Drupal \_ Découverte

Username : [serge@corpulence.fr](mailto:serge@corpulence.fr)

Password : Mfovp15-vM.asR95!

**Licences**

**🡪 GNU** General Public License (**GPL**), version 2 ou ultérieure

le **logiciel** et les **fichiers partagés** hébergés sur *Drupal.org*

🡪 **Creative Commons Attribution-ShareAlike 2.0**

**Contenu du site** drupal.org

**Rappel**

[**EasyPHP Devserver**](https://www.easyphp.org/easyphp-devserver.php)**-** Inclut à la fois Apache et Nginx.

[**Wamp.NET**](https://wamp.net/)**-** Inclut à la fois Apache et Nginx, ainsi que Mysql et MariaDB.

[**WampServer**](https://www.wampserver.com/en/)**-** Inclut à la fois MySql et MariaDB.

[**XAMPP**](https://www.apachefriends.org/index.html)**-** Inclut MariaDB mais pas MySQL. Des modules complémentaires sont disponibles, notamment Drupal.

**Recherches**

OpenAI 🡪 Consulter mon compte

**Todo list**

**A.A / Veille technologique**

**0-** Maquetter « DY-WO » et structurer les l’implémentation de la veille technologique.

**1-** Eclairer la distinction entre la plateforme Drupal et ses axes, Drupal 11 et Drupal CMS.

**2-** Comprendre ce en quoi Twig et Symfony sont différents. Apprécier le degré de liberté de Twig dans l’usage du Framework Symfony pour mieux combiner les complémentarités des zones architecturales du site.

**3-** Présenter un Template de définition avancé pour les outils suivants en veillant à initialiser l’identification de leur interopérabilité. Le résultat est un mini site dynamique qui pourra facilement s’encastrer dans les interfaces de management admin. Il sera ainsi plus facilement mis à jour et apparaîtra corrigé dans toutes les applications qui y feront appel. Il fera ainsi partie de la suite d’applications autonomes que nous voulons nommer “DY-WO”.



A white background with blue text and black text

AI-generated content may be incorrect.

|  |  |
| --- | --- |
|  |  |

**Todo list**

**A.A / Veille technologique**

**4-** Présenter un Template inclusif avancé pour les étendus de codes, commandes et mots clés de chaque langage revendiqué. L’outil traite Sens, Usage, Ramification et pertinence des exemples **sans aucun dynamise**, ce conformément aux modèle DY-WO et 7views.

**5-** Présenter un Template **de définition** avancé pour les synthèses procédurales. L’outil présente une synthèse pour chaque procédure accessible via un menu inclusif à définir. Chaque procédure sera détaillée dans un sous menu commun à toutes les procédures. L’objectif est de gagner en confiance procédurale et en vitesse de compréhension.

**6-** Présenter un Template avancé pour la mémoire du troubleshooting propre aux projets, avec un mécanisme de factorisation (BDD indexée, tables de relation, template de monitoring PHP et scripte JS concerné).

**7-** Définir le cheminement méthodologique protocolaire idéal pour une action d’enregistrement ayant un impact sur une des interfaces d gestion administratives ou de veille technologique.

**Todo list**

**A.B / Réussir le travail en équipe**

**1-** Créer tous les espaces de travail favorables à la réussite du travailler en groupe.

🡪 Double contrainte : manque de **temps** et **distance** entre les lieux de travail rescpectifs

🡪 Solution : soutenir l’approche DevOps par l’inclusion des plateformes actuelles.

° Décrir un schéma de redondance marqué par la compatibilité.

° effectuer les inscriptions et les invitations pour positionner le groupe admin.

° Recenser le potentiel d’action acquis et donc la redondance validée pour mieux centrer les effert sur le besoin réel que constitue le site web client et admin à créer (justification du cahier des charges).

