

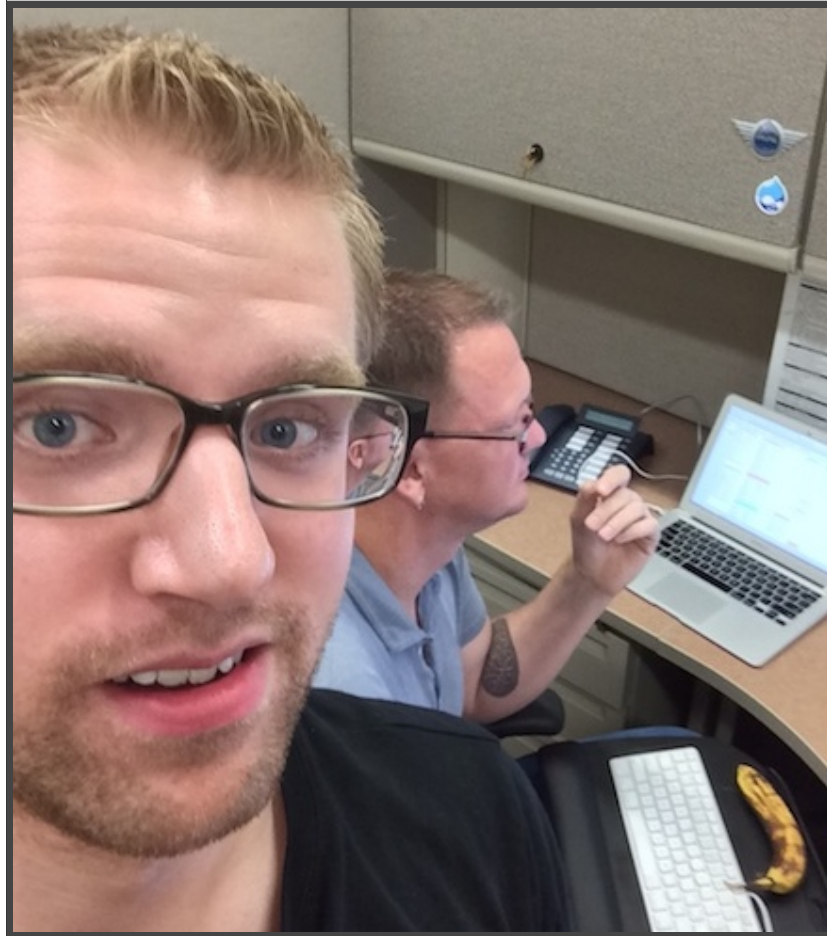
Custom Code *the* Drupal Way

Matt Mendonca

Me

- IT Specialist (i.e. Web Developer) at the [National Institute of Standards and Technology](#)
- Previously a Developer at [VOX Global \(FleishmanHillard\)](#) and [High Rock Studios](#)
- 3+ years professional experience with Drupal

A day in the life of...



(standard scale) ^

Contents

- What are modules
 - Why you should modularize your code
 - What makes up a module
- What is the hook system
 - How does that (magic) work
- Resources
- Questions

DISCLAIMER!

The Drupal 7 info is solid.
The Drupal 8 info is "solid".

New Concepts in Drupal 8

- MVC (Symfony)
 - https://symfony.com/doc/current/book/from_flat_php_to_symfony2.html
- PSR Standards (Autoloading)
- Composer / Packagist
 - Package management / Autoloader

Modules Are

- Building blocks

Modules Can Be

- Encapsulated functionality
- Frameworks
- API's

Why Modularize Your Code

- Code encapsulation
 - Enable / disable code
 - Cleanly extend core and / or contributed modules
 - Prevents hacking other's code
 - Preserves upgrade path
 - Doesn't Hack Core
- Allows you to share your code
- Allows your code to be extendable

Modules and Themes

- Make things *work* with modules
- Make things *pretty* with themes
- Developing modules allows you to keep logic out of your themes
 - Separation of concerns
 - decoupling
 - synergy
 - word
 - other words...

