

MIGRATING DATA TO **DRUPAL 8**

8.1.x
Ready!

about.me

Nacho Sánchez
CTO@**HACKITY**



@isholgueras

nacho@letshackity.com

drupal.org/u/isholgueras

<http://www.isholgueras.com>



MIEMBRO
DE LA
ASOCIACIÓN
ESPAÑOLA
DE DRUPAL





STEPS

1. Requirements
2. Anatomy of a migration
3. Migration Framework
4. D2D migrations



1.

REQUIREMENTS

Migrate is in core!

Three new modules

- ▶ **Migrate**
Handles migrations Framework.
 - ▶ **Migrate Drupal**
Contains migrations from D6 & D7.
 - ▶ **Migrate Drupal UI**
The older migrate update (new in 8.1.x).
- Take care!
It's in experimental!*

Migrate in core!

1. REQUIREMENTS

But... how can I execute that migration.

UI is not ready?

No Drush command?



1. REQUIREMENTS

So...
Contrib!



Needed

- ▶ Drupal 8.1.x
- ▶ Drush 8
- ▶ Migrate tools (contrib)
- ▶ Migrate Plus (contrib)

1.

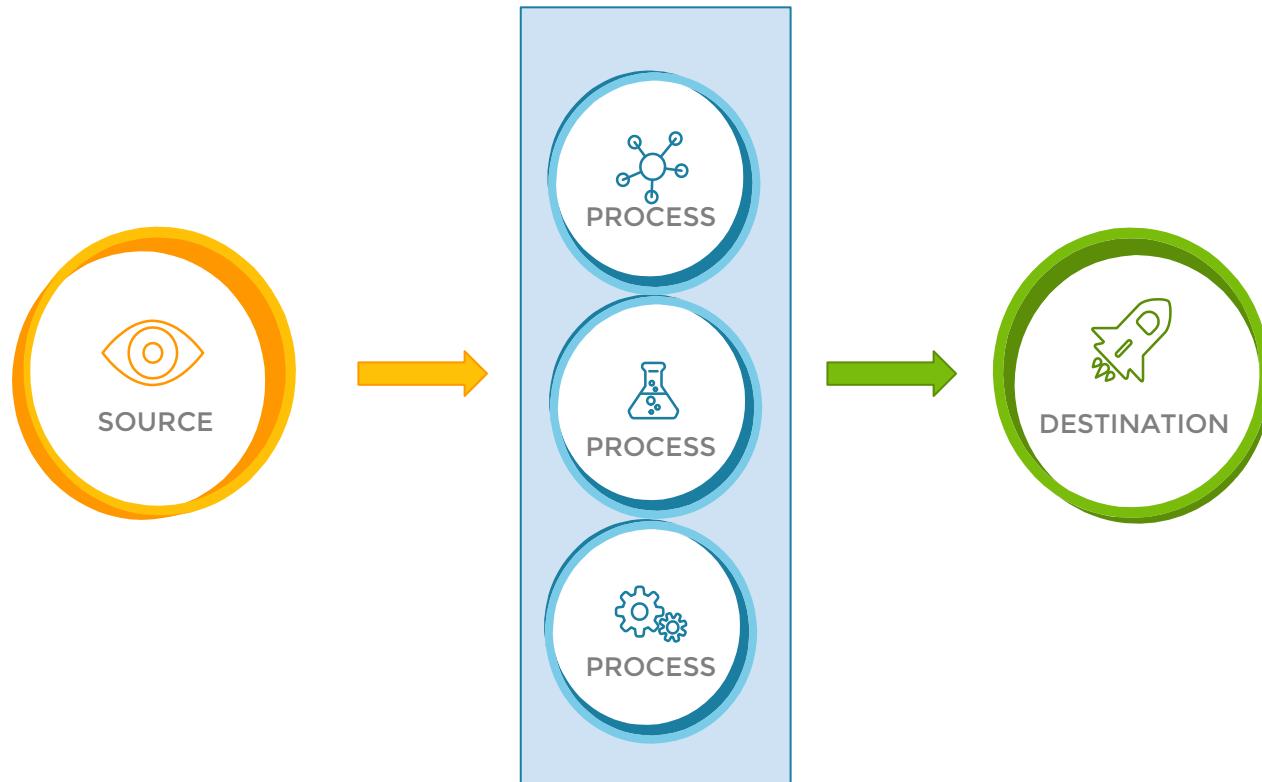
REQUIREMENTS

2.

