

# What's New in Drupal 8

Configuration Management (CMI)



# What We're Talking About

- ▶ What is configuration management?
- ▶ What problems does it solve?
- ▶ Changes for end-users and developers
- ▶ What you're going to need to learn



# What Is Configuration Management?

- ▶ Allows configuration to be **synchronized** between environments
- ▶ Provides a **consistent API** for developers to save and retrieve configuration

# Why Configuration Management?

- ▶ Content vs. Configuration
- ▶ Ability to **synchronize configuration** between environments
- ▶ No consistent way to **store configuration** in Drupal 7
- ▶ It's a **best practice**, but it's hard to follow

# Configuration Management

- ▶ Look to others for inspiration
- ▶ "Simple things should be simple, and complex things should be possible."

# CMI for End Users

- ▶ UI for importing, exporting, comparing, and synchronizing configuration
- ▶ Drush commands for import, export, compare
- ▶ Changes are in VCS; you have a history of changes
- ▶ Works with multi-lingual configuration
- ▶ Sites own configuration, not modules



# Synchronize

- Synchronize
- Single Import/Export
- Full Import/Export

Home » Administration » Configuration » Development

⚠ Your current configuration has changed. Changes to these configuration items will be lost on the next synchronization:

- system.site

Import configuration that is placed in your staging directory. All changes, deletions, renames, and additions are listed below.

## 1 changed

NAME	OPERATIONS
system.site	<a href="#">View differences</a>

View changes of system.site

OLD		NEW
uuid: 4fb85091-09f7-4139-a600-ed6c51931503		uuid: 4fb85091-09f7-4139-a600-ed6c51931503
- name: <b>Drupal</b>	+	name: <b>'Magical Robots'</b>
mail: admin@example.com		mail: admin@example.com
- slogan: "	+	slogan: <b>'Where lazars meet glitter'</b>
page:		page:
403: "		403: "

[Back to 'Synchronize configuration' page.](#)

# CMI for Developers

- ▶ Single, unified API for managing configuration
- ▶ Default configuration is stored as YAML files
- ▶ Simple configuration vs. Configuration Entities
- ▶ Changes are in VCS
- ▶ Configuration belongs to the project not the module



# core/modules/system/config/install/system.site.yml

```
uuid: ''
name: Magical Robots
mail: ''
slogan: 'Where Lasers Meet Glitter'
page:
  403: ''
  404: ''
  front: user/login
admin_compact_mode: false
weight_select_max: 100
langcode: en
```

# Example API Usage

```
// Load configuration.  
$config = \Drupal::config('system.performance');  
  
// Get a specific value.  
$value = $config->get('cache.page.enabled');  
  
// Set a scalar value.  
$config->set('cache.page.enabled', 1);  
  
// Save your data to the file system.  
$config->save();
```

# What Do I Need to Know?

- ▶ Memorize this: “**Export > Commit > Deploy > Import**”
- ▶ Learn the CMI API, `$config->get()`, `$config->set()`, and `$config->save()`
- ▶ YAML
- ▶ Understand the difference between simple config and configuration entities and know which to use when

# Recap

The configuration management initiative allows configuration to be **easily synchronized** between environments by providing a **unified API** that developers can use to read/write configuration, and tools for users to **import/export configuration changes** and transport them.

