Paradigm Shift: Drupal 7 Hooks to Drupal 8 Plugins

Jennifer Hodgdon (jhodgdon)

Pacific Northwest Drupal Summit

Vancouver, BC

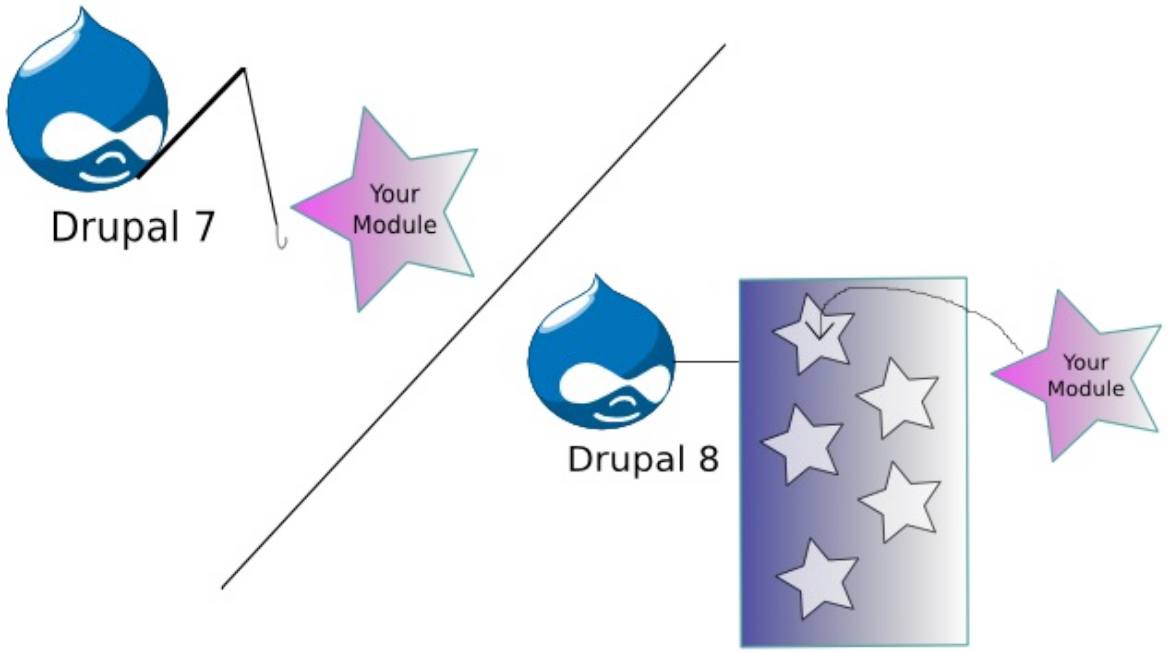
October 5-6, 2013

What do Hooks and Plugins Do?

- Allow modules to add functionality to Drupal Core or other modules.
- Allow modules to alter how Drupal Core or other modules work.
- Define things like Blocks, Text Formats, Content Types, etc. for Drupal Core or other modules to use or display.

Poplar ProductivityWare

How Hooks and Plugins Interact with Drupal



How Hooks Work (Functional Code in D7)

- Module A (example: Block module) ***defines*** a hook (implicitly), and chooses a hook name like "`hook_block_foo`". Often, there are sets of related hooks, such as `hook_block_info()`, `hook_block_view()`, `hook_block_configure()`, etc.
- Module A (hopefully) ***documents*** the hook's parameters/return value in its `block.api.php` file.
- Module B (example: My Module) ***implements*** the hook by defining a function `mymodule_block_foo()` in its `mymodule.module` file.
- Module A uses `module_invoke($module, 'block_foo')` or `module_invoke_all('block_foo')` to ***invoke*** the hook on a module or all implementing modules at appropriate times.

Example of Hooks in Code (D7)

```
// Block module (making list of module-supplied blocks):
foreach (module_implements('block_info') as $module) {
  $blocks[$module] = module_invoke($module, 'block_info');
}

// Your module (telling Block module about your blocks):
function mymodule_block_info() {
  $blocks = array();
  $blocks['first_block'] = array(
    'info' => t('First block from My Module'),
  );
  return $blocks;
}
```

Example of Hooks in Code (D7) [cont.]

```
// Block module (viewing a block):
$array = module_invoke($block->module, 'block_view', $block->delta);

// Your module (providing block output):
function mymodule_block_view($delta = '') {
  if ($delta == 'first_block') {
    return array(
      // The block's title.
      'subject' => t('First block'),
      // The block's content.
      'content' => t('Hello World'),
    );
  }
}
```

Problems with Hook Idea

- Rather ad-hoc: a bunch of related functions that may not all be defined by implementing module.
- Trying to be object-oriented without using object-oriented code (left over from the old days of PHP).
- Hooks may not be documented (function signatures, return values).
- Discovery of a hook implementation is basically `function_exists($module . '_' . $hook_name)`, so module C could implement a hook on behalf of module B, uck!

How Plugins Work (Object-Oriented Code D8)

- Module A (example: Block) defines a PHP interface, an annotation scheme (PHP class), and (hopefully) a base class to help implementation.
- Classes use namespaces and PSR-0 for auto-loading.
- Module B (example: My Module) creates a class with an annotation block, which implements the interface.
- Module A uses a Plugin Manager class to collect plugin classes, instantiate them, and call interface methods as appropriate.

Example of Plugins in Code (D8)

```
// Block module — annotation class
// core/modules/block/lib/Drupal/block/Annotation/Block.php
class Block extends Plugin {
  // Your unique plugin ID.
  public $id;
  // Administrative label of your block type.
  public $admin_label;
}

// Block module - plugin interface
// core/modules/block/lib/Drupal/block/BlockPluginInterface.php
interface BlockPluginInterface {
  // Define who can see the block.
  public function access();
  // Output the block.
  public function build();
  // etc.
}

// There is also a base class: BlockBase
```

Example of Plugins in Code (D8) [cont.]

```
// Your module – defining the block.
// File: mymodule/lib/Drupal/mymodule/Plugin/Block/MyModuleFirstBlock.php
namespace Drupal\mymodule\Plugin\Block;

use Drupal\block\BlockBase;
use Drupal\block\Annotation\Block;
use Drupal\Core\Annotation\Translation;

/**
 * Provides a sample block.
 *
 * @Block(
 *   id = "mymodule_first_block",
 *   admin_label = @Translation("First block from My Module")
 * )
 */
class MyModuleFirstBlock extends BlockBase {
  public function build() {
    return array(
      '#markup' => t('Hello World'),
    );
  }
}
```

Example of Plugins in Code (D8) [cont.]

```
// Block module: Making a list of available block plugins

// Defining the Plugin Manager pluggable service's default class
// in block.services.yml :
services:
  plugin.manager.block:
    class: Drupal\block\Plugin\Type\BlockManager

// Constructing the Block List controller
// In class  \Drupal\block\BlockListController create()/construct():
$this->blockManager = $container->get('plugin.manager.block');

// In buildForm() method:
$plugins = $this->blockManager->getDefinitions();
foreach ($plugins as $plugin_id => $plugin_definition) {
  // …
}
```