ANATOMY OF A MIGRATION

Workflow

2. ANATOMY OF A MIGRATION



In files

DEFINITIONS

Yaml Files.

Custom files

PLUGINS

PHP Files

Core files or custom files

Types:

- Source
- Process
- Destination
- Builder
- ID Map

2.

ANATOMY OF A MIGRATION

The easiest example

2.
ANATOMY OF
A MIGRATION

DEFINITIONS

config/install/migrate_plus.migration.article_node.yml

```
id: article_node
label: Migrate posts from CakePHP to Drupal 8

source:
  plugin: article_node
  key: legacy
  #target: legacy
destination:
  plugin: entity:node
process:
  type:
    plugin: default_value
    default_value: article
  nid: id
  title: title
  'body/value': description
  uid: user_id
  status:
    plugin: default_value
    default_value: true
  created:
    plugin: callback
    source: created
    callable: strtotime
migration_dependencies: {} #new in 8.1.x
```

2.

ANATOMY OF
A MIGRATION

PLUGINS

src/Plugin/migrate/source/ArticleNode.php

2.

ANATOMY OF A MIGRATION

```
<?php
namespace ..
use ..
/**
 * Source plugin for beer content.
 *
 * @MigrateSource(
 *     id = "article_node"
 * )
 */
class ArticleNode extends SqlBase {

    public function query() {
        $query = $this->select
('articles','a')
        ->fields('a', [
            'id',
            'user_id',
            'title',
            'description',
            'created',
        ]);
        return $query;
    }
}
```

```
public function fields() {
    $fields = [
        'id' => $this->t("Article ID"),
        // ...
    ];
    return $fields;
}
public function getIds() {
    return [
        'id' => [
            'type' => 'integer',
            'alias' => 'a',
        ],
    ];
}
```

2. ANATOMY OF A MIGRATION

CONFIGURATION

sites/local/settings.php

```
<?php
//By default ['default']['migration']
$databases['legacy']['default'] = array(
  'database' => 'old_app',
  'username' => 'dev',
  'password' => 'dev',
  'prefix' => '',
  'host' => 'localhost',
  'port' => '3306',
  'namespace' => 'Drupal\\Core\\Database\\Driver\\mysql',
  'driver' => 'mysql',
);
```

EXECUTION

Only with Drush8 (8.1-dev) and migrate_tools enabled

2.

ANATOMY OF A MIGRATION

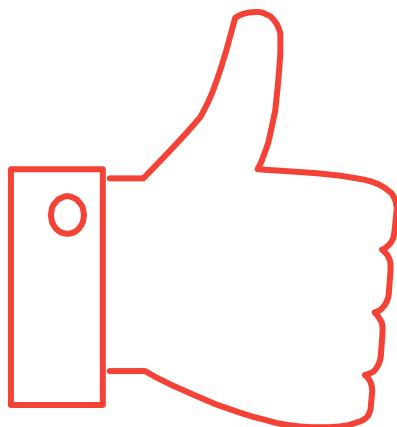
```
vagrant@dev $ drush8 migrate-status
Group: my_group    Status   Total   Imported   Unprocessed   Last imported
article_node      Idle     128     0          128

vagrant@dev $ drush8 migrate-import article_node; d8 ms
Processed 128 item (128 created, 0 updated, 0 failed, 0 ignored) - done with
'article_node'                                [status]
Group: my_group    Status   Total   Imported   Unprocessed   Last imported
article_node      Idle     128     128        0            2016-02-22 12:34:38

vagrant@dev $ drush8 migrate-rollback article_node; d8 ms
Rolled back 128 items - done with 'article_node'
Group: my_group    Status   Total   Imported   Unprocessed   Last imported
article_node      Idle     128     0          128           2016-02-22 12:34:38
```

2. ANATOMY OF A MIGRATION

Too easy!



