

# Initiation a Drupal

## slides

Les slides de la présentation sont disponible à cet url

[bit.ly/drupalini](http://bit.ly/drupalini)  
[initiation-a-drupal.pdf](#)

## Agenda

- Drush
- Module et hook
- API

## Drush

### Drupal shell

Drush c'est le drupal shell, c'est l'outil en ligne de commande de drupal.

[Documentation](#) | [drush.org](http://drush.org)

### Project manager

Drush contiens un projet manager, qui peut télécharger et mettre à jours le core de drupal, les thèmes et les modules.

```
$ cd ~/Sites
$ drush pm-download
Project drupal (7.34) downloaded to /Users/dl/Sites/drupal-7.34
```

### Installation

Drush permet d'installer drupal, ce qui créé ou écrase la base de données et les fichier de configuration.

```
$ cd /Users/dl/Sites/drupal-7.34
$ drush site-install standard \
```

## Mise à jours

On peut faire les mises à jour du code des modules et du core ainsi que la base de données par drush.

```
$ drush pm-updatecode
No code updates available.
```

```
$ drush updatedb
No database updates required
```

## Variables

Drush permet d'inspecter et de modifier les variables de drupal.

```
$ drush variable-get name

$ drush variable-set site_name new-name
site_name was set to "new-name".
```

## script

On peut utiliser drush pour executer des scripts php ce qui est pratique pour explorer l'api.

```
$ echo "<?php var_dump(menu_tree_page_data('main-menu')));" \
> test.php
$ drush php-script test.php
array(1) {
  '50000 Home 218' =>
  [...]
}
```

## Gestion des modules

On peut aussi gérer les téléchargements, installations et mise à jour des modules

Avec la commande suivante on télécharge le projet exemple de drupal. Ce sont des exemples de module pour apprendre à travailler avec les api backend de drupal.

```
$ drush pm-download example
Project examples (7.x-1.x-dev) downloaded to sites/all/modules/examples.
```

[Examples for Developers | drupal.org](#)

## block\_example

Nous allons étudier le module 'block\_example' qui démontre comment créer un block custom.

Au minimum un module contient un fichier de configuration block\_example.info et un fichier avec le code block\_example.module.

Le fichier block\_example.install contient le script d'installation et le block\_example.test contient les tests d'intégration drupal avec Simple Test.

```
$ cd sites/all/modules/examples/block_example
$ ls
block_example.info
block_example.install
block_example.module
block_example.test
```

### block\_example.info

Le fichier info contient l'information du module qui seront affichés dans l'interface d'administration ou dans la commande `drush pm-info`.

```
$ head -4 block_example.info
name = Block Example
description = An example outlining how a module can define blocks.
package = Example modules
core = 7.x
```

### HOOK\_block\_info

Dans le fichier .module on définit des hook pour que drupal puisse appeler notre code. Les hook sont des fonctions qui débutent avec le nom du module.

Par exemple pour implémenter hook\_block\_info on définit une fonction avec le nom block\_example\_block\_info().

```
/**
 * Implements hook_block_info().
 *
 * This hook declares what blocks are provided by the module.
 */
function block_example_block_info() {
```

Le hook block\_info() retourne les informations de notre block à drupal.

```

$blocks['example_uppercase'] = array(
  'info' => t('Example: uppercase this please'),
  'status' => TRUE,
  'region' => 'sidebar_first',
);
return $blocks;
}

```

Les valeurs possible de l'array de block est documenté sur le site de drupal.org

[hook\\_block\\_info | drupal.org](#)

[hooks | drupal.org](#)

Lorsque drupal as les information du block, on peut l'assigner à une région dans l'interface administrateur.

[example.local/admin/structure/block](#)

[Show row weights](#)

Block	Region	Operations
<b>Content</b>		
⊕ Main page content	Content ▼	<a href="#">configure</a>
<b>Content pre</b>		

Figure 1:

## hook\_block\_view

Le hook block view est appelé par drupal avant d'afficher les block et le machine name du block est passé en argument.

```

function block_example_block_view($delta = '') {
  switch ($delta) {
    case 'example_configurable_text':

```

On as simplement à ajouter la string qu'on veut afficher comme valeur à la clé 'content' pour qu'elle s'affiche dans le block. ### example\_empty

```

    case 'example_empty':
      $block['subject'] = t('Title of second block');
      $block['content'] = block_example_contents($delta);
      break;

```

```
drush dl mothership drush cc all
```

drush |grep mothership Other commands: (make,mothership) mothership Create a mothership sub-theme. drush mothership test Mothership subtheme "test" created in: /Users/dl/Sites/drupal-7.34/sites/all/themes/test Visit your themes settings page and configure it to your liking: <http://default/admin/appearance/settings/test>  
## Système de région créer une nouvelle région Associer un block à une région  
## Système d'ouverture de template dans drupal

## hook\_\_menu

Les routes dans drupal sont déclaré en utilisant le hook menu. Ça porte à confusion au début mais pour les développeur drupal un menu ce n'est pas un menu c'est une route, le path pour accéder à une page custom.

Nous allons regarder les exemples fournis par drupal.

```
$cd ~/Sites/drupal-7.34/sites/all/modules/examples/menu_example
```

```
$ls
```

```
menu_example.info  menu_example.module menu_example.test
```

```
/**  
 * Implements hook_menu().  
 *  
 * A simple example which defines a page callback and a menu entry.  
 */  
function menu_example_menu() {
```

Dans le fichier menu\_example.module on définit le hook\_\_menu que nous allons utiliser pour retourner les informations sur nos routes à drupal.

```
example.local/examples/menu_example
```

```
function menu_example_menu() {  
  $items['examples/menu_example'] = array(  
    'page callback' => '_menu_example_basic_instructions',  
    'access callback' => TRUE,  
  );  
  return $items;  
}
```

Chaque clés de l'array que nous allons retourner seras utilisé comme route.

