Migrate API in Drupal 8
Upgrading from Drupal 6 / 7 to Drupal 8

Sven Decabooter
@sdecabooter

2dotstwice
connecting the dots

UC Leuven Limburg
MOVING MINDS

AMPLEXOR
Pure Content Management

Migrate API in Drupal 8

Upgrading from Drupal 6 / 7 to Drupal 8

•••

Sven Decabooter
@sdecabooter

About me

- Freelance Drupal developer
- President of Drupal User Group Belgium vzw
- Co-organiser of Drupalcamp Leuven
- Contributor to Migrate in Drupal 8
- https://www.drupal.org/user/35369
- https://twitter.com/sdecabooter

Drupal 8 release party in Ghent

- Thursday 19 November 2015 - Vooruit Gent
- http://www.meetup.com/DUG-BE

Presentation outline

- Getting Migrate in Drupal 8 core
- Drupal 8 Migrate modules
- Drupal 8 Migrate concepts
- Demo migration
- Migrate pipeline
- Advanced topics (time permitting)
- How can you help?
- Resources

Getting Migrate in Drupal 8 core

History of Migrate

- Drupal 7:
  - Migrate & Drupal-to-Drupal data migration (migrate_d2d)
  - contrib modules by Mike Ryan
- Drupal 8:
  - “Migrate in Core” initiative started around Drupalcon Prague
  - Aim: to provide a better upgrade path - rather than upgrade.php
  - Be able to migrate directly from D6 or D7 to D8

Current status

- Migrate & Migrate Drupal modules will be in Drupal 8.0.0
- Contains:
  - API functionality
  - D6 > D8 migration (pretty stable & complete)
  - D7 > D8 migration (users, nodes, terms, comments, blocks and some more)
- Marked as “Experimental”
  - At least until the 8.1.0 release
  - Still needs lots of testing ‘in the wild’

Current status

- Migrate UI & Drush support
  - Currently still in contrib: https://www.drupal.org/project/migrate_upgrade
  - Still unsure when UI will go into Drupal core
  - Drush command will become part of core Drush at some point
- What will be migrated by Drupal Migrate in core?
  - Core content and configuration
  - For modules that ship with Drupal 8
    - also if source in D6 / D7 was contrib, e.g. CCK, ImageCache, Link, ...
    - but still some holes: e.g. Views, i18n content, ... - see https://www.drupal.org/node/2167633
  - Contrib modules will need to provide their own migrations

Drupal 8 Migrate modules

Drupal 8 core

- Migrate module
  - API to migrate content & configuration into Drupal 8
- Migrate Drupal module
  - Functionality to migrate content & configuration from Drupal 6 / 7 to Drupal 8
- Migration plugin classes & migration templates
  - Inside each core module's directory

Drupal 8 contrib

- Drupal Upgrade (https://www.drupal.org/project/migrate_upgrade)
  - UI to upgrade from Drupal 6 or 7 (Batch upgrading at /upgrade)
  - Drush command `drush migrate-upgrade`
- Migrate Plus (https://www.drupal.org/project/migrate_plus)
  - Optional functionality that is not added to Migrate in D8 core
  - e.g. CSV source, additional Drush commands, ...
  - Example migrations

Drupal 8 Migrate concepts

Migration configuration

- A core or contrib module can define migration configuration
- Migrate configuration is stored in configuration entities
- 2 ways to define the configuration entities:
  - As a standard configuration file
    - [module]/config/optional/migrate.migration.[thing].yml
    - recommended for contrib modules & custom migrations
  - In a migration template
    - [module]/migration_templates/d7_[thing].yml
    - recommended for core
    - templates get extra processing before being turned into config entities
- In both cases YAML file with same syntax

www > core > modules > node > migration_templates > d6_node.yml

Project

- node
  - config
  - css
  - migration_templates
    - d6_node.yml
    - d6_node_revision.yml
    - d6_node_setting_promote.yml
    - d6_node_setting_status.yml
    - d6_node_setting_sticky.yml
    - d6_node_settings.yml
    - d6_node_type.yml
    - d6_view_modes.yml
    - d7_node.yml
    - d7_node_revision.yml
    - d7_node_settings.yml
    - d7_node_title_label.yml
    - d7_node_type.yml

d6_node.yml

```yaml
id: d6_node
label: Nodes
migration_tags:
  - Drupal 6
builder:
  plugin: d6_node
source:
  plugin: d6_node
process:
  nid: nid
  vid: vid
  type: type
  langcode:
    plugin: default_value
    source: language
    default_value: "und"
  title: title
  uid: node_uid
  status: status
  created: created
  changed: changed
```