* Plateforme Drupal
* GitLab
* Azure DevOps
* GitHub
* Slack
* Teams
* Corpulence (service web de plateforme pédagogique et de stockage dynamique des données : BDD + storage + Backup)

**Start Points**

**🡪** Security support for Drupal 7 ended on 5 January 2025.

**Git**

DrupalCode Access

**Internanal login recognition**

🡪 Drupal fonction sur GitLab

🡪 Settings management accès : **git.drupalcode.org**

**🡪** My git.drupalcode.org username : **sergeHounhouayenou**

**Extgernal login recognition**

🡪 Recommended account information configuration :

git config --global user.name 'Hounhouayenou Serge'  
git config --global user.email 'serge@corpulence.fr'

**Emails déclarés sur git.drupalcode.org**

[serge@corpulence.fr](mailto:serge@corpulence.fr) 🡪 vérifié : statut OK

[hoserge@gmail.com](mailto:hoserge@gmail.com) 🡪 vérifié : statut OK

**Slack**

https://app.slack.com/client/T06GX3JTS

**Azure DevOps**

(Pilines)

**A.A-1**

**Start point**

🡪 Eclairer la distinction entre la plateforme Drupal et ses axes, Drupal 11 et Drupal CMS

**Champ lexical à définir**

- Drupal (Plateforme ; site web ; Drupal Core ; Drupal 11)

- Drupal CMS

- Twig

- Symfony

- Composer

- PHP Unit

- PHP Stan

**A.A-1.1 : Drupal\_ Constellation**

**La constellation**

🡪 S’exécute idéalement sur la pile backend « **LAMP** » mais reste compatible

* Linux, Windows ou Mac
* Apache, Nginx, IIS
* MySQL, PostgreSQL, SQLite ; MariadB ; Percona
* Etc. Contrainte : Drupal est écrit en PHP

🡪 Les fonctionalités sont dans les “**modules**” et les layouts sont dans les “**thèmes**”

Un simple fichier CSS suffit à personnaliser son thème. Pour plus de précisions : *https://www.drupal.org/docs/theming-drupal*

🡪 Chaque module et thème installé augmente la durée de chargement des pages. Il convient donc de désinstaller ceux qui ne sont pas utilisés.

**Drupal Core**

🡪 C’est le noyau du logiciel. Equivalent à docker Engine pour Docker, il établit la relation initiale avec l’OS sur lequel il s’installe pour créer l’environnement interne d’exécution des instructions. Il contient ainsi les fonctionalités de base pour les actions élémentaires du logiciel Drupal.

🡪 Cela dit, Drupal Core s’administre aussi bien via le Terminal grace aux outils tels que “Drush” qu’avec son interface graphique utilisateurs, propre à chaque site, accéssible via le navigateur à l’url du site.

🡪 Fonctionalités de base :

* Gestion des comptes utilisateurs
* Gestion du contenu initial : module Node pricipal (pages) et Champs relatifs (module “field principal”, module “field UI”, modules “field types”).
* Gestion ds menus de navigation
* Gestion d’affichages listes, grilles et blocs pour de contenus exiistants (modules principaux “Views” et “Views UI”)
* Téléchargement de modules complémentaires sur Drupal.org
* Créaon de modules personnalisés.

**Drupal CMS**

🡪 C’est un CMS à usage générale. Il sera donc moins adapté au besoins très spécifiques.

🡪 Il constitue une base libre solide que l’on peut développer à des besoins spécifiques.

**Les Distributions**

Complètes dans le sens qu’elles intègrent tous les besoins spécifique de site générique représenté, ou autres selon leurs singularités ou leur caractère partiel, les distributions sont des « paquages projet » qui incluent la version Drupal de Base aux configurations modules et thèmes complémentaires nécessaires à paramétrer (Cf. distribution Linux = Kernel + fonctionnalités spécifiques).

**A.A-1**

**End point**

**Apprendre Drupal**

**🡪** Udemy proposent des cours dédiés à Drupal, allant du niveau débutant à avancé.

**Guides et tutoriels** : Il existe de nombreux guides et tutoriels en ligne qui peuvent vous aider à comprendre les bases de Drupal ainsi que des fonctionnalités plus avancées. Par exemple, un guide complet pour débutants et experts est disponible et couvre les éléments essentiels à connaître avant de commencer une formation.

