

DCLannion 27-29 octobre 2017

Merci aux sponsors et à l'équipe de bénévoles !

Hashtag officiel #dclannion // Concours sur Instagram pour un cadeau surprise. Concours Twitter pour @OWS pour gagner un cadeau !

Ont participé à ce pad (**ajoutez votre nom**) :
qde, Artusamak, Q__nt_n, c4ilus, saisai, Bes, chloecorformat

J1 - Vendredi 27 octobre

11h15 - Faut-il être un développeur pour se lancer en Drupal 8 ? Par webreizh29

Refonte de 11 sites internet sous SPIP3 vers un seul sous Drupal 8
Analyse entre 6 CMS : Wordpress, SPIP, Joomla, EzPublish, Typo3, Drupal
Œil contributeur / webmestre

"J'hésite à me lancer sur Drupal"

* Pour quel type de site ? (Petit site, asso, plaquette entreprise ou gro site). Voir les slides pour identifier les outils recommandés selon les typologies de sites.

Quelle réputation pour Drupal sur la toile : CMS très puissant, modulaire mais complexe. Performant à plus d'un titre car personnalisé. Logique de briques. Apprécié des développeurs car fortement personnalisable.
CMS PHP, Open source

Adapté à plusieurs typologies de sites.

Utilisation d'un outil comme Simplytest.me pour tester très rapidement un site Drupal sans avoir à l'installer (site temporaire pour 24h).

Vous pouvez installer, tester Drupal sans avoir besoin de compétences de développement. Responsive par défaut (compatible téléphones portables).

Les contenus en ligne et la communauté permettent de trouver des informations et de se débloquer sans avoir de connaissance avancées au préalable.

"Apprendre vite et sans galère, possible ?" Oui si l'on se donne le temps de chercher les informations et de s'intéresser.

Flexibilité grâce aux types de contenus configurables via l'interface et des modules complémentaires publiés par la communauté

Il faut oser mettre les mains dans le cambouis

Progresser sans code : jusqu'à une certaine limite. Cela dépend du site à créer. L'expert technique viendra pour vous renforcer et vous aider à produire le site selon vos besoins très personnalisés

En 1 mois, sans ligne de code on est capable de bien maîtriser Drupal

De débutant à expert, des étapes et du chemin tu franchiras.

Profils écosystème Drupal : webmaster (administrateur du contenu), site builder (développeur à la souris : il connaît les capacités offertes par le cœur et les modules et sait configurer tout cela), intégrateur / thémeur : maîtrise le HTML / CSS et connaît le PHP pour extraire les variables

Il est possible de faire un site sans code

Il est possible de s'approprier l'outil en partant de 0

Une communauté d'entraide existe et est volontaire en France
Tout n'est pas possible sans coder, cela peut être un frein mais ne devrait pas vous arrêter
Communication pas toujours orientée pour les débutants

11h15 - De Scald à Media Entity dans le code D8 par @slybud (OWS)

https://www.dropbox.com/s/ga9c7qwkp8eume/OWS_DCLANNION_prez_media_20171027.pdf?dl=0

Historique de la gestion es médias de D6 => D8

Evolutions :

- de plus en plus de medias, son etc
- de + en + dématérialisés ou sur les réseaux sociaux ou dans un DAM

Le coeur n'y répondait pas, avec souvent uniquement le "use case" 1 contenu = 1 champ image pour remonter dans des listes

Asset média ? => Tout ce qui n'est pas texte est un asset média.

- Insertion depuis un champ ou un CKEDITOR
- Possibilité de les partager / importer / réutiliser

Un Média dans Drupal :

- Objet drupal natif / entité
- Organisé en bundle
- Exemple de bundle : images, vidéos, instagram, tweet

D6

- Un simple champ image
- Début de media (complètement refondu après la v1)
- Scald beaucoup trop marginal

D7 : 3 solutions concurrentes : Media (tout asset est un fichier, ou est compatible File Entity) / Scald (challenger de Media, plus porté par la communauté, mais qui a tout de même réussi à créer une communauté autour) / Asset (peu de communautés créée autour)

D8 : nombreuses discussions (DCon Portland 2012) sur la bonne approche "on peut parler de combat" !
Choix en 2013 (DCon Prague) de partir sur une Media Entity, solution unique, différents modules réutilisables

Gros effort de collaboration et de négociation pour le rentrer dans le core (MD Systems, Acquia, OWS, + contributeurs autonomes)

Media Entity : une suite de composants indépendants (mais qui vont bien ensemble)

Statut :

- les gros travaux actuels ont porté sur l'architecture + intégration dans le core
- en version 8.4, à l'occasion de l'intégration du module Media dans le core (ex Media Entity) il y a eu des renommages qui obligent les modules communautaires à s'adapter ; l'ancien Media Entity Document intégré à travers le module File
- roadmap ambitieuse pour les versions 8.5 et 8.6 pour améliorer l'expérience utilisateur et faire en sorte que cela marche "out of the box" pour des cas d'usage simple sans la grosse configuration manuelle actuelle de la suite des modules (<https://www.drupal.org/node/2825215>)
 - dépréciation des types de champ Fichier et Image, tout devient média (<https://www.drupal.org/node/2831940> et <https://www.drupal.org/node/2831941>)
 - intégration Entity Embed dans le core (<https://www.drupal.org/node/2801307>)
 - intégration de Entity Browser (<https://www.drupal.org/node/2834729>)
 - simplification de l'UI de mapping des champs

Bilan : environ 2 ans de retard (on visait D8.0) mais ca y est, l'architecture technique est en place, c'est le bon moment de contribuer pour finaliser

12h15 - Headless Drupal en pratique par @garphy (icilalune)

Pourquoi Drupal pour faire du headless ?

