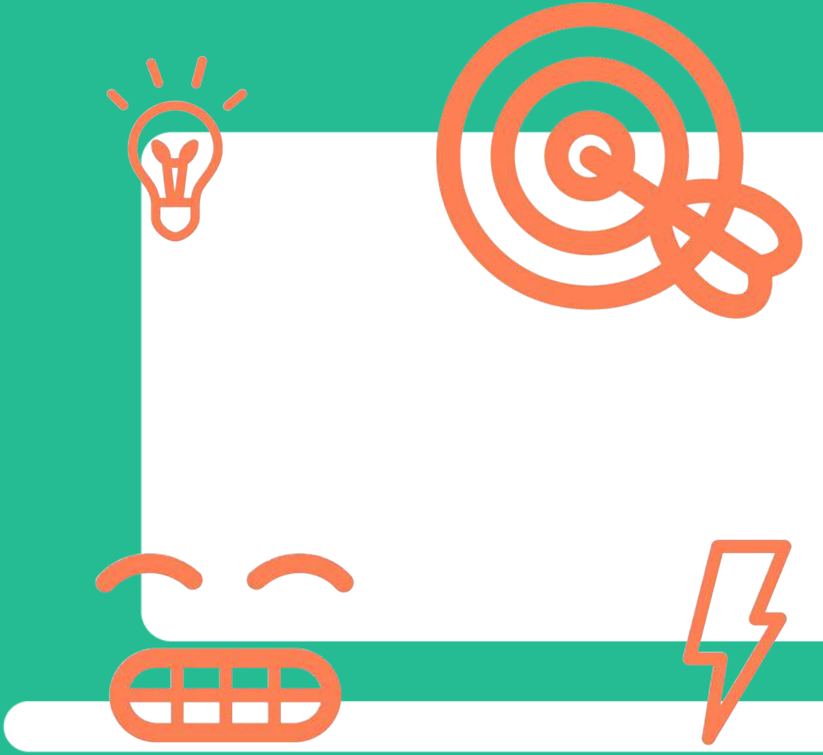
Deploy your content with Entity Share

BoF Solutions for content deployment by @Florent_Torre
DrupalCon Amsterdam 2019

WHY SHARING CONTENT ?