**🡪 Documentation officielle** : Consultez la documentation officielle de Drupal. Elle est très complète et vous fournira des informations détaillées sur toutes les fonctionnalités de la plateforme.

🡪 PGF en ligne : https://www.drupal.org/files/ebooks-en-10.2.x.tgz

Aide Générique proposé par le GRETTA : <https://www.youtube.com/@certifacademy/videos>

**Certification Drupal**

**🡪 Certifications Acquia.**

* Acquia Certified Site Builder : Pour ceux qui construisent des sites Drupal.
* Acquia Certified Developer : Pour les développeurs Drupal.
* Acquia Certified Front End Specialist : Pour les spécialistes du front-end.
* Acquia Certified Back End Specialist : Pour les spécialistes du back-end.

🡪 formations publiques et privées

🡪 ressources d'apprentissage gratuites via l'Acquia Academy.

🡪 cours en ligne, guides d'étude et autres ressources éducatives

<https://docs.acquia.com/acquia-academy/study-guides>

https://docs.acquia.com/acquia-academy

<https://www.drupal.org/docs/user_guide/en/index.html>

**Les Types chez Drupal**

**Content**

Information : textes, images, etc… destinés à être affichés aux utilisateurs clients.

**Configuration**

Information de configuration personnelle visible sur le site, les espaces personnels ou les interfaces admin.

**State**

Informations de nature temporaire sur l'état actuel du site, telles que l'heure à laquelle les tâches ont été exécutées pour la dernière fois, etc…

**Session**

Informations sur les interactions individuelles des visiteurs avec le site. Exemple : ils sont connectés et ils ont des cookies. C’est un sous-type de « State » car temporaires.

**Les Régions chez Drupal**

🡪 Les régions sont les parties principales qui constituent la structure de la page Html. Leur choix de configuration par défaut caractérisent les thèmes. A l’image du Head et du Body dans le script Html en lui même, seul la Région “Content” est obligatoire en Drupal. C’est elle qui contient tout le contenu et donc génère les obligations minimales de script.

**Les entités de contenu et de champ chez Drupal**

🡪 Les “Content Entity” sont les entités de types “content” qui définnissent les unités de données de contenu destinés à l’affichage du visiteur, ce quels que soient leurs sous-types ou leurs positions dans les ligatures. Des données telles que les balises htmls en elles mêmes peuvent ainsi être concernées, du fait de devoir être affichées.

🡪 Le Drupal Core, logiciel principal, peut défiir les les entités de contenus de base. On passer par l’installation des modules spécifiques pour les autres besoins.

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity type** | **Entity subtype** | **Defining Module** | **Main uses** |
| **Content item** | Content type | Node module | Content intended to be the main page area for pages on the site |
| **Comment** | Comment type | Comment module | Commentary added to content entities (typically to Content item entities) |
| **User profile** | (none) | User module | Data related to a person with a user account (login access) on the site |
| **Custom block** | Block type | Custom Block module | Text and images in smaller chunks, often displayed in the site header, footer, or sidebar |
| **Taxonomy term** | Vocabulary | Taxonomy module | Used to classify other types of content |
| **File** | (none) | File module | An image or attachment file that is tracked and managed by the site, often attached to other types of content |
| **Contact form** | Form type | Contact module | A form that lets site visitors contact site owners |

**Entity** 🡪 **Fields** : one type of data for each ((defined by Drupal core or additional Modules)

**Les contenus modulaires chez Drupal**

🡪 C’est l’exploitation de la spécificité des sites dynamques quant à leur relation avec une BDD. Les contenues modulaires sont ainsi les tables, les entités mais cette fois ci au sens propre aux BDD. Le but recherché est ainsi l’identification etla factorisation des besoins d’affichage, de dynamisme, d’animation et d’opérations formelles dont le vivsiteur à besoin pour les structurer de manière fractale : sans redondance.

