Migrate API in Drupal 8

Sven Decabooter
@sdecabooter

dazzle
the best the web can get

About me

- Freelance Drupal developer
- Doing Drupal for over 10 years
- Working for Dazzle - agency in Brussels, Belgium
- President of Drupal User Group Belgium vzw

https://www.drupal.org/user/35369

https://twitter.com/sdecabooter

dazzle™
the best the web can get


DE ENGHEL BAR & RESTAURANT
photo byTom Roeleveld

Dazzle is looking for international partners

- Only senior profiles with 6+ years experience
- Transparant & no bullshit
- Interesting rates
- https://www.drupal.org/dazzle
- Talk to me or contact luc.claeys@dazzle.be

dazzle
the best the web can get

Presentation outline

- Getting Migrate in Drupal 8 core
- Basic Migrate concepts
- Demo migrations
- Drupal 8 contrib Migrate modules
- Creating Migrate configuration files
- How can you help?
- Resources
- Writing your own plugins

Getting Migrate in Drupal 8 core

dazzle
the best the web can get

History of Migrate

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

History of Migrate

- Drupal 8.0.x:
  - Modules added to core:
    - Migrate: API functionality
    - Migrate Drupal: D6 → D8 / D7 → D8 migrations
  - Migration plugin classes & migration templates
    - Inside each core module's directory

History of Migrate

- Drupal 8.1.x:
  - Bugfixes and more default migrations
  - Modules added to core:
    - Migrate Drupal UI: UI for D6 → D8 / D7 → D8 migrations
  - Migrations are now Plugins (no longer config entities)

Migrate in Drupal 8

- Marked as “Experimental”
  - Still needs lots of testing ‘in the wild’
  - Still experimental in the 8.2.0 release
    - but possibly “alpha” → “beta” stability level
    - see https://www.drupal.org/core/experimental

Drupal 8 core scope

- 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 missing functionality: e.g. Views, D6 files … - see https://www.drupal.org/node/2167633
  - Contrib modules will need to provide their own migrations

Basic Migration concepts

Migration configuration

- Each Migration is defined in a YAML config file
- Usually you define a config file for each distinct data set / object type
  - e.g. Drupal 8 core contains YAML template files per module for migrating specific data items from D6/D7 to Drupal 8
- YAML file provides configuration for:
  - Migration pipeline: Source → Process → Destination
  - Metadata and dependencies

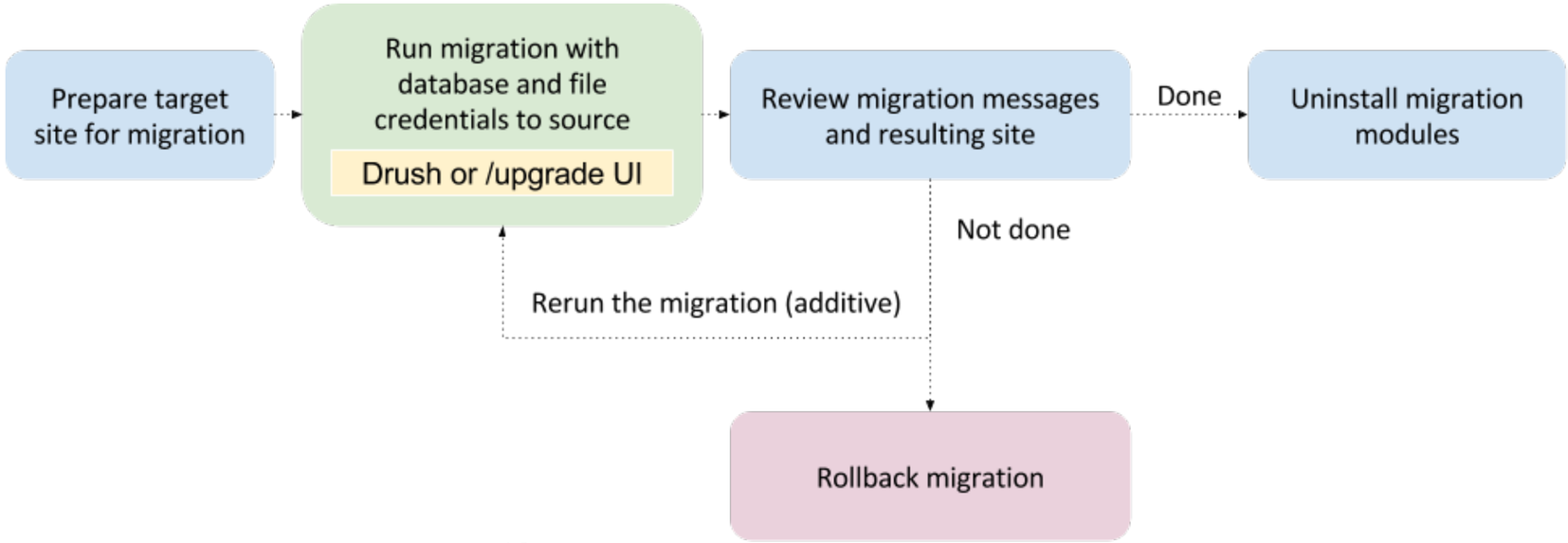
d6_file.yml

```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
```

