

## SENSIOLABS

92-98 Boulevard Victor Hugo

92115 CLICHY Cedex

Phone : +33(0) 140 998 109

Fax: +33(0) 140 998 230

SIRET Number : 752 893 842 00019

Web: <http://training.sensiolabs.com>

**Sensio**Labs



## WEB DEVELOPMENT WITH SYMFONY2

### **Goals:**

In four days, discover the essential of the Symfony framework with the best experts in the world. You will learn how to master the MVC design pattern, forms management, authentication principles and of course the best practices with unit testing and HTTP caching.

**Duration:** 4 days - 28 hours

**Price:** €1,990.00 / person

**Code:** SF2C4

**Public:** PHP web Developers

**Requirements:** Mastering OOP in PHP

### **OBJECT ORIENTED PROGRAMMING (ON DEMAND)**

- Classes and objects
- Properties and methods
- Constructor and destructor methods
- Class inheritance
- Attributes and methods scopes (visibility)
- Error handling with exceptions
- Namespaces

### **GENERAL INTRODUCTION TO THE SYMFONY FRAMEWORK AND ITS TOOLKIT**

- Introducing the Symfony framework and its ecosystem
- Installing the Symfony « Standard Edition Distribution »
- Discovering the Symfony project architecture (application, bundles, libraries, bundles etc.)
- Understanding the HTTP request processing workflow
- Understanding the concept of environment (development, production and testing)
- Introducing the coding and files naming conventions
- Discovering the concept of a « Bundle »

SensioLabs

92 / 98 bld Victor Hugo, 92115 Clichy Cedex, France

+33 (0)140 99 82 05 - <http://sensiolabs.com>

## DEBUGGING THE CODE

- Getting help from the « Web Debug Toolbar »
- Introspecting the recorded logs
- Profiling HTTP requests thanks to the « Web Profiler » application
- Spotting problems thanks to explicit errors and exceptions stack traces

## CONFIGURING THE APPLICATION

- Comparing supported built-in configuration: XML, YAML and PHP
- Configuring the application thanks to global settings
- Configuration application's routes and URLs
- Overriding the configuration defined by a built-in or third-party « bundle »

## GENERATING PAGES FOR THE END USER

- Introducing the Twig templates rendering engine
- Comparing Twig and raw PHP templating strategies
- Accessing global variables from Twig templates (request, session, user)
- Using tags, filters and function to generate and format contents in a Twig template
- Making templates truly extensible thanks to the template inheritance feature
- Splitting templates into small templates chunks to avoid code duplications
- Rendering a Symfony controller from within a Twig template
- Generating URLs based on the application's routes configuration
- Rendering Symfony forms with custom Twig view helpers

## DESIGNING CONTROLLERS AND ANALYZING THE REQUEST

- Designing a controller class with its action methods
- Mapping URL patterns to Symfony controllers thanks to annotations configuration
- Generating a raw response or a response based on a Twig template
- Accessing client's information from the Request object
- Reading and writing session data
- Reading and writing cookies' data
- Triggering a 404 error page when the resource is not found
- Triggering an internal redirect to another controller to avoid code duplication
- Redirecting the user to another page

## INTERACTING WITH THE END USER THANKS TO FORMS

- Discovering the « Form » component
- Designing and processing simple web forms
- Adding validation constraints on submitted form fields' data
- Prototyping the form rendering thanks to Twig view helpers
- Gathering the form submitted data and processing them
- Customizing form rendering with built-in and custom Twig form themes
- Configuring form field types as services
- Applying built-in validation constraint rules on submitted data
- Applying custom validation constraint rules on submitted data
- Contextualizing the data validation thanks to validation groups
- Sending the form validated and filtered data by e-mail to some recipient

## INTERNATIONALIZING AND LOCALIZING THE USER INTERFACE

- Discovering the « Translation » component
- Changing the default user's locale
- Defining custom abstract translation keys in Twig templates
- Generating XLIFF translation catalogues skeletons
- Translating abstract translation keys in XLIFF catalogues
- Dealing with dynamic translations including pluralized sentences

## DEPENDENCY INJECTION AND SERVICE CONTAINER

- Understanding the dependency injection principle
- Getting started with the Symfony service container
- Listing all built-in Symfony services thanks to the command line tools
- Registering new custom business services in the dependency injection container
- Accessing a registered service from the service container
- Creating and accessing global configuration parameters from the container

## CONTRACTING A QUALITY INSURANCE THANKS TO AUTOMATED TESTS

- Discovering the tests automation framework PHPUnit
- Understanding the goals of automating tests (unit and functional tests)
- Configuring an automated tests suite
- Designing and executing a unit tests suite
- Designing and executing a functional tests suite
- Generating code coverage reports

## RESTRICTING ACCESS TO SECURED PAGES OF THE APPLICATION

- Getting started with authentication and authorization principles
- Securing some application's pages thanks to a form based authentication strategy
- Setting up an access control policy based on user's roles
- Testing user's permissions from a controller and a Twig template
- Registering custom authorization rules thanks to the security « voters »

## IMPROVING PERFORMANCES WITH HTTP CACHING AND ESI

- Getting started with the HTTP caching workflows
- Discovering the different kind of cache systems
- Comparing the expiration and validation cache strategies
- Applying expiration and validation cache constraints to generated web pages
- Caching page fragments thanks to the ESI technology (« Edge Side Includes »)
- Installing and configuring the Symfony's built-in reverse-proxy cache

## INTERACTING WITH A RELATIONAL DATABASE SYSTEM WITH DOCTRINE

- Discovering the basics of the Doctrine ORM
- Understanding the concepts of « entities » and « repositories »
- Automating PHP classes and SQL queries code generation with Doctrine
- Persisting data to the database thanks to the Doctrine ORM entity manager
- Fetching data back from the database thanks to Doctrine
- Building and executing custom SQL queries thanks to the Doctrine Query Builder object