



Partagez vos contenus avec Entity Share

Présentation au Drupalcamp Lannion par @Florent_Torre & @vmaucorps

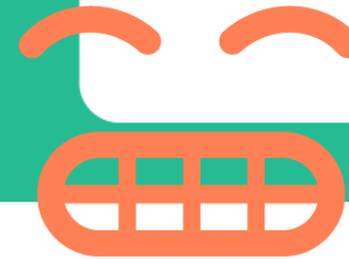




1. Pourquoi le partage de contenu ?
 - Un composant majeur d'une usine à sites
 - Les 2 cas d'usage du partage de contenu
2. Pourquoi le module Entity Share ?
 - Différences avec le module Webfactory
 - Différences avec l'écosystème Deploy
3. Architecture du module
 - Utilisation de la JSON API
 - Architecture
 - Problèmes connus
 - Roadmap
4. Démonstration

POURQUOI LE PARTAGE DE CONTENU ?

MISE EN CONTEXTE



L'USINE À SITES;

LE GRAAL DES DSI



Contexte...

- La multiplication des sites coûte cher aux entreprises
- Les entreprises ont des besoins hétérogènes (Editorial, Intranet, RSE, E-commerce...)

... et enjeux

- Comment rationaliser les coûts?
- Comment capitaliser les fonctionnalités?
- Comment industrialiser les déploiements? La maintenance?



PARTAGER DU CONTENU:

