



Database abstraction with PHP & Doctrine 2

Duration: 3 days - 21 hours

Price : 1 490€ / person

Code: DOC2C1

GOALS

During three days, you will learn how to increase your productivity thanks to Doctrine 2, the PHP ORM framework.

PRE-REQUISITES: Mastering PHP 5.3 & Object Oriented Programming

PUBLIC: PHP Developers

FULL PROGRAM

INTERACT WITH A RELATIONAL DATABASE THANKS TO PHP AND THE PDO EXTENSION

- Introduction to PDO (DSN, prepared statements, transactions...)
- Establishing a MySQL or SQLite connection with PDO
- Executing a SQL query on the database
- Securing critical SQL statements with a transaction
- Refactoring the business logic into model classes

INTRODUCTION TO THE DOCTRINE 2 PHP LIBRARY

- Introduction to the Doctrine2 library
- Installing Doctrine2 from Git and PEAR
- Discovering database abstraction pros and cons
- Exploring the Doctrine2 architecture (Common / DBAL / ORM)

DATABASES ABSTRACTION WITH THE DOCTRINE 2 DBAL COMPONENT

- Introduction to the database abstraction principle
- Connecting to a MySQL or SQLite databases
- Selecting records from a database table
- Inserting new records in a database table
- Updating some database records
- Deleting records from the database
- Executing SQL queries in a transaction to ensure security
- Manipulating a database schema (tables, columns, indexes...)

OBJECT RELATIONAL MAPPING WITH THE DOCTRINE 2 ORM COMPONENT

- Discovering the object relational mapping principle
- Designing a database schema by defining PHP classes
- Adding relations between models (1-n, m-n, 1-1...)
- Adding table indexes
- Persisting business objects to the database
- Querying a database thanks to the QueryBuilder object

- Overloading business objects with custom methods
- Detaching objects from their persistence manager
- Executing batch SQL queries
- Writing abstract SQL queries thanks to the DQL language

MIGRATING A DATABASE SCHEMA THANKS TO THE MIGRATIONS COMPONENT

- Introduction to the Doctrine 2 Migrations component
- Creating migration classes by hand
- Generating migration classes from an existing database schema
- Automating migration procedures from the command line interface