**La gestion du Workflow chez Drupal**

🡪 Les **statuts** **“publié/non-publié”** permmettent de définir qui peut voir les contenus selon les **rôles** des **visiteurs** et des **administrateurs**. Par exemple, les visiteurs n’ont accès qu’aux contenus “publiés”. Les “créateurs” ont accès aux contenus “non publiés”.

🡪 Le **suivi des révisions**, un genre de snapshot intégré, est pris en charge par certains types d’entités de contenus. Cela peut devenir un critère de structuration des ligatures primales de manière à pour mieux stabiliser la structure : du fait de la rendre plus flexible.

🡪 Pour “content item” et “custom blocks”, le module “**Workflow**” basic définit les “**State**” et les “**Transition**”. Le module complémentaire “**Content Modération**”, lui, il attribue des autorisations et des rôles aux “Transition” .

🡪 Ne pas oublier que “custom block” ne pourra rendre visible son contenu sur le site qu’une fois que le dit “custom block” sera prêt. En revanche, s’il ne dispose pas des avantages de prévisualisation dynamique live pré implémentés dans les modèles, il offre toute latitude de personnalisation.

**La gestion des langues chez Drupal**

🡪 La langue de base est l’anglais. Pour utiliser une autre langue que l’angliq sur le site il faut installer le module “language”. Il ne faut pas alors confondre l’acte de traduire les contenus initialement en anglais (ex : GUI) avec le fait de disposer volontairement dans les pages de paragraphes en français. C’est le problème de la reconnaissance efficace des mots et des sens pour que les traductions soient performantes qui se pose. Il faut faire les bons choix de point de départ. Par exemple, le jargon technique de l’administrateur et des créateurs sera toujours mieux compris en anglqis q’en français.

🡪 Les modules complémentaires à télécharger sont :

**1-** Interface Translation module ; **2-** Configuration Translation module ; **3-** Content Translation module

**L’installation de Drupal**

Lien : Exigences PHP : https://www.drupal.org/docs/getting-started/system-requirements/php-requirements

**Configuration des dépendances**

🡪 PHP 8.3 ;

🡪 Apache 2.4.7 (“**mode\_rewrite**” activé et “**Virtualhost**” doté de la directive ***AllowOverride All***)

🡪 Nginx 0.7 ou supérieur (“ngx\_http\_rewrite\_module” activé)

🡪 **MySQL - 8.0+** [ (**MariaDB** 10.6+, Percona 8.0+) avec un moteur de stockage principal compatible InnoDB ] **OU** **PostgreSQL - 16** ou supérieur avec l'extension pg\_trgm **OU SQLite - 3.45+**.

**Autres outils utiles**

🡪 **Git**

See below for more about version control tools.

🡪 **Composer**

[Composer](https://getcomposer.org/) est un outil de **gestion des dépendances PHP**, où le développeur spécifie quelle version de chaque bibliothèque externe est nécessaire, et Composer gère le processus de **téléchargement et d'installation des bibliothèques**.

🡪 **Drush**

Les CLI offrent des fonctionnalités supplémentaires et permettent d’automatiser des tâches plus complexes en écrivant des scriptes qui combinent des commandes directement liées au site (PHP) à des commandes sur le serveur, ce sans oublier les requêtes SQL. Par nature, la GUI est limitée à l’environnement d’exécution du site. Attention à l’analogie logicielle « serveurs locaux » / « serveur production ». **Installation** :

**composer** **require** **drush/drush**

drush site:install standard --db-url='mysql://DB\_USER:DB\_PASS@localhost/DB\_NAME' --site-name=example

🡪**Coder**

[Coder](https://www.drupal.org/project/coder) est un outil en ligne de commande qui **vérifie** si vos **modules et thèmes** sont conformes aux normes de codage et autres bonnes pratiques. Il peut également corriger les violations des normes de codage.

🡪 **Devel**

Débogage, inspection de code et génération de contenu factice.

https://www.drupal.org/project/devel

**Drupal Core**

🡪 il s’installe localement via Composer ou via un environnement préconfiguré contenant Drupal (exemple : une machine virtuelle).

