Configuration System Intro

Drupal's Configuration System

- Why do we need to manage configuration?
- About Drupal's configuration system
- A new workflow: configuration synchronization

drupalize me

Why Configuration Management?

- **Problem:** To store site configuration in code required contributed modules like Features or CTools
- **Objective:** Provide a way out-of-the-box to sync and store configuration in code and in the database
- **Goal:** Make the configuration system accessible to a wider audience: developer and non-developer

About the Configuration System

- Announced as a Drupal 8 initiative in March 2011
- Known as Configuration Management Initiative (CMI) during Drupal 8's development cycle
- Provides a central way for your site and modules to store configuration data
- Accessible via Drupal's administrative interface and via the Drupal command-line tool, Drush


Development
Staging
Production

Configuration Workflow

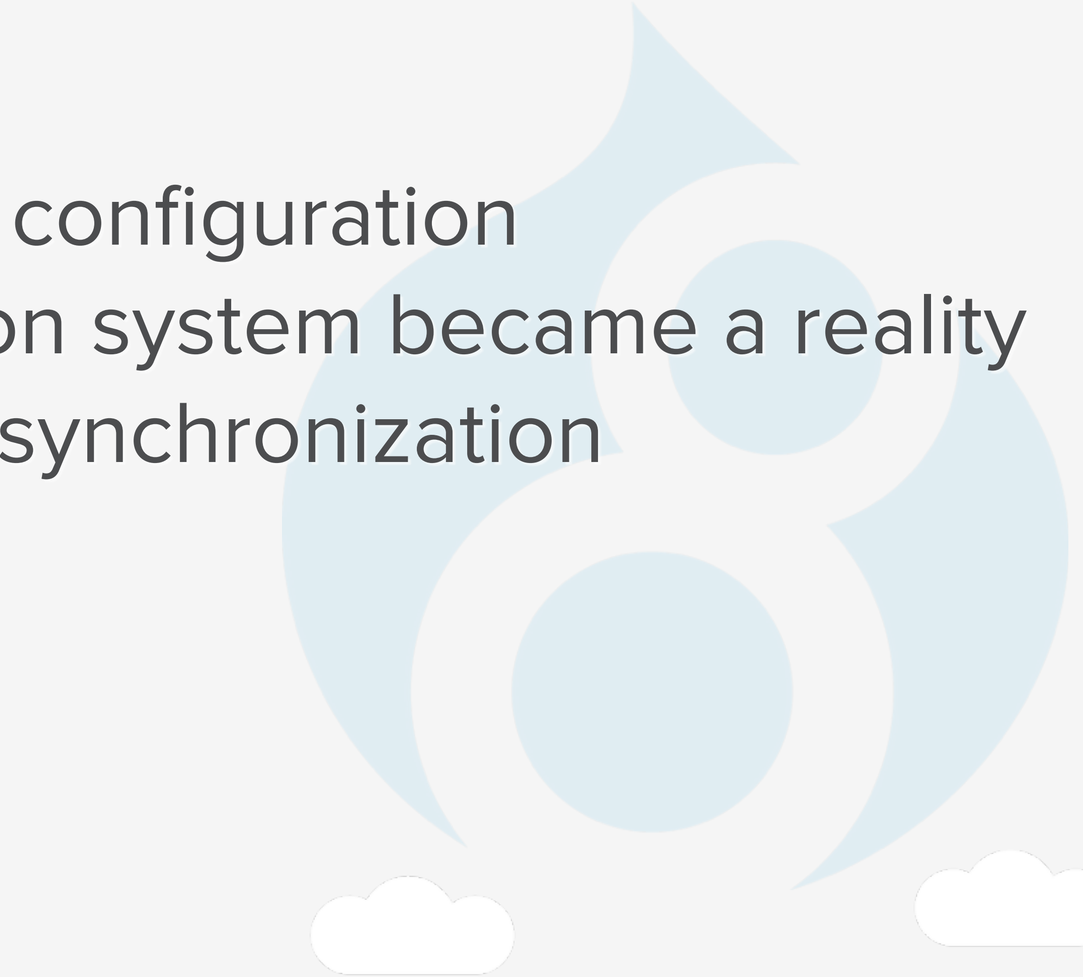
- Make configuration change
- Export configuration
- Import configuration