3.

.Migrations Framework

0.- KNOWLEDGE

Source Plugin:

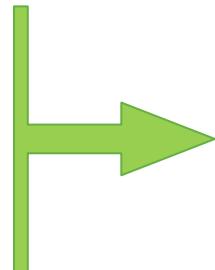
- SqlBase

Process Plugin:

- ProcessPluginBase

Destination Plugin:

- DestinationBase



**Most
used**

More plugins in: core/modules/migrate/src/Plugin/migrate

3.

**MIGRATION
FRAMEWORK**

0.- KNOWLEDGE

prepareRow **!==** transform

- extends SqlBase
- Before execution

- extends ProcessPluginBase
- During execution

3.

MIGRATION FRAMEWORK

1.- SOURCE

We tell here to SqlBase:

- Which Database is the Source.

```
#migrate_plus.migration.article_node.yml
source:
  plugin: article_node
  key: legacy
#target: legacy
```

```
#settings.php
$databases['key']['target'] = array(
```

3.

MIGRATION
FRAMEWORK

1.- SOURCE

- And which Plugin will make the Query

```
<?php
namespace
Drupal\cm_migrate\Plugin\migrate\source;
use ...
/**
 * Source plugin for articles content.
 *
 * @MigrateSource(
 *   id = "article_node"
 * )
 */
class ArticleNode extends SqlBase {

  public function query() {
    $query = $this->select('articles', 'a')
      ->fields('a', [
        'id',
        'user_id',
        'title',
        'description',
        'created',
      ]);
    return $query;
  }

  public function fields() {
    $fields = [
      'id' => $this->t("Article ID"),
      // ...
    ];
    return $fields;
  }

  public function getIds() {
    return [
      'id' => [
        'type' => 'integer',
        'alias' => 'a',
      ],
    ];
  }

  public function prepareRow(Row $row) {
    $id = $row->getSourceProperty('id');
    //print $id . PHP_EOL;
    //print_r($row);
    $row->setSourceProperty('user_id', 1);
    return parent::prepareRow($row);
  }
}
```

3. MIGRATION FRAMEWORK

2.- DESTINATION

How and where to save the data

- **Where** to save the data.

```
#migrate_plus.migration.article_node.yml
destination:
  plugin: entity:node
```

- entity:<place-here-an-entity>

3.

MIGRATION FRAMEWORK

2.- DESTINATION

Need more destination plugins?

Search for “destination:” in core

3.

MIGRATION
FRAMEWORK

3.

MIGRATION FRAMEWORK

3.- ID MAPPING

How Migrate associate old row with new row.

#	sourceid1	destid1	source_row_status	rollback_action	last_imported	hash
1	53878802-b218-4239-9605-2e1032577816	4	0	0	0	
2	538d0096-01c0-4ab6-92cd-2caa32577816	5	0	0	0	
3	538d84fb-c010-4675-9e18-2d3932577816	6	0			
4	539eadf1-d91c-4d95-b0d2-64f532577816	7	0			
5	53a003c9-2e54-41f3-af3e-78c132577816	8	0			
6	53b7db07-ba20-4d35-9f97-043d32577816	9	0			
7	53ba6d47-b2bc-4e37-ab81-39b832577816	10	0			
8	53bbf80d-6c30-4e4d-9713-440032577816	11	0			
9	53c30e11-38c8-48e3-8a0d-77e532577816	12	0			
10	53c3a17f-ef58-4869-8872-132832577816	13	0			
11	53c4c27d-510c-4873-8767-1e3132577816	14	0			

```
public function getIds() {
    return [
        'id' => [
            'type' => 'string',
            'alias' => 'u',
        ],
    ];
}
```

3.