- Modélisation du contenu ultra flexible 80
- Richesse fonctionnelle
- Multilingue
- E-commerce
- Le backoffice pour gérer son contenu etc..
-

L'avantage du headless est de pouvoir utiliser des technos front (React, Angular ..) sur les différents canaux de communication (Web, application etc..)

Nous voulons afficher partout un contenu géré depuis 1 backend.

- contenu géré depuis 1 backend ==> CMS ==> Drupal
- affichage multisupport / responsive ==> Framework Front

L'équation headless "1 site headless = 2 projets" donc :

- prévoir plus de temps surtout au début
- hébergement disparate
- recrutement plus large

Solutions en D8

- REST (core) => **ne pas s'en servir !** se base sur Entity normalizer, sort en XML, JSON, CRUB mais pas de collections, il faut passer par Views :(+ "plein de drupalisms" dans la sortie JSON qu'on impose au dev front, pas mal de bugs + couverture de test perfectible
- JSON API => projet extérieur à Drupal (norme <http://jsonapi.org>), implémentée en Drupal
- GraphQL => "le SQL des API", un langage d'interrogation (spec Facebook), beta depuis cette semaine \o/ => son adoption va augmenter, redonne le contrôle des données au dev front-end, sans avoir besoin de communiquer avec le dev back-end

Zoom sur JSONAPI :

- CRUD, support des collections, pagination, filtrage, gestion des inclusions (entité dans une entité => 1 seule requête)
- accès via URL du type `/jsonapi/node/article`
- => pas d'URL pour des collections mixant 2 entités (travail en cours)
- se reporter aux slides pour des exemples de requêtes avec filtrage etc
- doc de bonne qualité <https://www.drupal.org/docs/8/modules/json-api> (mainteneur fait des vidéos etc)

Distributions (basées sur JSONAPI pour le moment, pas encore GraphQL) :

- Contenta (par le créateur du module JSONAPI)
- Reservoir (Aquia)

Préco : partir "from scratch" sans passer par ces distributions

Les modules JSONAPI et GraphQL sont fait de manière à pouvoir switcher sans trop de difficultés de l'un à l'autre.

Niveau 2 : le routing

Problématique du "deep linking" = le besoin d'avoir des URLs différentes en front afin de ne pas briser les logiques de navigation, bookmarks et partage de contenu sur les réseaux sociaux

Pour contrecarrer ce problème de single URL, utilisation des URL Fragments (obsolète) ou de l'History API

Il y a des solutions de routing côté front dans chaque framework, mais les motifs d'URL sont alors codées dans la partie front, ce qui est dommage puisque le CMS produit déjà des URLs => URL est une propriété du contenu, et qui est gérée côté back.

=> besoin d'un mécanisme de résolution d'URL (dans Angular, il y a UI router), la clef n'est pas l'URL + côté Drupal, on expose une API (par exemple via module Services), exemple dans les slides `/api/path?path=/xxx`
=> retourne l'ID à interroger ensuite + code côté front pour "réveiller le bon contexte" côté front
obsolète

Niveau 3 : les crawlers

Le problème qui se pose alors est celui de la SEO, comment est-ce que le crawling reste cohérent ? Une des solutions est l'utilisation de Prerender, une solution qui va permettre d'intercepter la requete envoyé par le robot d'indexation, et effectuer un prerender du DOM côté serveur un peu plus "crawler friendly." Besoin de distinguer le crawler de l'internaute "normal", l'ancienne solution poussée par Google est dépréciée, il y a une balise meta permettant de distinguer le crawler, sinon retour sur User-Agent. Ce prerendering est ensuite retraité par le framework Front qui effectue le rendu final.

Niveau 4 (pas couvert par cette prez) :

- authentification
- caching
- jsonapi versus graphql
- quel techno JS front ?

12h15 - Paragraphs - la rédaction de contenu sous stéroïdes par @flocondetoile & @opidentica

(Slides : <https://opi.github.io/dclannion-paragraphs/>)

Avantages :

- réutilisation de composants
- composition riche de pages complexes
- atomic design Wahou
- support des révisions / multilinguisme

Apprécié par les dev pour la robustesse et par les webmasters pour la simplicité

Inconvénients :

- Contenu & configuration parfois mixés dans le paragraphe
- Multiplication des champs & des tables dans la bdd
- UI/UX

Bah alors, ça suit pas beaucoup dans la salle du bas ! -> en fait, c'est pour pas reprendre complètement les slides, vu que j'ai mis le lien.

ID : utiliser les entity reference pour permettre à l'utilisateur des teasers sur une page éditoriales

14h50 - Galen Framework ou comment tester ses sites responsive @RaphMo

Notions de base :

Le responsive : rendre la navigation la plus confortable possible quelque soit le support (ordinateur, mobile, tablette, etc). Pour cela on utilise beaucoup les grilles et flexbox

Css : permet de mettre en forme les pages, se base sur le DOM du code HTML

Javascript : permet de dynamiser nos pages, se base aussi sur le DOM

Pourquoi mettre en place des tests automatisés :

- Éviter les régressions
- Faciliter la maintenance dans le temps
- Gagner du temps et donc de l'argent

Galen c'est un framework du monde node basé sur Selenium issu du monde Java

On s'en sert en cli :

- galen check // lance un test
- galen test // lance une suite de test

Définition des objets :

- On peut déclarer différents éléments du DOM via id, css, xpath, etc..
- On peut déclarer des objets multiples ou regrouper des objets

Que peut-on tester ?