🡪 Démo en ligne gratuite : <https://www.drupal.org/try-drupal>

🡪 vérifier si l’hébergeur n’offre pas une solution d’installation en clic avant de déployer.

🡪 Faire un test local avec un site temporaire simplifié :

**cd** Mon\_Répetoire\_Voulu

**php** **-d** memory\_limit=256M **core/scripts/drupal** **quick-start** **demo\_umami**

**Machine Virtuelle Bitnami**

A screenshot of a computer

AI-generated content may be incorrect.

🡪 Préconfiguré avec les modules Ctools et Views.

🡪 Configuration automatique de Drush et Let's Encrypt.

A screenshot of a computer

AI-generated content may be incorrect.

**Synchronisation des dossiers** : Utiliser différents mécanismes de synchronisation des dossiers peut avoir un impact significatif sur les performances de votre site Drupal. Par exemple, NFS offre un bon compromis entre performance et facilité d'utilisation, tandis que Rsync offre les meilleures performances à l'intérieur de la VM.

**Optimisation des performances** : utiliser les plugins comme vagrant-cachier pour partager le cache des packages Composer entre toutes les VM.

**Pas à pas pour VirtualBox à partir du fichier .ova**

Dans VirtualBox

🡪 Fichier 🡪Importer un appareil 🡪 "Choisir un fichier" et sélectionnez votre fichier

🡪 Suivant 🡪 configurer selon performance 🡪 Cliquer sur Importateur et patienter

Une fois l'importation terminée, votre VM apparaît dans la liste de VirtualBox.

Le login/mot de passe par défaut est souvent **bitnami/bitnami** (vérifier la documentation)

🡪 voir l'IP locale de la VM 🡪 **ip a**

**Interface Drupal** : **Accès** : Machine hôte : http://[IP\_VM]

Cela devrait vous donner accès à l'interface Drupal.

(Optionnel) Configurer le Réseau

Si vous voulez accéder à Drupal via localhost, modifiez le réseau :

1. Éteindre la VM .
2. aler dans Configuration → Réseau .
3. Mettre "Accès par pont" pour qu'elle ait une IP sur le même réseau que ton PC.

**Réseau actuel**

A screenshot of a computer

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

🡪 Il arrive que le password soit stocké dans ce fichier :

**cat** /opt/bitnami/var/data/mariadb\_root\_password **//** **OU**

**cat** /opt/bitnami/mariadb/logs/mysqld.log | grep password

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Procédure de prise de contrôle** (à adapter et à déveloper)

🡪 Passer en root sans le mot de passe en espérant que l’utilisateur par défaut à la création soit assez admin (sécuriser cela après)

**Sudo** **su** **-**

🡪 Changer le mot de passe

**Passwd** **root**

**🡪 activer la connexion ssh**

**Sur Debian**

sudo rm -f /etc/ssh/sshd\_not\_to\_be\_run

sudo systemctl enable ssh

sudo systemctl start ssh

**Sur Ubuntu**

sudo mv /etc/init/ssh.conf.back /etc/init/ssh.conf

sudo start ssh

**Etablir la connexion ssh**

**Debian**

**nano** ~/.ssh/authorized\_keys

🡪 Penser à configurer le fichier /etc/ssh/sshd\_config

Sujets concernés : PubkeyAuthentication ; PasswordAuthentication ; ports ; etc.

**Ensuite :** vérifier qu’elle configuration est utilisée (serveur, BDD, etc.)

🡪 Se connecter à la gui web admin pour faire tourner les processus responsables

**ps** aux | grep -E "apache|nginx|caddy|lighttpd|php"

🡪 lire la configuration de la BDD

**cat** /opt/bitnami/drupal/sites/default/settings.php | grep 'database'

🡪 se connecter à la BDD

**/opt/bitnami/mariadb/bin/mariadb** -u **root** -p

🡪 lister les utilisateurs Drupal

**USE** bitnami\_drupal;

**SELECT** uid, name, mail FROM users\_field\_data;

**SELECT** \* FROM user\_\_roles WHERE uid = 1;

🡪 Changer le mot de passe de l’utilisateur « user »