Example: Drupal 8 core - node module migration templates

www > modules > migrate_plus > migrate_example > config > install > migrate.migration.beer_comment.yml

Project

- core
- modules
  - custom
  - migrate_plus
    - config
    - migrate_example
      - config
        - install
          - migrate.migration.beer_comment.yml
          - migrate.migration.beer_node.yml
          - migrate.migration.beer_term.yml
          - migrate.migration.beer_user.yml
          - migrate_plus.migration_group.beer.yml

migrate.migration.beer_comment.yml

```yaml
# Migration configuration for beer comments.
id: beer_comment
label: Comments on beers
migration_group: beer
source:
  plugin: beer_comment
destination:
  plugin: entity:comment
process:
  pid:
    plugin: migration
    migration: beer_comment
    source: cid_parent
  entity_id:
    plugin: migration
    migration: beer_node
```

Example: Drupal 8 contrib example migration config

Migration workflow

- Execution of a migration:
  - Load the appropriate Migrate configuration entities
  - Push each data migration through the Migrate pipeline:
    - Source: retrieve the source data
    - Process: prepare the source data for import
    - Destination: save the data to Drupal 8
- More details later in this presentation

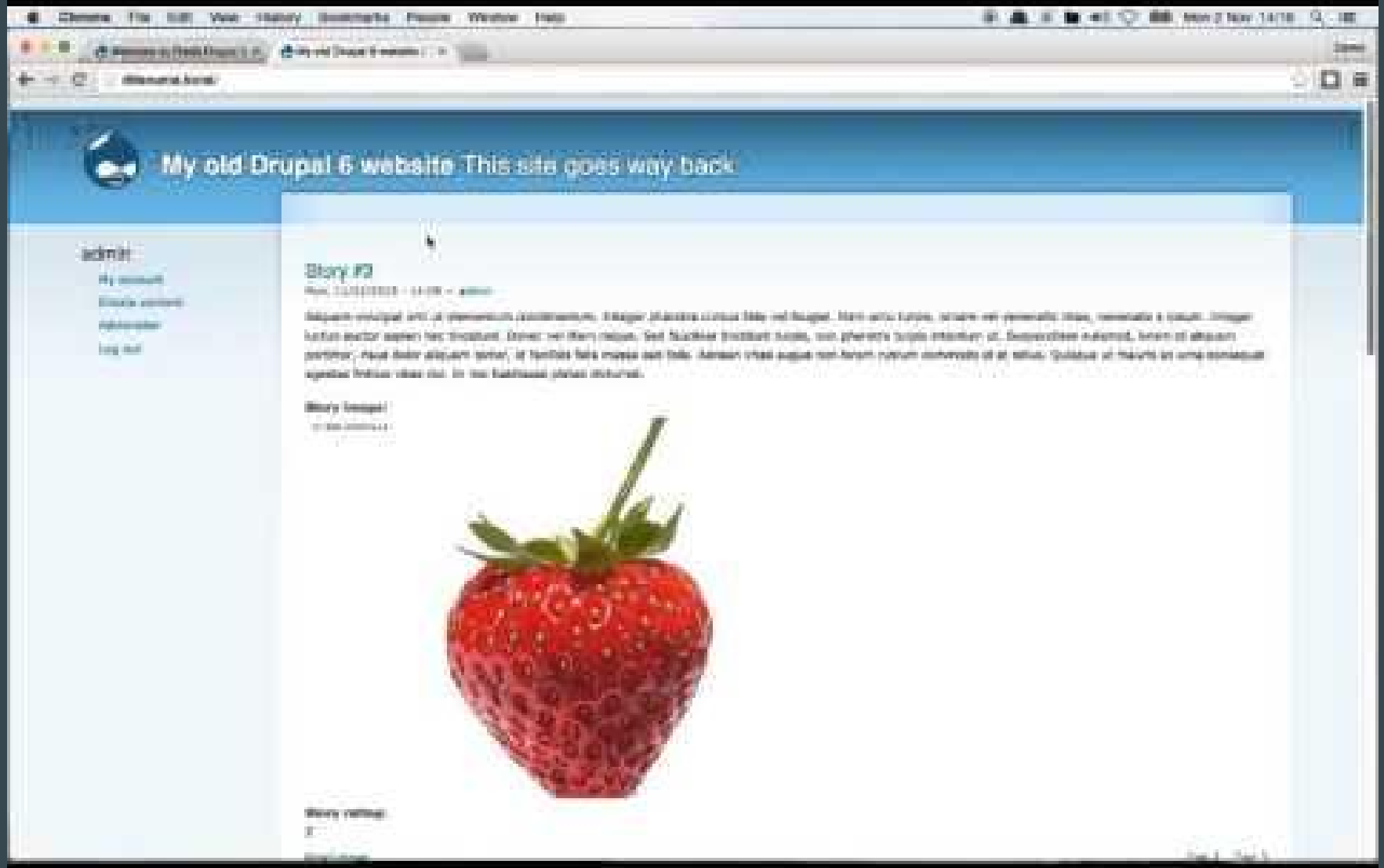
Other Migrate concepts

- Mapping tables keep track of source IDs and linked target ID
  - Migrations can be run multiple times
  - New content will be added
  - Unchanged content will be ignored
  - Updated content will be re-imported
  - Content deleted from source will remain in the destination
- Migration groups (in migrate_plus contrib, not in core)
  - Group migrations together
  - To execute them together
  - To provide shared configuration between migrations in group

Other Migrate concepts

- Stubs
  - Placeholder destination objects
  - Also added to mapping table for consistent IDs
  - To be updated with real data later in the migration process
  - Example:
    - Migration of a child taxonomy term that has a parent.
    - Parent has not yet been migrated: stub taxonomy term entity gets created
    - Stub gets a real ID, but gibberish data (e.g. name)
    - Stub ID gets added to mapping table for later retrieval
    - Later on parent taxonomy term gets migrated with real data into term entity with stub ID

Demo: Drupal 6 > Drupal 8 Migration


My old Drupal 6 website This site goes way back