UN COMPOSANT
MAJEUR D'UNE
USINE À SITES

1



FACTORISATION

- DU CODE
- DES FONCTIONNALITÉS



2

DÉPLOYER RAPIDEMENT

- DES CLONES
- DES SITES
ÉVÉNEMENTIELS
- UNE COMMUNICATION
DE CRISE



3

PARTAGER DU CONTENU

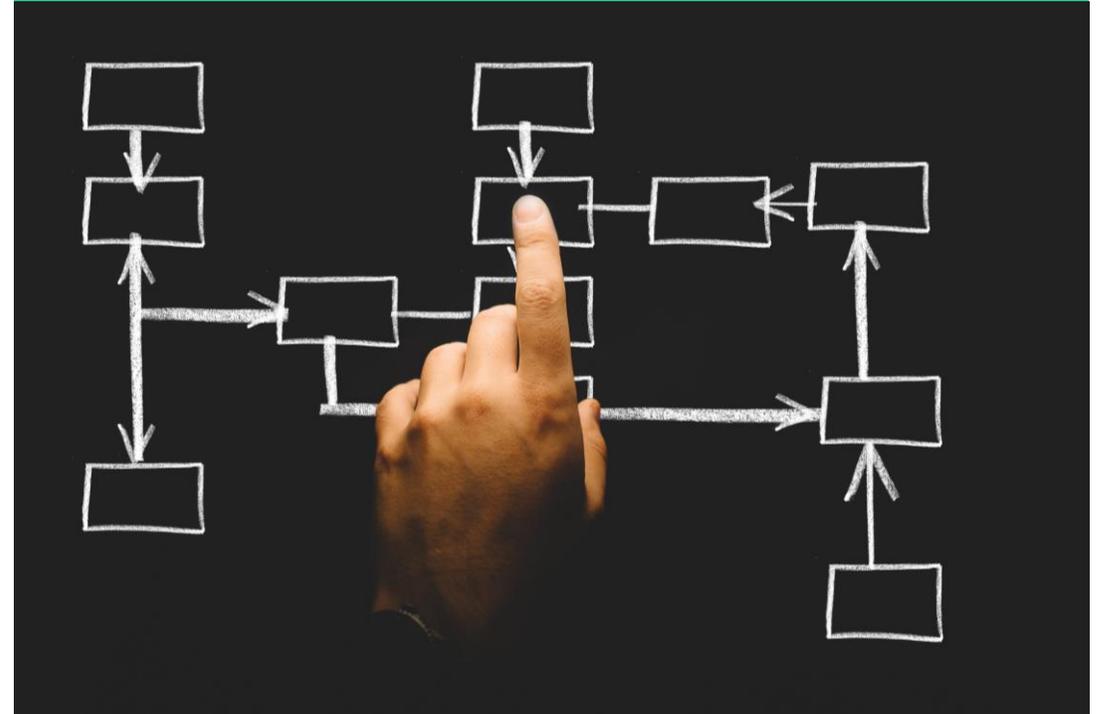


LES 2 CAS D'USAGE DU PARTAGE DE CONTENU

DÉPLOYER DU CONTENU SUR PLUSIEURS SITES



DANS LES PROCESS D'INDUSTRIALISATION

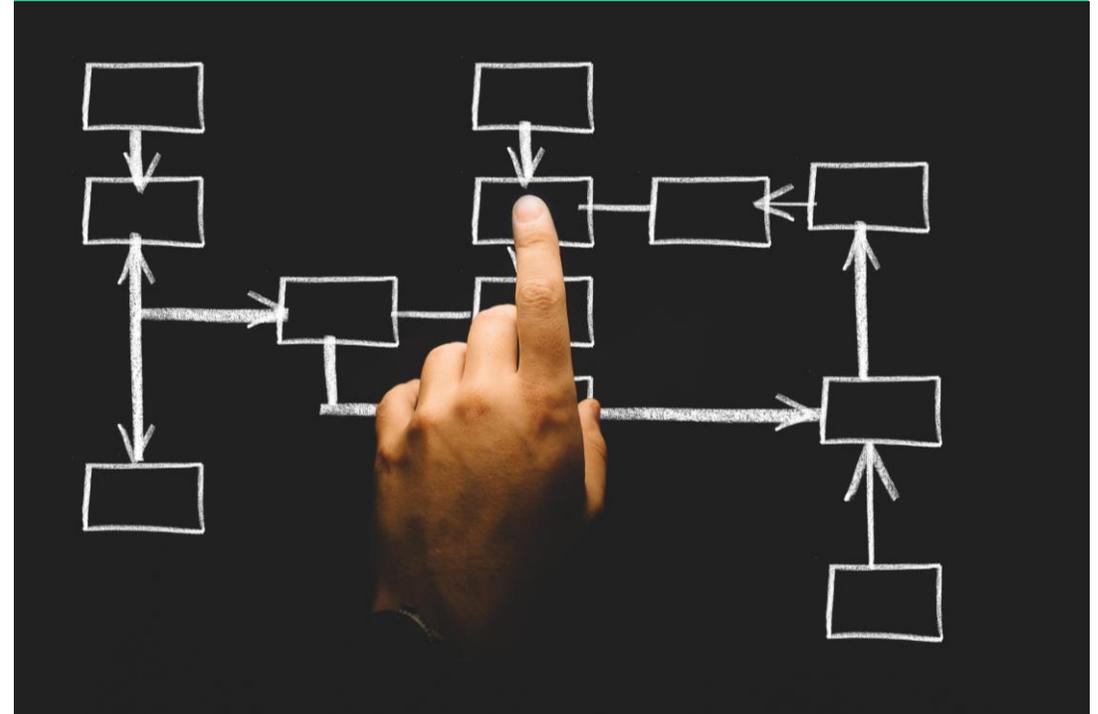


LES 2 CAS D'USAGE DU PARTAGE DE CONTENU

DÉPLOYER DU CONTENU SUR PLUSIEURS SITES

- Soit architecture avec hub de contenu
- Soit communications croisées entre différents sites

DANS LES PROCESS D'INDUSTRIALISATION



LES 2 CAS D'USAGE DU PARTAGE DE CONTENU

DÉPLOYER DU CONTENU SUR PLUSIEURS SITES



DANS LES PROCESS D'INDUSTRIALISATION

- Déployer du contenu d'une préprod vers la prod (staging de contenu), revient au cas du hub de contenu
- Récupérer du contenu de production vers les environnements de développement

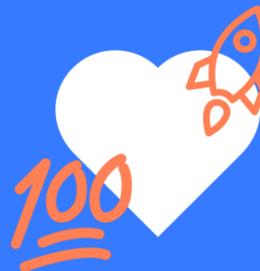
POURQUOI ENTITY SHARE ?