Configuration Synchronization

- The act of exporting or importing Drupal's configuration from one instance of the site to another
- Your site’s active configuration is stored in the database
- Exporting your configuration encapsulates site configuration in YAML files

Recap

- Why we need to manage configuration
- How Drupal's configuration system became a reality
- Introduced configuration synchronization



Cloning Drupal

drupalize me

Clone a Drupal Site

- Why clone your site?
- How to retrieve your site's UUID
- View an example directory structure
- Clone a Drupal site with Git

drupalize me

Why Clone Your Site?

- Configuration can only be imported or exported to a copy of a site or project
- Configuration system utilizes the **system.site uuid** — generated when you install Drupal 8

Retrieve Site UUID with Drush

```
$ drush cget system.site uuid
```

Example directory structure

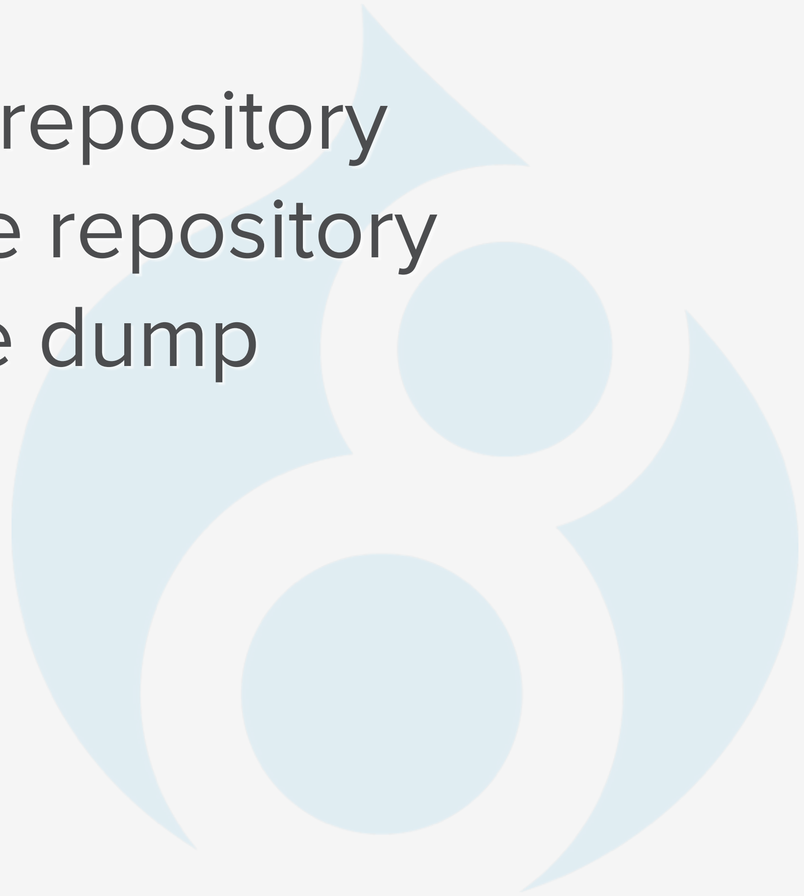
```
/project_root
/project_root/config // Not web accessible
/project_root/docroot // Drupal source, web accessible
/project_root/tests
```

Cloning your Site

- Initialize your git repository
- Create a .gitignore file from the Drupal example.gitignore
- Stage files for commit
- Add commit message
- Dump your database