Example Migrate YAML file

How to migrate to Drupal 8?

- **Source = Drupal 6 or Drupal 7 site**
  - **Lots of migration configuration files already defined in core**
  - **You only need custom migration config files or custom code for edge cases or missing support**
  - **Process:**



Drupal 6/7 to 8 process

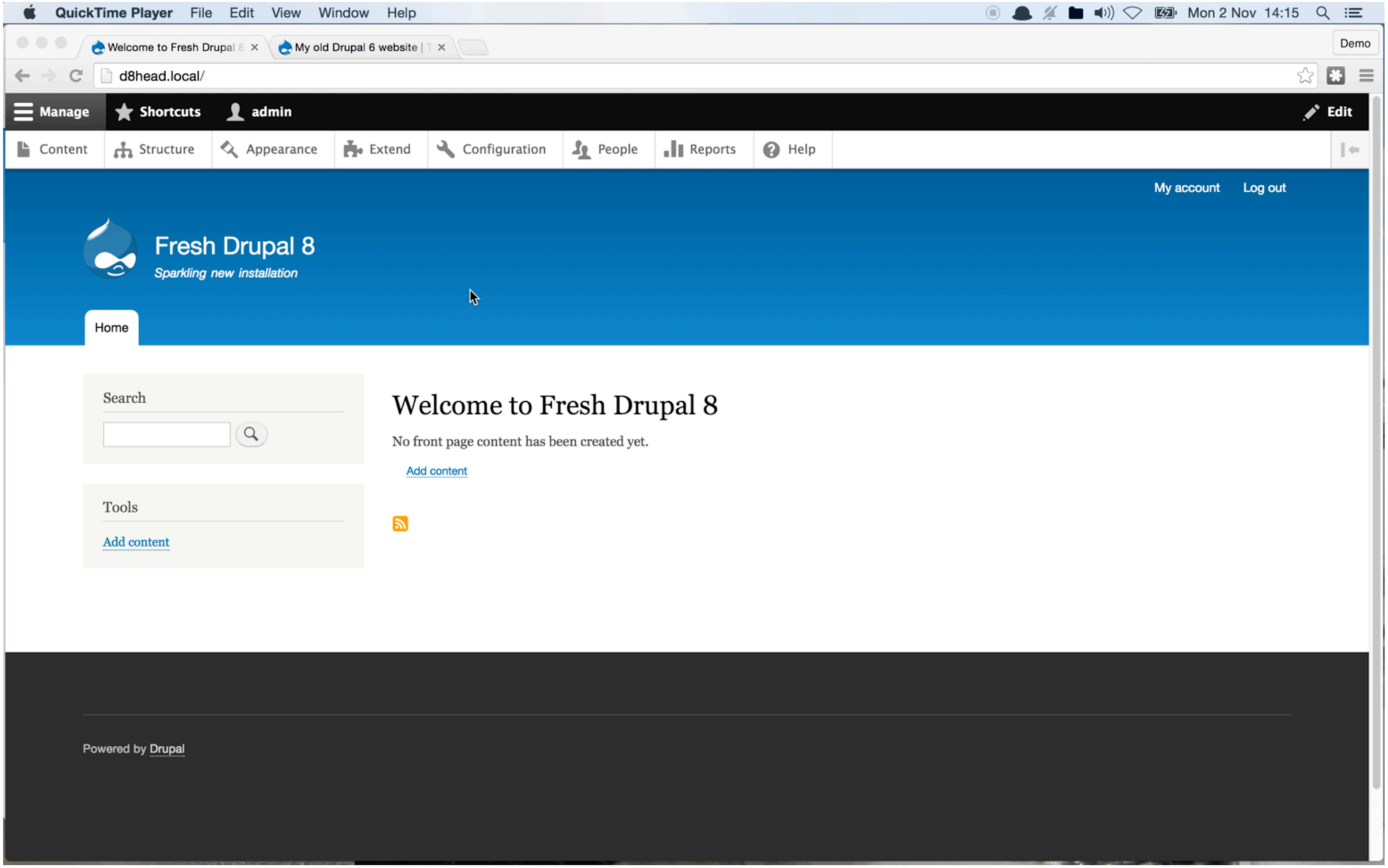
- Prepare migration
  - start from a fresh Drupal 8 installation
  - define an additional DB connection in your D8 settings.php ($databases variable) to your old Drupal 6/7 site
  - enable required modules on source & destination site
- Run migration via Drush / UI
- Review, optionally roll back and migrate again
- When done: uninstall Migrate modules