Et la clé 'page callback' contiens la chaine de caractère qui seras appelé pour cette route.

```
function _menu_example_basic_instructions($content = NULL) {
  $base_content = t(
    'This is the base page of the Menu Example. There are a number of examples
    here, from the most basic (like this one) to extravagant mappings of loaded
    placeholder arguments. Enjoy!');
  return '<div>' . $base_content . '</div><br /><div>' . $content . '</div>';
}
```

La fonction callback définie dans le hook\_menu retourne le markup à afficher sur dans le contenu de la route.

Voici la documentation du hook\_menu et celle du hook\_i18n\_translate\_path qui est pratique pour associer les routes des différentes langues pour que drupal puisse construire le language switcher.

[hook\\_menu | drupal.org](#)

[hook\\_i18n\\_translate\\_path | drupalcontrib.org](#)

## hook\_theme

Le module theming\_example.module démontre différent exemples d'utilisation du hook\_theme. Souvent on l'utilise simplement pour créer un template custom dans un module.

On n'as qu'à retourner la clé de notre thème et le nom du fichier template au hook.

```
function my_example_theme() {
  return [
    my_theme_key => [
      'template' => 'my-template-name'
    ]
  ];
}
```

Puis lorsqu'on utilise cette clé dans la fonction thème ça nous retourne une chaine de caractère contenant le markup du fichier template my-template-name.tpl.php

```
function _menu_callback() {
  return theme('my_theme_key', ['title'=> "variable value"]);
}
```

Le deuxième argument de la fonction thème contient les variables du template et les clés seront les noms des variables.

my-template-name.tpl.php

```
<h1><?php print $title ?></h1>
```

## hook\_preprocess\_hook

En déclarant le hook\_preprocess\_hook un module peut modifier les variables qu'il passe à un template.

Comme premier exemple le html.tpl.php est le premier template de drupal, il contient par exemple la balise

.

```
function my_example_preprocess_html( &$variables ) {  
    $variables['google_analytics_id'] = "1234543";  
}
```

Le template page.tpl.php contient le layout général du site, les régions, les menus, le header, le footer.

```
function my_example_preprocess_page( &$variables ) {  
    $variables['header'] = _get_custom_header();  
}  
  
function my_example_preprocess_node( &$var ) {  
    if ($var['type'] == 'my_content_type'){  
        $var['my_variables'] = _my_processing($var['node']);  
    }  
}
```

[hook\\_preprocess\\_HOOK | drupal.org](#)

## Variable

Drupal a un array de variable "persistante" qui est enregistré dans la base de données.

On peut utiliser la fonction variable\_get() avec la clé de la variable pour obtenir sa valeur.

```
$variables['google_analytics_id'] = variable_get('my_ga_id');
```

il y a un module nommé variable qui permet de déclarer des pages de configuration dans l'interface administrateur de drupal pour les variables.

```
$drush dl variable
```

```
Project variable (7.x-2.5) downloaded to sites/all/modules/variable.
```

```
Project variable contains 6 modules: variable_views, variable_store, variable_realm, variable_example
```

```
$ cd ~/sites/drupal-7.34/sites/all/modules/variable/variable_example
```

Pour obtenir une page de configuration pour notre variable on commence par le hook `variable_info` pour configurer les métadonnées de notre variable et on l'associe à un groupe de variable.

```
//variable_example.variable.inc
function variable_example_variable_info($options) {
  $variables['variable_example_text'] = array(
    'type' => 'text',
    'title' => t('Simple text', array(), $options),
    'default' => 'Example text.',
    'description' => t('Example of text variable.', array(), $options),
    'required' => TRUE,
    'group' => 'variable_example',
  );
}
```

Ensuite on déclare les informations du groupe de variable qu'on va afficher sur notre page de configuration et on l'associe à un path.

```
function variable_example_variable_group_info() {
  $groups['variable_example'] = array(
    'title' => t('Examples'),
    'description' => t('Variable examples of different types.'),
    'access' => 'administer site configuration',
    'path' => array('admin/config/system/variable/example'),
  );
}
```

Finalement on utilise `variable_group_info` pour déclarer la route custom pour la page d'administration de notre groupe de variables.

On ajoute le callback `drupal_get_form` et l'argument `variable_group_form` ainsi que le groupe de notre.

```
//variable_example.module
function variable_example_menu() {
  'path' => array('admin/config/system/variable/example'),
  'title' => 'Variable example',
}
```



```

    'description' => 'Example of auto generated settings form.',
    'page callback' => 'drupal_get_form',
    'page arguments' => array('variable_group_form', 'variable_example'),
    'access arguments' => array('administer site configuration'),
  );

```

## Api de contenu

EntityFieldQuery Entity api ## Création d'un type de content ## Utilisation du module field collection ## Gestion des menus

Modication de la portion backend ## Gestions des librairies et autoloader (composer) ## Exemples de code et test automatisé (phpspec) ## Traitement des variables des éléments de conteu Drupal adapter Arrays functions Fields functions Menu

Utilisation de git et release cycle ## Fonction des différentes branches ## Fonction des différents Environnements ## Déploiement du code et mise à jours de la configuration

Gestion de la configurations et mise à jours automatisé

## Configuration drupal par script automatisés

```

fichier.install
hook_update_n

```

```

## gestion des variables et des modules ## gestion des roles et permissions
## Export de la configuration par features ## Module d'intégration

```