

Drupal (V7) : principes et petite expérience



Jean-Luc Archimbaud CNRS/Mathdoc
<http://jl.archimbaud.free.fr/>

Présentation réunion SARI Drupal 4 oct 2012
V1.0

Corrigez mon discours ou complétez...

Licence de cette présentation : Creative Common BY-NC-ND

1

Mon expérience Drupal



- Plate-forme PLUME (<http://www.projet-plume.org>)
 - Initiateur, responsable du projet, rédacteur en chef (2006 – avril 2012)
 - Participation à la décision du choix drupal (mai 2007) - avec un cahier des charges
 - <https://www.projet-plume.org/ressource/description-du-processus-de-choix-dun-cms-pour-le-projet-plume>
 - Étudiés : Drupal e107, eZpublish, Jaws, Joomla, Lodel, MODx, MySource, SPIP, Textpattern, Tiki CMS, Typo3, Xaraya, Xoops
 - Short liste : Joomla, Drupal, SPIP, Typo3, MODx
 - D5 - D6 : contenus, rôles et accès, menus, vues, workflow, taxonomy, bilingue, migration
- Web laboratoire mathdoc (<http://www.mathdoc.fr>) : mai 2012-...
 - Admin, web design, responsable éditorial
 - Choix avant mon arrivée, reprise du travail d'un CDD
 - D7 : installation modules, CSS... + idem PLUME
- Futur: projet portail math : sept 2012 - ...
 - Projet : http://www.mathdoc.fr/projet_portail_math
 - Maquette D7 en court d'ouverture pour création du portail en mode collaboratif

Le logiciel



- CMS (Content Management System)
 - et framework de programmation (API : 4 000 fonctions)
- Logiciel (<http://drupal.org/>)
 - Programmeur à l'origine : Dries Buytaert, université d'Anvers, 2000
 - Drupal (en anglais) : druppel (en hollandais) : goutte
 - Logiciel libre : licence GNU GPL avec très grosse communauté
 - Ecrit en PHP avec fonctions JavaScript (bibliothèque JQuery)
 - Tourne sous Linux, Windows, MacOS
 - Cœur + modules obligatoires + 1000 modules optionnels (D7)
 - Arborescence dans l'espace disque du système, souvent un répertoire par module
 - Les modules peuvent être installés, mis à jour, (des)activés séparément
 - Multilingue : contenu et interface d'administration
 - Traduction de l'interface d'administration 'communautaire' (dynamique)
 - Une instance peut gérer n sites web

Le logiciel



- Nécessite
 - PHP
 - Base de données MySQL ou PostgreSql ou SQLite
 - Données, infos utilisateurs, paramétrage...
 - Remarque : certains paramètres et CSS... sont dans l'arborescence système
 - Apache ou IIS
- Une instance drupal = BdD + arborescence de fichiers
 - On ne peut pas séparer les deux
- PLUME et Mathdoc : MySQL, Apache et modules ajoutés :
 - Backup, Diff (entre versions), Help, Path (alias URL), Search, Spanspan, Statistics, Superfish (menus), Update manager, Views UI (éditeur de vues), workflow actions....,...



- Méthodes diverses pour
 - Créer les comptes : formulaire en accès public (choix PLUME), par l'administrateur uniquement (choix Mathdoc)...
 - Authentifier : Login-mdp (choix Mathdoc, PLUME), OpenId (possibilité PLUME)...
- Chaque utilisateur a un ou plusieurs rôles
- On peut créer des rôles, chacun avec des accès spécifiques comme
 - Installation-MAJ du logiciel, admin drupal, création de menus, création-modification-lecture de chaque type de contenu/menu/bloc, actions possibles durant workflow...
 - Mathdoc : membre Mathdoc (accès en lecture-modification sur toutes les pages) - resp éditorial (en + : menus, création de pages) - admin (en + : installation-MAJ modules)
 - PLUME : membre (lecture fiches logiciel à valider, commentaires...) - contributeur (en + : rédaction-modif de ses fiches, relecture de certaines fiches...) - resp thème (en + : création fiches, droits / workflow, lecture de tout contenu non publié...) - rédac chef (en + : modifs menus, modifs workflow, modification de tout contenu...) - admin (en + : instal modules...)