How to migrate to Drupal 8?

- Source = custom (legacy CMS, exported data, …)
  - Create custom module to configure migration
  - Optionally enable contrib modules to get source data
  - Create custom migration config YAML files in your module
  - Run migration via Drush / UI
  - Review, optionally roll back and migrate again
  - When done: uninstall Migrate modules

Demo: Drupal 6 → Drupal 8 Migration

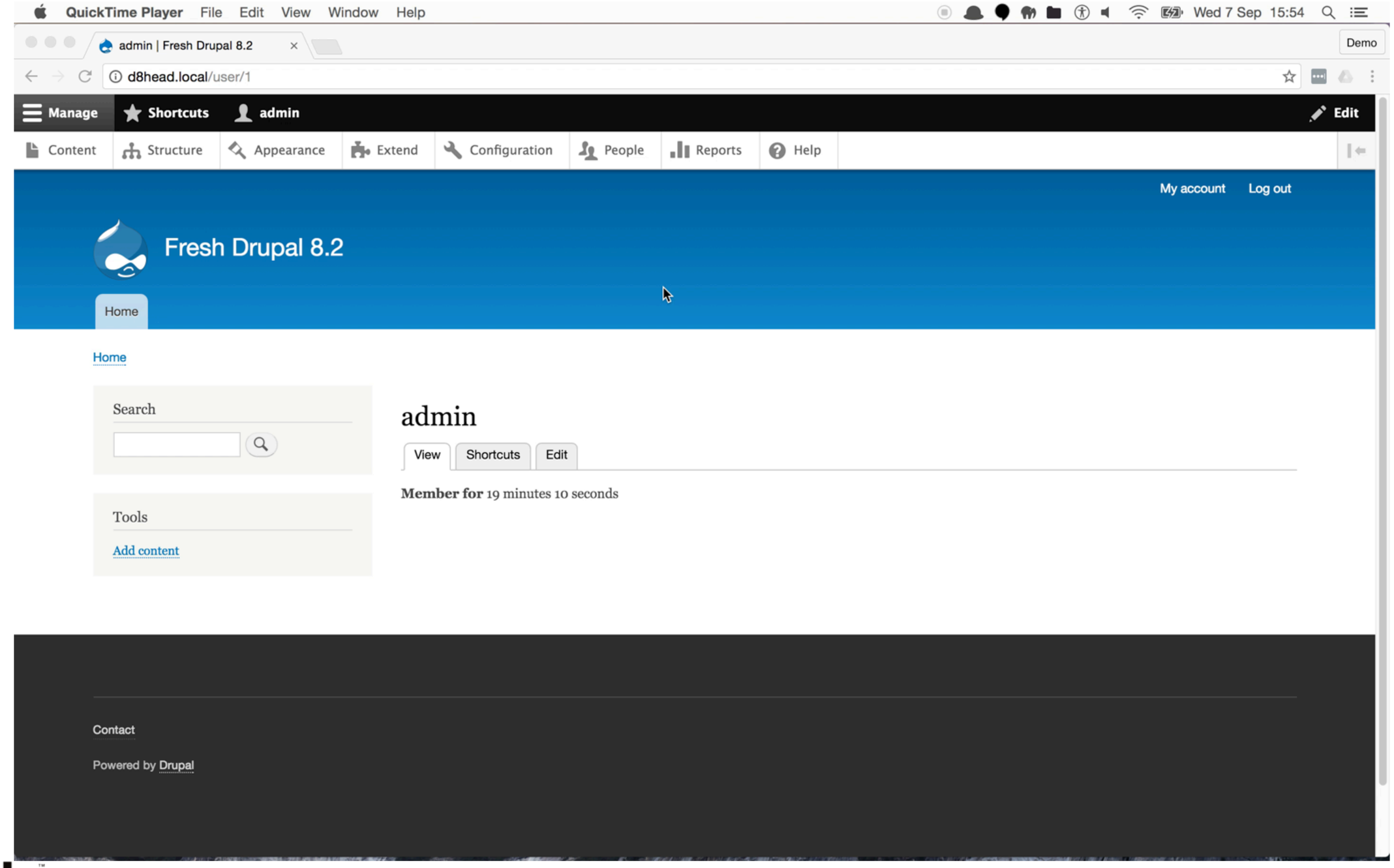
dazzle
the best the web can get

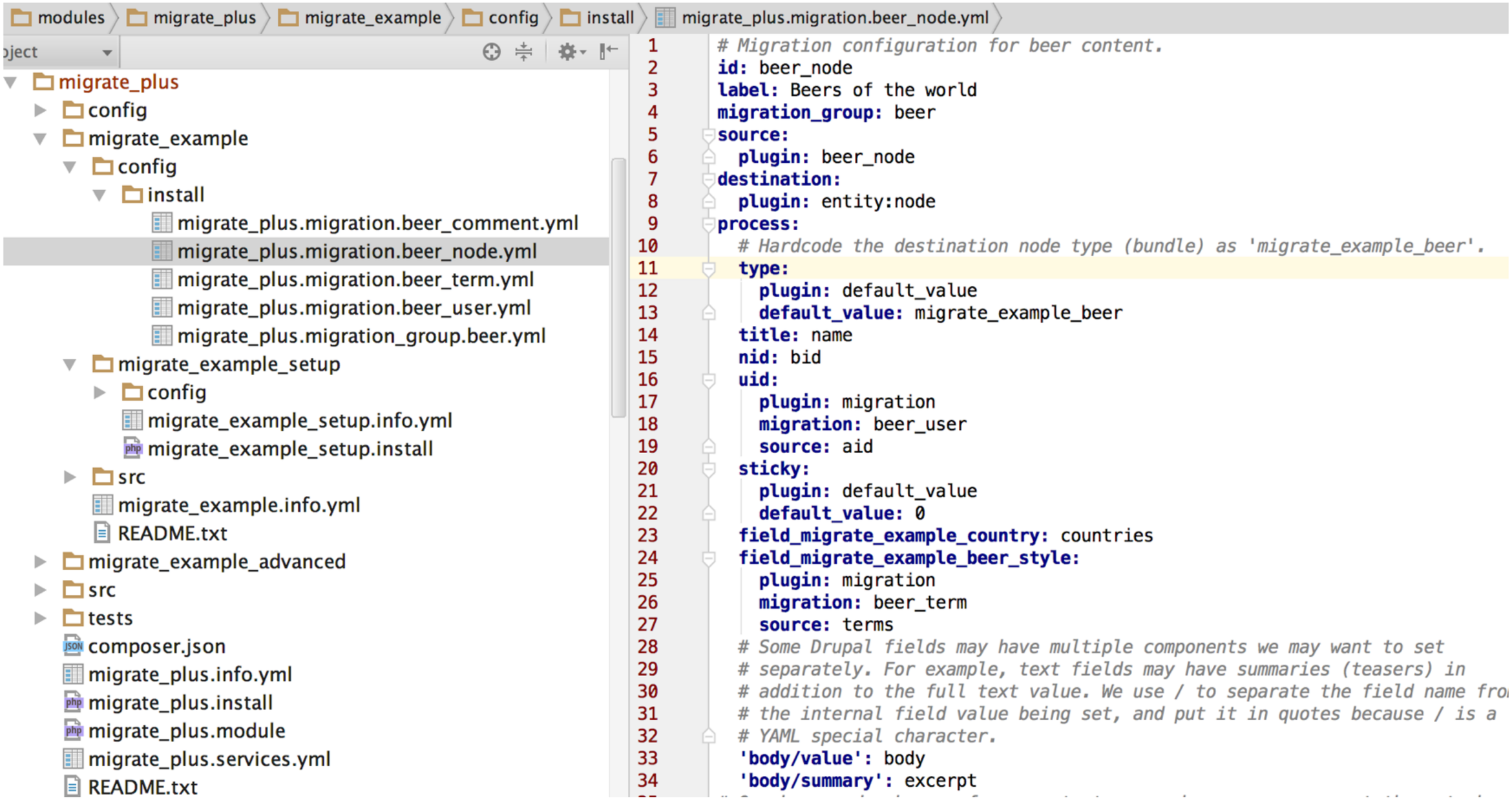

QuickTime Player File Edit View Window Help
Mon 2 Nov 14:15
Welcome to Fresh Drupal 8
My old Drupal 6 website
Demo
d8head.local/
Manage
Shortcuts
admin
Edit
Content
Structure
Appearance
Extend
Configuration
People
Reports
Help
My account
Log out
Fresh Drupal 8
Sparkling new installation
Home
Search
Tools
Add content
Welcome to Fresh Drupal 8
No front page content has been created yet.
Add content
Powered by Drupal

