

CRÉER UN MODULE DRUPAL 8

DE DRUPAL 7 À DRUPAL 8 : KILLING THE HOOKS !



A PROPOS



- Dominique CLAUSE @DomClause
- Ingénieur de développement
- Contributeur Drupal : <https://www.drupal.org/u/dom>.
- Freelance Drupal (ACINO) :
 - www.acino.fr
 - @AcinoContact

CRÉER UN MODULE DRUPAL 8

DE DRUPAL 7 À DRUPAL 8 : KILLING THE HOOKS !

CETTE PRÉSENTATION N'EST PAS

- un tuto module Hello World
 - => <http://drupal-addict.com/mon-premier-module-drupal-8-hello-world>
 - => <http://flocondetoile.fr/blog/creer-un-module-drupal-8-en-30-secondes>
- un tuto migration D7 à D8
 - => <http://drupal-addict.com/aide-a-la-migration-automatisee-d-un-module-drupal-7-vers-drupal-8>
- un cours de programmation PHP objet

COMPARAISON DES PRINCIPES D7 VS D8

- configuration drupal > configuration YAML
- hook() > configuration YAML
- hook() > hook()
- hook() > événement
- hook() > service
- hook() > annotation
- hook() > objet



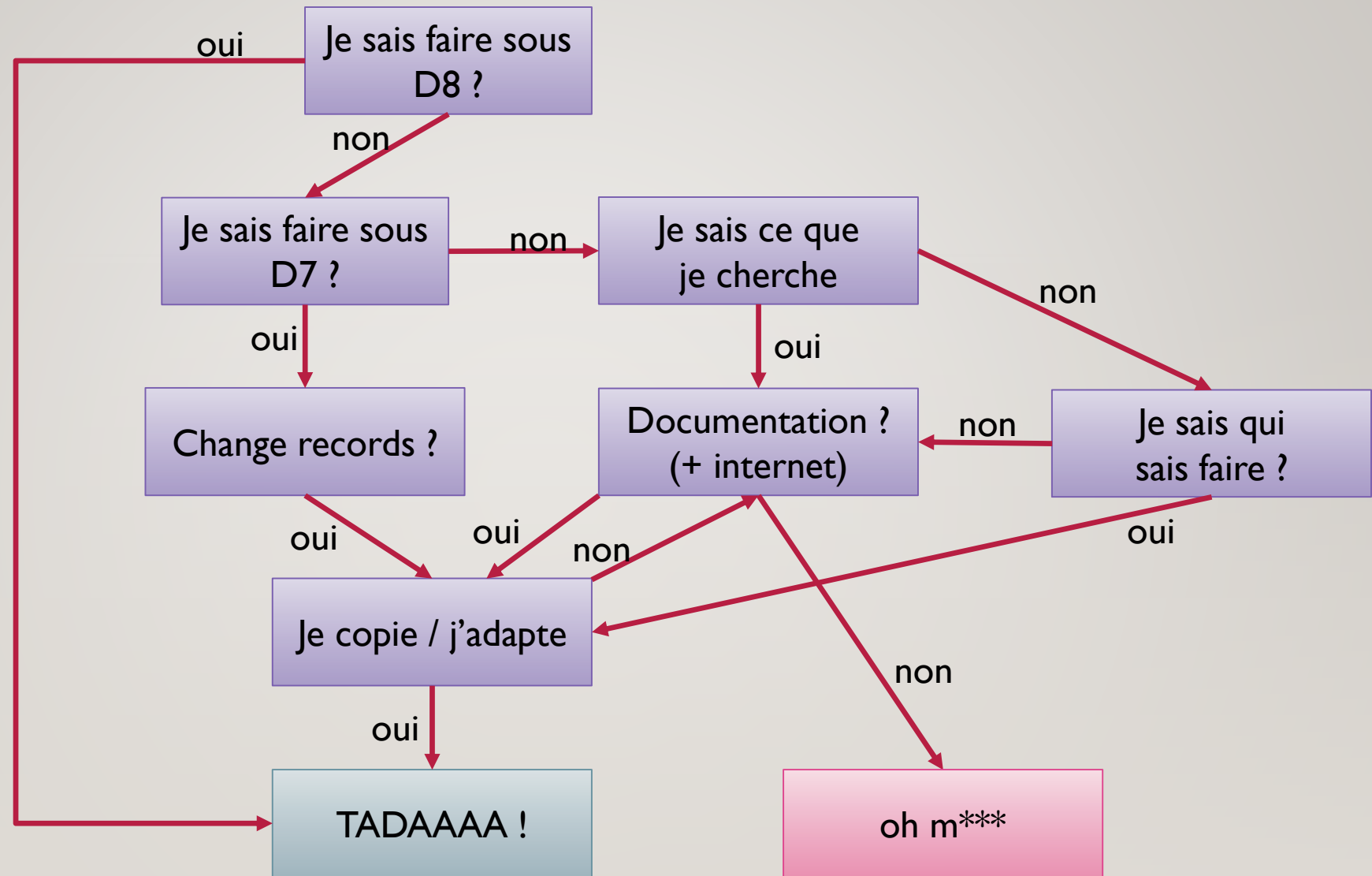
QUEL EST LE PROBLÈME ?

