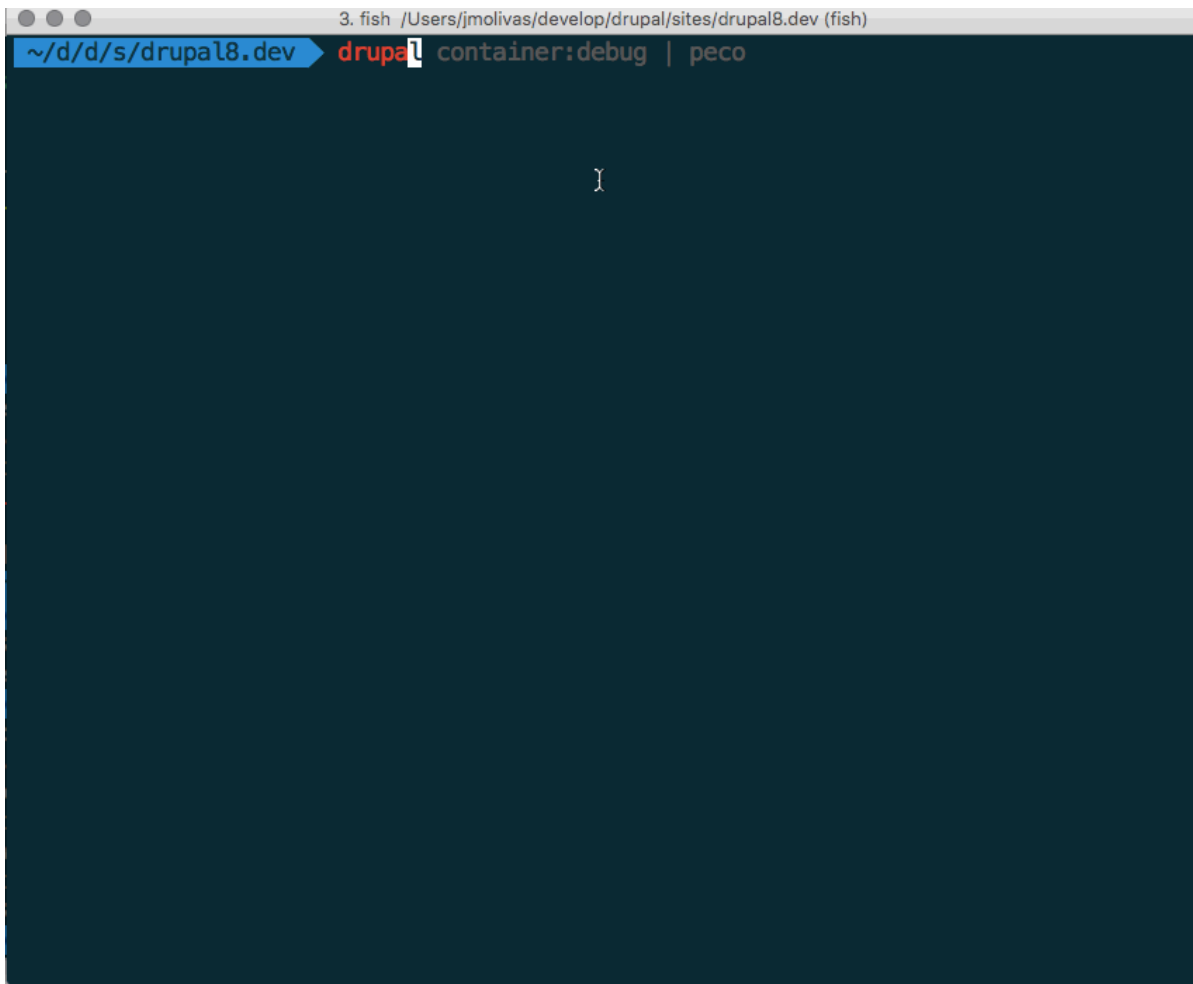
Width of the textfield (in characters) [64]:
>