Demo: custom Migration

dazzle

the best the web can get


QuickTime Player File Edit View Window Help
Wed 7 Sep 15:54
admin | Fresh Drupal 8.2
Demo
d8head.local/user/1
Manage
Shortcuts
admin
Edit
Content
Structure
Appearance
Extend
Configuration
People
Reports
Help
My account
Log out
Fresh Drupal 8.2
Home
Home
Search
Tools
Add content
admin
View
Shortcuts
Edit
Member for 19 minutes 10 seconds
Contact
Powered by Drupal



Configuration files defining the custom migration

Drupal 8 contrib Migrate modules

dazzle
the best the web can get

Drupal 8 contrib

- Migrate Tools (https://www.drupal.org/project/migrate_tools)
  - General purpose Drush commands for migrations
    - `drush migrate-status / drush migrate-import / drush migrate-rollback`
    - `drush migrate-stop / drush migrate-reset-status / drush migrate-messages`
  - UI to list migrations and show messages

Drupal 8 contrib

- Drupal Upgrade (https://www.drupal.org/project/migrate_upgrade)
  - provides Drush commands for Drupal 6/7 to Drupal 8 upgrades
    - `drush migrate-upgrade / drush migrate-upgrade-rollback`
- Drush commands will maybe become part of core Drush at some point

Drupal 8 contrib

- Migrate Plus (https://www.drupal.org/project/migrate_plus)
  - Extends core functionality
    - allows grouping migrations together
    - XML & JSON parser plugins
    - extra classes for advanced usage
  - Example modules with extensive documentation

Drupal 8 contrib

- Migrate Manifest (https://www.drupal.org/project/migrate_manifest)
  - Run migrations as defined in manifest file
    - specify list of migrations & config in a YAML file
    - useful when you only want to perform selected Drupal-to-Drupal core migrations

```
- d6_file:
  source:
    conf_path: sites/assets
  destination:
    source_base_path: destination/base/path
    destination_path_property: uri
- d6_action_settings
```