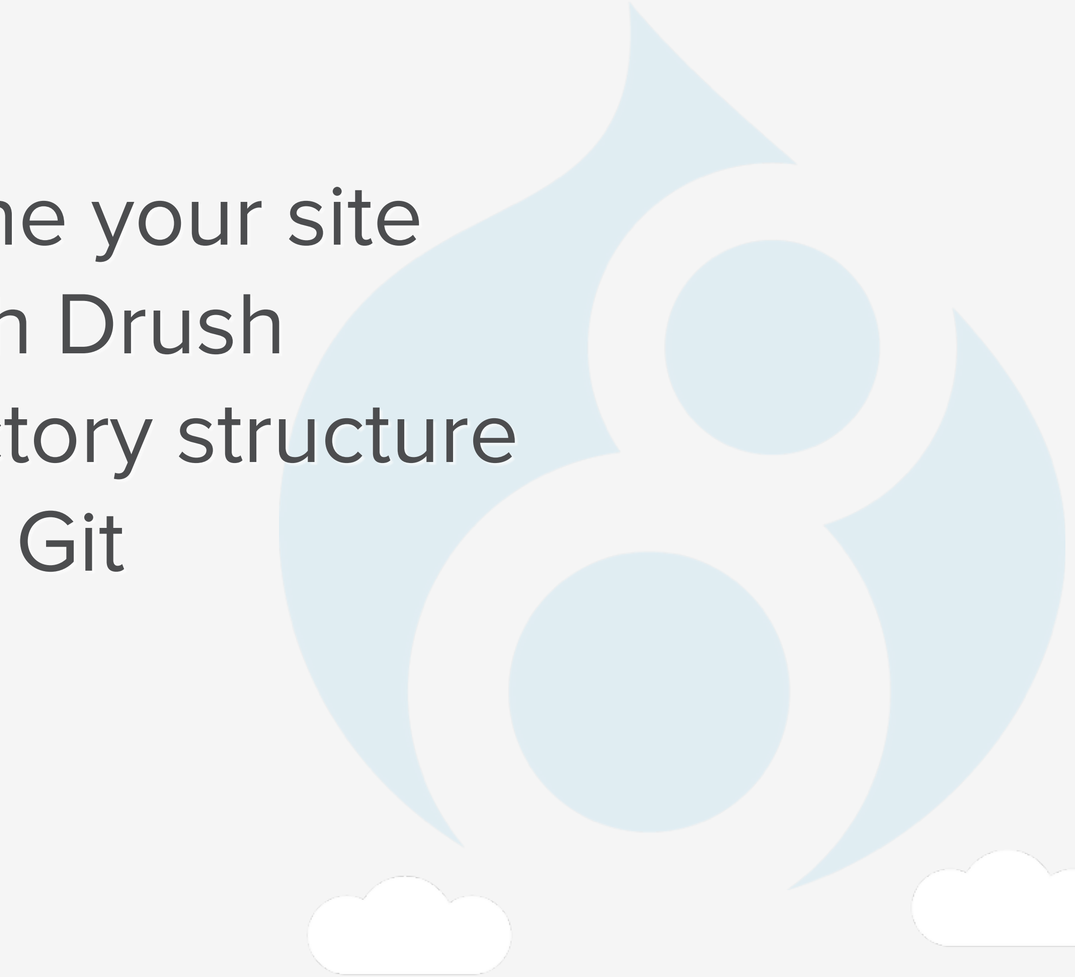
Cloning your Site

- Push to remote repository
- Pull from remote repository
- Import database dump



Recap

- Why it's necessary to clone your site
- Retrieved a site UUID with Drush
- Viewed an example directory structure
- Cloned a Drupal site with Git



Configuration Types

Configuration Data Storage

- Four **data storage types** in Drupal:
  - Content, Session, State, Configuration
- Two **configuration data storage types**:
  - Simple Configuration
  - Configuration Entities
- Configuration data stored in **database**
  - Export to **YAML** files

Data Storage Types in Drupal

Content: Information meant to be displayed on your site: articles, images, files, etc.

Data Storage Types in Drupal

Session: Information about individual users' interactions with the site, such as whether they are logged in.

Data Storage Types in Drupal

State: Information of a temporary nature about the current state of your site. For example, the time when Cron was last run.

Data Storage Types in Drupal

Configuration: Information about your site that is not content and is meant to be more permanent, such as the name of your site, the content types and views you have defined.