Description [ ]:
>

Default value [ ]:
```

Debug Services

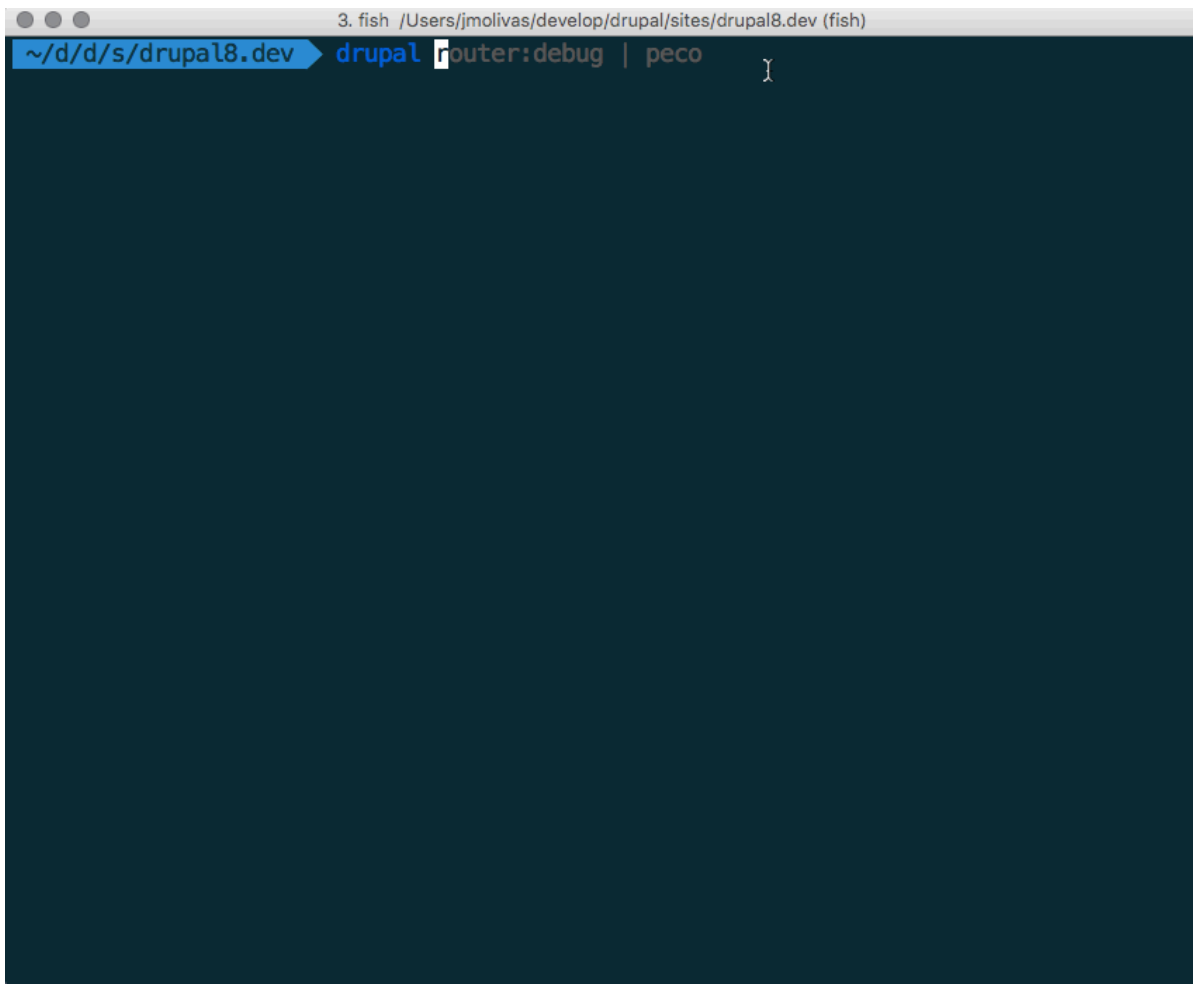
The ***container:debug*** command displays the currently registered services for an application. Drupal contains several services registered out-of-the-box plus the services added by custom and contributed modules, for that reason I will use [peco](#) a simplistic interactive filtering tool to make this debug easier.



```
3. fish /Users/jmolivas/develop/drupal/sites/drupal8.dev (fish)
~/d/d/s/drupal8.dev > drupal container:debug | peco
```

Debug Routes

The ***router:debug*** command displays the currently registered routes for an application. Similar to debugging services. In this example, I will use `peco` to make this debugging task easier.

A terminal window with a dark blue background. The title bar shows "3. fish /Users/jmolivas/develop/drupal/sites/drupal8.dev (fish)". The prompt is "~ /d/d/s/drupal8.dev" followed by a blue arrow pointing to "drupal". The command "router:debug | peco" is entered, with a cursor at the end. The rest of the terminal is empty.

