# LESSONS LEARNED: DRUPAL 8 MODULE PORTING

DRUPAL NORTH

CivicActions

## LESSONS LEARNED: D8 MODULE PORTING | OUTLINE

CivicActions

# Nerdstein (Adam)

- → Associate Director of Engineering, CivicActions
- → Masters of Science, Information Systems Security
- → Drupal 8 enthusiast

CivicActions

## LESSONS LEARNED: D8 MODULE PORTING | OUTLINE

- ➔ Password Policy, Password Strength, Taxonomy Menu
- ➔ Services, Multi Select, SecurePages
- ➔ Key, Encrypt, Field Encrypt, File Encrypt
- ➔ Two Factor Auth, Security Review
- ➔ Google Summer of Code: Google Authenticator Login, PubKey Encrypt (ownCloud)
- ➔ Composer Deploy (just began working on this)
- ➔ New sandbox project: Vault

# Porting modules is an incredible way to learn Drupal 8 and give back

## You can make an incredible D8 module port if you do it right

YOU HAD ME AT
MEAT TORNADO
DIYLOL.COM

## Where do you start?

CivicActions

## LESSONS LEARNED: D8 MODULE PORTING | WHERE DO YOU START?

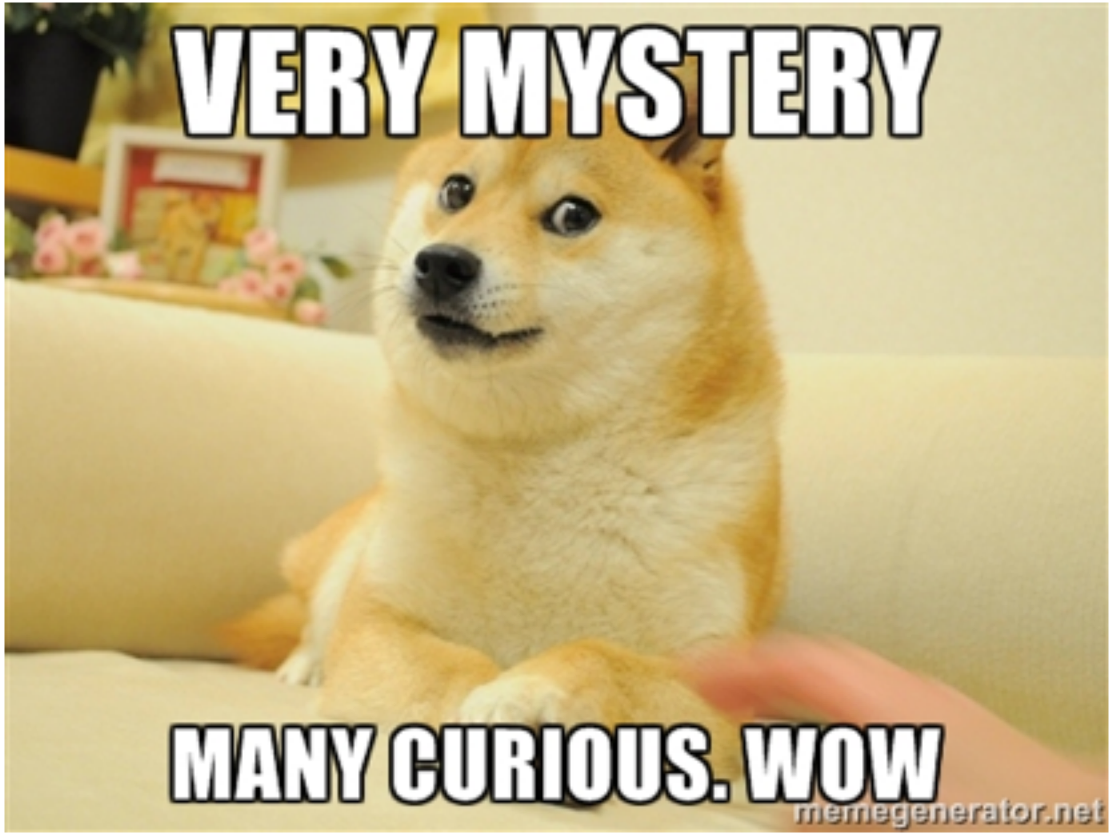
1. Start with new branch and no code
2. Get familiar D8 architecture
3. Get familiar with project needs
4. Find a mentor
5. Plan an architecture and review it
6. Don’t assume a port is needed