MIGRATION FRAMEWORK

4.- PROCESS

How we **transform** each field, each file or data.

- **Map fields:** Same value as origin.
- **Modify:** Change o process the value
- **Add:** Create new fields from other fields or calculate these fields.

3.

MIGRATION FRAMEWORK

4.- PROCESS

Map fields. Values are equal in both sides

```
public function query() {  
    $query = $this  
    ->select('articles', 'a')  
    ->fields('a', [  
        'created',  
        'title',  
        'id',  
        'body',  
        'user_id',  
    ]);  
    return $query;  
}
```

```
#common mapping  
process:  
    title: title
```

3.

MIGRATION FRAMEWORK

4.- PROCESS

DefaultValue. Add a default value

```
public function query() {  
    $query = $this  
    ->select('articles', 'a')  
    ->fields('a', [  
        'created',  
        'title',  
        'id',  
        'body',  
        'user_id',  
    ]);  
    return $query;  
}
```

```
#default value  
process:  
    type:  
        plugin: default_value  
        default_value: article  
    'body/format':  
        plugin: default_value  
        default_value: plain_text
```

3.

Migration Framework

4.- PROCESS

Callable. Values are similars but a transform is needed with a **php function**

```
public function query() {  
    $query = $this  
    ->select('articles', 'a')  
    ->fields('a', [  
        'created_date',  
        'title',  
        'id',  
        'body',  
        'user_id',  
    ]);  
    return $query;  
}
```

```
'Callable.php' core plugin  
process:  
    created:  
        plugin: callback  
        source: created_date  
        callable: strtotime
```

3.

MIGRATION FRAMEWORK

4.- PROCESS

DedupeEntity. Values in destination cannot be equals, but in origin could be.

```
public function query() {  
    $query = $this  
        ->select('users', 'u')  
        ->fields('u', [  
            'user_id',  
            'user_name',  
            'mail',  
        ]);  
    return $query;  
}
```

```
# DedupeEntity.php' core  
# plugin  
process:  
name:  
    plugin: dedupe_entity  
    source: user_name  
    entity_type: user  
    field: name  
    postfix: _  
    # admin_1, _2, ...
```

3.

MIGRATION FRAMEWORK

4.- PROCESS

Migration. Values from other migration.

Magic!

```
# ArticleNode.php
public function query() {
    $query = $this
        ->select('articles', 'a')
        ->fields('a', [
            'created_date',
            'title',
            'id',
            'body',
            'user_id',
        ]);
    return $query;
}
```

```
# Migration as plugin
process:
    field_tags:
        plugin: migration
        migration: tags_node
        source: terms ←
migration_dependencies:
required:
    - tags_node
```

fields()

3.

Migration Framework

4.- PROCESS

Migration. Values from other migration.

Magic!

```
# ArticleNode.php
public function fields() {
    $fields = [
        'terms' => $this->t
("New field terms"),
        // ...
    ];
    return $fields;
}
```

```
# ArticleNode.php
public function prepareRow(\\Row $row) {
    $terms = $this->select('terms')
        //...
        ->fetchCol();
    //print_r($terms);
    $row->setSourceProperty('terms',
$terms);
    return parent::prepareRow($row);
}
```

3.

MIGRATION FRAMEWORK

4.- PROCESS

Migration. Values from other migration.

Magic!

```
# TagsNode.php
public function query() {
    // code
    ->fields('terms', [
        'parent_term',
        'term']);
}
```

```
#migrate_plus.migration.
tags_node.yml
source:
    plugin: tags_node
destination:
    plugin: entity:taxonomy_term
process:
    name: term
    parent:
        plugin: migration
        migration: tags_node
        source: parent_term
```

3.

