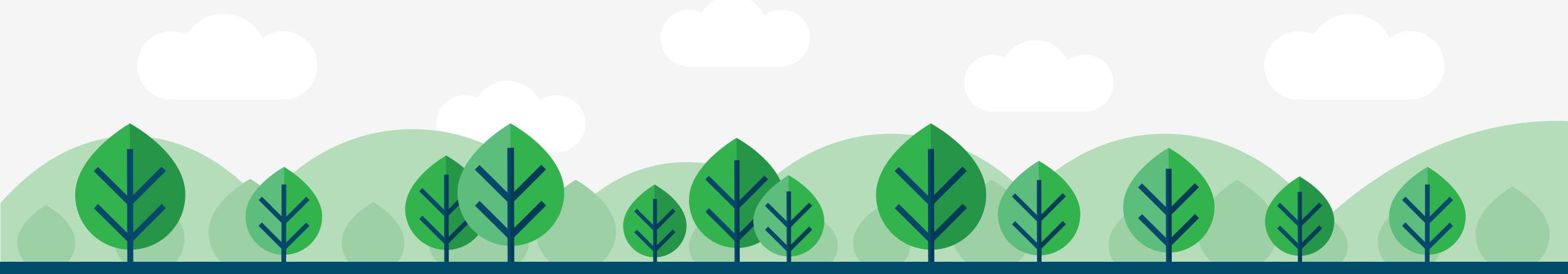




What's New in Drupal 8

Extending & Enhancing Drupal



What We're Talking About

- Where did all the hooks go?
- The role of hooks in Drupal 7
- New methods used in Drupal 8
- What you can do to prepare





Drupal 7, 6, 5 ... Use Hooks

- Specially named PHP functions
- Allows Drupal to work with older versions of PHP
- Hooks answer questions, announce events, alter
 Drupal, and extend Drupal
- Problematic because of namespace collisions, hard to discover, and they're a Drupalism



Drupal 8

- Modernizes Drupal's use of PHP
- Builds on work already done in Symfony

Methods for Extending Drupal 8:

- PluginsServices
- RoutesEntities
- Events Hooks



Drupal 8: Plugins

- Small pieces of interchangeable functionality
- Manageable through the UI
- Examples: blocks, field types
- Plugins constitute a Drupal specific, object-oriented, design pattern



Drupal 8: Routes

- Map request to a PHP callable
- Replace Drupal 7 hook_menu()
- ▶ Example: When a user visits example.com/about call MyController::aboutPage() to get the content
- Builds on Symfony's routing

Excerpt from system.routing.yml

```
system.status:
   path: '/admin/reports/status'
   defaults:
        _controller: '\Drupal\system\Controller\SystemInfoController::status'
        _title: 'Status report'
   requirements:
        _permission: 'administer site configuration'
```



Drupal 8: Events

- Listen for events, a.k.a. subscribe
- Dispatch your own events
- Builds on Symfony's event system
- ▶ Example: hook_init() is now KernelEvents::REQUEST
- More events likely in future versions

Partial example of registering an event listener:

```
public function getSubscribedEvents() {
    // Subscribe to kernel terminate with priority 100.
    $events[KernelEvents::TERMINATE][] = array('onTerminate', 100);
    // Subscribe to kernel request with default priority of 0.
    $events[KernelEvents::REQUEST][] = array('onRequest');
    return $events;
}
```



Drupal 8: Services

- Decouples reusable functionality
- ► Example: replace global drupal_mail() function with a mail sending service
- Vastly improves testability of code
- Provide new functionality that you, and others, can make use of

Example: Sending mail with mail service

```
$mail_service = \Drupal::service('plugin.manager.mail');
$result = $mail_service->mail('update', 'status_notify', $to, $langcode, $params);
```



Drupal 8: Entities

- A more complete version of the ideas started in Drupal 7
- Store content using the Entity API, not directly in the database



Drupal 8: Hooks

- Still the dominant way to enhance Drupal
- Work just like Drupal 7 hooks
- Will likely continue to be phased out in future versions of Drupal



Where Should I Start?

- Brush up on your OOP-PHP and PSR-4 knowledge
- Watch our videos about dependency injection and service containers
- Learn basic YAML syntax
- Browse https://www.drupal.org/list-changes



Recap: Ways to Enhance Drupal

- Plugins
- Routes
- Events
- Services
- Entities
- Hooks