IN CONTEXT



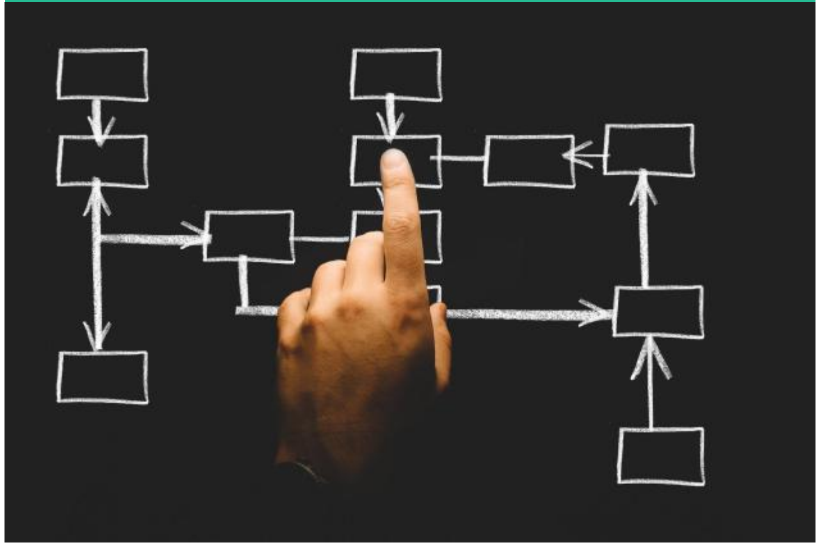
SMILE

I.T IS OPEN

THE 2 MAIN USE CASES OF SHARING CONTENT

DEPLOY CONTENT ON MULTIPLE WEBSITES

IN INDUSTRIALIZATION PROCESS

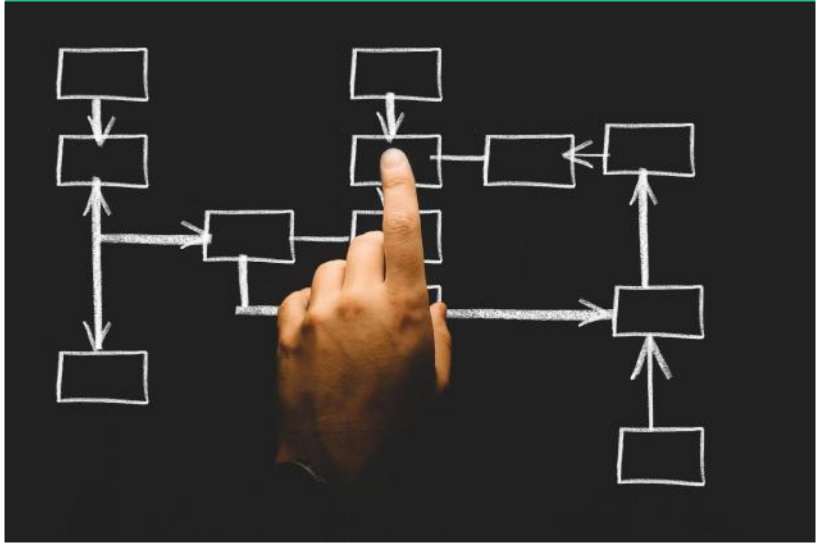


THE 2 MAIN USE CASES OF SHARING CONTENT

DEPLOY CONTENT ON MULTIPLE WEBSITES

- Either architecture with content hub
- Or cross-communications between different sites

IN INDUSTRIALIZATION PROCESS



THE 2 MAIN USE CASES OF SHARING CONTENT

DEPLOY CONTENT ON MULTIPLE WEBSITES

IN INDUSTRIALIZATION PROCESS

- Deploy content from preproduction to production (content staging), same as the content hub case
- Retrieve content from production to development environments

WHY ENTITY SHARE?

ALTERNATIVES
DISCOVERY

I.T IS OPEN

Drupal™

FORCES

Contrib modules

- Webfactory
- Deploy

Acquia has a turnkey solution for

- Websites creation
- Websites cloning
- Sharing users
- Sharing content

IN SHORT

- Created in 2001
- PHP
- 100% developed by the community (3000)

Very big ecosystem, composed from a wide range of actors in size and type

CHALLENGES

The product **Acquia Content Hub** is binding

- Content storage outside of Drupal
- Non-obvious usage
- High costs

WEBFACTORY MODULE LIMITATIONS

Allows multi-site management from a central website

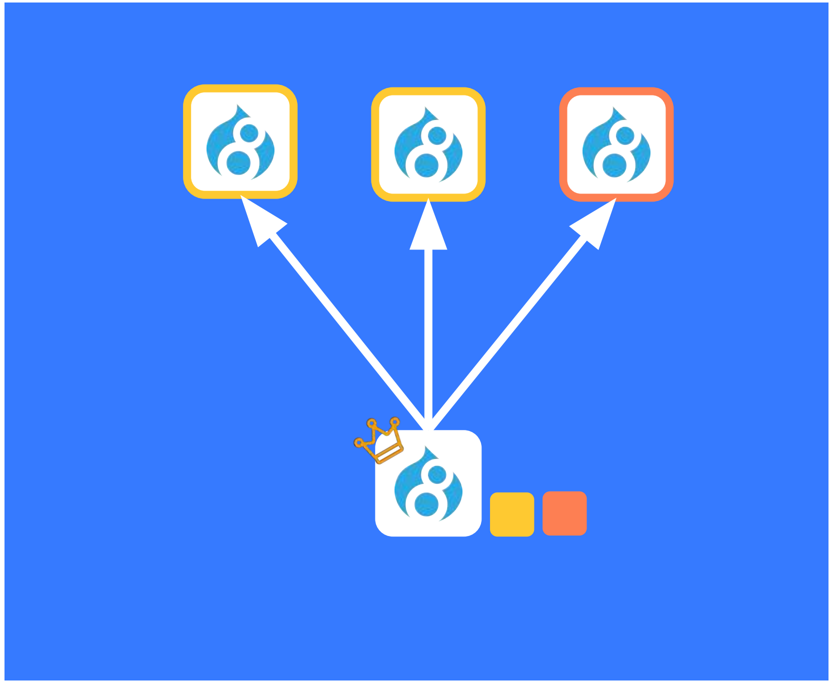
Facilitates the deployment of new sites

- Deploy a new “profile” directly from the backoffice

Sharing content from the central website

- Share entities: node
- A "channel" mechanism allows only certain entities to be shared at certain websites

Usage of Core webservices



THE DEPLOY ECOSYSTEM

Pushed by the community

Quite unstable

- At least at the beginning of 2017

Binding at the workflow level

More for content staging

ENTITY SHARE'S ARCHITECTURE

And how to help us!