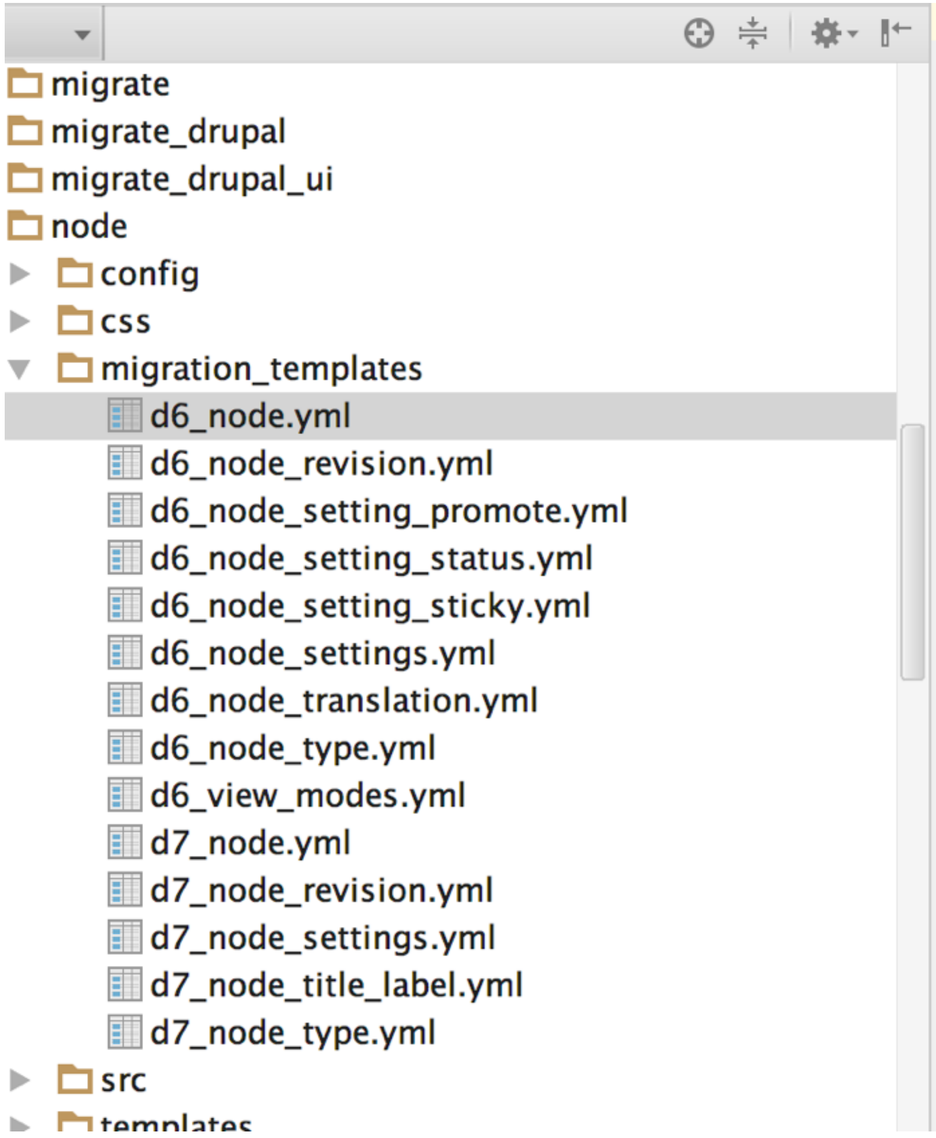
Drupal 8 contrib

- Wordpress Migrate (https://www.drupal.org/project/wordpress_migrate)
  - Migrates Wordpress blog exports (WXR) to Drupal 8
  - -dev release

Creating Migrate configuration files

Migrate configuration files

- Content & configuration migrations are defined in YAML files
- Core - for Drupal to Drupal migrations
  - Plugin manager looks for plugin config YAML files in directories
  - in migrations/*.yml
  - or migration_templates/*.yml (backwards compatibility)
- E.g. core/modules/[node]/migration_templates/[d6_node].yml



