

symfony Workshop

From flat file to MVC project

- understanding layers structure
- split business logic in classes
- get parameters from request
- passing variables to the view
- alternate syntaxe
- split projects into applications, modules, actions
- browsing a project's file structure
- decorate templates with a layout
- linking pages together
- understanding MVC

Templating with symfony

- using helpers
- build a form with helpers
- playing with templates, slots, partial and components
- configuring the view : view.yml

Working environment

- using several project environments
- understanding a symfony query with the web debug toolbar
- more informations with the log files
- the configuration cascade
- the Yaml format
- managing the cache
- using symfony command line
- symfony's exceptions

Routing and HTTP protocol

- understanding URLs
- GET and POST requests
- apache URL rewriting
- using a helper to deal with outgoing URLs
- understanding front controller
- understanding bi directional routing
- dealing with routing.yml
- forward and redirect
- use explicit URLs
- use named routes

Object Relational Abstraction

- create and manipulate business objects
- protect against SQL injection
- abstract SQL queries

- schema and SQL generation
- connecting an application to a database
- using generated getters and setters
- using generated methods to retrieve objects from the database
- translate a SQL query into object query
- using a criteria
- re-factor code from actions to the model

Installing symfony

- from a sandbox, an archive or a package
- symfony's SVN architecture
- initialize a project, an application
- Configure the web server
- dealing with versioning tools
- updating symfony
- deploying symfony

Admin generator

- generate or init a module
- differences between scaffolding and administration
- initiating an administration
- browse the generated code
- configuring the admin generator
- modifying the list page
- modifying the edit page
- overloading the action
- using themes
- understanding the templates templates

Using the session

- storing attributes and temporary flash messages
- authenticating, logging in and out a user
- configuring and testing credentials

Tests

- validating code elements with unit tests
- lime and symfony command line
- feeding a database with test data : the fixtures
- validating a functionality : functional testing
- sfTestBrowser : how to simulate a browsing session
- validating elements with the DomCssSelector
- Automating tests

The cache

- sending the result of a query in the cache
- sending the layout in the cache
- using the cache with components and partials
- explore the cache content