Simple Configuration

- Active Configuration stored in the database, and exported in YAML format
- Example: **system.site name: 'Configuration Management'**
- Stored in **system.site.yml** upon configuration export

Configuration Entities

- More complex to implement
- Requires a module
- The Configuration Entity API allows for storage of multiple sets of configuration
- Example: Views, Content Types and Image styles in core

Recap

- Four **data storage types** in Drupal:
  - Content, Session, State, Configuration
- Two **configuration data storage types**:
  - Simple Configuration
  - Configuration Entities
- Configuration data stored in **database**
  - Export to **YAML** files

Defining your Workflow

Configuration Sync Workflows

- Development to staging to production workflow
- Configuration change management with Git
- Configuration Read-Only Mode (contributed module)

A Typical Workflow

- Development to staging to production
- Clone your Site
- Make configuration changes
- Export configuration changes to YAML files
- Import configuration changes
- Configuration changes can be imported and exported between environments easily

Configuration Management with Git

- Provides fine-grained control of configuration using version control
- Simplifies the process of sharing configuration
- Provides a history of your configuration

Configuration Read-Only

- A contributed module that prevents configuration changes via the Drupal UI.
- Useful when configuration changes should only be made via configuration import, from development or staging environments

Recap

- Development to staging to production workflow
- Configuration change management with Git
- Configuration Read-Only Mode (contributed module)

Using Simple Configuration with your Module

drupalize me

Simple Configuration in a Module

- Define Simple Configuration for a module
- Access the configuration data with dependency injection
- Access the configuration data without dependency injection

Accessing configuration

```
$config = \Drupal::configFactory('module_name.settings');
```

Recap

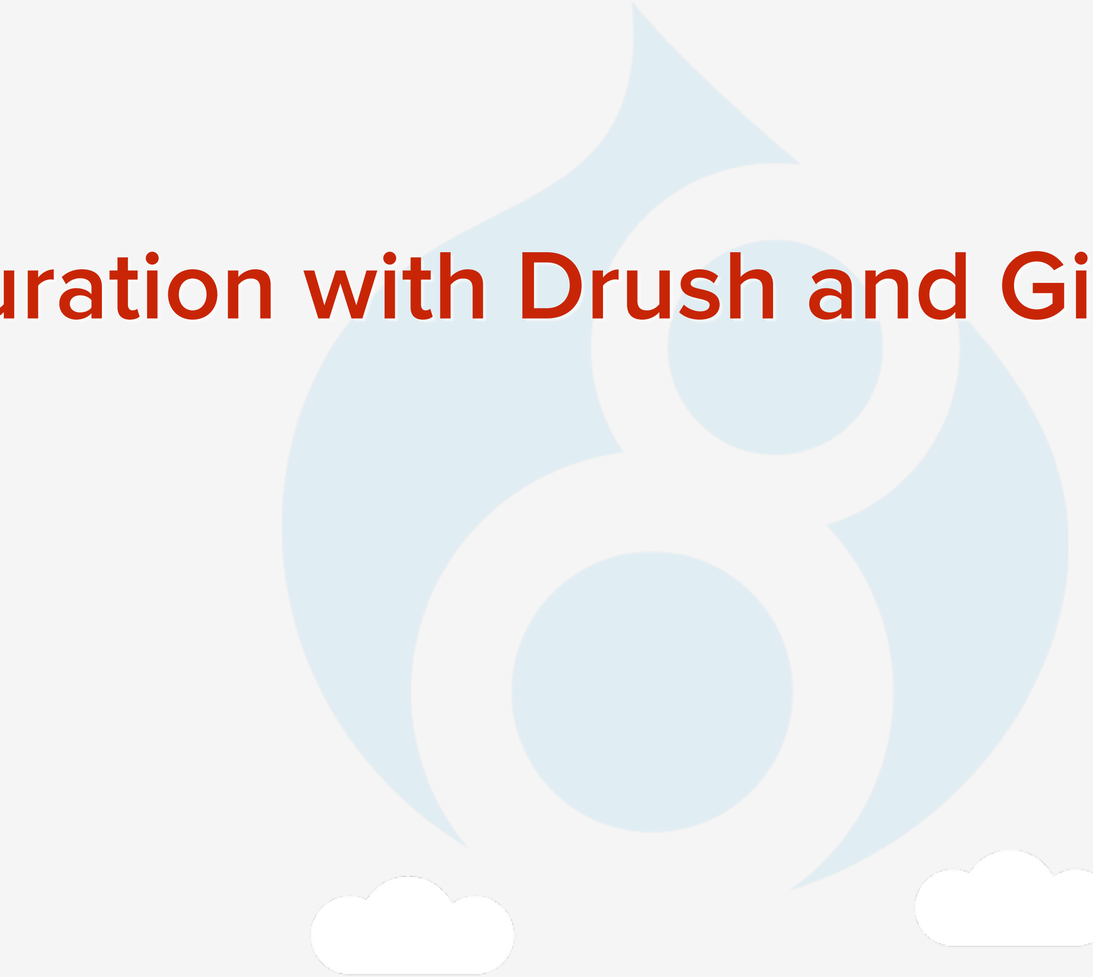
- Defined Simple Configuration for a module
- Accessed the configuration data with dependency injection
- Accessed the configuration data without dependency injection