Migration Framework

4.- PROCESS

CustomPlugins. Values are related but a custom transform for data is needed

```
public function query() {  
    $query = $this  
        ->select('articles', 'a')  
        ->fields('a', [  
            'created_date',  
            'title',  
            'id',  
            'body',  
            'user_id',  
        ]);  
    return $query;  
}
```

```
#CustomUUID.php' custom plugin  
process:  
    uid:  
        plugin: custom_uuid  
        source: user_id
```

3.

MIGRATION FRAMEWORK

4.- PROCESS

```
<?php
/**
 * @MigrateProcessPlugin(id = "custom_uuid")
 */
class CustomUUID extends ProcessPluginBase {

    public function transform($value, MigrateExecutableInterface
$migrate_executable, Row $row, $destination_property) {

        $uid = 0;
        $uids = \Drupal::entityQuery('user')
            ->condition('field_user_old_uuid', $row->getSourceProperty
("user_id"))
            ->execute();
        if (count($uids) > 0) {
            $uid = array_values($uids)[0];
        }
        return $uid;
    }
}
```

3.

MIGRATION FRAMEWORK

4.- PROCESS

CustomPlugins. We need to copy some files

```
public function query() {  
    $query = $this  
        ->select('files', 'f')  
        ->fields('f', [  
            'id',  
            'post_id',  
            'name',  
            'path',  
            'dir',  
        ]);  
    return $query;  
}
```

```
#CustomFiles.php' custom plugin  
destination:  
    plugin: entity:file  
process:  
    fid: id  
    filename: name  
    uri:  
        plugin: custom_file_uri  
        source:  
            - path  
            - name
```

3.

MIGRATION FRAMEWORK

4.- PROCESS

```
// In CustomFiles.php. source plugin.
public function prepareRow(Row $row) {
    // Set the complete external path to the image.
    $local_path = '/var/www/webroot/files/image/attachments/';
    $attachment = $row->getSourceProperty('path');
    $dir = $row->getSourceProperty('dir') . "/";

    $filepath = $local_path . $dir . $attachment;
    $row->setSourceProperty('path', $filepath);

    $file_name = basename($attachment);
    // Set filename. Not OK in every origin row.
    $row->setSourceProperty('name', $file_name);

    return parent::prepareRow($row);
}
```

3.

MIGRATION FRAMEWORK

4.- PROCESS

```
<?php
/**
 * @MigrateProcessPlugin(id = "custom_file_uri")
 */
class CustomFileUri extends ProcessPluginBase {

    public function transform($value, MigrateExecutableInterface
    $migrate_executable, Row $row, $destination_property) {

        list($filepath, $filename) = $value;
        $destination_base_uri = 'public://articles/';
        print "TRANSFORM Filepath: $filepath" . PHP_EOL;
        print "TRANSFORM Destination: $destination_base_uri" .
"$filename" . PHP_EOL;
        // public://articles/photo (2).jpg

        return $destination_base_uri . $filename;
    }
}
```

3.

MIGRATION FRAMEWORK

4.- PROCESS

More core plugins:

- concat
- extract
- flatten
- get
- iterator
- machine_name
- menu_link_parent
- route
- skip_on_empty
- skip_row_if_not_set
- static_map

Need an example?
Core is full of examples!

Search in core!

- “plugin: concat”
- “plugin: get”
- ...



3.

MIGRATION FRAMEWORK

5.- WANT MORE?

Take a look at:

- core/modules/migrate/src/Plugin/migrate
- core/modules/<any>/src/Plugin/migrate
- core/modules/<any>/config/migration_templates

Or search in core:

- “@MigrateSource(“
- “destination:”
- “migration_tags:”

3. MIGRATION FRAMEWORK

Drupal 8 core is
full of examples

Really!

4.

DRUPAL 2 DRUPAL MIGRATION

THINK THAT MIGRATE IS COMPLEX?

Maybe. And sometimes is difficult to debug.

