
Data Generation in Drupal

Problems, Considerations, and Solutions

Adam Bergstein, Gerardo Gonzalez

Contents

- Problems
- Considerations
- Solutions

Problems

- Data types and structures
 - Associations to other data
 - Quantity of data (scale)
 - Data constraints (bounds)
 - Selectable data / options
 - Machine generated vs. relevant samples
 - Commands and Operations
 - General purpose reuse
 - APIs
 - Clients engagement and relevancy
-

Considerations

- Testing performance with data at scale
 - Automated testing
 - Theming and Visual Regression
 - Demos
 - Installations (profiles, fresh installs)
-

Solutions

Devel Generate

Quick and dirty
You get what you get

- Built into widely used Devel
- UI-based and Drush enabled
- Respects Drupal field data types
- Content is totally random
- No API
- Supports scale

Drupal Faker

Coding a configurable solution
Robust library of data types
Limited integration

- Faker library has robust set of data types for more discretionary generated data
 - Mappings are coded, can be more discretionary
 - Lacks Drush commands
-

Realistic Dummy Content

Perfect for known content
API-based
Recipes for creating sets of
content by use-case

- Similar to Migrate
 - Known content, not machine generated
 - Leverages file structure of a module to search for dummy content:
 - Entity, bundle, and field hierarchy
 - Populate text files of known sample content
 - Recipes in similar file structure
 - Can create sets of related data for demos and full story telling
-

Feeds

More lightweight than Migrate

Can use UI

Leverages structured content

Can use Features to produce code

Can invoke from Drush

- Good for simple data importing in the same entity
 - Falls down with related data (menus, etc)
 - Triggered Feed operations with Drush
 - Leverages known structured data formats
 - Build in a UI, export as Feature, not purely a development task
-

Migrate

Fully-baked framework
Highly extensible, coded
Command-driven (rollback)
Strong CI

- Full scaffolding for operations, parsing/translation logic, and high level of customization
 - Very heavy in code, very robust supporting infrastructure for ongoing operations
 - Often requires an existing data source, preferred structured
 - Full Drush integration
 - Commerce Kickstart uses this for demo content
-

Migrate with Generated Source Hybrid

Custom api to generate a csv data source, and an api to generate migrations, all brought together in a few drush commands.

Custom solution

Too many complex components

Hard to maintain the continuously evolving data from a in-development project

Behat

It not only generates data, it also tests stuff.

The Drupal extension supports generation for the basic types of content in Drupal

Since the data is for testing, the data is pretty sparse

To get robust data the user still has to do quite a bit of work

Gather Content

Create your content model, and a content workflow, then pull it into a Drupal site when needed.

- Popular tool for content strategists and writers
 - No code
 - Lacks drush
 - Great for creating an editorial workflow
 - Not so good for clients who are not ready to think about their content in a structured form (better to start in a simple google doc)
 - Drupal module ([github link](#))
-

—

<u>Solution</u>	<u>Code/UI</u>	<u>Random/Fixed</u>	<u>Complexity</u>	<u>Drush</u>
Devel Generate	UI	Random	Low	Yes
Drupal Faker	Code	Random, API for discretionary data types	Medium	No
Realistic Dummy Content	Code	Fixed, non conventional formats	Medium	No
Feeds	UI	Fixed, conventional formats	Low	Yes
Migrate (and hybrids)	Code	Fixed, conventional formats	High	Yes
Behat	Code	Fixed, mimic steps	High	No
Gather Content	UI	Fixed, third party integration with content tool	Low	No

Summary Matrix

Questions