```
3. fish /Users/jmolivas/develop/drupal/sites/drupal8.dev (fish)
~/d/d/s/drupal8.dev ➤ drupal router:debug | peco
```

Create Data

The ***create:nodes*** command creates dummy nodes for your application.

```

3. fish /Users/jmolivas/develop/drupal/sites/drupal8.dev (fish)
~/d/d/s/drupal8.dev drupal create:nodes

// Welcome to the Drupal nodes generator

Select content type(s) to be used on node creation:
[0] Article
[1] Basic page
> 0,1

Enter how many nodes would you like to generate [25]:
> 5

Enter the maximum number of words in titles [5]:
>

How far back in time should the nodes be dated?:
[0] N | Now
[1] H | 1 hour ago
[2] D | 1 day ago
[3] W | 1 week ago
[4] M | 1 month ago
[5] Y | 1 year ago
> 5

```

Node Id	Content type	Title	Created Time
1	Basic page	Augue Immitto Minim Nostrud Patria	2015-08-01 05:53:17
2	Article	Enim Jumentum Saepius	2016-03-05 05:36:08
3	Basic page	Capto Ille	2015-05-18 05:12:55
4	Article	Pecus	2016-04-14 09:25:46
5	Basic page	Aliquip Amet Damnum Ideo	2015-11-25 09:45:58

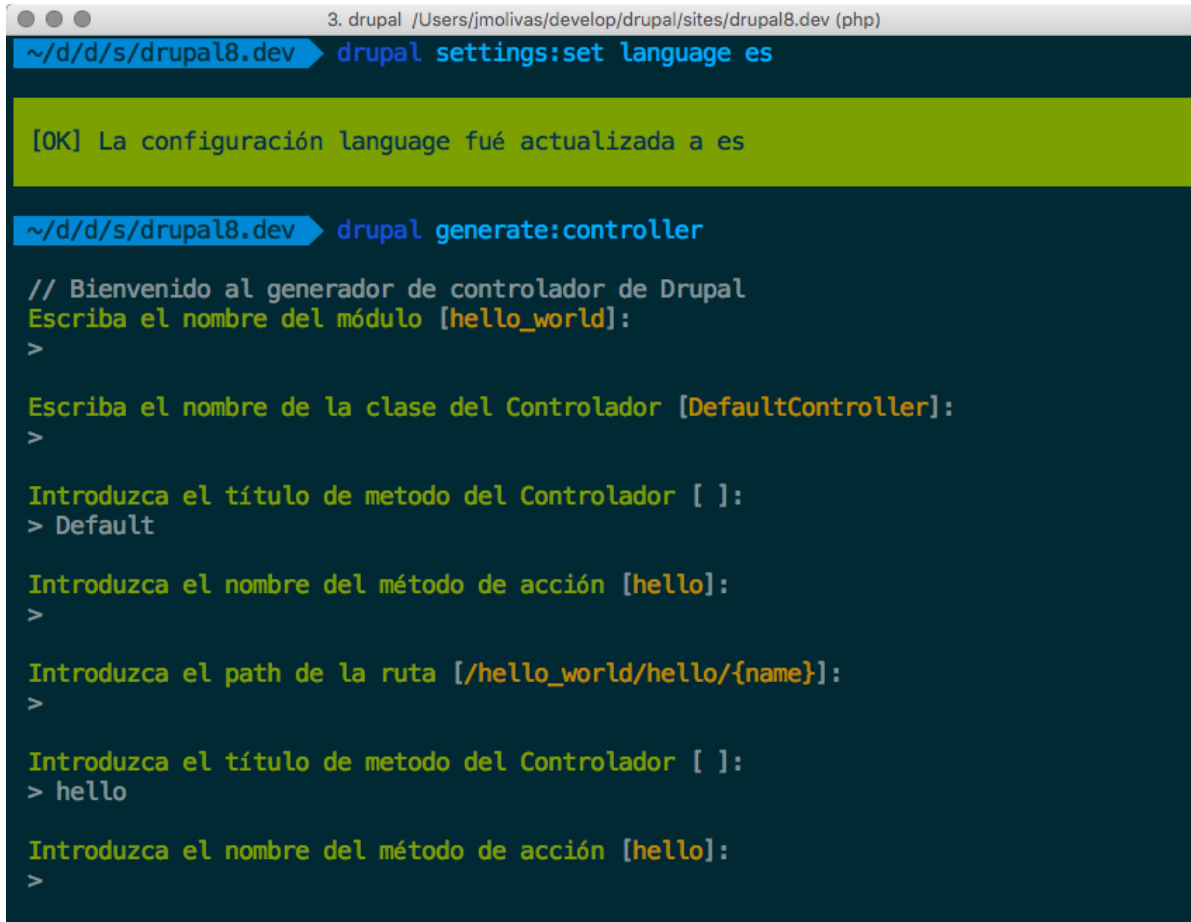
Drupal Console provides a YAML to execute using the **chain** command. This file contains instructions to execute *create:users*, *create:vocabularies*, *create:terms* and *create:nodes* command using one command.