- Contre les hooks :
 - Code procédural
 - Difficile de faire un test unitaire
 - Contrôle de l'ordre dans lequel un hook est invoqué
 - Invoker deux fois le même hook dans un même module
 - Stopper la propagation d'un hook

MÉTHODOLOGIE PERSONNELLE



MÉTHODOLOGIE PERSONNELLE



CRÉATION D'UN MODULE DRUPAL 8

search_autocomplete.info

```
name = Search Autocomplete
description = Provides autocompletion
for Drupal search forms and input
fields in your site.
core = 7.x
package = Search
files[] =
views/plugins/search_autocomplete.views
_plugin_style.inc
dependencies[] = views
configure =
admin/config/search/search_autocomplete
```

search_autocomplete.info.yml

```
name: Search Autocomplete
type: module
description: Provides autocompletion
feature for fields in your Drupal site.
This module ships with pre-existing
autocompletion configurations.
core: 8.x
package: Search
dependencies:
- views
configure:
autocompletion_configuration.list
```

<https://www.drupal.org/node/1935708>

HOOK() > CONFIGURATION YAML

hook_permissions()

```
/**
 * Implements hook_permission().
 *
 * Valid permissions for this module.
 */
function search_autocomplete_permission() {
  return array(
    'administer Search Autocomplete' => array(
      'title' => t('Administer Search Autocomplete'),
      'description' => t('Access administration panel for
autocomplete settings.'),
    ),
    'use Search Autocomplete' => array(
      'title' => t('Use Search Autocomplete'),
      'description' => t('Allow usage of autocomplete on
forms.'),
    ),
  );
}
```

\$module.permissions.yml

```
administer search autocomplete:
  title: Administer Search Autocomplete
  description: Access administration panel for
autocomplete settings.
use search autocomplete:
  title: Use Search Autocomplete
  description: Allow usage of autocomplete on configured
fields.
```

HOOK() > CONFIGURATION YAML

- `hook_menu()` > `*.routing.yml`, `*.links.menu.yml`, `*.links.action.yml`, `*.links.contextual.yml`, `*.links.task.yml` (<https://www.drupal.org/node/1800686>)

Plus de détails sur : <https://www.drupal.org/node/2118147>

Tutoriel:

- <http://drupal-addict.com/premier-module-hello-world-creer-une-nouvelle-page>
- <http://drupal-addict.com/premier-module-hello-world-ajouter-un-lien-dans-le-menu-du-site>

HOOK() > HOOK()

hook_page_build()

```
/**
 * Implements hook_page_build().
 */
function search_autocomplete_page_build() {

  // Checkout if user have authorization use admin tools.
  if (user_access('administer Search Autocomplete') &&
variable_get('sa_admin_helper', TRUE)) {
    // Add the admin tools.
    drupal_add_js(drupal_get_path('module',
'search_autocomplete') . '/js/search_autocomplete.admin.js');
    drupal_add_css(drupal_get_path('module',
'search_autocomplete') . '/css/search_autocomplete.admin.css');
  }
}
```

hook_page_attachments()

```
/**
 * Implements hook_page_attachments().
 */
function search_autocomplete_page_attachments(array
&$attachments) {

  // Load module settings.
  $module_settings =
\Drupal::config('search_autocomplete.settings');

  // Checkout if user have authorization use admin tools
and, if set: include it.
  if (\Drupal::currentUser()->hasPermission('administer
Search Autocomplete') && $module_settings->
get('admin_helper')) {
    $attachments['#attached']['library'][] =
'search_autocomplete/search_autocomplete.admin.helper';
  }
}
```

