

Curso Mastering Symfony

Descripción: PHP introdujo la Programación Orientada a Objetos desde su versión 5, hoy en día su versión 7 soporta nuevas características pero los frameworks como Symfony enriquecen el lenguaje con un amplio conjunto de componentes de gran reutilización, ideal para grandes proyectos.

Audiencia: Desarrolladores de PHP.

Prerrequisitos: Conocimientos del lenguaje PHP y programación orientada a Objetos.

Duración: 30 horas.

Contenido

Chapter 1. Installing and Configuring Symfony

- Why Symfony?
 - Influenced by Symfony
 - How bright is the future?
- Installation
- Composer and Packagist
 - Installing Symfony via Composer
 - The road map
 - Checking the installation

Chapter 2. The Request and Response Life Cycle

- The big picture
- Anatomy of a bundle
 - Generating a new bundle
 - Best practices
- Custom bundles versus AppBundle
- Creating templates with TWIG
 - Controller/View interaction
 - Conditional templates
 - Make it dynamic
 - Database configurations
 - Generating an entity
 - Installing bundles created by others
- Creating data fixtures
 - Loading data fixtures
- Dynamic templates and controllers
- The big picture with MVC

Chapter 3. Setting Up the Environment

- The importance of Continuous Integration
- Amazon Web Services
- Elastic Compute Cloud
 - Creating a new instance
- Setting up the server
 - Installing Apache
 - Installing MySQL and PHP
 - Installing Jenkins
- Setting up security and installing plugins in Jenkins
- Simple Email Service
 - Configuring Jenkins
- Installing PHP tools
 - Sniff Symfony codes
- Orchestrating the build process
- Creating a new job in Jenkins
- Running the first build
 - How does GitHub alert Jenkins about new pushes?
- Do I need CI?

Chapter 4. Using Behavior-Driven Development in Symfony

- Getting started with BDD
 - Is BDD a replacement for TDD?
- What is Behat?
 - Installing and configuring Behat
 - The features we need for this project
 - More information about the project
 - Gherkin – the common language
- Writing a scenario for the about page
 - Seeing it in action
 - Headless versus zombie
 - Using the Selenium2 controller for automated tests
- The about page does not follow BDD
 - A scenario to show the user's details
 - Implementing the user's details scenario
 - Testing the scenario
- More about the acceptance test flow in Mink
- Defining and prioritizing features
- Codeception – the BDD style testing framework
 - Installing the Codeception framework
 - Bootstrapping Codeception
 - Test suits
 - The testers
 - Adding sample tests
 - Running the tests

Chapter 5. Business Logic

- Choosing between creating a Model or entity
- So where does the business logic live?
- Reviewing the facts and building entity relationships
 - Creating ERDs using MySQL Workbench
 - Adding a new entity
 - Adding a new relationship
 - Creating actual tables from a diagram
 - Generating entities
 - Data fixtures
- Some business logic features and scenarios
- TDD and BDD with Codeception
 - Step one – creating a functional test
 - Developing the missing code
 - Step two – creating the unit tests
 - Setting up the database for a test environment in the right way
 - Dropping and recreating the database for each test
 - Creating unit tests
 - Writing the code to pass the test
 - Running functional and unit tests
 - Step three – creating an acceptance test

Chapter 6. Dashboard and Security

- How security is organized in Symfony?
 - Authentication
 - Authorization
- Handling users with FOSUserBundle
 - Security settings
 - Adding the required configurations to FOSUserBundle
 - Adding routes
 - Updating the tables
 - A simple road test
- Generating automated data fixtures
 - Introducing AliceBundle
 - Creating data fixtures with Alice
 - Relationship with Alice
 - Setting up the login redirection
 - Creating tests for the new controller
 - Creating the Dashboard Controller
 - Securing the dashboard
- The Sonata project
 - Installing and configuring Sonata bundle
 - Adding contents to the dashboard
 - Creating admin feature for entities with relations
- Integrating FOSUserBundle into the admin area
 - Installing SonataUserBundle
 - SonataUserBundle configuration
 - Updating the routes
 - Setting the security
 - Checking the installation
 - Putting SonataUserBundle in charge
- User dashboard
- Generating CRUD
 - Modifying the forms

Chapter 7. The Presentation Layer

- How assets are organized
- Asset management
- How templates are organized
- Let's mold the clay
- To navigate or not to navigate
- What is Bootstrap?
- MopaBootstrapBundle
 - Bootstrap configuration
- Creating your first menu
 - Rendering the menu in a template
- The Dashboard template
- Overriding templates
 - Profile-related templates
- Changing the backend logo

Chapter 8. Project Review

- The dashboard's contents
 - Visual blocks that provide statistics about tasks
 - A feature file for the finished tasks block
 - Implementing the finished tasks block
 - Implementing the dashboard controller
- Uploading files with SonataMediaBundle
 - Adding an attachment feature to the Task entity
- Team and team members
 - The Team entity
- Adding a notification system
 - Adding time tracking properties
- The notification business logic
 - Events, event dispatchers, and event listeners
 - The Notifier event listener

Chapter 9. Services and Service Containers

- How to create a service
- How are services beneficial to our projects?
- How to call a service
- How to configure a service
- Why is it called a Dependency Injection Container?
- Why didn't we import services inside the bundle?
- How to create and load services via autowiring
- Organizing services with tags

Chapter 10. Custom User Commands

- Creating and registering commands
- Creating commands for tasks
 - The configuration part
 - The execution part
- Adding interactivity to commands
- Console helpers

Chapter 11. More about Dev, Test and Prod Environments

- Why do we need different environments?
 - The environment configuration file
- Processing configuration files
- Creating a new environment
 - The config file
 - The front controller

Chapter 12. Caching in Symfony

- Definition of a cache
 - Characteristics of a good cache
- Caches in a Symfony project
- Key players in the HTTP cache header
- Using the Symfony reverse proxy cache
 - Set expiration for dashboard page
- Validation strategy
- How to mix expiration and validation strategies
 - Doctrine cache
- Putting it all together
- ESI for selective caching
- Sophisticated bundles

Forma de pago: Este pago debe realizarse inicio en una sola exhibición dos semanas antes de la fecha de inicio de la siguiente manera:

- ❖ Depósito en Banamex cuenta 4923239 Sucursal 575 a nombre de Desarrollo y Capacitación en Internet, S. A. de C. V. (CLABE en caso de transferencia electrónica vía Internet 002180057549232394)



[Informes](#)

DCInternet