admin
Story #2

Migrate pipeline

Migrate configuration files

- Content & configuration migrations are defined in YAML files
- Recap:
  - Normal configuration entity file in config/optional - for most cases
  - Migration template file in migration_templates - for more advanced cases
- Contains:
  - identifying data (id, label, migration tags, migration groups, ...)
  - source, process & destination configuration
  - dependencies (required / optional)

d6_file.yml

```yaml
# Every migration that saves into {file_managed} must have the d6_file
# migration as an optional dependency to ensure d6_file runs first.
id: d6_file
label: Files
migration_tags:
  - Drupal 6
source:
  plugin: d6_file
process:
  fid: fid
  filename: filename
  uri:
    plugin: file_uri
    source:
      - filepath
      - file_directory_path
      - temp_directory_path
      - is_public
  filemime: filemime
  filesize: filesize
  status: status
  changed: timestamp
  uid: uid
destination:
  plugin: entity:file
```

Migrate pipeline

- Source → Process → Destination
- These are all Drupal 8 Plugins
- Source Plugins provide rows of source data (unprocessed)
- Process Plugins prepare & manipulate data for import
- Destination Plugins save data to Drupal 8 targets
  - e.g. content entity, configuration entity, plugin, ...

Source Plugin

- Provides unprocessed rows of source data
- Can be retrieved from different sources:
  - Drupal database (D6 / D7) - use DrupalSqlBase class (D8 core)
  - regular SQL database - use SqlBase class (D8 core)
  - CSV file - use CSV class (D8 Migrate Plus contrib module)
- Iterates over each source row
- Returns the desired fields for each row

Process Plugin

- Describes how to manipulate source data
- Simplest form: copy as-is from source to destination

  `<target property>: <source property>`

```
process:
  name: site_name
  mail: site_mail
  slogan: site_slogan
```

Process Plugin

- One or more process plugins can be executed for each source field
- Key = target property
- Value = (array of) process plugins

```yaml
process:
  d8_target:
    -
      plugin: <first_plugin_name>
      source: d6_source
    -
      plugin: <second_plugin_name>
      some_setting: TRUE
```

Process Plugin: default_value

- Sets a fixed default value
- Also useful to set a default value if a previous process plugin returned no value

```yaml
langcode:
  plugin: default_value
  source: language
  default_value: "und"
```