Providing Default Configuration

Written Tutorial

drupalize me

Managing your Configuration with Drush and Git



drupalize me

Manage Configuration with Command Line Tools

- List site configuration with Drush
- Change site configuration with Drush
- Export site configuration with Drush
- Import site configuration with Drush
- Manage configuration changes with Git

Configuration Export

```
$config_directories['sync'] = '../config';
```

Recap

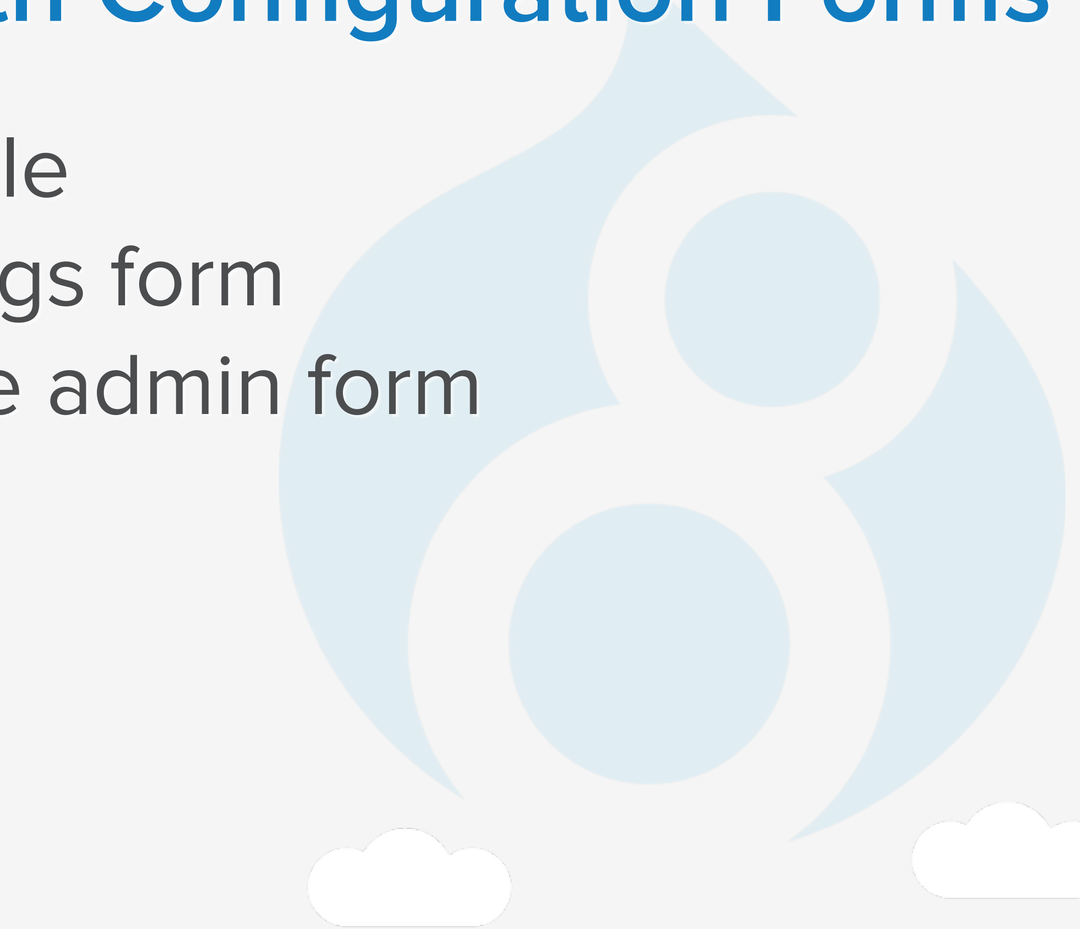
- Listed site configuration with Drush
- Changed site configuration with Drush
- Exported site configuration with Drush
- Imported site configuration with Drush
- Managed configuration changes with Git

