

Drupal Module Development

Course Summary

Description

Drupal is the most popular modular content managements system (CMS) in the PHP eco-system. Its wealth of modules gives it power and flexibility beyond any other CMS. In this course you learn how to develop modules for Drupal 8 and 9.

Topics

::Development Perspective	::Drupal Admin GUI
::What is Drupal?	::Module Development Considerations
::Drupal Layers	::PHP Knowledge Prerequisites
::Drupal Installation	::Module Development Overview
::Drupal Admin GUI	

Audience

This course is designed for beginning to intermediate level PHP developers looking to break into the Drupal module development market. It also attracts in-house corporate developer groups who have a need to create or customize Drupal modules.

Prerequisites

Students need to be familiar with the Drupal graphical admin interface at a very basic level. Students would also need a base level understanding of PHP and object oriented programming.

Duration

One day

Drupal Module Development

Course Outline

I. Module Development Quick Start

One of the best ways to get started quickly with Drupal module development is to use an appropriate development tool.

- A. Module Development Quickstart
- B. Development Tools: Drupal Console
- C. Development Tools: Drush
- D. Development Tools: DCG
- E. Module Development Quickstart
- F. Module Development Quickstart: Generator
- G. Module Development Quickstart: Module Type
- H. Module Development Quickstart: Module Type
- I. Module Development Quickstart: *.info.yml
- J. Module Development Quickstart: Other *.yml
- K. Module Development Quickstart: Done
- L. Module Development Quickstart: Followup
- M. Module Development Quickstart: Module Structure
- N. Lab: Quickstart
- O. Lab: Quickstart Continued

II. Module Infrastructure

Drupal modules require a specific infrastructure: a place to put module files, configuration files and so forth.

- A. Module Infrastructure
- B. Module Naming
- C. Module Placement
- D. Module Registration
- E. *.info.yml File
- F. *.info.yml File: Compatibility
- G. *.info.yml File Example
- H. composer.json File
- I. Example composer.json File
- J. Does the Module Appear?
- K. Lab: Infrastructure

III. Drupal Controllers

In this unit you learn about Drupal controllers and how to add one to a module.

- A. Module Controller
- B. What is a Controller?
- C. Controller Class: Naming and Placement
- D. ControllerBase Class
- E. Returning Output
- F. Controller Skeleton
- G. Grabbing Request Parameters

- H. Request Parameter Example
- I. Lab: Controllers

IV. Drupal Routing

A website visitor needs to have a way to access your new module and controller. This is accomplished through a process referred to as `_routing_`.

- A. Drupal Routing
- B. What is Routing?
- C. Routing Example
- D. Route Parameters
- E. Routing Parameters Example
- F. Routing Parameters and the Controller
- G. Route Parameter Control
- H. Advanced Routing
- I. Routing Based on HTTP Method
- J. Routing Based on HTTP Method
- K. Lab: Drupal Routing

V. Drupal Menu Links

Once you have created a controller, and have defined a route, often the next step is to add a menu link.

- A. Drupal Menu Links
- B. About Menu Links
- C. Menu Via Admin GUI
- D. Menu Via Admin GUI
- E. Menu Via Admin GUI
- F. Menu Via Admin GUI
- G. Permanent Link
- H. Permanent Link Options
- I. Permanent Link Example
- J. Permanent Link Example
- K. Lab: Menu Links

VI. Drupal Forms

One of the primary ways web developers capture information from website visitors is to present an HTML `_form_`. In addition, Drupal uses forms to configure modules.

- A. Drupal Forms
 - B. Forms Overview: Interface
 - C. Forms Overview: Base Classes
 - D. Forms Overview: Generic Structure
 - E. Forms Building: buildForm()
 - F. Forms Building: buildForm()
 - G. Forms Building: Elements
 - H. Forms Building: Element Example
 - I. Form Validation
- Form Validation: Per Form

Drupal Module Development

Course Outline (cont'd)

- J. Form Validation: Per Element
- K. Form Submission
- L. Form Routing
- M. Form Example
- N. Forms and Controllers
- O. Lab: Forms

VII. Database Access

You can access the Drupal database itself, or any other databases using the Drupal Database API.

- A. Database Access
- B. Database Configuration
- C. Replicated Database Configuration
- D. Other Database Configuration
- E. Database Connection
- F. Drupal\Core\Database\Connection
- G. Using the Connection
- H. Using the Connection Continued
- I. Dynamic Queries
- J. Dynamic Queries: select()
- K. Dynamic Queries: select()
- L. Dynamic Queries: select()
- M. Dynamic Queries: Simple Conditions
- N. Dynamic Queries: Simple Conditions
- O. Dynamic Queries: Complex Conditions
- P. Inserting Data
- Q. Dynamic Queries: update()
- R. Dynamic Queries: update()
- S. Dynamic Queries: delete()
- T. Lab: Database

VIII. Custom Blocks

Blocks are a convenient way of encapsulating functionality that can later easily be managed via the Drupal Admin GUI.

- A. Custom Blocks
- B. What is a Drupal Block?
- C. Creating a Custom Block
- D. Custom Block Placement 1
- E. Custom Block Placement 2
- F. Custom Block Placement 3
- G. Custom Block Placement 4
- H. Custom Block Placement 5
- I. Block Form
- J. LoginBlock Class
- K. LoginBlock Class Continued
- L. Login Form Block Output
- M. Lab: Custom Blocks

IX. Module Configuration

Module configuration in Drupal can take two forms: static default configuration hard coded into a configuration file, or dynamic configuration accessible via the admin GUI.

- A. Module Configuration
- B. Default Configuration
- C. Default Configuration Usage
- D. Default Configuration Example
- E. Dynamic Configuration
- F. Building the Settings Form
- G. Creating the Settings Form with Drush
- H. Settings Form Route
- I. Settings Form Menu Link
- J. Settings Form Class
- K. Settings Form Class: buildForm()
- L. Settings Form Class: validateForm()
- M. Settings Form Class: submitForm()
- N. Settings Form Placement
- O. Settings Form Display
- P. Lab: Module Configuration