

SENSIO LABS

Advanced Unit Testing with PHPUnit

Sensio

92-98 boulevard Victor Hugo

92115 Clichy Cédex



TABLE OF CONTENTS

- Introduction 3
- The PHPUnit framework 3
- Training’s goals..... 3
- Audience and prerequisites 4
- Duration and price 4
- The training course 4
- The trainer..... 5
- Evaluation 5
- Training conditions..... 5
- Full program..... 5
 - Introduction to Unit Tests 5
 - Introduction to PHP Unit..... 6
 - Code coverage analysis 6
 - Data providers and fixtures..... 6
 - Mock objects..... 6
 - Results exports..... 7
 - Continuous integration tools 7

INTRODUCTION

The **Advanced Unit Testing with PHPUnit** e-learning training helps you to discover in **three hours** how to implement code coverage analysis and continuous integration tools like phpUnderControl with the PHPUnit framework.

During three hours, you will be trained by a Sensio Labs consultant, who's an expert in PHP 5 and symfony web development. He'll lead you in the discovery of advanced unit testing usages with PHP Unit. You will discover how to generate and analyze a code coverage report, to .use mock objects and to configure a continuous integration tool like phpUnderControl.

THE PHPUNIT FRAMEWORK

PHP Unit is an open-source **PHP 5 unit testing framework** created by Sebastian Bergmann, one of the PHP Quality Team members. PHP Unit is a complete unit testing tool, which provides the following features out of the box:

- API to write unit test suites
- Command line interface to run the test suite and generate reports
- Code coverage analysis
- JUnit and TAP compliant
- XML, JSON, TAP, Clover or HTML outputs
- Fixtures, skeleton generators, subs and mock objects
- Integration with Selenium RC
- Integration with Continuous Integration tools like Bamboo, PHPUnitControl or CruiseControl

Unit tests are part of .agile methodologies mantra and code quality. They ensure the code works as expected and suits the project's specifications.

TRAINING'S GOALS

The main goal of this training is to help attendees to learn advanced unit testing usages with PHP Unit in order to be able to write unit tests for complex needs and to setup continuous integration

tools. To do that, this training follows a pedagogic scheme based on practice to help attendees to discover basics of PHP Unit tools.

AUDIENCE AND PREREQUISITES

This training is dedicated to advanced php developers having a good knowledge of the PHP language, object oriented programming and PHPUnit basic usages. These prerequisites are fundamental to be able to take benefit from this three hours training. If you don't have these prerequisites, we advise you to follow the **Getting Started with PHPUnit** training session first.

DURATION AND PRICE

The training will take place during an only **three hours** online session.

The price for one participant is **120 euros (taxes not included)** and **100 euros (taxes not included)** for more than 5 attendees.

Commercial contact : Nicolas BLIN – +33 1 40 99 81 09

Trainings Manager : Hugo HAMON – +33 1 40 99 82 11

THE TRAINING COURSE

The training will be managed by a php expert. It's given online on a dedicated e-learning platform, on which attendees and trainer get connected from their web browser. This e-learning application allows the trainer to give his talk in the best conditions, including a white board on which training materials are displayed, sound, pictures, text messaging and video. Attendees can take part to the training with a "chat" and their microphones.

Training materials are displayed along the training on the application in order to allow attendees to follow the course and guide them between theory and practice. These training materials contain theoretic concepts to master and exercises with their solutions.

Attendees will receive a convocation by email indicating the time and links to handle their connection on the e-learning platform. This convocation also indicates they have to setup their personal laptop with the following software:

- Java runtime activated
- Voice and sound activated (speakers or headphones)
- PHP 5.2.4 (or greater) with PDO
- An IDE (Eclipse, Netbeans...) or a simple text editor (Notepad++)

- MySQL 5

Lots of exercises are given along the training session. Attendees will be asked to develop by themselves their unit tests suites thanks to the help of our php expert.

THE TRAINER

The trainer is a php expert. Our trainers are all consultants and know php very well as they worked with it on big companies' projects.

EVALUATION

Evaluation is managed by the trainer, who's on charge to lead each participant during the whole training. Moreover, he invites attendees to get involved in the session by chatting with all people in the virtual room.

Exercises are made with the whole group and each attendee takes part to the session's animation. All attendees have their own working copy on their personal computer.

TRAINING CONDITIONS

The training takes place during one three hours online session, on an e-learning platform. This application is accessible from a web browser just after having clicked on the link sent by email.

The virtual room offers attendees and trainers lots of tools to interact with each other. The trainer owns a virtual white board on which he can display training materials, a webcam and a microphone to speak during the training session.

Attendees own an instant text messaging system to allow them to chat with each other and the trainer. They also can chat by voice with the trainer if they own a microphone.

FULL PROGRAM

INTRODUCTION TO UNIT TESTS

- ✓ What are unit tests?
- ✓ Pros and cons

- ✓ The FIRST rule

INTRODUCTION TO PHP UNIT

- ✓ Introduction to PHP Unit
- ✓ Open Source PHP projects use PHP Unit
- ✓ IDE integration
- ✓ PHPUnit installation from PEAR
- ✓ **Lab: Install PHPUnit with PEAR**

CODE COVERAGE ANALYSIS

- ✓ Understanding the code coverage concept
- ✓ Check XDebug extension is installed on the web server
- ✓ Generate a complete code coverage report from the command line interface
- ✓ Analyzing and understanding generated reports and statistics
- ✓ Understanding and using the special @covers annotation
- ✓ Ignore code blocks from coverage process thanks to @codeCoverageIgnore(Start|End) annotation
- ✓ **Lab: Implement new methods and their tests to validate the code coverage rate**

DATA PROVIDERS AND FIXTURES

- ✓ Understanding the data providers and fixtures concepts
- ✓ Create a new data provider method to run a test with several values
- ✓ **Lab: implement a data provider method to test a method's edge cases**

MOCK OBJECTS

- ✓ Discovering mock objects
- ✓ Creating mock objects with PHPUnit
- ✓ Test data model coupled to mock objects
- ✓ Check code coverage rate thanks to advanced unit tests
- ✓ **Lab: implement mock objects to simulate real objects' behaviors**

RESULTS EXPORTS

- ✓ Export results as XML
- ✓ Export results as JSON
- ✓ Export results as TAP
- ✓ Check code coverage rate thanks to advanced unit tests
- ✓ **Lab: export results as XML, JSON and TAP**

CONTINUOUS INTEGRATION TOOLS

- ✓ Discovering continuous integration
- ✓ Installing CruiseControl and phpUnderControl thanks to PEAR
- ✓ Initializing a new continuous integration project for CruiseControl
- ✓ Discovering the CruiseControl's build.xml file
- ✓ Discovering the CruiseControl's config.xml file
- ✓ Using phpUnderControl to ease the creation of a new CruiseControl's project
- ✓ **Lab: install and configure phpUnderControl to make builds of the code**