- La valeur d'un attribut (largeur, présence, etc)
- La distance entre plusieurs éléments
- la précedence des éléments
- L'inclusion des éléments
- Alignement
- Valeur de texte
- Propriété CSS
- etc
-

Ici une démo

Lors du lancement de galen, on voit en direct le navigateur s'ouvrir et faire les actions demandées par les tests.

Ses conseils :

- faites le dès le début du projet

L'industrialisation des tests est possible à travers Jenkins par exemple.

14h50 - Retour d'expérience sur Migrate en Drupal 8 @simongeorges

<http://simongeorges.github.io/slides/drupalcamp-rex-migrate-drupal8/#/>

Une migration c'est une source, une destination, un process.

Exemples d'usage : Import de flux RSS (Stop waiting for Feeds (lien billet de blog)), import massif de données, migration d'un site à un autre. Cas prévu pour le cœur de Drupal à Drupal. Pour les projets clients, c'est souvent d'une source autre qu'un Drupal (BDD, fichiers, etc) vers Drupal

Extraction des données lorsque l'on migre un Drupal pour s'épargner les migrations de structure. Cela devient une source comme une autre.

Modules importants

Migrate Tools : pour aller plus vite (commandes drush)

Migrate Plus : pour aller plus loin (Complétion du framework : Entité de migration, "SourcePluginExtension" pour définir des CSV comme sources ou un plugin source Url. Plugins complémentaires de traitement des

données : EntityLookup (rattachement d'entités entre elles) / EntityGenerate (Création à la volée de contenus si pas de correspondances trouvées. Par ex les taxos associées à un contenu).

Mais aussi des types de plugins (Data fetch(File, HTTP (si couche d'authentification HTTP ou Oauth)), data parser (simple mxl, xml, json, soap)

Exemples de code : attention ça éclabousse.

Source des données Exemple d'une migration avec source CSV. Format de description des données assez simple. Deuxième exemple avec un import de données RSS. Description des données plus avancée.

Possibilité d'importer plusieurs urls (array).

Destination : rattachement à un type d'entité par exemple (fichier de 3 lignes)

Process simple : pas de transfo, ou correspondance entre source et destination

Si transformation des données : Il est possible de chaîner les transformations (scinde les valeurs selon un délimiteurs, arrête toi qui c'est vide, trouve les correspondance pour les termes associés)

Exemples de plugins utiles

Format de date (transforme la date selon son format)

"Skip on empty" Faux champ pour ajouter le tester d'une valeur vide indiquant d'ignorer la ligne si l'entrée est vide

Table de correspondance (static map) entre chaines

Exemple de code pour injecter des paramètres dynamiquement

Exemple pour lancer une migration depuis du code

Principal problème des migrations : l'état des données récupérées (formats incorrects, inconsistants...).

Analysez et nettoyez vos données avant de les importées.

Loggez vos données autant que possible cela vous aidera à debugger vos migrations

Snippet de code pour recharger le code d'une migration // "drush credit" pour éditer une config directement sans avoir à réinstaller pour les micros ajustements. Sinon commande drush

2 hooks à dispo pour modifier les données de la source

Cas courant "rien ne se passe quand je lance la migration" : réponse : avez vous activé la source "migrate source csv" ? (Remplacer CSV par l'élément)

Je veux importer aussi les éléments qui ont changé, utiliser un flag "track_changes:true"

Mon plugin de process doit se lancer au début de l'import (code pour retrouver ses habitudes D7)

Exemple de modules complémentaires :

S'il n'y a qu'un élément à retenir : LISEZ LE CODE. Les exemples sont de qualité.

Modules à voir aussi : default_content / entity_staging (notamment pour exporter des contenus)

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

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

15h40 - La dette technique d'un site sous Drupal 7 : état des lieux @Laboroug

<https://drive.google.com/file/d/0B7bgXNFblemlbkVqZGZmY19PYVvk/view>

15h40 - Questions/réponses : venez avec vos interrogations, on essaie d'y répondre en direct @Goz @simongeorges @flocondetoile

Questions autour du workflow :

- module Content Moderation (ex Workbench Moderation) intégré dans le core + vues pour que les contributeurs sachent quel contenu valider

- module state_machine (pas de UI) utilisé par Commerce

16h45 - Mais en vrai on test quoi ? @Artusamak - Happyculture

<https://lannion2017.drupalcamp.bzh/programme/sessions/mais-en-vrai-test-quoi>

C'est commencé ? à l'instant

Mince c'était la fin de journée et personne n'a pris de notes.