CivicActions

# D8 establishes new architectural paradigms you will want to use, your old code does not

CivicActions

## LESSONS LEARNED: D8 MODULE PORTING | WHERE DO YOU START?



## LESSONS LEARNED: D8 MODULE PORTING | WHERE DO YOU START?

1. Start with new branch and no code
2. **Get familiar D8 architecture**
3. Get familiar with project needs
4. Find a mentor
5. Plan an architecture and review it
6. Don’t assume a port is needed

CivicActions

## You can’t make the right choices if you don’t know what choices to make

CivicActions

## LESSONS LEARNED: D8 MODULE PORTING | WHERE DO YOU START?

1. Start with new branch and no code
2. Get familiar D8 architecture
3. **Get familiar with project needs**
4. Find a mentor
5. Plan an architecture and review it
6. Don’t assume a port is needed

## LESSONS LEARNED: D8 MODULE PORTING | WHERE DO YOU START?

1. Start with new branch and no code
2. Get familiar D8 architecture
3. Get familiar with project needs
4. **Find a mentor**
5. Plan an architecture and review it
6. Don’t assume a port is needed

## LESSONS LEARNED: D8 MODULE PORTING | WHERE DO YOU START?

1. Start with new branch and no code
2. Get familiar D8 architecture
3. Get familiar with project needs
4. Find a mentor
5. **Plan an architecture and review it**
6. Don't assume a port is needed

## LESSONS LEARNED: D8 MODULE PORTING | WHERE DO YOU START?

1. Start with new branch and no code
2. Get familiar D8 architecture
3. Get familiar with project needs
4. Find a mentor
5. Plan an architecture and review it
6. **Don't assume a port is needed**

CivicActions

## D8 has a host of new features that might make a potential port irrelevant (Menu Block, Webform)

## Symfony and Object Oriented

CivicActions

1. **Namespaces, interfaces, and conventions**
2. Containers and dependency injection
3. Controllers, services, and entity operations
4. Derivers

1. Namespaces, interfaces, and conventions
2. **Containers and dependency injection**
3. Controllers, services, and entity operations
4. Derivers

CivicActions

## LESSONS LEARNED: D8 MODULE PORTING | SYMFONY AND OBJECT ORIENTED

**The Symfony Book**

1. Namespaces, interfaces, and conventions
2. Containers and dependency injection
3. **Controllers, services, and entity operations**
4. Derivers

## LESSONS LEARNED: D8 MODULE PORTING | SYMFONY AND OBJECT ORIENTED

1. Namespaces, interfaces, and conventions
2. Containers and dependency injection
3. Controllers, services, and entity operations
4. **Derivers**

CivicActions

## Configuration Management

CivicActions

1. **No more features**
2. **YML format**
3. **Usage within projects**

## Totes excited

CivicActions

1. No more features
2. **YML format**
3. Usage within projects

CivicActions

## LESSONS LEARNED: D8 MODULE PORTING | CONFIGURATION MANAGEMENT

- → **Key values**
- → **Nested trees**
- → **Arrays and objects**

1. No more features
2. YML format
3. **Usage within projects**

## LESSONS LEARNED: D8 MODULE PORTING | CONFIGURATION MANAGEMENT

- ➔ Static module settings (info file, services file)
- ➔ Default installation configuration (settings, config/install)
- ➔ Configuration entities or the project’s entity schemas
- ➔ Any CMI moves into site wide configuration and overridable

