

Python based API and Microservices Fundamentals

Duration: 4 days

Prerequisites: Knowledge of Python Programm

Day 1 – API Fundamentals & REST with Python

Theory Modules

1. Introduction to APIs & Microservices
 - Monolithic vs Microservices architecture
 - Real-world enterprise API use cases
 - API-first development approach
2. REST Architecture Fundamentals
 - REST principles
 - HTTP methods (GET, POST, PUT, DELETE)
 - HTTP status codes & headers
3. Python API Framework Overview
 - Why FastAPI (enterprise readiness)
 - Project structure & setup
 - Request–response lifecycle

Lab Modules

- Lab 1: Setup Python API development environment
- Lab 2: Create first REST API using FastAPI
- Lab 3: Test APIs using Postman / curl

Outcome:

Ability to build and test basic Python REST APIs.

Day 2 – API Design, Validation & Documentation

Theory Modules

1. API Design Best Practices
 - Resource naming
 - Clean endpoint structure
2. Request & Response Models
 - JSON schema design
 - Request validation
 - Response formatting
3. Parameters Handling
 - Path parameters
 - Query parameters
 - Request body
4. Error Handling
 - HTTP exceptions
 - Custom error responses

5. API Documentation
 - OpenAPI & Swagger UI

Lab Modules

- Lab 1: Design structured API models
- Lab 2: Implement validation & error handling
- Lab 3: Generate auto-documented APIs

Outcome:

Participants design clean, validated, and documented APIs.

Day 3 – Database Integration & Microservices Concepts

Theory Modules

1. Microservices Architecture Fundamentals
 - Service independence
 - Database-per-service concept
2. Database Integration
 - ORM concepts
 - SQLAlchemy overview
3. CRUD Operations
 - Create, Read, Update, Delete APIs
4. Service Layer Architecture
 - Controller vs service separation
 - Business logic isolation
5. Configuration Management
 - Environment variables
 - Config files

Lab Modules

- Lab 1: Connect API to relational database
- Lab 2: Implement CRUD microservice
- Lab 3: Refactor API using service-layer architecture

Outcome:

Ability to build data-driven microservices.

Day 4 – Microservices Communication, Security & Deployment

Theory Modules

1. Inter-Service Communication
 - REST-based service-to-service calls
 - Sync vs async communication (conceptual)
2. API Versioning Strategies

3. Security Fundamentals
 - API keys
 - Token-based authentication (intro)
4. Logging & Monitoring Basics
5. Deployment Fundamentals
 - Introduction to Docker
 - Containerizing Python APIs
 - Local deployment strategies

Lab Modules

- Lab 1: Create multiple microservices
- Lab 2: Enable service-to-service communication
- Lab 3: Secure APIs with basic authentication
- Lab 4: Dockerize and run Python microservices

Outcome:

Participants understand how microservices communicate, secure, and deploy.