<https://www.drupal.org/node/23577555>

GESTION DES ASSETS JS ET CSS

- Déclarer des libraires : *.libraries.yml

```
search_autocomplete.admin.helper:  
  version: VERSION  
  js:  
    js/search_autocomplete.admin.js: {}  
  css:  
    theme:  
      css/admin_helpers/search_autocomplete.admin.css: {}  
  dependencies:  
    - core/jquery  
    - core/drupal
```

- Librairies dynamique : hook_library_info_build()

GESTION DES ASSETS JS ET CSS

- Utiliser ma librairie : via #attached sur un render array.

```
$attachments ['#attached'] ['library'] [] =  
'search_autocomplete/search_autocomplete.admin.helper';
```

```
$form['mon_textifield'] = array(  
  '#type' => 'textfield',  
  '#attached' => array(  
    'library' => array(  
      'search_autocomplete/search_autocomplete.admin.helper',  
    ),  
  ),  
);
```

- Note special D7 > 7.36:
 - \$conf['javascript_always_use_jquery'] = FALSE;
 - drupal_add_js('mylibrary.js', array('requires_jquery' => FALSE));

<https://www.drupal.org/node/2169605>

HOOK() > EVENEMENT

hook_menu_site_status_alter()

```
/**
 * Implements hook_menu_site_status_alter().
 */
function
mymodule_menu_site_status_alter(&$menu_site_status,
$path) {
  if ($menu_site_status == MENU_SITE_OFFLINE) {
    // Do stuff.
  }
}
```

onKernelRequestMaintenance

```
/**
 * Do stuff
 *
 * @param \Symfony\Component\HttpKernel\Event\GetResponseEvent
$event
 * The event to process.
 */
public function onKernelRequestMaintenance(GetResponseEvent $event)
{
  $route_match = RouteMatch::createFromRequest($event->
getRequest());
  if ($this->maintenanceMode->applies($route_match)) {
    // Do stuff;
  }
}
```

UN PEU PLUS LOIN

- Liste des événements : class *Events

- Exemple: `ConfigEvents`

- Déclenchement des événements: `eventDispatcher->dispatch(EVENTNAME Event);`

- Exemple:

```
$this->eventDispatcher->dispatch(ConfigEvents::SAVE, new ConfigCrudEvent($this));
```

Supprimez vos hooks, et utilisez des événements ! : <https://www.drupal.org/node/230265> |

SERVICE-ORIENTED ARCHITECTURE

- Definition:
 - A Service is any PHP object that performs some sort of "global" task. (Symfony documentation)

Déclaré via *.services.yml

http://symfony.com/doc/current/book/service_container.html

SERVICE CONTAINER

- Definition:
 - A *[Service Container](#)* (or dependency injection container) is simply a PHP object that manages the instantiation of services (i.e. objects). (Symfony documentation)

http://symfony.com/doc/current/book/service_container.html

EXAMPLES

coding time !

TRAITS

- Exemple:

```
use RfcLoggerTrait;
```

HOOK() > ANNOTATIONS

hook_block_info()

```
/**
 * Implements hook_block_info().
 */
function search_block_info() {
  $blocks['form'] = array(
    'info' => t('Search form'),
  );
  return $blocks;
}
```

```
/**
 * Implements hook_block_view().
 */
function search_block_view($delta = '') {
  if (user_access('search content')) {
    $block['content'] =
drupal_get_form('search_block_form');
    return $block;
  }
}
```

```
/**
 * Provides a 'Search form' block.
 *
 * @Block(
 *   id = "search_form_block",
 *   admin_label = @Translation("Search form"),
 *   category = @Translation("Forms")
 * )
 */
class SearchBlock extends BlockBase {

  protected function blockAccess(AccountInterface $account) {
    return AccessResult::allowedIfHasPermission($account, 'search
content');
  }

  public function build() {
    return \Drupal::formBuilder()->
getForm('Drupal\search\Form\SearchBlockForm');
  }
}
```

MERCI À TOUS !

Vos questions ?