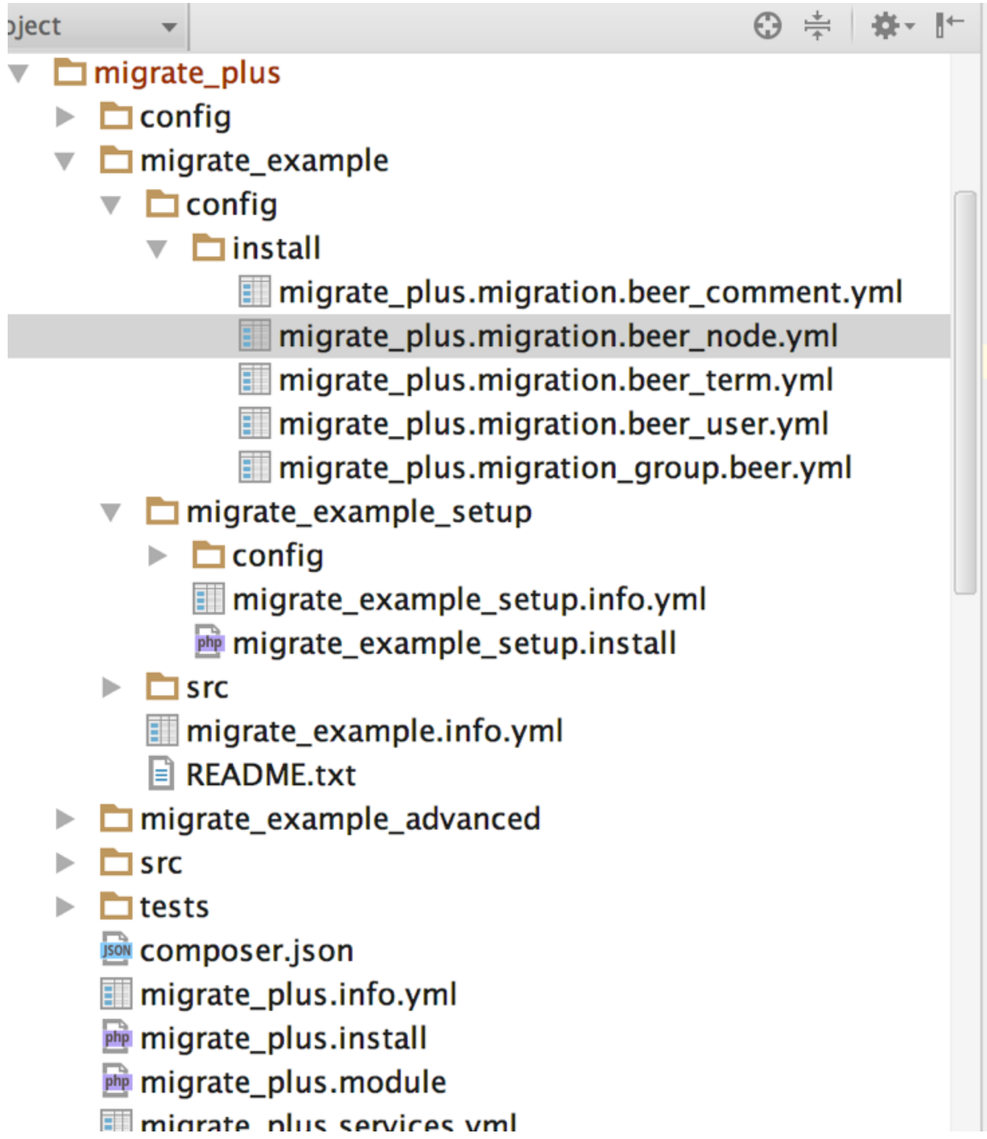

```yaml
id: d6_node
label: Nodes
migration_tags:
  - Drupal 6
deriver: Drupal\node\Plugin\migrate\D6NodeDeriver
source:
  plugin: d6_node
process:
  # In D6, nodes always have a tnid, but it's zero for untranslated nodes
  # We normalize it to equal the nid in that case.
  # @see \Drupal\node\Plugin\migrate\source\d6\Node::prepareRow().
  nid: tnid
  vid: vid
  langcode:
    plugin: default_value
    source: language
    default_value: "und"
  title: title
  uid: node_uid
  status: status
  created: created
  changed: changed
  promote: promote
  sticky: sticky
  'body/format':
    plugin: migration
    migration: d6_filter_format
    source: format
  'body/value': body
  'body/summary': teaser
```