Working with Configuration Forms

drupalize me

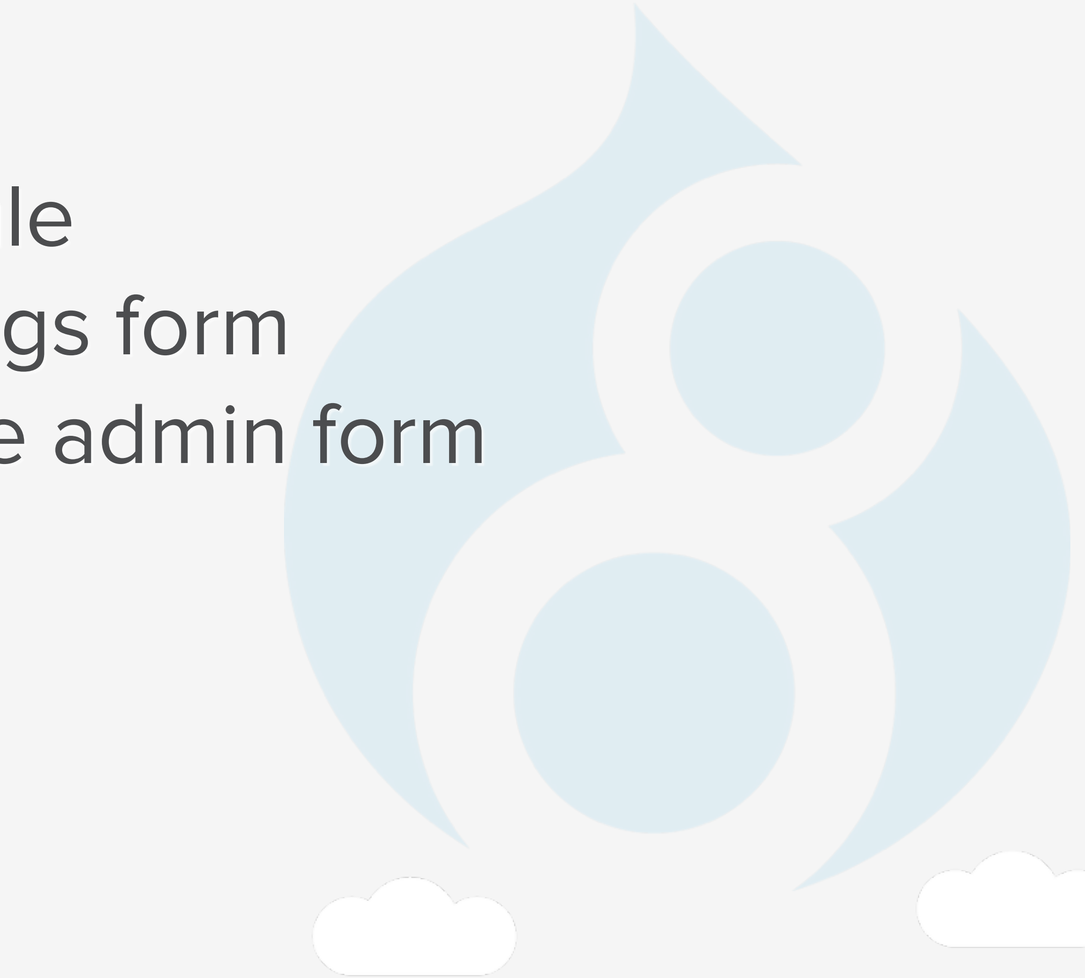
Save User Settings with Configuration Forms