```
drupal chain --file=/Users/jmolivas/.console/chain/create-data.yml
```

Modify the Drupal Console language

The *settings:set* command helps to change the application configuration in this example using the arguments *language es* we can set Spanish as the application language. After switching the

default language the interface is translated.



```
3. drupal /Users/jmolivas/develop/drupal/sites/drupal8.dev (php)
~/d/d/s/drupal8.dev ➤ drupal settings:set language es

[OK] La configuración language fué actualizada a es

~/d/d/s/drupal8.dev ➤ drupal generate:controller

// Bienvenido al generador de controlador de Drupal
Escriba el nombre del módulo [hello_world]:
>

Escriba el nombre de la clase del Controlador [DefaultController]:
>

Introduzca el título de metodo del Controlador [ ]:
> Default

Introduzca el nombre del método de acción [hello]:
>

Introduzca el path de la ruta [/hello_world/hello/{name}]:
>

Introduzca el título de metodo del Controlador [ ]:
> hello

Introduzca el nombre del método de acción [hello]:
>
```

All of the available commands

The *list* command can be used to show all of the available commands. A print screen was not included because the more than 130 commands make the image huge to show on this blog post.

For the full list of commands, you can visit the documentation page at <http://docs.drupalconsole.com/en/commands/available-commands.html> or visit the quick reference guide (cheat sheet) at

<http://drupalconsole.com/cheatsheet/>

The screenshot shows a web browser displaying the Drupal Console Cheatsheet. The page is organized into a sidebar and a main content area. The sidebar contains a 'Check All' button and a grid of buttons for various languages (en, es, ca, fr, de, ko, hi, hu, id, ja, pt_br, ro, ru, si, vn, zh_hans) and tools (cache, chain, config, container, create, cron, database, generate, image, locale, migrate, module, multisite, node, none, plugin, rest, router, settings, site, state, test, theme, translation, update, user, views, yaml). The main content area is divided into several panels, each containing a list of commands and their descriptions. The panels include: 'none' (displaying basic information about the project, chain command execution, system requirement checker, help, copy configuration files, and available commands), 'cron' (listing modules implementing a cron, executing cron implementations, and releasing the cron system lock), 'cache' (rebuilding and clearing all site caches), 'config' (showing current configuration, deleting it, and exporting it), 'chain' (listing available chain files), 'container' (displaying current services for an application), 'create' (creating dummy comments, nodes, terms, users, and vocabularies), and 'database' (launching a DB client if available).

What makes Drupal Console unique

- Has been built to utilize the same modern PHP practices adopted by Drupal 8.
- Generates the code and files required by Drupal 8 modules and components.
- Facilitate Drupal 8 adoption while making development and interaction more efficient and enjoyable.
- Allow individuals and teams to develop smarter and faster on Drupal 8.

- Help developers to understand Drupal 8 with the "--learning" flag.
- Fully multilingual and translatable, just like Drupal 8 itself.

Links and resources

- [Drupal Console landing page](#)
- [Drupal Console documentation](#)
- [Drupal Console support channel](#)
- [Drupal Console project page at d.o.](#)
- [Drupal Console project at GitHub](#)
- [Drupal 8 Module of the Week: Drupal Console](#)
- [Drupal Console Deep Dive: How to Develop Faster and Smarter on Drupal 8](#)