Updated 10/02/2025

Sign up

FastAPI training

3 days (21 hours)

Presentation

, easy to learn and fast to code, we'll teach you the tool with our FastAPI training. Discover this fast, modern Web framework for creating APIs with Python 3.8+.

With this tool, you'll learn to master the creation asynchronous APIs, automatic data validation, authorization and authentication.

You'll explore dependency management with Pydantic, automatic JSON schema generation and GraphQL compatibility.

This course will equip you with skills such as robust API development, efficient use of asynchronous Python, data validation and interactive documentation.

The course will be presented with the latest FastAPI features, version 0.110 (at the time of publication).

Objectives

- Use Pydantic to model queries, validate data and implement security mechanisms
- Create your first FastAPI application
- Integrating SQL databases with FastAPI

Target audience

Developers.

PREREQUISITES

- Basic knowledge of Python
- Understanding web development concepts

OUR FastAPI TRAINING PROGRAM

INTRODUCTION TO FASTAPI AND ASYNCHRONOUS IN PYTHON

- What is FastAPI and why use it?
- Comparison with other Python Web frameworks
- Introduction to asynchronous programming in Python
- Understanding async and await
- Creating a development environment for FastAPI

FIRST STEPS WITH FASTAPI

- Installing FastAPI and Uvicorn
- Creating a first application
- Defining routes and path operations
- JSON responses and status code management
- Automatic execution and reloading of the development server

PATH AND QUERY PARAMETERS

- Using path parameters to capture values
- Validation and conversion of path parameter types
- Understanding and validating query parameters
- Combining path and query parameters in a route
- Management of default values and optional parameters

Parameter management with Pydantic

- Introduction to Pydantic and data models
- Validating query bodies with Pydantic models
- Using nested models to structure complex data
- Declaration of query data examples
- Using Pydantic's Field and validators

SECURITY AND AUTHENTICATION

- Understanding FastAPI security mechanisms
- Implementing authentication with OAuth2 and JWT tokens
- Route protection and obtaining the current user
- Using dependencies for security management
- Configuring CORS for cross-origin resource sharing

DATA MANAGEMENT AND DATABASES

- SQL database integration with FastAPI
- Using SQLAlchemy for database operations
- Creating Pydantic database models and schemas
- Implementing CRUD operations in the API
- Database migrations with Alembic

DEPENDENCIES AND MIDDLEWARE

- Understanding dependencies in FastAPI
- Using classes and sub-dependencies
- Custom middleware creation
- Background task management
- Configuring responses with custom headers and cookies

TESTING AND DEBUGGING

- Writing tests with pytest
- Testing routes and data models
- Debugging FastAPI applications
- Using lifecycle events to test startup and shutdown
- Asynchronous test configuration

DEPLOYMENT AND BEST PRACTICES

- Deployment concepts for FastAPI
- Using Docker containers for FastAPI
- Configuring production servers like Gunicorn with Uvicorn
- Continuous deployment using tools such as GitHub Actions
- Review of best practices for maintaining and upgrading FastAPI applications

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is confirmed, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the forthcoming course, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or learning difficulties.

in-company security (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.