So... Here we have:

Migrate Drupal

Migrate Drupal UI

And:

- Migrate Plus
- Migrate Tools

4.
D2D
MIGRATION

*experimental

4. D2D MIGRATION

REQUIREMENTS

- Fresh install of Drupal 8
- Modules enabled:
 - Migrate,
 - Drupal Migrate,
 - Drupal Migrate UI
- Access to D6 or D7 database
- Access to D6 or D7 files

SET UP

Go to: “/upgrade”.

4. D2D MIGRATION

Drupal Upgrade

Drupal Upgrade

Upgrade a Drupal site by importing it into a clean and empty new install of Drupal 8. You will lose any existing configuration once you import your site into it. See the [upgrading handbook](#) for more detailed information.

1. **Back up the database for this site.** Upgrade will change the database for this site.
2. Make sure that the host this site is on has access to the database for your previous site.
3. If your previous site has private files to be migrated, a copy of your files directory must be accessible on the host this site is on.
4. In general, enable all modules on this site that are enabled on the previous site. For example, if you have used the book module on the previous site then you must enable the book module on this site for that data to be available on this site.
5. Put this site into [maintenance mode](#).

This upgrade can take a long time. It is better to import a local copy of your site instead of directly importing from your live site.

[Continue](#)

SET UP

▼ Source database
Provide credentials for the database of the Drupal site you want to upgrade.

Database type *

MySQL, MariaDB, Percona Server, or equivalent
 PostgreSQL
 SQLite

Database host *
localhost

Database name *
[empty input field]

Database username *
[empty input field]

Database password
[empty input field]

► Advanced options

▼ Source files

Files directory
[empty input field]
To import files from your current Drupal site, enter a local file directory containing your site (e.g. /var/www/docroot), or your site address (e.g. http://example.com).

Review upgrade

4. D2D MIGRATION

EXECUTE

4. D2D MIGRATION

Are you sure?

Are you sure?

- 12 available upgrade paths
- 30 missing upgrade paths

Missing upgrade paths

The following items will not be upgraded. For more information see [Upgrading from Drupal 6 or 7 to Drupal 8](#).

Source	Destination
book	Missing
codefilter	Missing
color	Missing
comment	Missing
comment_notify	Missing
contextual	Missing
field_sql_storage	Missing
field_ui	Missing
gravatar	Missing
help	Missing
jquery_update	Missing
list	Missing

EXECUTE

4. D2D MIGRATION

▼ Available upgrade paths

Source	Destination
block	block block_content
contact	contact
dblog	dblog
field	field
file	file
filter	filter
menu	system
node	node
text	text
update	update
user	user

[Perform upgrade](#) [Cancel](#)

EXECUTE

4. D2D MIGRATION

Currently upgrading Nodes (Feed) (29 of 40 total tasks)
Upgraded Nodes (Book page) (processed 11 items total)
Upgraded User accounts (processed 1 item total)
Migration user_picture_entity_form_display did not meet the requirements. Missing migrations user_picture_field_instance. requirements: user_picture_field_instance.
Operation on User picture form display configuration failed
Migration user_picture_entity_display did not meet the requirements. Missing migrations user_picture_field_instance. requirements: user_picture_field_instance.
Operation on User picture display configuration failed
Migration user_picture_field_instance did not meet the requirements. Missing migrations user_picture_field. requirements: user_picture_field.
Operation on User picture field instance configuration failed
Source ID 986: File 'files/LocalizationClientDrupal8Alpha1Submission.png' does not exist.
Source ID 981: File 'files/LocalizationClientDrupal8Alpha1.png' does not exist.
Source ID 976: File 'files/Drupal8MenuInSidebar.png' does not exist.
Source ID 971: File 'files/Drupal8MenuItemItems.png' does not exist.
Source ID 966: File 'files/Drupal8FrenchArticle.png' does not exist.
Source ID 961: File 'files/Drupal8EditArticle.png' does not exist.
Source ID 956: File 'files/Drupal8ArticleFieldsCustomized.png' does not exist.
Source ID 951: File 'files/Drupal8MultilingualModulesOverlap.png' does not exist.
Source ID 946: File 'files/Drupal8TranslationSearch.png' does not exist.
Source ID 941: File 'files/Drupal8OutdatedWorkflow2.png' does not exist.
Source ID 936: File 'files/Drupal8OutdatedWorkflow1.png' does not exist.
Source ID 931: File 'files/Drupal8TranslationPermissions.png' does not exist.