Example: Drupal 8 core - node module migration templates

dazzle
the best the web can get

Migrate configuration files

- Content & configuration migrations are defined in YAML files
- Custom - for migrations defined in your custom module
  - Use “migrate_plus” contrib module
  - Defines Migration configuration entity concept for discovery & configuration persistence
  - Configuration entities are created at module install time, from config/install directory or generated via a UI or Drush
- E.g. modules/[foobar_migrate]/config/install/migrate_plus.migration.[foobar].yml




```yaml
# Migration configuration for beer content.
id: beer_node
label: Beers of the world
migration_group: beer
source:
  plugin: beer_node
destination:
  plugin: entity:node
process:
  # Hardcode the destination node type (bundle) as 'migrate_example_beer'.
  type:
    plugin: default_value
    default_value: migrate_example_beer
  title: name
  nid: bid
  uid:
    plugin: migration
    migration: beer_user
    source: aid
  sticky:
    plugin: default_value
    default_value: 0
  field_migrate_example_country: countries
  field_migrate_example_beer_style:
    plugin: migration
    migration: beer_term
    source: terms
  # Some Drupal fields may have multiple components we may want to set
  # separately. For example, text fields may have summaries (teasers) in
  # addition to the full text value. We use / to separate the field name fro
  # the internal field value being set, and put it in quotes because / is a
  # YAML special character.
  'body/value': body
```

Example: Drupal 8 example custom migration config

dazzle
the best the web can get

Contents of Migrate config files

- identifying data (id, label, migration tags, migration groups, …)
- source, process & destination configuration
- dependencies (required / optional)
  - Example: profile field migration would depend on user migration

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
```

Example Migrate YAML 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 contrib module)
  - XML file - use XML class (D8 contrib module)
  - etc...
- Iterates over each source row
- Returns the desired properties 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 on each source property
- Key = target property
- Value = (array of) process plugins

```
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

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

dazzle
the best the web can get

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 property, or array of properties

```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
- example:

source:

bid	module	delta	theme	status	weight	region	custom	visibility	p
21	user	online	bartik	1	0	sidebar_first	0	0	

destination:

id
bartik_user_online

```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
  - e.g: 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
```

TABLES

	source_ids_hash	sourceid1	destid1
migrate_map_d7_taxonomy_vocabulary	f2c7bc26b113ec8e7fc…	123	456

dazzle
the best the web can get

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 & custom modules can define Migration configuration files
- YAML files created in module directory:
  - *migrations/[migration_name].yml* for Drupal core & contrib (using Drupal Migrate)
  - *config/install/migrate_plus.migration.[migration_name].yml* for custom migration (using Migrate Plus)
- They describe migration pipeline to be executed:
  - From what source to get the data
  - How to process the source data
  - How to save the processed data in your Drupal 8 site

How can you help?

dazzle
the best the web can get

Getting involved

- Test the upgrade path with your site(s) and report issues:
  - https://www.drupal.org/upgrade/migrate (instructions)
  - Report in https://www.drupal.org/project/issues/drupal (select “migration system” component)
- Help improve D6 / D7 core migrations or other outstanding issues
- Add migrations to contrib D8 modules to provide upgrade path
- 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
- Blogs:
    - http://mikeryan.name/ - http://virtuoso-performance.com/
    - https://www.advomatic.com/blog/drupal-8-migrate-from-drupal-6-with-a-custom-process-plugin
- Twitter: @MigrateDrupal

Writing your own plugins

dazzle

the best the web can get

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

```
58    /**
59     * {@inheritdoc}
60     */
61    public function fields() {
62      return array(
63        'fid' => $this->t('File ID'),
64        'uid' => $this->t('The {users}.uid who added the file. If set to 0, this file was added by an anonymous user.'),
65        'filename' => $this->t('File name'),
66        'filepath' => $this->t('File path'),
67        'filemime' => $this->t('File Mime Type'),
68        'status' => $this->t('The published status of a file.'),
69        'timestamp' => $this->t('The time that the file was added.'),
70        'file_directory_path' => $this->t('The Drupal files path.'),
71        'is_public' => $this->t('TRUE if the files directory is public otherwise FALSE.'),
72      );
73    }
74    /**
75     * {@inheritdoc}
76     */
77    public function getIds() {
78      $ids['fid']['type'] = 'integer';
79      return $ids;
80    }
81
82  }
```

Writing your own plugins - Process plugin

- in [modulename]/src/Plugin/migrate/process/[name].php
- Extend ProcessPluginBase
- Implement 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()

Questions?

Sven Decabooter
@sdecabooter