- Tous les contenus sont des nœuds (nodes) : /node/xxx
 - Mais il existe un système d'alias : /equipe → /node/12
 - Chaque contenu a un type (une structure définie)
 - Pas d'arborescence !
- Type de contenu (content type)
 - Suite de champs typés (texte, liste, booléen, sélection...)
 - Texte : HTML (full ou filtered) ou texte ASCII ou markdown
 - Masque de saisie du contenu
 - Un par défaut qui est modifiable
 - Affichage du contenu
 - Un par défaut qui est modifiable
 - Chaque contenu (node) a
 - Un propriétaire (créateur), dates création et dernière modification...
 - En option : accès particuliers, workflow, commentaires, versions conservées, fichiers attachés...
 - On peut définir plusieurs types de contenus. Exemples :
 - Mathdoc : page (publique, sans commentaire), page privée (accès limité rôle Mathdoc, commentaires possibles)
 - PLUME : page publique, page privée, fiche logiciel, fiche ressource, ticket, news...

- La page est découpée en blocs (rectangles)
 - répartis dans des barres horizontales (entête, pied de page...) ou des colonnes (à gauche, à droite...) ou dans le pavé central
- Chaque bloc peut avoir une présentation particulière (encadrement, fond, polices...) et on peut déplacer les blocs dans les barres-colonnes...
- Des canevas de page sont fournis : thèmes
- Généralement un site choisit un thème fourni puis le modifie
 - Mathdoc : Bartik → Bartik_mathdoc
- Un site peut avoir plusieurs thèmes (suivant le rôle de l'utilisateur...)
- Des blocs peuvent contenir des menus
- L'affichage d'un bloc peut être conditionnel
 - Suivant la page (node) affichée, le rôle du visiteur...

Fonctionnalités utilisées : vues, recherche, taxonomy, workflow

- Une vue
 - Liste les nodes qui répondent à certains critères que l'on définit (type de contenu, dates, propriété, mots-clés, publié ou non...) et affichent certaines infos des nodes (que l'on choisit)
 - Affichage en liste, tableau... : dynamique, sans besoin de configuration pour la présentation
 - PLUME : Fiches par thème, chronologiques...
- Recherche de texte intégral : sur le contenu HTML-texte, pas sur le PDF des fichiers attachés
- Mots clés (taxonomy)
 - On peut définir des mots clés (et les classer en ensembles, éventuellement hiérarchiques)
 - Les affecter ou non aux différents types de contenu
 - Faire une recherche via ces mots-clés
 - PLUME : de très nombreux mots clés dans des ensembles
- Création de workflows : modules workflow
 - Affecté à certains types de contenu, avec des états - des droits suivant le rôle pour changer d'état, lire-modifier suivant l'état - des actions (envoi de mails...) quand passage d'un état à l'autre
 - PLUME : processus de création-relecture-publication de fiche, de MAJ de fiche...

Exploitation : 3 sites Mathdoc



- Logiciel et données sont liés : on ne peut pas les faire évoluer séparément
- 3 sites – instances drupal à Mathdoc (similaire dans PLUME)
 - Production (accessible depuis internet)
 - Développement (pour développer)
 - Test (pour essais pouvant être destructifs)
- Le site de référence : le site de production
- Ajout/modif de contenu - petite modif config : directement sur le site de production
- Modif importante - installation/MAJ de modules :
 - Essai sur site de test ou de développement
 - Rejoue sur site de production
 - Car : pas de retour en arrière (undo) possible dans les MAJ de modules
- Périodiquement (mais pas dans le cron) : copie production → test ou développement

Mon avis



- Drupal demande une période d'apprentissage assez longue
 - Pas adapté pour réaliser (rapidement) un petit site
- Avantages Drupal
 - Logiciel libre, grande communauté efficace, évolue rapidement (trop?)
 - Très large utilisation internationale
 - Hyper robuste
 - Souple (présentation-ergonomie, types de contenu, modules, menus...)
 - Dynamique (changement dans menus, création-modification de type de contenu, traduction...)
- Désavantages
 - Complexe (car beaucoup de fonctionnalités)
 - Données et configuration pas vraiment séparées
 - Changement de version, MAJ de modules : difficile (expérience PLUME)
- Recommandation : keep it simple
 - N'installer un module que si on en a vraiment besoin
 - Modifier au minimum
 - Développer au minimum
 - Pas de configuration trop complexe : ne pas multiplier les rôles...
 - Recommandation pour tout site Web : le design en dernier ! Le contenu d'abord