Bon alors :

- 1) présentation des différents types de test
- 2) zoom sur la partie Behat
 - Notion de Driver et d'API
 - Définition des phrases clef (steps)
 - Mink (bas niveau) + extension Drupal qui ajoute des phrases supplémentaires

Quelques points à retenir :

- certains points délicats (chaînes de texte de connexion/déconnexion à configurer dans le behat.config.yml) confusion possible sur certains appel type création de noeud qui ne se comportent pas pareil
- ne PAS tout tester (trop cher, trop long), partir du principe que le core ou autre sont testés/safe et se focaliser sur les fonctions clef/métier (celles qui nous valent des appels le week end lorsque le pb est découvert car pas évident)

J2 - Samedi 28 octobre

Allez les gens on se motive :)

11h15 - Gérer et versionner son site avec composer et Git by @Kgaut

11h15 - Mon site est hacké ! Et maintenant ? by @fgm

<https://speakerdeck.com/fgm/mon-site-est-hacke-que-faire>

Discussion libre en préambule à la présentation sur les sites piratés, l'importance du bon niveau de communication des failles, exemple du Drupalgeddon qui a fait l'objet d'une communication préalable (annonce de la faille plusieurs jours avant la publication).

Intro : comme un lundi

Un PRI c'est un plan de reprise sur Incident. C'est la procédure que l'on sort le jour où l'incident survient pour ne pas paniquer. On suit à la lettre le plan.

RGPD : Règlement Général sur la protection des données. Posture à adopter le jour où survient une faille de sécu. s'applique en mai 2018 => clients qu'il va falloir notifier (art 33.1)

Lisez <https://www.drupal.org/drupal-security-team/your-drupal-site-got-hacked-now-what> si vous n'avez vraiment aucun plan de préparé.

Premier réflexe : votre avocat

Prendre 2 blocs :

- 1) journal d'observations (horodatées !) (observations / découvertes, étapes du processus)
- 2) idées de corrections (référence svers numéro du bloc 1 idées de fixes, idées de durcissement (? pas eu le temps de lire)

Prendre un cliché

Première tentation : restaurer et reprendre le service, mais le site est toujours vulnérable (voir plus, puisque l'attaquant sait qu'il est vulnérable) !!!!

Attention, analyser implique de modifier => Il faut prendre un cliché !

Comment faire ?

- débrancher ! (certaines attaques nettoient les traces de l'attaque sur le signal d'extinction => forcer l'arrêt)

Débrancher réellement la prise si c'est un serveur physique. Si on a une VM, on peut envoyer le signal d'arrêt (on peut aussi snapshotter afin de rester en ligne pour ne pas divulguer qu'on a vu l'attaque).

De quoi prendre des clichés ? Pas que la DB (reverse proxy, frontaux web, serveurs DB autres / fichiers...

Journaux tiers (SaaS), transactions externes, logs IDS (parefeu)

Le site peut aussi avoir été utilisé pour faire un rebond, vous n'êtes pas forcément la cible de l'attaque par le site ouaibe. Cela peut servir à viser un autre élément du réseau / de vos services.

Maintenir une présence en ligne

L'avantage à laisser le site en ligne : ne pas dévoiler que l'attaque a été remarquée (cela peut inciter les intrus à continuer à agir et donc vous donner plus d'indices).

Inconvénient : dommages accrus, responsabilité légale, financière, morale...

Attaque évoluée :

Pénétrer, chercher quoi exploiter sur la machine après être rentré, implanter un zombie, attendre l'archivage des versions zombie (plusieurs semaines, voire mois), activer et profiter. (Commentaire de prise de note : les vicieux bâtardssssssssss)

Attaque éclair :

Exporter sans tarder, tant que ça dure avec des pertes plus faibles

Attaque masquée (ohé ohé):

Contenu valorisé, récupération des données perso et nettoyage derrière l'attaque pour s'assurer d'être le dernier à passer.

Mode de repli sécurisé préparé en amont, plusieurs possibilités :

- Site statique limité avec un cache de reverse proxy (peut réduire la charge si seul le RP renvoi les pages, ça permet de gagner un peu de temps)
- Site réel limité en ressources sur une infra et/ou technologie alternative

Problème du repli sécurisé : s'il y a du contenu créé pendant cette période, il faudra le reporter sur le site d'origine une fois revenu en ligne

Quand tout a échoué (ou s'il n'y a rien de prévu dans votre stratégie)

Les réseaux sociaux peuvent servir (leur downtime est presque inexistant). Communiquer sur FB / Twitter vous permet de faire passer un message et de gérer la communication de crise

Communiquer pendant la crise

Communiquer entre techniciens : il faut impliquer les parties prenantes, remonter très vite, très haut (jusqu'aux décideurs)

Une peur des représailles peut survenir (client non compréhensif, menace de poursuites...)

Protection en France :

Statut de lanceur d'alerte (sapin 2)

Communiquer vers l'exécutif

Conseil juridique avec les avocats (ce n'est pas à vous de vous préoccuper de cette réponse en tant que techniciens)

Forces de l'ordre (unités cybercrime, ANSSI en FR, joignable par son site <http://www.ssi.gouv.fr>)

Il reste 15 min, Fred va accélérer seules 25% des slides ont été faites.

Vie privée : RGPD : en cas de pénétration, à partir de mai 2018 vous devez communiquer à la CNIL le bilan de l'attaque

Il vaut mieux communiquer sur ce qu'il s'est passé, expliquer pourquoi c'est arrivé et expliquer ce qu'il a été mis en place pour éviter que cela se reproduise.

Reconstruire pas réparer

Quoi faire après la résolution de l'urgence ?

Ne pas restaurer, la faille est connue, plus facile pour un attaquant

Garder et corriger le site ? Cela demande du temps et un effort de revue de code.

Jeter ? possible d'en profiter pour changer le site ou la techno pour se débarrasser de l'historique, passer à un simple site statique...