* + **Récupérer le hash**

**php** -r 'echo password\_hash("NouveauMotDePasse", PASSWORD\_DEFAULT) . "\n";'

* + **Mettre à jour le passe avec le hash**

**UPDATE** users\_field\_data

**SET** pass = '[HASH\_GÉNÉRÉ]' // renseigner le has généré

**WHERE** name = 'user';

Appliquer les changements effectués

**FLUSH** PRIVILEGES;

**EXIT**;

**Pas à pas pour l’installation sur Docker**

**Ubuntu**

1. **Prérequis**

🡪 vérifier que Docker, Docker Compose et Composer sont installés sur la machine.

Docker Compose sert à orchestrer les conteneurs tandis que Composer est utilisé pour gérer les dépendances PHP et, ici, créer le projet Drupal.

Pour Docker et Docker Compose il vaut mieux reprendre un pas à pas spécifique.

**Composer**

**Installer les dépendances nécessaires** :

**sudo** **apt** update

**sudo** **apt** upgrade

**sudo** **apt** install curl php-cli php-mbstring git unzip

**Télécharger et installer Composer** :

**cd** ~

**curl** -sS https://getcomposer.org/installer -o composer-setup.php

**php** composer-setup.php --install-dir=/usr/local/bin --filename=composer

**Vérifier l'installation** :

**composer** –version

**UFW**

🡪 Configurer le pare-feu et autorisez les ports nécessaires :

**sudo ufw** allow OpenSSH

**sudo ufw** allow http

**sudo ufw** allow https

**sudo ufw** enable

1. **Créer un projet Drupal**

🡪 Utilisez Composer pour créer un projet Drupal (recommandé) :

**composer** create-project drupal/recommended-project my\_drupal

**cd** my\_drupal

1. **Configurer Docker Compose**

🡪 Créez un fichier docker-compose.yml dans le dossier projet avec le contenu suivant :

* + version: '3.7'
  + services:
  + drupal:
  + image: drupal:latest
  + ports:
  + - 8080:80
  + volumes:
  + - ./web:/var/www/html
  + depends\_on:
  + - db
  + db:
  + image: mysql:5.7
  + environment:
  + MYSQL\_ROOT\_PASSWORD: example
  + MYSQL\_DATABASE: drupal
  + MYSQL\_USER: drupal
  + MYSQL\_PASSWORD: drupal
  + volumes:
  + - db\_data:/var/lib/mysql
  + volumes:
  + db\_data:

1. **Démarrer les conteneurs**

🡪 Lancez les conteneurs Docker :

**docker-compose** up -d

1. **Accéder à Drupal**

🡪 navigateur et accédez à http://localhost:8080

🡪 terminer l'installation

**Windows**

1. **Prérequis**

🡪 Installez Docker Desktop pour Windows depuis le site officiel de Docker.

🡪 Installer Composer

**Composer**

Téléchargez et installez depuis le site officiel de Composer **ou** utilisez **PowerShell**

* + cd ~
  + curl -sS https://getcomposer.org/installer -o composer-setup.php
  + php composer-setup.php --install-dir=/usr/local/bin --filename=composer
  + composer –version

1. **Configurer Docker Compose**

🡪 Créez un répertoire pour votre projet Drupal et un fichier docker-compose.yml avec le même contenu que pour Ubuntu.

1. **Créer un projet Drupal**

🡪 Ouvrez PowerShell ou l'invite de commandes et utilisez Composer pour créer un projet Drupal recommandé :

* + composer create-project drupal/recommended-project my\_drupal
  + cd my\_drupal

1. **Démarrer les conteneurs**

🡪 Lancez les conteneurs Docker :

* + docker-compose up -d

1. **Accéder à Drupal**

🡪 Ouvrez votre navigateur et accédez à http://localhost:8080 pour terminer l'installation de Drupal via l'interface web.