...



70%

EXECUTE

4. D2D MIGRATION

Recent log messages

▼ Filter log messages

Type	Severity
cron	Emergency
migrate_upgrade	Alert
php	Critical
system	Error
user	Warning
	Notice
	Info
	Debug

Filter **Reset**

► Clear log messages

Type	Date ▾	Message	User	Operations
migrate_upgrade	02/02/2016 – 15:14	Upgraded Node title label (processed 6 items total)	Gábor Hoitsy	
migrate_upgrade	02/02/2016 – 15:14	Upgraded Node revisions (Tweet) (processed 0 items...)	Gábor Hoitsy	
migrate_upgrade	02/02/2016 – 15:14	Upgraded Node revisions (Post) (processed 834 items...)	Gábor Hoitsy	
migrate_upgrade	02/02/2016 – 15:14	Upgraded Node revisions (Issue) (processed 0 items...)	Gábor Hoitsy	
migrate_upgrade	02/02/2016 – 15:14	Upgraded Node revisions (Feed item) (processed 0 items...)	Gábor Hoitsy	

4.
D2D
MIGRATION

Can I code?

Sure!

1.- SOURCE

4. D2D MIGRATION

We tell here to **DrupalSqlBase**:

- Which Database is the Source.

```
#migrate_plus.migration.article_node.yml
source:
  plugin: custom_user
  key: legacy

#equal as standard migration
```

1.- SOURCE

4.

D2D MIGRATION

Class now extends from DrupalSqlBase

```
class User extends DrupalSqlBase
    implements SourceEntityInterface {

public function query() {
    return $this->select('users', 'u')
        ->fields('u', [ //Fields ])
        ->condition('uid', 0, '>');
}
```

1.- SOURCE

- Can use baseFields and add more fields

```
public function fields() {  
    $fields = $this->baseFields();  
    $fields['first_name'] = $this->t('First Name');  
    $fields['last_name'] = $this->t('Last Name');  
    $fields['biography'] = $this->t('Biography');  
    return $fields;  
}
```

4. D2D MIGRATION

4.

D2D MIGRATION

1.- SOURCE

- Prepare the Rows

```
public function prepareRow(Row $row) {  
    $uid = $row->getSourceProperty('uid');  
  
    // first_name  
    $result = $this->getDatabase()->query('SELECT fld.field_first_name_value  
        FROM {field_data_field_first_name} fld  
        WHERE fld.entity_id = :uid  
' , array(':uid' => $uid));  
    foreach ($result as $record) {  
        $row->setSourceProperty('first_name',  
            $record->field_first_name_value );  
    }  
}
```

4. D2D MIGRATION

Or give a try
to ...

4.
D2D
MIGRATION

migrate-manifest

```
drush migrate-manifest --legacy-db-url=... \  
D6Manifest-User.yml
```

<https://www.youtube.com/watch?v=l4m5msEY-Jg>

Manifest

List of migrations

```
#D6Manifest-User.yml
- d6_user
- d6_user_profile_field
- d6_user_profile_field_instance
- d6_user_profile_entity_display
- d6_user_profile_entity_form_display
- d6_profile_values:user
- d6_filter_format
- d6_user_role
- d6_user_picture_entity_display
- d6_user_picture_entity_form_display
- d6_user_picture_file
- d6_user_picture_field
- d6_user_picture_field_instance
```