Parfois, on ne peut pas revenir à un état sain (cas de l'attaque dormante pendant une longue durée), stratégie de sauvegarde à définir. Pérénnité des backups sur un temps long. Définir ce qu'il est acceptable de perdre

Parfois, il faut reprendre une version antérieure sans pouvoir facilement identifier ce qui est mauvais dans le diff avec la version en prod attaquée => se poser la question de ce qu'il est acceptable de perdre

La preprod peut parfois être une source à partir de laquelle partir

En refaisant tout il y a un risque d'oubli de config, mais c'est très chronophage et coûteux

Enquête * musique de the who *

(5 last minutes \o/)

Comment on a su ?

Qu'est-ce qui a rendu cela possible (cause) ? bulletin de sécurité des composants utilisés, OWASP, chercher sur Google ce qu'on voit affiché ;)

Cherche à revenir à la cause du problème, (5 pourquoi).

Improbable n'est pas impossible. La plupart du temps, le plus évident est la bonne explication mais il arrive que les attaquants soient vraiment vicieux

Si par exemple sur un blog hacké vous identifiez des snippets, un peu de Google fu peut vous apporter une explication / solution rapidement (il est probable qu'elle ait été utilisée ailleurs)

****tout**** peut avoir été nettoyé ... mais c'est rarement le cas ! L'enquête peut faire perdre des infos (historique bash...) => travailler sur des clichés

Repartir d'une source vierge après avoir exploré une piste permet de ne pas altérer les indices & traces laissées par les attaquants

Basculer d'Apache à Nginx ouvre une porte à des failles car Drupal ne supporte pas officiellement Nginx, les règles de protections des htaccess seront donc à réécrire à la main à la sauce Nginx

L'horodatage des fichiers modifiés après la date du dernier déploiement peut vous aider à trouver les fichiers modifiés.

Outils utiles : Hacked! (autres en D7)

CA VA TROP VITEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE, LA PORTE, VITEEEEE, LA PORTE !!!!!

Pour FGM, conserver le stockage persistant des sessions (ne pas les externaliser dans Redis Memache comme c'est souvent fait pour des raisons de perf) car on perd l'horodatage qui permet de donner des indices

Les logs sont une source inestimable d'infos. On ne peut pas supprimer les traces s'ils sont en écriture seule hors site

Le RGPD demande :

- qu'on ait un registre de l'ensemble des traitements, qui va devoir indiquer entre autres où sont les logs / journaux + backups
- qu'on produise le rapport d'intrusion

Cracking Drupal, seul livre qui parle sécu Drupal même s'il date un peu.

MOOC Ansi gratuit à visionner.

Reprendre les 2 journaux mentionnés en début de présentation pour faire un bilan du temps passé sur les différentes étapes, et voir ce qui peut être amélioré.

LEs horodatages de vos actions vous permet de mesurer combien de temps vous mettez à faire vos opérations de réparation et comparer d'une fois sur l'autre votre temps de réaction.

Il faut s'entraîner et répéter ça et surtout cela va devoir devenir obligatoire.

12h15 - Présentation du module Entity Share by @greamreaper & @vmaucorps

12h15 - Drupal 8 sans modules, si si c'est possible by @pounard

<https://github.com/makinacorpus/lannion-drupal-sans-modules>

Pounard ne connaît pas ses raccourcis clavier :)

L'ami des poneys et des chatons n'aime pas la complexité et cherche à s'affranchir de sa définition de la complexité.

Pourquoi penser modulaire pour un projet client sachant qu'il ne demande jamais à activer / désactiver un

module

Objectif mapper l'organisation du repo Drupal à la sauce Symfony

Création d'un dossier app/ où se retrouvent les fichiers de config de YML (export de config, services, paramètres)

"Symfonismes" : Déplacement des caches dans app/cache, dossier de config dans app/config

Globalement illustration de pas mal de fonctions Symfony permettant de remapper la config, les caches, le chargement des annotations (Plugins) dans un emplacement unique à la racine du projet plutôt que dans X modules selon le layout standard d'un projet Drupal.

14h15 - ZURB Foundation : un starter kit de qualité by @FlorisM

ZURB Foundation

- Pour les emails ;
- Pour les sites.

Pour avoir une idée de ce qui est proposé par le framework : <https://foundation.zurb.com/sites/docs/kitchen-sink.html>

(Système de grille : Pas de conteneur : on démarre directement sur des colonnes.)

Dans Drupal, il existe un thème contrib zurb_foundation (beaucoup de mouvement sur le projet actuellement).

Il faut utiliser le dossier STARTER pour débiter et créer un thème enfant.

Il existe une commande Drush pour créer un thème enfant : drush fst

On délaisse souvent le front, alors qu'il y a des bonnes pratiques.

Foundation = Good HTML

- markup particulier
- balisage sémantique
- attributs "aria-", "role", "data-"

SMACSS pour découper son scss (nomenclature à la fois des fichiers & des sélecteurs) : cela permet de travailler à plusieurs sur un projet sans conflit.

Il existe un linter scss.

DEMO

Pourquoi SASS et pas LESS ? A priori plus de mixins dans SASS.

Dans la salle : Les grilles de Bootstrap 4 vachement mieux que Bootstrap 3.

Les versions 5 & 6 de Foundation existent pour Drupal, mais la version 6 est beaucoup mieux.

Moins d'options que dans Bootstrap, ZURB est moins paramétrable mais du coup pas de souci de "ça j'en

veux pas"

Webpack c'est cool.

Courage Floris !!!