EXPLORATIONS DES
ALTERNATIVES



Drupal™

FORCES



Modules **communautaires**

- Webfactory
- Deploy

Acquia propose une solution clé en main

- Création de sites
- Clonage de sites
- Partager des utilisateurs
- Partage de contenu

EN BREF



- Créé en 2001
- PHP
- 100% développé par la communauté (3000)

Très large écosystème en France, composés d'acteurs de tous types et tailles

CHALLENGES



Le produit **Acquia Content Hub** est contraignant

- Stockage des contenus hors Drupal
- Utilisation contre intuitive
- Investissement financier non négligeable

LIMITES DU MODULE WEBFACTORY

Permet la gestion multisite depuis un central

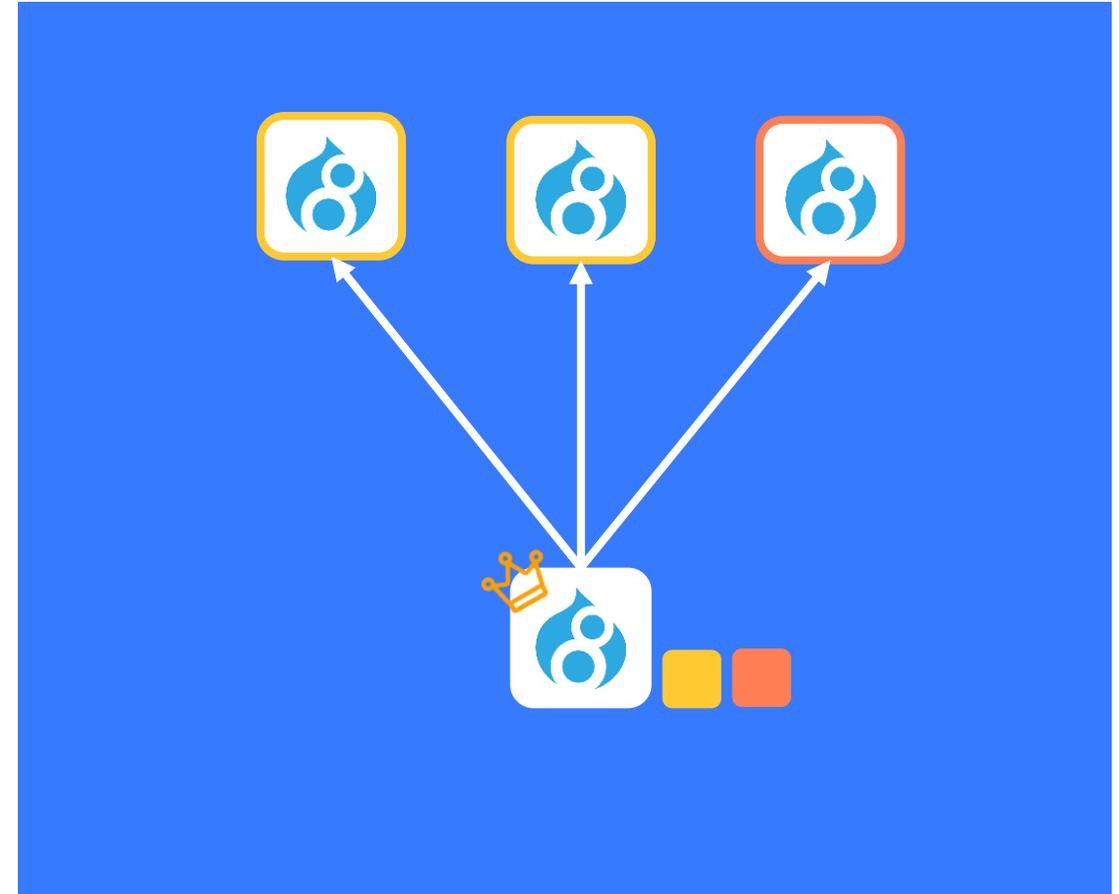
Facilite le déploiement de nouveaux sites

- Déployer un nouveau « profil » directement depuis le backoffice

Le partage de contenu depuis le central

- Partage des entités : node
- Un mécanisme de « channel » permet de ne partager que certaines entités à certains sites

Utilisation des webservices natifs



L'ÉCOSYSTÈME DEPLOY



Poussé par la communauté

Assez instable

- En tout cas en début d'année

Contraignant niveau workflow

Plus pour du staging de contenu

ARCHITECTURE DU MODULE

Et comment nous aider !



