

SENSIOLABS

92-98 Boulevard Victor Hugo
92115 CLICHY Cedex
Tel: +33 1 40 99 82 05
Fax: +33 1 40 99 82 30
Web: <http://training.sensiolabs.com>

SensioLabs



EXTENDING & HACKING SYMFONY2

Goals:

In two days, you will learn how to become a Symfony framework expert by tackling advanced techniques like audit, code optimization, tools customization and framework extension.

Duration: 2 days - 14 hours

Price: €2,100.00 / person

Code: SF2C3

Public: PHP web Developers

Requirements: Mastering Symfony

LEVERAGING THE SYMFONY CONSOLE COMPONENT

- Discovering the Symfony Console component
- Automating heavy tasks thanks to the command line
- Designing and enabling interactive
- Using the « Formatter » helper to colorize the console
- Using the « ProgressBar » helper to develop progression bars
- Using the « Table » helper pour display tabular data in the console output

AUDITER SON APPLICATION AVEC LE PROFILER

- Analyzing collected data from the profiler
- Analyzing collected data from the application's code
- Configuring the web profiler for the production environment
- Designing custom data collectors to extend the Symfony Profiler
- Extending the « Web Debug Toolbar » and the « Web Profiler » with new debug panels

SensioLabs

92 / 98 bld Victor Hugo, 92115 Clichy Cedex, France
+33 (0)140 99 82 05 - <http://sensiolabs.com>

DEPENDENCY INJECTION AND SERVICE CONTAINER

- Reminding some basics about the dependency injection principles
- Registering custom business services in the service container
- Understanding the concept of services « tags » to extend the framework built-in tools
- Developing custom « compiler passes » objects to optimize the container compilation

SECURITY: AUTHENTICATION AND ACCESS CONTROL

- Developing a custom user provider class
- Recording the last connection date and time of the user
- Triggering custom business code when the user impersonates another user
- Setting up a fine grained resources access control policy thanks to security voters

FORMS AND DATA VALIDATION

- Understanding the internals of the « Form » component
- Extending or overriding built-in form field types
- Developing custom form field types
- Registering new form field types as services
- Developing new custom data transformers
- Modifying the form's data and structure thanks to form events listeners
- Displaying form fields depending on user's permissions thanks to form field types extensions
- Developing new custom data constraint validators

CODE DECOUPLING THANKS TO EVENT DISPATCHING

- Discovering the kernel events and other components built-in events
- Extending the kernel behaviors with custom kernel events listeners
- Dispatching custom business events to make the code truly extensible
- Connecting custom event listeners to custom business events

DOCTRINE ADVANCED TECHNIQUES (ON DEMAND)

- Creating new custom storage form field types (ie. « Money » column type)
- Extending the default DQL syntax with custom DQL functions