Process Plugin: static_map

- Provide a map of static "source: destination" values
- Searches map to set destination property based on given source
- "source" can be one field, or array of fields

```yaml
id: d6_view_modes
label: View modes
migration_tags:
  - Drupal 6
source:
  plugin: d6_view_mode
  constants:
    targetEntityType: node
    status: true
process:
  mode:
    plugin: static_map
    source: view_mode
    map:
      0: normal
      1: preview
      2: search_index
      3: search_result
      4: rss
      5: print
      teaser: teaser
      full: full
```

Process Plugin: concat

- Concatenate array of source properties into single destination property

```yaml
id: d7_block
label: Blocks
migration_tags:
  - Drupal 7
source:
  plugin: block
process:
  # Block status is not a thing
  # disabled blocks.
  status:
    plugin: skip_on_empty
    method: row
    source: status
  id:
    -
      plugin: concat
      source:
        - theme
        - module
        - delta
      delimiter: _
```

Process Plugin: migration

- Gets mapped ID from Migrate mapping tables
- Example:
  - D6 site source has “vid” 123
  - d7_taxonomy_vocabulary migration has migrated this to “vid” 456 on D8 site
  - migration plugin returns 456 for given vid 123

d7_taxonomy_term.yml

```yaml
id: d7_taxonomy_term
label: Taxonomy terms
migration_tags:
  - Drupal 7
source:
  plugin: taxonomy_term
process:
  tid: tid
  vid:
    plugin: migration
    migration: d7_taxonomy_vocabulary
    source: vid
  name: name
  description: description
  weight: weight
  parent:
    plugin: migration
    migration: d7_taxonomy_term
    source: parent
  changed: timestamp
destination:
  plugin: entity:taxonomy_term
migration_dependencies:
  required:
    - d7_taxonomy_vocabulary
```

Process Plugins

- More info & additional process plugins: https://www.drupal.org/node/2129651
- Writing your own plugin: see advanced topics

Destination Plugin

- Specify plugin that takes care of saving the destination value
- Can be a Drupal 8 entity

```
destination:
  plugin: entity:user
```

- Or a Drupal 8 config entity

```
destination:
  plugin: config
  config_name: book.settings
```

- Or a custom destination Plugin

Recap

- Core & contrib modules can have Migration configuration files
- They describe:
  - From what source where to get the data
  - How to process the source data
  - How to save the data in your Drupal 8 site
- Use the migrate_upgrade module to run a full migration
  - via Drush: drush migrate-upgrade
  - via UI: http://example.com/upgrade
- You can write your own migration scripts

Advanced topics

Writing your own plugins - Source plugin

- in [modulename]/src/Plugin/migrate/source/[name].php
- Extend existing source base class:
  - \Drupal\migrate\Plugin\migrate\source\SqlBase
  - \Drupal\migrate_drupal\Plugin\migrate\source\DrupalSqlBase
  - \Drupal\migrate_drupal\Plugin\migrate\source\Variable
  - \Drupal\migrate_drupal\Plugin\migrate\source\VariableMultiRow
  - \Drupal\migrate_drupal\Plugin\migrate\source\d7\FieldableEntity

Writing your own plugins - Source plugin

- In case of (Drupal)SqlBase, implement:
  - public function query()
  - public function fields()
  - public function getIds()

File.php ×

\Drupal\file\Plugin\migrate\source\d6\File

```php
<?php

/**
 * @file
 * Contains \Drupal\file\Plugin\migrate\source\d6\File.
 */

namespace Drupal\file\Plugin\migrate\source\d6;

use Drupal\migrate\Row;
use Drupal\migrate_drupal\Plugin\migrate\source\DrupalSqlBase;

/**
 * Drupal 6 file source from database.
 *
 * @MigrateSource(
 *   id = "d6_file"
 * )
 */
class File extends DrupalSqlBase {

  /**
   * {@inheritdoc}
   */
  public function query() {
    return $this->select('files', 'f')
      ->fields('f')
      ->orderBy('timestamp')
      // If two or more files have the same timestamp, they'll end up in a
      // non-deterministic order. Ordering by fid (or any other unique field)
      // will prevent this.
      ->orderBy('fid');
  }
```

