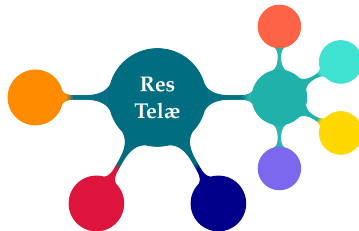


# Profils d'installation sous Drupal 7

Felip Manyer i Ballester



18 décembre 2013

# Plan

- 1 Profils d'installation
  - Introduction
  - Distributions
  - Déploiement
- 2 Anatomie d'un profil d'installation
  - Il était une fois les profils d'installation
  - Tâches d'installation
  - Makefiles
- 3 Un exemple de processus d'industrialisation
  - My Life and Work
  - Moving Forward

- 1 Profils d'installation
  - Introduction
  - Distributions
  - Déploiement
- 2 Anatomie d'un profil d'installation
  - Il était une fois les profils d'installation
  - Tâches d'installation
  - Makefiles
- 3 Un exemple de processus d'industrialisation
  - My Life and Work
  - Moving Forward

# Introduction

## Profil d'installation

Permet de personnaliser Drupal à l'installation : étapes d'installation, modules activés, configuration. Utile pour **automatiser** et éviter les tâches répétitives.

Vous utilisez déjà des profils d'installation : choix entre `minimal` et `standard` sous D7, uniquement `default` sous D6.

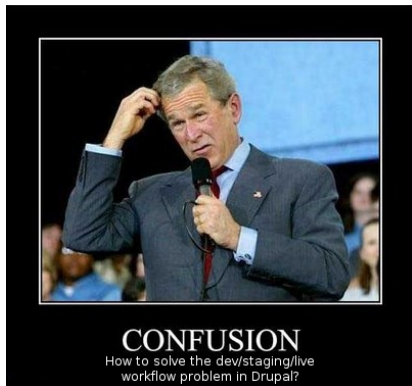
- Mise en place de sites complexes, répondant à un cas d'utilisation spécifique : les **distributions** ;
- un outil précieux pour l'**industrialisation** : construisez votre **kit de développement** ;
- ne livrez plus « un site », mais un profil d'installation.

# Distributions

Cf. la présentation de Simon Georges au drupalcamp Paris : *État des lieux des distributions Drupal*.

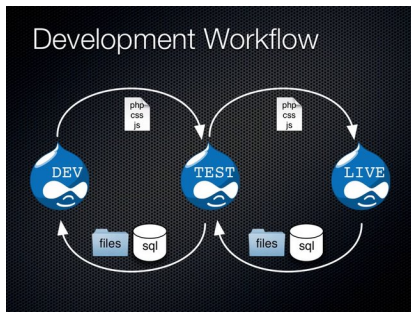


# Déploiement sous Drupal



- Instances différentes, par ordre de stabilité croissante : dév, test, pré-prod, prod.
- Péchés originels de Drupal : configuration dans la base → comment conserver une trace du processus de développement ?

# Déploiement sous Drupal



- Le code migre de **dev** à **prod**.
- Le contenu (BD) migre **dans l'autre sens**.
- Rôle central du système de **gestion de versions**.
- Un pas plus loin :  
intégration continue.

# Profils d'installation et industrialisation

- Features, puis CMI (D8) vous permettent de **baser votre processus de développement sur le code** (code driven development), et de déployer proprement de l'instance de développement à l'instance de production.
- Construire un site avec Drupal revient à créer une **application**, s'appuyant sur le cœur et les modules contributifs, constituée de modules personnalisés et de *features*, ainsi que d'un thème ⇒ c'est elle que vous placez sous **gestion de version**.
- Cette application est un **profil d'installation** ; un site est une **instance de cette application** (cf. mig5) ⇒ fonctionnalités **multi-sites**.
- Une installation Drupal vierge est une **plate-forme**, pouvant par exemple être décrite avec des **makefiles** (cf. **drush**).



- 1 Profils d'installation
  - Introduction
  - Distributions
  - Déploiement
- 2 Anatomie d'un profil d'installation
  - Il était une fois les profils d'installation
  - Tâches d'installation
  - Makefiles
- 3 Un exemple de processus d'industrialisation
  - My Life and Work
  - Moving Forward

## Vue générale

- Les profils d'installation vivent (heureux) dans le répertoire `profiles/`, en compagnie de `minimal` et `standard`, fournis par défaut.
- Depuis Drupal 7, ils sont semblables à des modules et sont constitués de trois fichiers : `.info`, `.profile` et `.install`.
- Ils peuvent embarquer des modules (dont des *features*) et des thèmes.
- Définition de tâches d'installation, modification des tâches existantes.

## Arborescence pour les modules et thèmes

- Il existe un ordre de précedence : `sites/fqdn/modules` est pris en compte avant `sites/all/modules` et `profiles/modules`.
- Placez les modules contribués dans le sous-répertoire `contrib`, vos modules dans `custom`, et les features dans... `features`, vous y verrez plus clair.

## Fichier .info

- Nom et description du profil.
- Modules à activer à l'installation (dépendances). Dès que possible, je préfère déclarer des modules comme dépendances de mes features.
- Ces modules (+ thèmes, bibliothèques et traductions) sont listés dans un makefile et téléchargés avec Drush♥
- Annotations supplémentaires (date, version) ajoutées automatiquement dans le cas des distributions de drupal.org.

