# Drupal... at your service!

By Mauro Vigliotti

# About me

Mauro Vigliotti

- I’m from Argentina (South America)
- Drupal developer since 2005
- Working for **The Brick Factory**
- Developer by day…
- … rocker by night
- 2018 DrupalCon Official guitar winner!

# About *The Brick Factory*

The Brick Factory is a Washington, DC-based digital agency that plans and executes digital campaigns for nonprofits, trade associations, advocacy groups and brands

- Digital Strategy
- Digital Marketing
- Design
- Development
- Hosting and Systems
- Products

# Get ready to learn about one of Drupal 8’s greatest new features!

# Overview

**What’s a Service**

**The Service Container**

**Service Definition**

**Tags**

**Accessing a Service**

**Samples:**

- Page redirection
- Controlling user sessions
- Overriding an existing service
- Adding service using DI

# What will you get from this talk?

A good approach to D8 Services

Sample code to start producing your own services, it’s so easy!

Understanding about Service Container, controller classes and the new Symfony approach on Drupal Core

# This talk is not intended to:

Go deeply into Symfony

Provide *all* the services that are available within the Drupal core

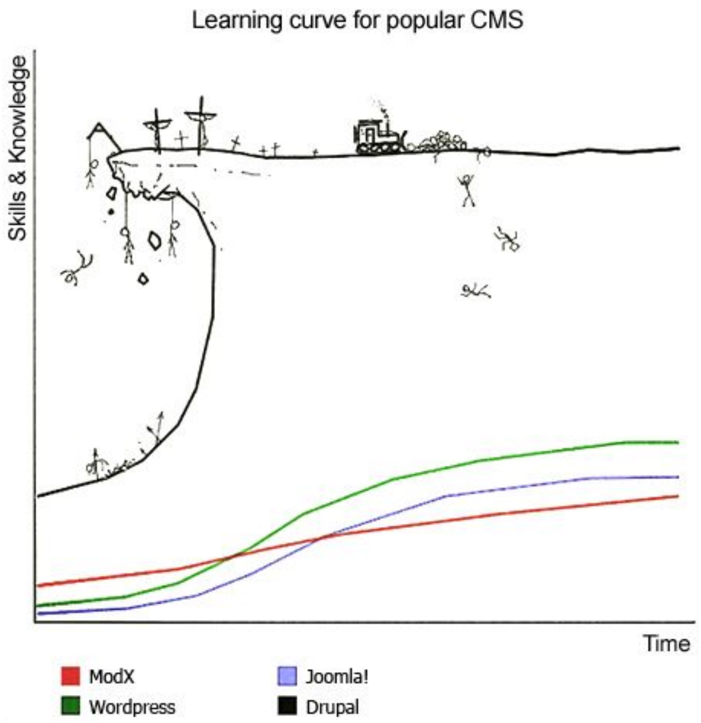
Solve *all* your Drupal 8 problems

Talk about REST (in Drupal 7, Services' meaning is different)

# Important note! Why do I say this talk is targeted to beginners?

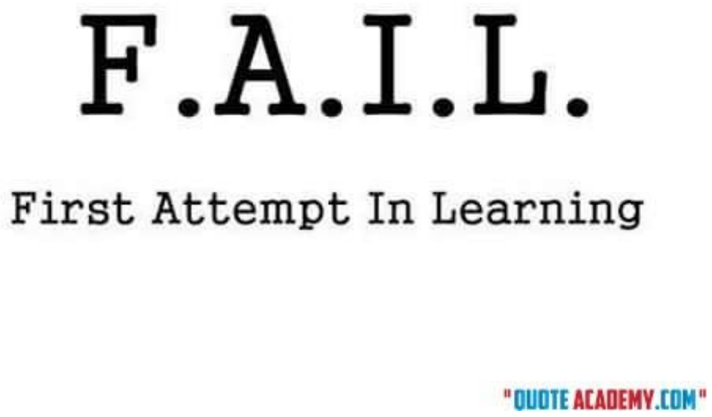
# Target: Drupal < 8 developers

The learning curve for most of the developers we've been working on previous Drupal versions before can be hard, mostly because the CMS had a hook-based approach, and now is moving to an Object-Oriented Paradigm.



# Target: Drupal freshmen.

If you're brand new on Drupal, finding out there's so many things to learn could be the first blocker, whether if you're a content manager, a designer, a front-end or back-end developer.



F.A.I.L.

First Attempt In Learning

"QUOTE ACADEMY.COM"

