

Course : Laravel, improvement

Practical course - 3d - 21h00 - Ref. LVL

Price : 1830 € E.T.

★★★★★ 5 / 5

This course will enable you to finalize a Laravel project. You'll learn how to manage the security of your application with authentication and authorizations, manage e-mail dispatch and queue management with Queue Worker. You'll also see how to deploy a site in production using Docker.

Teaching objectives

At the end of the training, the participant will be able to:

- ✓ Manage authentication
- ✓ Creating unit tests
- ✓ Develop e-mail application modules
- ✓ Deploying web applications with Docker

Intended audience

Developers, architects.

Prerequisites

Basic knowledge of Laravel - Good knowledge of PHP. Basic knowledge of object-oriented programming.

Practical details

Hands-on work

Training alternates theory and practice. Everything we learn is put into practice.

Teaching methods

Assessment of learning is carried out throughout the session through a series of exercises (50-70% of the time).

Course schedule

PARTICIPANTS

Developers, architects.

PREREQUISITES

Basic knowledge of Laravel - Good knowledge of PHP. Basic knowledge of object-oriented programming.

TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS

The trainer evaluates each participant's academic progress throughout the training using multiple choice, scenarios, hands-on work and more. Participants also complete a placement test before and after the course to measure the skills they've developed.

1 Authorizations and authentication

- Advanced settings.
- Authentication methods.
- Disconnection.
- Safety concepts.
- Authorizations.
- Protecting roads.

Hands-on work

Create accounts with different rights.

2 Cache and session

- Sessions.
- Use Redis.
- Store data.
- The cache.
- File storage.
- Database.
- Store, retrieve and delete data.

Hands-on work

Session management with Redis.

3 Queue management

- Key concepts.
- Configuration.
- Pilot.
- Queue Worker.
- Create and dispatch tasks.
- Monitoring.
- Error handling.

Hands-on work

Create queue tasks.

4 E-mails

- Local development.
- Prepare e-mails.
- Send e-mails.
- Use notifications.
- Choose a driver in a production environment.
- Amazon SES, Mailgun, SMTP.

Hands-on work

Development of an e-mail application.

TEACHING AIDS AND TECHNICAL RESOURCES

- The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.
- At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.
- A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

TERMS AND DEADLINES

Registration must be completed 24 hours before the start of the training.

ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@orsys.fr to review your request and its feasibility.

5 The tests

- Unit testing.
- Database.
- Functionality testing.
- Laravel Dusk.
- Simulation.
- Automatiser le lancement des tests. Automatiser le lancement des tests

Hands-on work

Create unit tests.

6 The components

- Integrate a component.
- Useful, high-quality components.
- Create your own components.

Hands-on work

Component creation.

7 Assistants

- Handle strings.
- Functions on tables.
- Other assistant functions.
- Collections.
- Tinker.

Hands-on work

String manipulation.

8 Deploying a site in production

- Considerations.
- Laravel Forge.
- Prepare a server manually.
- Continuous integration and deployment.
- Index.

Hands-on work

Setting up a production site.

9 Laravel development environment with Docker

- Why dock the project?
- Setting up.
- Services.
- Docker-compose.
- Service start-up

Hands-on work

Application deployment with Docker.