

## Symphony and Doctrine – web applications (code: php-symfony-doctrine)

### Overview

The PHP ecosystem was for a long time known for an abundance of frameworks and libraries. As it became increasingly mature, a group of indisputable technological leaders emerged, among them Zend Framework, Symfony, CakePHP and CodeIgniter. The wide-reaching changes in PHP – especially namespaces introduced in 5.3.0 – led to a second generation of these popular frameworks. This time, there is **only one unquestionable leader, Symfony**. It would hardly be an overstatement to claim that **familiarity with SF2 and its idioms is crucial for a modern PHP programmer**.

Symfony is not only the most popular framework for modern web-based PHP applications, but also a comprehensive development philosophy, consisting of:

- building applications and managing autoloading with a composer,
- Symfony 2 way of modularizing applications and configuring the modules,
- injecting dependencies,
- handling the view layer with Twig library,
- handling relational databases with Doctrine 2 library,
- using JSR-303 validation,
- using Symfony built-in forms components.

Our training covers all Symfony components mentioned above – and shows how they interact in a finished product. We impart both practical tips (from programmers who use SF in their daily work) and theoretical groundwork, such as design patterns and reasons behind design decisions made by SF's authors. The deeper understanding of the framework and its philosophy allows training participants not only to create modern web applications, but also to easily adapt them to changes that will be brought by future versions of the framework.

### Duration

3 days

### Agenda

1. Symfony – an overview
  - architecture of a typical application
  - fundamental and additional components (Request, Response, HttpKernel, Controller, Service Container)
  - modularity, most commonly used bundles
  - installation and configuration of a typical SF application; using the composer tool
2. Basic components of an applications
  - Twig template engine
    - templates, template inheritance
    - tags and filters; custom Twig extensions
    - Twig and caching
    - bundling templates
    - active templates; calling the controller from inside the template, client- and server-side
  - routing and controllers
    - calling actions: URL and parameters
    - services provided by the standard controller

### Ask for details

Phone +44 203 608 6289

info@alx.academy

- \* session handling: normal sessions, flash sessions
- \* forwarding and redirecting
- \* displaying templates
- error handling
- 3. Forms and validation
  - JSR-303 and its Symfony implementation
  - a form and its model; FormBuilder and typical use cases
  - native and custom field types (e.g., text fields, buttons, calendars, combo boxes...)
  - validation and forms: handling forms, guessing the field type
  - Twig and forms: form helper, custom themes, automatic and manual form display
  - complex models: embedded forms, collections of embedded forms, model transformers
  - dynamically modified forms
  - protection against CSRF
- 4. Authorization and authentication
  - authentication configuration – users, groups, roles
  - accessing application modules: securing paths, methods and controllers
  - accessing data
  - fundamental ACL (Access Control Lists) and Voters strategies
- 5. Doctrine
  - ORM models and their consequences: Active Record, Transparent Persistence
  - workflow, design patterns (unit of work, identity map, repository / dao)
  - configuration and project integration
  - generating entities from database and databases from entities
  - mappings: manual creation and modification:
    - entities, primary keys (simple and composite), primary key generators
    - simple fields and associations: uni- and bidirectional associations, owning and inverse side
  - working with entities: adding, removing and modifying data in the database
  - queries: Doctrine Query Language, Criteria, repositories
  - efficiency and optimization – lazy and eager relation loading, loading additional data with queries and relations, paging strategies, operation cascading
- 6. Advanced topics
  - registering and using services, Service Container; alternative solutions: YAML, XML, code
  - direct configuration and Container Extension configuration
  - Expression Language
  - services and their dependencies - dependency injection (constructor, setter, property)
  - services as controllers

## Ask for details

Phone +44 203 608 6289

info@alx.academy

## Target audience and prerequisites

The course is designed for PHP programmers who feel somewhat comfortable with the fundamental parts of the language and are familiar with class, method and object syntax. Furthermore, familiarity with basic HTML, CSS and SQL is required, but advanced object-oriented programming techniques and design patterns are **not required**.

## Certificates

Course participants receive completion certificates signed by ALX.

## Locations

- Warsaw (English) – Jasna 14/16A
- Online (English) – your home, office or wherever you want
- any other location (London, UK, EU) on request

## Ask for details

Phone +44 203 608 6289  
info@alx.academy

## Price

990 EUR

The price includes:

- course materials,
- snacks, coffee, tea and soft drinks,
- course completion certificate,
- one-time consultation with the instructor after course completion.