- Generate a custom module
- Generate an admin settings form
- Create a menu link for the admin form



Recap

- Generate a custom module
- Generate an admin settings form
- Create a menu link for the admin form



Configuration Overrides

Written Tutorial

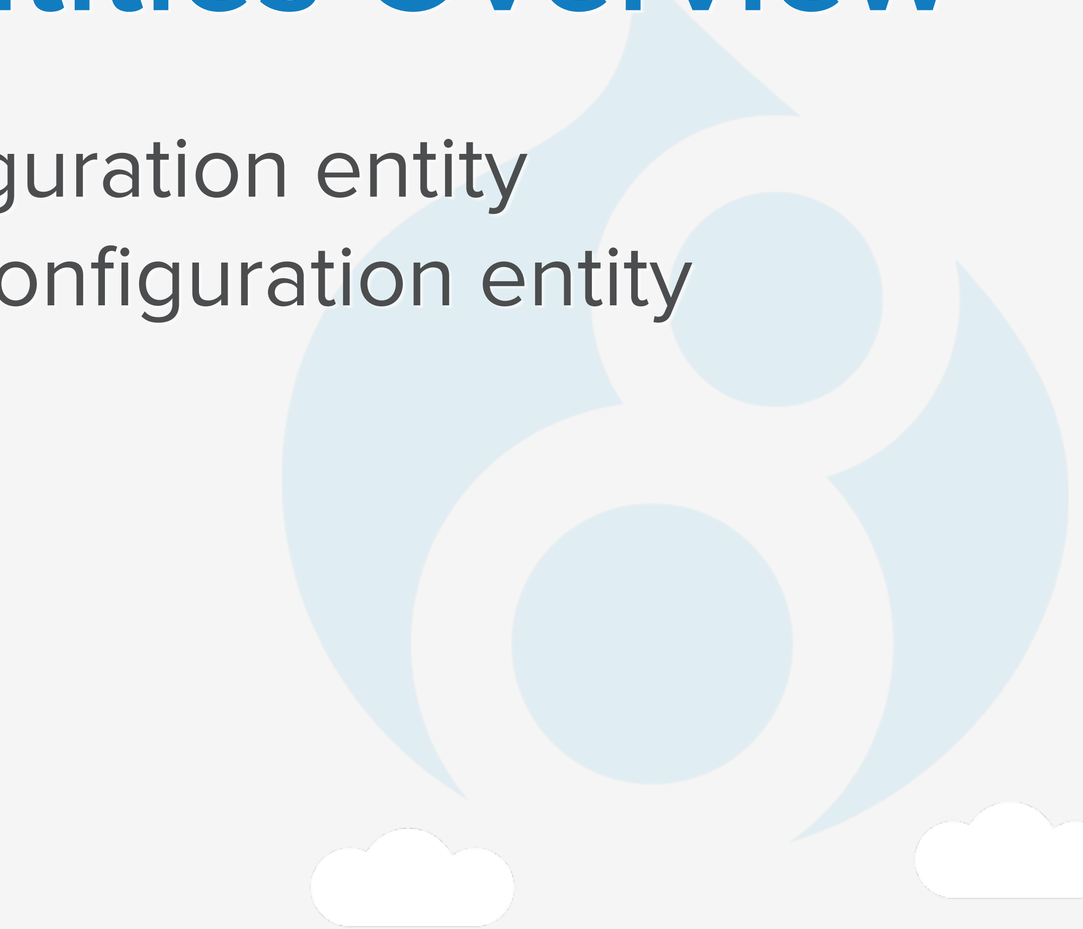
drupalize me

Creating Configuration Entities

drupalize me

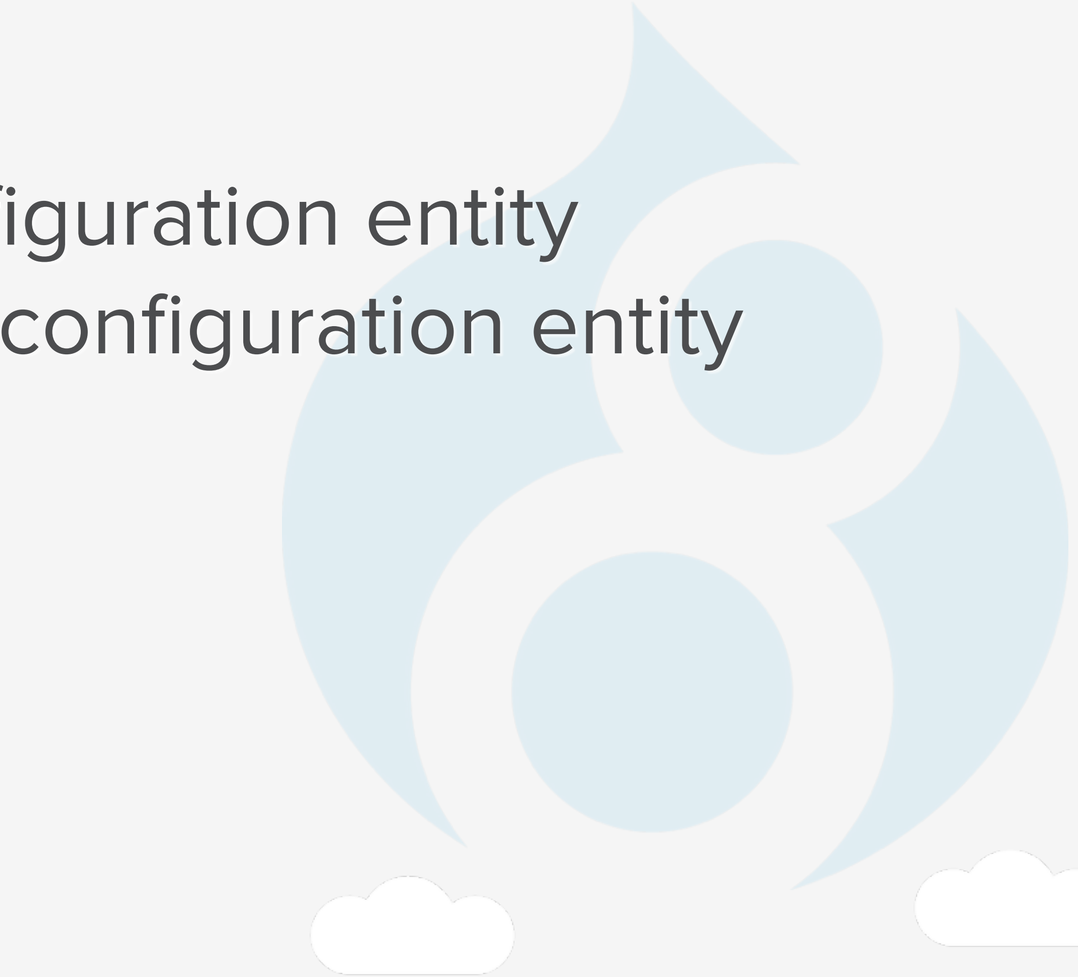
Configuration Entities Overview

- Generate a custom configuration entity
- Examine the generated configuration entity
- Create some entities



Recap

- Generated a custom configuration entity
- Examined the generated configuration entity
- Created some entities



Adding Properties to a Configuration Entity

Add Properties to a Configuration Entity

- Update configuration entity schema
- Update associated entity forms
- Update the EntityListBuilder
- Add get and set methods to implementation of ConfigEntityInterface

Recap

- Update configuration entity schema
- Update associated entity forms
- Update the EntityListBuilder
- Add get and set methods to implementation of ConfigEntityInterface

Working with Configuration Entity Data

drupalize me

Load and Save Configuration Entity Data

- Learn how to access a configuration entity via the EntityManager
- Update the settings form to load and save configuration entity data
- Save some simple configuration for a module with data from the configuration entity

Recap

- Learned how to access a configuration entity via the EntityManager
- Updated the settings form to load and save configuration entity data
- Saved some simple configuration for a module with data from the configuration entity