# Let’s go!

# What’s a Drupal 8 Service?

**Let’s throw the mystery away, a Service is a PHP Class (or a bunch of classes, depending on your implementation).**

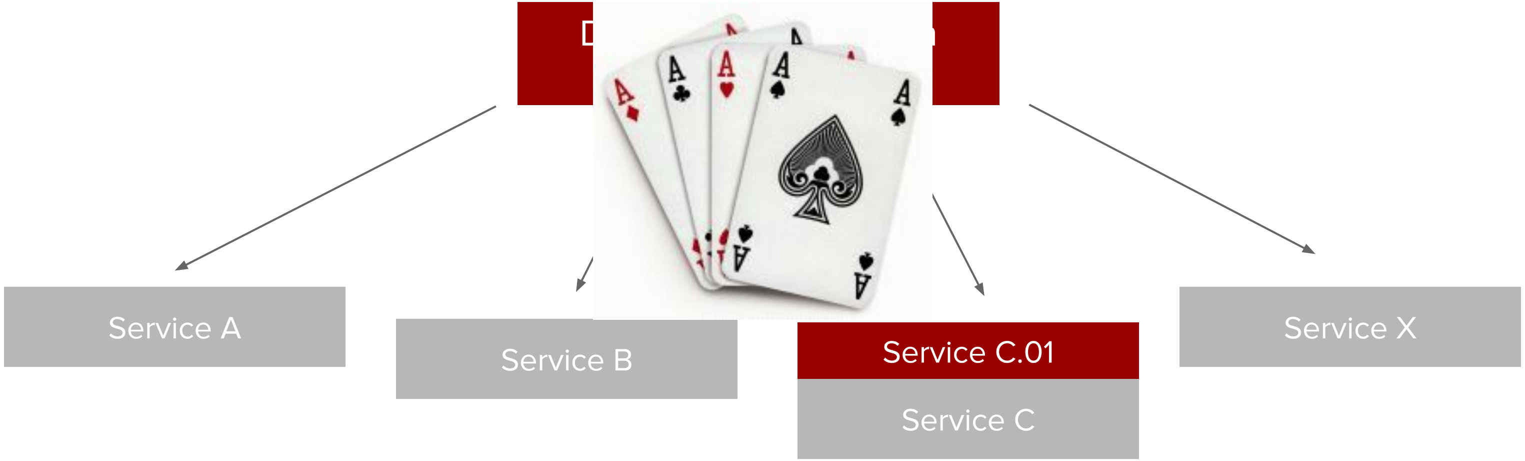
- Decoupled reusable functionality
- Plugged to your Drupal 8 at any time
- Introduced in Drupal by Symfony.

# What’s a Drupal 8 Service?

# Drupal Services Out-of-the-box

- More than 100 services available.
- Sample services:
    - Accessing the current logged in user
    - Sending an email
    - Asking for a page redirection
    - Altering a http response
    - Performing a database query
    - Translating a piece of text
- Full core services list here: https://api.drupal.org/api/drupal/core%21core.services.yml/8.2.x ...or opening the core/core.services.yml file.

# The Service Container (a.k.a. Dependency Injection Container)



Service A

Service B

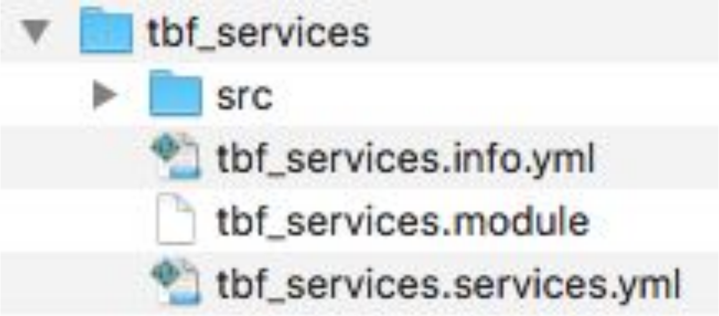
Service C.01

Service C

Service X

# Service definition

## YAML file in root of our custom module



```
▼ tbf_services
    ▶ src
      tbf_services.info.yml
      tbf_services.module
      tbf_services.services.yml
```

# Service definition

## Service definition format

Service name

Designated class

Arguments

Tags

```
user.current_user_context:
 class: Drupal\user\ContextProvider\CurrentUserContext
 arguments: ['@current_user', '@entity.manager']
 tags:
   - { name: 'context_provider' }
```

https://www.drupal.org/docs/8/api/services-and-dependency-injection/structure-of-a-service-file

# Tags

- Define a group of related services (remember the card deck concept)
- Tag properties
  - name
  - priority
  - global
  - alias, and many others

- Some sample tag names:
  - access_check
  - cache.bin
  - context_provider
  - event_subscriber
  - theme_negotiator

https://www.drupal.org/docs/8/api/services-and-dependency-injection/structure-of-a-service-file

# Accessing a Service

# Accessing a Service: YAML Service file

```yaml
user.current_user_context:
 class: Drupal\user\ContextProvider\CurrentUserContext
 arguments: ['@current_user', '@entity.manager']
 tags:
   - { name: 'context_provider' }
```

# Accessing a Service: $container object

```
// EntityViewBuilder.php

public static function createInstance(ContainerInterface $container,
EntityTypeInterface $entity_type) {
 return new static(
   $entity_type,
   $container->get('entity.manager'),
   $container->get('language_manager'),
   $container->get('theme.registry')
 );
}
```

# Accessing a Service everywhere: Global \Drupal class

```
$foobar = \Drupal::service('foo.bar');
```

# Okay… enough talking…
# Show me the code!

# Sample Code

- Page redirection
- Controlling user sessions
- Overriding an existing service
- Adding service using DI

# Questions?

# Thank you!

Get in touch if you have any questions

- Drupal.org: https://www.drupal.org/u/mauro_ptt

Presentation and code ready to download at:

- Presentation: https://goo.gl/Vr8DTK
- Sample code: https://github.com/maurotbf/d8services