14h15 - Gérer son site Drupal avec Ansible @SebCorbin

<https://cdn.rawgit.com/makinacorporus/makina-slides/master/docs/ansible-drupal.html>

Ansible sert à faire :

- du déploiement multi-node
- exec de tâche
- gestion de conf

Utilise Python ce qui permet de l'installer partout et utilise ssh pour communiquer avec les serveurs

Déploiement / orchestration

On administre un serveur à la main et puis le temps faisant patatra.

On oublie, on se rate, on se mélange, on est plus lent qu'un script.

Il existe plusieurs orchestrateur, puppet, chef, salt, et donc ansible.

Ansible basé sur YAML (Language), Jinja2 (Templating) similaire à twig

Contrairement a ses "concurrents" Ansible n'a pas besoin que l'on installe un agent sur les serveurs pour qu'il fonctionne.

Un peu de vocabulaire:

- Les directives sont des "Tasks"
- Les Scripts sont des "Playbooks"
- Un "Master" contrôle des "Slaves"

Un playbooks est composé de tâches ou de plays. Une tâche c'est un nom machine, des commandes, des variables, etc

Les tâches sont toujours exécutées dans l'ordre les unes après les autres. 4 status possibles: ok, changed, fail et skipped. Un fail arrête l'exécution d'un playbook.

Une tâche spécifique qui doit être exécuté précisément après une autre est un "Handler".

15h15 - Cache sous Drupal 8 @Eiffel_Alec

Améliorer les performances en stockant le résultat de processus coûteux pour ne pas les recalculer.

Activé par défaut dans Drupal 8.

Premier cache : navigateur

2nd : Proxy (au niveau du réseau), CDN (content delivery networks)
Plus bas niveau : Varnish. Choix classique avec Drupal (peut être piloté par Drupal)

Cache anonyme de page:

- module "Internal page Cache"
- cache intégral du rendu html d'une page
- la clé est l'url complète
- utilise les cache tags

Cache dynamique de page (nouveau Drupal 8)

- plus fin que le cache anonyme
- agrège le cache des éléments d'une page

Auto-placeholder

- utiliser les render array très changeantes sans invalider le cache parent

Les éléments qui composent le cache dynamique sont les render array de Drupal.

Les métadonnées de cache

- Cache context => avoir des variantes de cache en fonction du contexte
- Cache tags => invalider des types de cache
- Max age => invalidation en fonction d'un temps en secondes (par défaut : -1 = permanent). Autre option : 0 => toujours désactivé.

Pour les utilisateurs, site-builders => le cache est géré par Drupal !

Pour les devs ... HEAVY BREATH

- cache static dans la durée de vie d'une page : `drupal_static()` dans une méthode, ou si OO les "services" sont singletons => attributs persistents
- `Drupal::cache()->set / ->get`
- `->delete` pour l'invalidation (déconseillé)
- Enregistrement => stock de la somme (checksum) des compteurs d'invalidation des caches concernés
- Appel du cache => recalcul cette somme.
- Possibilité d'utiliser un cache périmé (le temps qu'il soit reconstruit) `Drupal::lock()->acquire`

4 types de conteneurs de cache (bins) : default / data / discovery / Null. Possibilité d'en créer de nouveaux.

15h15 - La configuration en Drupal 8 : plusieurs approches @grimreaper

Avec la configuration en Drupal 8 sont apparues les entités de configuration (en plus des entités de contenu). Toute la configuration passe dans Drupal 8 par des fichiers Yaml.

Il est possible de surcharger la configuration dans le settings.php.

Chaque module peut livrer de la config optionnelle ou obligatoire (module, thème, profil d'installation)

- install pour activer de la configuration au moment de l'installation
- optional : prérequis de fonctionnement qui ne sera pas activé par défaut
- schema : pour permettre à l'API de mieux fonctionner

On oublie souvent de le rappeler mais la traduction de la config est à regrouper par sous-dossiers de langue (ndr : ahah retour à 1987 où on livrait de la trad dans le code :D)

Si vous faites un export complet de la config et tentez de faire son import directement, vous allez vous faire jeter à cause des UUIDs générés

Il est possible de surcharger de la configuration à la volée depuis le fichier settings.php.

Il existe dans le fichier 2 variables, \$settings et \$config. Le premier étant là pour modifier des paramètres non administrables, le second étant là pour surcharger les éléments de config exportable et administrable par l'UI. Faire de la surcharge à cet endroit ne se répercute pas dans la base de données. Dans les écrans d'admin, vous n'avez donc pas de visibilité sur le fait que la variable puisse être surchargée.

Feature un outil de packaging pour les fonctionnalités à réutiliser d'un projet à un autre et non un outil pour déployer la config de nos sites d'un environnement à un autre.

Feature ne permet toujours pas la suppression de la configuration (scripting nécessaire).

Drush language https://www.drupal.org/project/drush_language // Module pour activer / désactiver des langues et importer des .po custom

L'autodetection des composants d'une Feature est pratique mais assez aléatoire

Config split fonctionne avec Config Filter et permet de séparer les exports de config par groupes (souvent des environnements). On peut ainsi exporter ou non certains fichiers de config dans certains environnements, activer ou non certains modules. Grâce à cela on peut par exemple avoir tous nos modules d'UI désactivés par défaut et sur un environnement de dev, on peut activer les modules d'UI ou de dblog par exemple.

Grâce à la partie graylist de Config Split le client peut modifier certaines config sans tout lui interdire ou tout lui autoriser.