SMILE

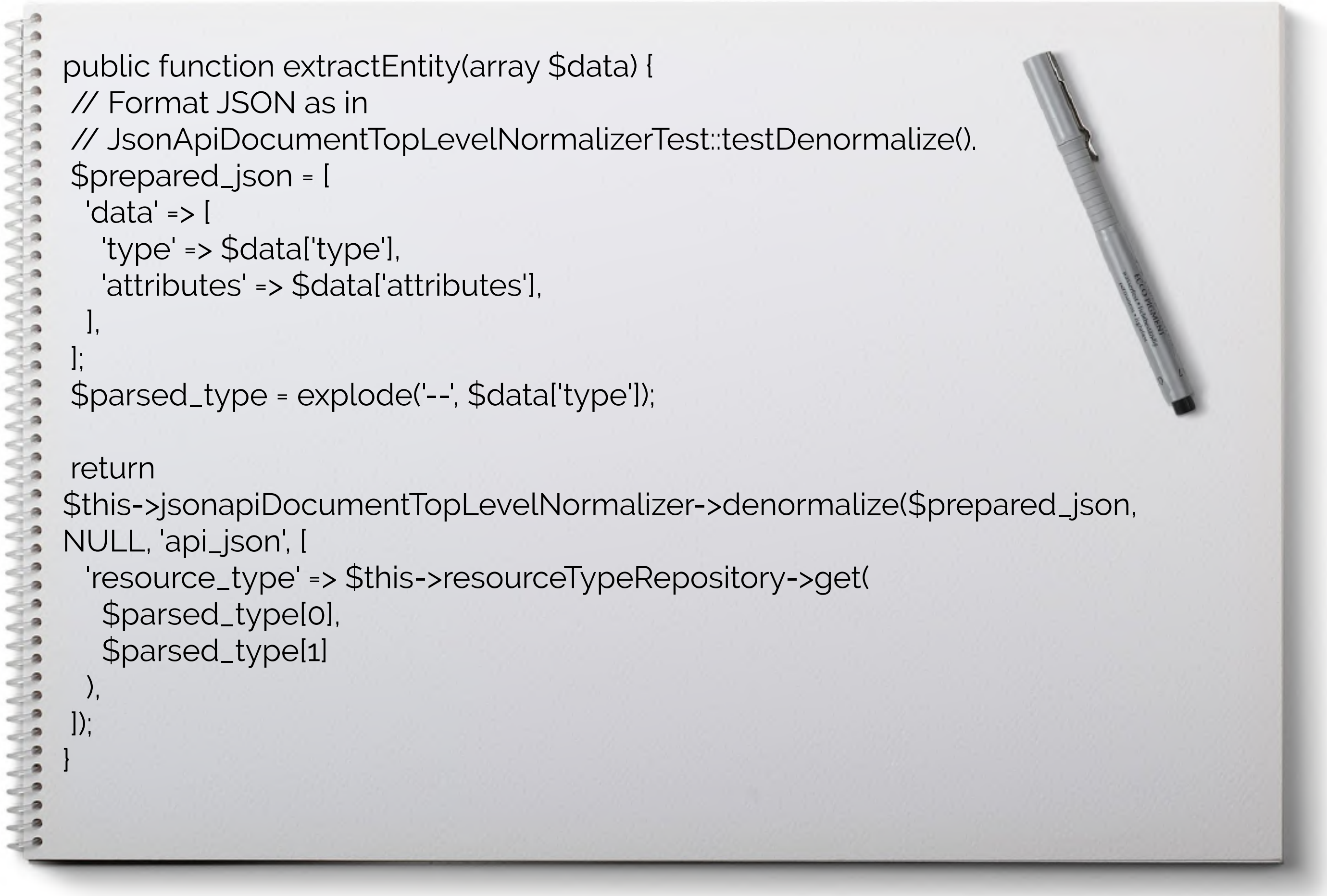
IT IS OPEN

JSON:API USAGE

- To provide entities listings
- More stable and documented than GraphQL (At the beginning of 2017)
- At the beginning of 2017, already an initiative for JSON:API to become a Core module
- Easy to use