Module Dissected

D7

- Folder designated by the machine name of the module (/sites/all/modules/dev/my_module)*
 - Info file designated by the machine name of the module (my_module.info)
 - Module file designated by the machine name of the module (my_module.module)

* Drupal's module scanning is recursive; you can add folders within the module folder and it will find it. E.g. /sites/all/module/contrib | /sites/all/module/dev | etc.

Module Dissected

D8

- Folder designated by the machine name of the module (/modules/dev/my_module)*
 - Info YAML file designated by the machine name of the module (my_module.info.yml)
 - Routing YAML file designated by the machine name of the module (my_module.routing.yml)
 - Composer json file to list package dependencies (composer.json)
 - Module file designated by the machine name of the module (my_module.module)
 - Controller files designated by [PSR4](#) naming convention (./src/Controller/UiPageController.php)

Info File (D7 & D8)

- The .info file describes your module
- At a minimum describe name, description, package, type (D8), and core

Info File: Name

- **Name** is the user friendly name of your module

```
name = "My Module"
```

Info File: Description

- **Description** is the description of your module

```
description = "This my module."
```


Info File: Package

- **Package** is the grouping of your module
- Groups modules together on the module admin page

```
package = "User Interface"
```


Info File: Core

- **Core** is the major version of Drupal core that your module is designed for



```
core = 7.x
```

Info File: Version

- **Version** describes which version your module is at
- By convention, version of Core followed by version of the module
- *Note: this is only to be filled out when your module is not on Drupal.org*

```
version = 7.x-1.0
```

Info File: Dependencies

- **Dependencies** is an array listing all the modules that your module requires
- You do not need to list the dependencies of your dependencies

```
D7
dependencies[] = views
dependencies[] = panels

D8
dependencies:
- rules
```

Info File: Configure

- **Configure** specifies the url (D7) or route (D8) of the module's configuration page (if any)
- Adds a configure link on the module admin page

```
D7
configure = admin/config/content/my-module

D8
configure: my_module.admin_page
```

Info File: Type (D8)

- Indicates the type of the extension
- E.g. module, theme, profile

```
D8  
type: module
```

Example D7 Info File

```
name = "Matt's Module"  
description = "This is Matt's module"  
core = 7.x  
package = "Matt"  
version = 7.x-1.x
```

Example D8 Info File

```
name: "Matt's Module"  
description: "This is Matt's module"  
type: module  
core: 8.x  
package: "Matt"  
version: 1.0
```

Module File

- The module file is the file that Drupal always loads for enabled modules - **always**
- Should contain your bootstrap code / hooks (as little as possible)
- When you can, store code in .inc files

Note: D8 still uses module files for hooks, however, hooks are not used as much (routing, etc.).

(Captain) Hook

- The hooks system provides access points to Drupal's response cycle
- Used by Modules and Themes (preprocessing)
- Like a roll call

How? (D7)

- At certain points (an established hook) in Drupal's response cycle, Drupal (and module's with hooks) will run all functions that access that hook
- Further reading: <https://github.com/matt-mendonca/hooks-explained>

Modules with no hooks

- Use Case: loading a php library for use by other modules
 - Remember that Drupal will always load the contents of .module file into memory (global scope)

Resources

- [Google](#)
- [Drupal.org module development guide](#)
 - [D7 module tutorial](#)
 - [D8 module tutorial](#)
- [Drupal.org best practices](#)
- [Drupal API reference](#)
- [Drupal Answers \(Stack Exchange\)](#)
- [Stack Overflow \(questions tagged Drupal\)](#)

Questions?

[https://github.com/matt-mendonca/example-d7-
module](https://github.com/matt-mendonca/example-d7-module)

<https://github.com/matt-mendonca/Example-D8-Module>

<https://www.drupal.org/sandbox/mattm01/2487604>