Config installer (profil personnalisé) est nécessaire pour pouvoir faire une installation depuis une config (pas possible nativement pour le moment)

Config Ignore permet d'ignorer complètement certaines configuration.

Cas d'usages :

Usine à site : utiliser features (un CMI pour 30 sites peut être compliqué à maintenir).

Gestion de configurations différentes par environnement (Features + config split)

Bug features, pas de résolution des dépendances, activation des modules par ordre alphabétique... plutôt aléatoire et problématique.

La configuration doit être pensée dès le début du projet et doit être propre à ce projet (il est difficile de garder le même type de configuration pour plusieurs projets).

Il faut aussi définir les droits de modifications de la configuration par le client.

Config (idem aux environnements / ne change pas) VS Setting (propre aux environnements / ne change pas)
VS State (propre aux environnements / change)

hook_update_n() VS hook_postupdate_n() pour faire des opérations de changement garantissant que la

structure a bien été mise à jour.

16h50 - BOF : RGPD

<http://i.haza.fr/di/ZYQ3/bof-rgpd.png>

<https://www.drupal.org/node/2848974>

Loi : moyen, pas de précision sur le moyen. Obligation c'est fournir un moyen raisonnablement proportionné.

Peut être une adresse mail : envoyer moi mes données.

Peut être, page de listing des données, page de suppression des données, page de récupération des données.

Vraisemblablement ce qui sera utilisé par les gros sites.

Attention au logging,

Date de 2014, entre en vigueur en mai 2018, 90 pages, versionnée annotée de 120 pages en FR

article 94 règlement, du juridique pénible à lire.

Ancienne directive a priori assez proche, mais SANS les sanctions !

Lutte US vs Europe : sur les GAFAM, vous surtaxez nos banques, on va surtaxer vos techs.

Entreprises US se préparent depuis un moment pour cette échéances, d'où les MAJ de conditions d'utilisations récentes etc.

Question ouverte : est-ce que ce type de sanction ne va pas rester uniquement théorique ?

Pas possible de se faire condamner 2 fois sur la même infraction, certaines entreprises cherchent à se faire condamner tant que ça ne coûte pas trop cher.

Sanction max : 20 Md€ ou 4% du CA (lourd !)

Tout exploitant de données avec traitement de données automatisées. Besoin de mise en place des registres de traitement automatisés (comment les infos sont créés, gérées, détruites). S'applique potentiellement un peu partout en entreprise (téléphonie, annuaire LDAP, outils informatiques mis à disposition de nos clients).

Axe rassurant pour les pro : s'appuyer sur Article 40 de la CNIL : notion de guide des bonnes pratiques pour commencer à mettre cela en place

<https://www.cnil.fr/fr/principes-cles/reglement-europeen-se-preparer-en-6-etapes>

CNIL mais pas l'appelé CNIL, il y a un autre nom. ???

1ère obligation : identifier TOUS les traitements (outils mis en place, il y a une paye, une compta ; authentification sur les outils) => registre des traitements

Registre facultatif si sous 200 personnes dans l'entité qui gère le registre. Data Protection Officer (DPO), par défaut responsable de l'entreprise si pas nommé.

Être en obligation de prouver qu'on est en conformité.

Suppression de la déclaration CNIL sur les données simplifiée, remplacer par la notification à l'autorité pour les traitements sensibles.

Minimiser la collecte

La politique a changé pour les déclarations CNIL : De pas excessif "ne prenez pas trop de données sur les utilisateurs" à strictement nécessaire "prenez selon la plus petite donnée nécessaire pour faire fonctionner votre service. Si vous voulez plus, vous devez nous justifier pourquoi.

PIA preliminary impact assessment : à chaque traitement de collecte de données sensibles (santé par exemple),

Considéré comme sensible, tout croisement de fichiers, la taille des données, le croisement de données.

- Schemas and Types

Traçabilité du consentement doit être conservée.

Opt-in doit être actif : les cases à cocher pour donner des consentements doivent être décochées par défaut.

Protéger la donnée

Limiter même au sein de l'entreprise qui exploite la donnée Définir par segment qui a accès à quoi.

Prendre soin qu'un minimum de personnes aient accès aux données et les données soient chiffrées.

Une issue est ouverte pour dire "de quoi doit on s'occuper dans le coeur pour pouvoir continuer d'exploiter Drupal en Europe à l'entrée en vigueur du RGPD"

Potentiellement, il faut encrypter les données des fields.

Garantir système de récupération de la données en cas de perte => mise en place de backups.

Ce n'est pas une directive, c'est un règlement => ça se fait appliquer directement, ça n'a pas à être transposé dans chaque pays.

Garantir la portabilité des données

Notification des intrusions

l'état de piwik : <https://piwik.org/blog/2017/09/gdpr-potential-consequences-piwik/>

Ateliers

Drupal et GraphQL

Le problème de base (URL classique, trop complexe et peu lisible, avec JSONAPI) :

https://live-contentacms.pantheonsite.io/api/articles?format=api_json&page%5Blimit%5D=3&filter%5BisPromoted%5D%5Bpath%5D=isPromoted&filter%5BisPromoted%5D%5Bvalue%5D=1&filter%5BisPublished%5D%5Bpath%5D=isPublished&filter%5BisPublished%5D%5Bvalue%5D=1&include=contentType%2Cimage%2Cimage.thumbnail&fields%5Brecipies%5D=title%2Cdifficulty%2Cimage&fields%5Bimages%5D=name%2Cthumbnail&fields%5Bfiles%5D=filename%2Curl&fields%5BcontentType%5D=type&sort=-created

Commande :

```
composer require drupal/graphql # télécharge automatiquement la bibliothèque PHP associée, merci  
composer  
drush en graphql
```

2 types de requêtes :

- Query pour lire
- Mutation pour créer

La documentation est générée automatiquement (typage des requêtes).