JSON:API USAGE

ENTITY SHARE PROVIDES A UI ON TOP OF THE JSON:API



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

 return
$this->jsonapiDocumentTopLevelNormalizer->denormalize($prepared_json,
NULL, 'api_json', [
  'resource_type' => $this->resourceTypeRepository->get(
    $parsed_type[0],
    $parsed_type[1]
  ),
 ]);
}
```

ARCHITECTURE: 2 SUB-MODULES

THE 2 SUB-MODULES MAY BE ENABLED ON THE SAME SITE.

Entity share server

- Activate on the website that will provide the contents
- Provides the channel system
  - Prepare a JSON:API endpoint URL to call by the client website: entity type, bundle, language, filters, sorts
  - Plus listing of channels according to the authorized user

Entity share client

- Activate on the website that will pull (or push) content
- Allow to set the websites on which to connect to
- Provides a pull form (and push form but on an experimental branch)

ARCHITECTURE: GLOBAL VIEW OF THE PROCESS

WHEN PULLING, FOR EACH SELECTED ENTITY

Check if an entity exists with this UUID

- If no entity is found, create a new one
- If an entity exists, create or update the translation regarding the language in the JSON data.

Store the UUID in the processed entities list to avoid infinite loop

Manage entity reference fields

- For each "relationship" (JSON:API) field, request the endpoint showing the list of entities referenced by the field
  - For each of these entities, do the initial process
  - Put the ids of the processed entities in the entity reference field value

Manage physical files

- If the entity is a file, use its properties (URI) to get the content of the file

ENTITY SHARE'S ECOSYSTEM

Entity Share async (sub-module)

- To mark content to be synced later by a queue during cron execution

Entity Share cron (separated project)

- https://www.drupal.org/project/entity_share_cron
- Provides an UI to configure frequency of automated pull of channels
- For more complex usage, there is an example module in Entity Share: entity_share_client_test

KNOWN PROBLEMS

Workaround for Core limitation on link fields with internal link values: Use JSON:API Extras

Metatag field breaks the import (~~#3060702~~)

Not working when server website is behind HTTP authentication (#2856713)

Implementation relies on some JSON:API internal classes (#2939827)

Support of Dynamic Entity reference field is broken since JSON:API 2.x (#3056102)

LIMITATIONS

To avoid side effects, config entities and users are not handled

Impossible to import non-translatable entities (~~#2996220~~)

Impossible to import content in a language not enabled (#3064328)

Push form (#2856715):

- Impossible to PATCH translations (JSON:API)
- Impossible to PATCH file field (JSON:API)

FEATURE REQUESTS

Better Pull form (~~#3077808~~, ~~#3077810~~, ~~#3077815~~, #2891653, #2856719, ~~#3064252~~)

Avoid to synchronize already synced entities (#3080629, #3077976, #3009258)

Parse RTE to get referenced entities (#3056911) :

- Entity embed
- Linkit

Parse Link fields to get referenced entities (#3064276)

Allow an entity to be updated locally after being synchronized once (#2975806)

Compatibility with Block field (~~#3064331~~)

Manage Pathauto behavior (~~#3064320~~)

Better channel form (#2856717)

ROADMAP

Priority 1:

- Automated tests!!! (#2909022)
- bug reports

Priority 2:

- Rework services and tests for better architecture and DX (#3060694)
- No more depend on JSON:API internal classes (#2939827)

Priority 3:

- Feature requests depending on client needs and sponsored collaborations.
- Now a feature request must have automated tests before being merged!!!

ALL THE DETAILS ON THE PROJECT PAGE AND IN THE MODULE ISSUES QUEUE

PERSPECTIVES

Entity Share V2?:

- Availability to have multiple bundles per channel (JSON:API Cross Bundles)
- System of configurable plugins to control behavior:
  - Depth of handled relationships
  - Sync all translations at once
  - Parse RTE
  - Parse Link field
  - …

See this issue to give and discuss the ideas.

DEMO

SMILE

I.T IS OPEN

Thanks for your attention!

And thanks to all the contributors!

SMILE

I.T IS OPEN

SMILE

I.T IS OPEN