

# Migración de data a Drupal con Migrate

Drupal Summit Latino 2012  
Guadalajara, México



**Diego Tejera**

Email: [dtejera@rootstack.com](mailto:dtejera@rootstack.com)

Twitter: @di3gopa



# Migración de Data

# Temario

- Métodos de Migración
- Módulo Migrate
  - Introducción
  - Construcción de un script de migración
  - Ejemplo
  - Comandos de Drush
- Preguntas

# Métodos de Migración

# Migración Manual

## Ventajas

- No se necesita mucho conocimiento para implementarlo
- Se podría migrar sin problemas de fields a fields

## Desventajas

- Muy propicio al error
- Mucho personal para meter la data
- No es posible hacer varias iteraciones

Migration Army! #fail



# Otros Métodos

- Node Export: [http://drupal.org/project/node\\_export](http://drupal.org/project/node_export)
- Feeds: <http://drupal.org/project/feeds>
- Nuestro propio script “from scratch”

# Proyecto Migrate

- Posee varios tipos de fuentes de entrada de data definidos:
  - Bases de datos, JSON, XML, CSV
- Se puede Importar a diferentes tipos de contenido
  - Usuarios, Nodos, Comentarios, Taxonomías, Archivos (Entidades del Core)
  - Es posible definir nuestros propios handlers de importación para otro tipo de entidades.
- Es posible definir nuestros propios métodos para campos custom
- Existen módulos adicionales para complementarlo
  - Migrate Extras ([http://drupal.org/project/migrate\\_extras](http://drupal.org/project/migrate_extras))
- Integración con Drush
  - Rollback, Status, importaciones

Url: <http://drupal.org/project/migrate>

# Pasos para implementar una migración

- Avisarle a migrate sobre tu módulo de migración
- Construir una clase de migración
  - Darle una descripción
  - Definir de donde viene nuestro contenido
  - Definir a que tipo de contenido o entidad estamos migrando
  - Crear el mapeo de los campos
  - Implementar métodos extras de la clase de migración para manipular el contenido que va a entrar hacia Drupal.



# Primero Paso - Creación del hook

- Implementar el hook `hook_migrate_api`
  - Proveer la versión del api que vamos a utilizar, en este momento es la 2

```
function miModulo_migrate_api() {  
    $api = array(  
        'api' => 2,  
    );  
    return $api;  
}
```

# Segundo paso - Implementar clase de migración

```
class TipoDeContenidoMigration extends Migration {  
    public function __construct() {  
        parent::__construct();  
        ...  
    }  
  
    public function prepareRow($current_row) {  
        ...  
    }  
  
    public function complete ($node, stdClass $current_row) {  
        ...  
    }  
}
```

# Segundo paso - Información sobre la migración

```
$this->team = array(  
    new MigrateTeamMember('Diego Tejera', 'dtejera@rootstack.com', t('Lead Developer')),  
);  
  
$this->issuePattern = 'http://drupal.org/node/:id:';
```

# Segundo paso - Definición de la fuente de Información

```
$this->map = new MigrateSQLMap($this->machineName,  
    array(  
        'style' => array('type' => 'varchar',  
                        'length' => 255,  
                        'not null' => TRUE,  
                        'description' => 'Topic ID',  
                    )  
    ),  
    MigrateDestinationTerm::getKeySchema()  
);  
  
$query = Database::getConnection('default', 'beer')  
->select('migrate_example_beer_topic', 'met')  
->fields('met', array('style', 'details', 'style_parent', 'region', 'hoppiness'))  
// This sort assures that parents are saved before children.  
->orderBy('style_parent', 'ASC');  
  
$this->source = new MigrateSourceSQL($query);
```

# Segundo paso - Mapeo de Información

```
$this->destination = new MigrateDestinationTerm('migrate_example_beer_styles');

$this->addFieldMapping('name', 'style');
$this->addFieldMapping('description', 'details');

$this->addFieldMapping('parent_name', 'style_parent')
    ->description(t('The incoming style_parent field is the name of the term parent'));

$this->addFieldMapping(NULL, 'hoppiness')
    ->description(t('This info will not be maintained in Drupal'))
    ->issueGroup(t('DNM'));

$this->addFieldMapping(NULL, 'region')
    ->description('Will a field be added to the vocabulary for this?')
    ->issueGroup(t('Client Issues'))
    ->issuePriority(MigrateFieldMapping::ISSUE_PRIORITY_MEDIUM)
    ->issueNumber(770064);
```

# Segundo Paso - Métodos de manipulación

## Métodos más comunes

- `prepareRow($row)`
- `prepare($entity, stdClass $row)`
- `complete($entity, stdClass $row)`

## Métodos no tan utilizados

- `preImport()`
- `postImport()`
- `preRollback()`
- `postRollback()`

# Ejemplo de método de manipulación

```
public function prepareRow($current_row) {
    $source_id = $current_row->wineid;
    $result = db_select('migrate_example_wine_vintages', 'v')
        ->fields('v', array('vintage'))
        ->condition('wineid', $source_id)
        ->execute();
    foreach ($result as $row) {
        $current_row->best_vintages[] = $row->vintage;
    }
}
```

# Interfaz de Administración

Migrate ⊕

CONTENT COMMENTS **MIGRATE**

Migrate Configure

<input type="checkbox"/>	STATUS	MIGRATION	TOTAL ROWS	IMPORTED	UNIMPORTED	MESSAGES	THROUGHPUT	LAST IMPORTED
<input type="checkbox"/>	Idle	<a href="#">BeerTerm</a>	3	0	3	0	Unknown	
<input type="checkbox"/>	Idle	<a href="#">BeerUser</a>	4	0	4	0	Unknown	
<input type="checkbox"/>	Idle	<a href="#">BeerNode</a>	3	0	3	0	Unknown	
<input type="checkbox"/>	Idle	<a href="#">BeerComment</a>	5	0	5	0	Unknown	
<input type="checkbox"/>	Idle	<a href="#">WinePrep</a>	N/A	N/A	N/A	N/A	Unknown	
<input type="checkbox"/>	Idle	<a href="#">WineBestWith</a>	3	0	3	0	Unknown	
<input type="checkbox"/>	Idle	<a href="#">WineFileBlob</a>	1	0	1	0	Unknown	



# Comandos de Drush

drush ms – Lista las clases de migración y su status

drush mi <clase> - importa la clase

drush mr <clase> - Hace Rollback de la clase

## Opciones:

- --idlist = id1, id2... Importa contenido con Ids específicos
- --item-limit=n Cantidad de Items que migrar
- --feedback="n seconds" - Muestre reporte cada n segundos
- --feedback="n items" - Muestre reporte cada n items

# Comandos de Drush

drush ms – Lista las clases de migración y su status

drush mi <clase> - importa la clase

drush mr <clase> - Hace Rollback de la clase

## Opciones:

- --idlist = id1, id2... Importa contenido con Ids específicos
- --item-limit=n Cantidad de Items que migrar
- --feedback="n seconds" - Muestre reporte cada n segundos
- --feedback="n items" - Muestre reporte cada n items

**¿Preguntas?**

# ¡Gracias!

## Referencias:

<http://drupal.org/project/migrate>

[http://drupal.org/project/migrate\\_extras](http://drupal.org/project/migrate_extras)

<http://btmash.com/article/2011-03-16/presenting-migrate-module-los-angeles-downtown-drupal-meetup>

dtejera@rootstack.com