CivicActions

## Entities

CivicActions

1. **Instances and metadata**
2. Fields and Properties
3. Content Entities
4. Configuration Entities
5. Functions/operations

CivicActions

1. Instances and metadata
2. **Fields and Properties**
3. Content Entities
4. Configuration Entities
5. Functions/operations

1. Instances and metadata
2. Fields and Properties
3. **Content Entities**
4. Configuration Entities
5. Functions/operations

## Avoid creating database schemas, use entities

## LESSONS LEARNED: D8 MODULE PORTING | ENTITIES

CivicActions

1. Instances and metadata
2. Fields and Properties
3. Content Entities
4. **Configuration Entities**
5. Functions/operations

CivicActions

## Configuration:

## To export or not to export, that is the question

CivicActions

1. Instances and metadata
2. Fields and Properties
3. Content Entities
4. Configuration Entities
5. **Functions/operations**

## Plugin Systems

## Drupal 8 is not procedural

1. **Hooks as runtime overrides**
2. Purpose of plugin system
3. Event Subscribers

## LESSONS LEARNED: D8 MODULE PORTING | PLUGIN SYSTEMS

1. Hooks as runtime overrides
2. **Purpose of plugin system**
3. Event Subscribers

1. Hooks as runtime overrides
2. Purpose of plugin system
3. **Event Subscribers**

## Recent Exploration

CivicActions

1. **Composer based workflows**
2. Caching (BigPipe)

## LESSONS LEARNED: D8 MODULE PORTING | RECENT EXPLORATION

- ➔ **Packagist**
- ➔ **Deployment workflows**
- ➔ **Dependencies in and outside of Drupal**

1. Composer based workflows
2. **Caching (BigPipe)**

## LESSONS LEARNED: D8 MODULE PORTING | RECENT EXPLORATION

CivicActions

## Lessons Learned

CivicActions

1. **Planning before doing**
2. Community visibility and mentoring
3. Use of core
4. Automated testing
5. GitHub, Travis, DrupalTI
6. PHPStorm, Drupal Console, DrupalVM

## LESSONS LEARNED: D8 MODULE PORTING | LESSONS LEARNED

CivicActions

1. Planning before doing
2. **Community visibility and mentoring**
3. Use of core
4. Automated testing
5. GitHub, Travis, DrupalTI
6. PHPStorm, Drupal Console, DrupalVM

1. Planning before doing
2. Community visibility and mentoring
3. **Use of core**
4. Automated testing
5. GitHub, Travis, DrupalTI
6. PHPStorm, Drupal Console, DrupalVM

1. Planning before doing
2. Community visibility and mentoring
3. Use of core
4. **Automated testing**
5. GitHub, Travis, DrupalTI
6. PHPStorm, Drupal Console, DrupalVM

CivicActions

1. Planning before doing
2. Community visibility and mentoring
3. Use of core
4. Automated testing
5. **GitHub, TravisCI, DrupalTI**
6. PHPStorm, Drupal Console, DrupalVM

- ➔ **Pull requests help rapidly develop**
- ➔ **DrupalTI can bootstrap automated tests in TravisCI**
- ➔ **After alpha, deprecate Github, move to [drupal.org]()**

## LESSONS LEARNED: D8 MODULE PORTING | LESSONS LEARNED

1. Planning before doing
2. Community visibility and mentoring
3. Use of core
4. Automated testing
5. GitHub, Travis, DrupalTI
6. **PHPStorm, Drupal Console, DrupalVM**

## LESSONS LEARNED: D8 MODULE PORTING | LESSONS LEARNED

- → PHPStorm fully supports OO and as code standards built in
- → PHPStorm has xDebug integration
- → Console for code generation
- → DrupalVM has fully suite of tools for a sandbox (Drush, Console, Composer, etc)

CivicActions

## LESSONS LEARNED: D8 MODULE PORTING | LESSONS LEARNED

## Thank you, Drupal North!

## Questions?

CivicActions