# Fichier .install

- C'est ici qu'on implémente `hook_install`, comme dans un module 😊 : activation des thèmes, blocs « système » (content et help), activation features, filtres de texte...
- Pour ma part, je préfère en fait définir des tâches d'installation et utiliser la Batch API.
- (Autres blocs placés avec Context dans Features)

# Fichier .profile

- Équivalent du .module, deux utilisations.
  - Altérer le processus d'installation (ajouter ou altérer des formulaires).
  - implémenter des hooks comme n'importe quel module (mais vous savez créer des modules...).
- Ne me sert pas 😞.

## Tâches d'installation : formulaires



- ✓ Choose profile
- ✓ Choose language
- ✓ Verify requirements
- ✓ Set up database
- ✓ Install profile
- ✓ Configure site
- ▶ **Setup Client**

Finished

Setup your default client account below.

**Client Username \***

**Client E-mail Address \***

**Client Password**

Continue

# Tâches d'installation : lots (batches)

## Installing Drupal



- ✓ Choose profile
- ✓ Choose language
- ✓ Verify requirements
- ✓ Set up database
- ▶ **Install profile**
  - Configure site
  - Finished




## Tâches d'installation : `hook_install_tasks()`

- Ajouter (ou altérer) les tâches d'installation
  - entrée de données (formulaire). Exemple : données d'identification de l'organisation propriétaire du site (interfaçage possible avec Ægir).
  - définition de tâches supplémentaires. Exemple : à la fin de l'installation, activer les features, configurer le site, créer du contenu (pages types, mentions légales, Beans...).
- Ne pas hésiter à utiliser la Batch API (cf. barre de progression).
- Ne pas hésiter à copier un profil existant (Standard, Commerce Kickstart, etc.).
- Création de contenu : pourquoi pas utiliser Devel Generate ou Migrate ? (cf. Makina Corpus).

# Makefiles

- Format de fichier permettant de décrire une plate-forme Drupal, à la version près, et de la reconstituer (de manière reproductible) grâce à Drush.
- Récupération automatique du cœur, des modules, thèmes, bibliothèques, traductions, via HTTP, SVN, Git... éventuellement patchés (proprement !) à la volée.
- Mettez sous gestion de version la liste des modules contrib, pas les modules eux-mêmes au sein de votre profil d'installation.
- Ægir : mettre à jour  $\equiv$  migration entre plates-formes.

 Voir l'exemple

- 1 Profils d'installation
  - Introduction
  - Distributions
  - Déploiement
- 2 Anatomie d'un profil d'installation
  - Il était une fois les profils d'installation
  - Tâches d'installation
  - Makefiles
- 3 Un exemple de processus d'industrialisation
  - My Life and Work
  - Moving Forward

## Retour d'expérience

- J'utilise un profil d'installation de base (`fonamental`) comportant notamment mes features, modules et thèmes, ainsi que les makefiles permettant de reconstituer la plate-forme (+ script de construction, `build.sh`).
- Utilisation d'un préfixe différent, commun à l'ensemble du projet, pour chaque client (`stakhanov`).
- Déploiement basé sur Git et `Ægir`.
- Depuis peu, intégration continue avec Jenkins (interfaçage avec `Ægir`).



### Démonstration

## Conclusion générale

- Features, CMI : des outils pour capturer en code et enfin déployer et mettre sous gestion de versions proprement et efficacement.
- Profils d'installation : la bonne manière d'empaqueter votre application Drupal (instance d'une plate-forme multi-sites).
- Ouvre la porte à des outils tels qu'Ægir (gestion de plates-formes et d'instances Drupal avec Drush et le profil Hostmaster : automatisez la gestion de sites) et Jenkins (intégration continue).
- Munissez-vous d'outils vous permettant de collaborer, de vous concentrer sur votre métier et de gagner en efficacité.

## Références

- BUTCHER Matt & al., *Drupal 7 Module Development*, chapitre 12, Packt Publishing, décembre 2010
- COLOMBO Tristan, *Intégration continue avec Jenkins*, GNU/Linux Magazine/France (hors-série n° 69), novembre-décembre 2013
- CORBIN Sébastien, *Drush Make Driven Development*, <http://paris2013.drupalcamp.fr/node/237>
- GEORGES Simon, *État des lieux des distributions Drupal*, <http://paris2013.drupalcamp.fr/node/232>
- HADFIELD Scott, *Custom steps and forms in a D7 install profile*, <http://scotthadfield.ca/node/408>

## Références

- JACQ Miguel, *Drupal deployments & workflows with version control, drush\_make, and Ægir*, <http://mig5.net/node/115>
- JACQ Miguel, *Developing for Ægir — extending Ægir and communicating with install profiles*, <http://mig5.net/node/286>
- JACQ Miguel, *Zero-touch Drupal deployment with Jenkins, Aegir, Git, Fabric and Drush*, <http://mig5.net/node/342>
- MELANÇON & al., *The Definitive Guide to Drupal 7*, chapitre 34, Apress, 2011
- TOMLINSON Todd & al., *Pro Drupal 7 Development*, chapitre 24, Apress, 2010