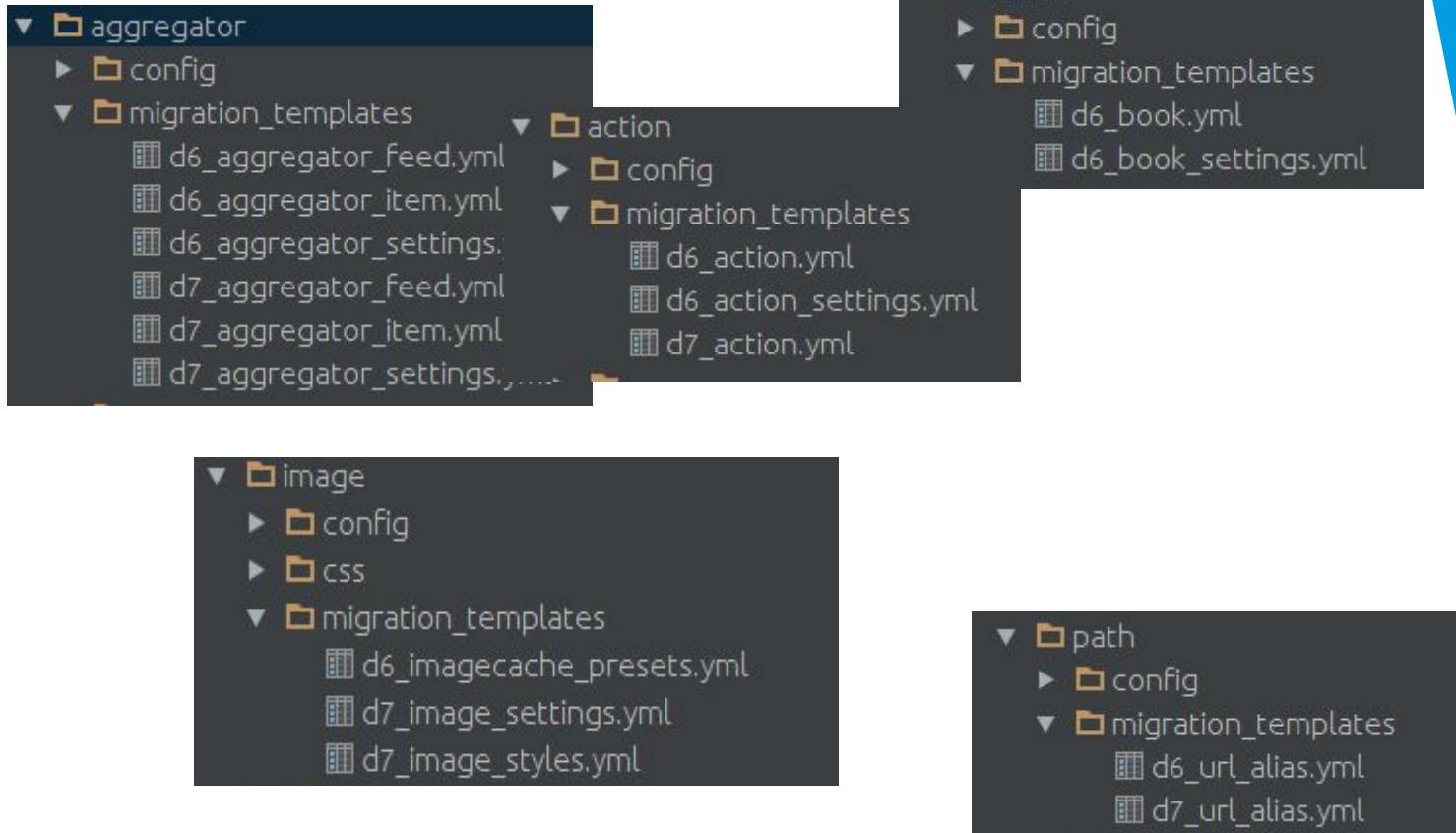
4.
D2D
MIGRATION

4.

D2D MIGRATION

Manifest

Migrations that are in core.



THANKS!

@isholgueras

nacho@letshackity.com