UTILISATION DE LA JSON API



- Pour fournir des listes d'entités
- Plus stable et documenté que GraphQL
- Initiative pour l'intégrer au noyau
- Simple d'utilisation

UTILISATION DE LA JSON API

ENTITY SHARE
FOURNIT UNE UI
AUTOUR DE LA JSON
API

```
// Format JSON as in
// JsonApiDocumentTopLevelNormalizerTest::testDenormalize().
$prepared_json = [
    'type' => $data['type'],
    'data' => [
        'attributes' => $data['attributes'],
    ],
];
$parsed_type = explode('--', $data['type']);

return $this->jsonapiSerializer->denormalize($prepared_json,
    JsonApiDocumentTopLevelNormalizer::class, 'api_json', [
        'resource_type' => $this->resourceTypeRepository->get(
            $parsed_type[0],
            $parsed_type[1]
        ),
    ],
);
```



DÉCOUPAGE EN 2 MODULES



LES 2 MODULES PEUVENT ÊTRE ACTIVÉS SUR UN MÊME SITE.

Entity share server

- À activer sur le site qui va fournir le contenu
- Fourni le système de channel
 - Préparer l'URL d'un endpoint JSON API à appeler par le site client : type d'entité, bundle, langue, filtres, tris
 - Plus listing des channels selon l'utilisateur autorisé

Entity share client

- A activer sur le site qui va puller (ou pusher) du contenu
- Permet de renseigner les sites sur lesquels se connecter
- Fourni le formulaire de pull (et push)



LORS D'UN PULL

POUR CHAQUE ENTITÉ SÉLECTIONNÉE

On regarde si une entité du même UUID existe

- Si aucune entité existe, on en crée une nouvelle
- Si une entité existe, on crée ou met à jour la traduction de l'entité par rapport à la langue indiquée dans les données récupérées. Et on stocke son ID dans la liste des entités traitées.

On stocke l'ID de l'entité comme ayant été traité

On s'occupe des champs entités références

- Pour chaque champ « relationship » (JSON API), on va requêter le endpoint représentant la liste des entités référencées par le champ
 - Pour chacune de ces entités on refait le même process initial
 - On mets les Ids des entités traités dans la valeur du champ entité référence

On s'occupe des fichiers physiques

- Si l'entité est un fichier, on utilise ses propriétés pour aller récupérer son contenu



PROBLÈMES CONNUS



Besoin d'un patch pour la hiérarchie de taxonomie (Core)

Solution de contournement pour les champs de type lien avec des liens internes (Core).
Utilisation de JSON API Extras + patch

Le tris multiple sur les channels a des effets de bords (JSON API)

Besoin d'un patch pour les groupes de filtres de champs imbriqués (JSON API)

Options des champs liens non présentes dans le JSON (Core)

ROADMAP

Tests automatisés !!!

Formulaire de push

- Pas possible de PATCHer pour les traductions (JSON API)
- Pas possible de PATCHer pour un champ de type image par exemple (JSON API)
- Pas possible de PATCHer les entités fichiers (JSON API file)

Améliorations du formulaire des channels

- Avoir un système d'arborescence avec drag&drop pour les groupes de filtres
- Lors du paramétrage d'un filtre avoir un outil de sélection du champ comme possède Search API



**TOUS LES DÉTAILS SUR LA PAGE DU MODULE
ET DANS L'ISSUE QUEUE**



DÉMONSTRATION

ET LÀ JE METS MA TÊTE DANS LA
GUEULE DU LION !





SMILE

The word "SMILE" is rendered in a bold, white, sans-serif font. Each letter is intricately decorated with blue line-art icons. The 'S' features a rocket ship. The 'M' is transformed into a smiling face with closed eyes and a wide mouth. The 'I' is a simple vertical bar with a small dot above it. The 'L' is a vertical bar with a jagged, lightning-bolt-like shape on its left side. The 'E' is a vertical bar with a speech bubble on its right side. The word is surrounded by several small blue stars and dots, creating a festive and celebratory atmosphere.

I.T IS OPEN