Possible de mixer les types de node demandés (10 derniers nodes quelque soit leur type), pas possible avec JSONAPI

Accéder à l'interface d'interrogation sur <http://drupal8.local/graphql/explorer> ; noter le bouton de doc en haut à droite

Une première requête :

```
{  
  nodeQuery{  
    entities {  
      entityLabel  
    }  
  }  
}
```

Avec des filtres :

```
{  
  nodeQuery(limit: 5, filter: {status: true, type: "blog_item"}) {  
    count,  
    entities {  
      entityLabel,  
      entityUuid,  
      entityType,  
      entityBundle,  
    }  
  }  
}
```

```
}
```

Limitation actuelle : si on doit filtrer de manière plus complexe (par une catégorie), il faut créer une vue de type GraphQL, dont les filtres exposés deviendront des filtres de la requête GraphQL

En faisant remonter des entités liées :

```
{
  nodeQuery(limit: 5, filter: {status: true, type: "article"}) {
    count
    entities {
      entityLabel
      entityUuid
      entityType
      entityBundle,
      ... on NodeArticle {
        fieldTags{
          entity{
            entityLabel
          }
        }
      }
    }
    entityOwner{
      entityLabel
    }
  }
}
```

Possibilité de réécrire les noms des champs (alias) avec :

```
title: entityLabel
```

=> on a title dans la réponse

Il y a un display mode GraphQL, il est donc possible d'exploiter les field formatters pour les réponses (exemple : appliquer un certain formatage sur des champs ou des image styles sur les images)

Ouverture sur la partie cliente (front) :

- en React, il y a Apollo qui permet de mapper un composant directement sur la requête sans passer par un codage de requête HTTP
- sinon possible de faire une requête classique, mais penser à faire un POST même en lecture en raison de la taille des paramètres

Noter l'effet "wow" sur <http://drupal8.local/graphql/voyager>

Au final :

plus puissant que JSONAPI car spec mieux faite, écosystème se forme pour profiter de cette puissance réduction de la taille des réponses possibles en spécifiant les champs à remonter

Dépôt "drupal8 graphql server" :

<https://github.com/nyl-auster/drupal8-graphql-server>

Lien utile :

<https://github.com/drupal-graphql/> : Certains modules, tels que GraphQL Views, de la bêta (disponible uniquement sur Github ne fonctionnent pas correctement)

Atelier Role de l'adhérent et Avantage de l'adhésion

Phillipe : Qu'est-ce que l'association Drupal France a comme spécificités
Est ce que je vais motiver les gens à participer, à adhérer. Qu'est ce que cela apporte.
Est ce que l'on cherche des membres et pour quoi faire ?

Leon: l'asso a des missions définies Faire la promotion de Drupal, la traduction, représenter la communauté auprès des, aider les communautés Francophone (voir le site) Ces missions peuvent évoluer dans le temps
Comment imagine tu l'asso, quelles devraient être ses missions ?

Phillipe : dans les besoins : la doc pratique, vulgariser Drupal, décrire commencer

Fabien :

Thierry :

Concernant la documentation : des exemples plus parlants se trouvent sur le planète.

Les avantages de l'adhésion :

- Sentiment d'appartenance à la communauté
- Être visible par la communauté dans l'annuaire
- Passer des offres et demandes d'emploi
- Voir des offres et demandes d'emploi
- Réduction sur les événements de l'asso ou d'autres événements de la communauté
- Ajouter un badge adhérent individuel/société, actif, sympathisant, supporter, ...
- Autosatisfaction de contribuer à la communauté
- Avoir un droit de vote
- Avoir accès au comptes rendus des actions et AG
- Être consulté ou au courant des évolutions et questionnements stratégiques
-

Actuellement aucune condition n'est demandée pour être membre sur la plateforme de l'association

Le site s'adresse à tous les utilisateurs potentiels sans tri. Peut être des pages : Je suis une institution, je débute, je suis une agence, ...

Annuaire

contient n'importe quoi actuellement (sos serrurier, sos plomberie ...)

Régions à actualiser

Ne mettre que les adhérents dans l'annuaire

Changer le titre de l'annuaire car il y a d'autres que des prestataires

Il peut y avoir plusieurs entrées de l'annuaire partenaire, sponsor, agence, pro, utilisateur,
On ne va pas exporter l'annuaire sur le nouveau site,
Les entreprises adhérentes sont dans l'annuaire,
L'accès à l'annuaire sera réservé aux adhérents

Il a été discuté la possibilité de diverses catégories de membres et/ou :

- Actif (votants) quelqu'un qui a envie d'être investi et bénévole,
- Sympathisant
- Utilisateur simple (non votants),
- Partenaires

Ateliers / Sprints

J'ai participé aux ateliers / sprints :

<nom> | < sujet >

Sprint infrastructure :

- statification des anciens sites de camp

Florent Torregrosa
Guillaume Bec

Sprint traduction

- - seb : debug de l10n_client
- - osinet : création de l'extension chrome sur le store
- - artusamak : Traduction du core