File.php

```php
  /**
   * {@inheritdoc}
   */
  public function fields() {
    return array(
      'fid' => $this->t('File ID'),
      'uid' => $this->t('The {users}.uid who added the file. If set to 0, this file was added by an anonymous user.'),
      'filename' => $this->t('File name'),
      'filepath' => $this->t('File path'),
      'filemime' => $this->t('File Mime Type'),
      'status' => $this->t('The published status of a file.'),
      'timestamp' => $this->t('The time that the file was added.'),
      'file_directory_path' => $this->t('The Drupal files path.'),
      'is_public' => $this->t('TRUE if the files directory is public otherwise FALSE.'),
    );
  }
  /**
   * {@inheritdoc}
   */
  public function getIds() {
    $ids['fid']['type'] = 'integer';
    return $ids;
  }

}
```

Writing your own plugins - Process plugin

- in [modulename]/src/Plugin/migrate/process/[name].php
- Extend ProcessPluginBase
- Override public function transform()

```php
namespace Drupal\image\Plugin\migrate\process\d6;

use ...

/**
 * @MigrateProcessPlugin(
 *   id = "d6_imagecache_actions"
 * )
 */
class ImageCacheActions extends ProcessPluginBase {

  /**
   * {@inheritdoc}
   */
  public function transform($value, MigrateExecutableInterface $migrate_executable, Row $row, $destination_property) {
    $effects = [];

    foreach($row->getSourceProperty('actions') as $action) {
      $id = preg_replace('/^imagecache/', 'image', $action['action']);

      if ($id === 'image_crop') {
        $action['data']['anchor'] = $action['data']['xoffset'] . '-' . $action['data']['yoffset'];

        if (!preg_match('/^[a-z]*\-[a-z]*/', $action['data']['anchor'])) {
          $migrate_executable->message->display(
            'The Drupal 8 image crop effect does not support numeric values for x and y offsets. Use keywords to set crop
            'error'
          );
        }

        unset($action['data']['xoffset']);
        unset($action['data']['yoffset']);
      }

      $effects[] = [
        'id'     => $id,
        'weight' => $action['weight'],
        'data'   => $action['data'],
      ];
    }

    return $effects;
  }
```

Writing your own plugins - Destination plugin

- in [modulename]/src/Plugin/migrate/destination/[name].php
- Extend DestinationBase
- to create Drupal entities:
  - extend EntityContentBase or EntityConfigBase
- Implement public function import()

Config entity files vs migration_templates

- Recap:
  - Config entity: [module]/config/optional/migrate.migration.[stuff].yml
  - Template: [module]/migration_templates/[stuff].yml
- Migration templates pass through extra layer - builder system before being turned into config entities
- Migration config entity definitions get created as-is

Config entity files vs migration_templates

- Migration templates
  - Used for “dynamic” migrations
    - where migration config entities need to be created on the fly
  - Workflow:
    - Builder Plugin parses template
    - Applies dynamic processing
    - Returns derived migration entities

How can you help?

Getting involved

- Test the upgrade path with your site(s) and report issues:
  - https://www.drupal.org/node/2257723 (documentation)
  - Report in https://www.drupal.org/project/issues/drupal (select “migration system” component)
- Help improve D6 / D7 core migrations
- Help get the Upgrade UI in core (frontend / UX)
- Add migrations to contrib D8 modules
- IRC: #drupal-migrate

Resources

Resources

- Upgrading from D6/D7 to D8 - docs:
  - https://www.drupal.org/upgrade/migrate
- D8 Migrate API docs:
  - https://www.drupal.org/node/2127611
- Migrate Example module in Migrate Plus
  - https://www.drupal.org/project/migrate_plus
- Blogposts:
  - https://www.advomatic.com/blog/drupal-8-migrate-from-drupal-6-with-a-custom-process-plugin
  - http://mikeryan.name/blog/mikeryan/update-on-migration-in-drupal-8

Questions?

Get in touch

Twitter: @sdecabooter
IRC: svendecabooter
Web: http://svendecabooter.be


ANNO
2015
DRUPAL
CAMP
Leuven


2dotstwice
connecting the dots


AMPLEXOR
Pure Content Management


UC Leuven
Limburg
MOVING MINDS


uniway
Business Critical Internet Solutions


ONE
AGENCY


TOBANIA


Dropsolid
Makkelijk Digitaal Ondernemen


dazzle
the best the web can get


Combell
Your host on the internet


ACQUIA
THINK AHEAD