



## Building API Clients and Servers with Python

- 5 Days
- Lecture and Hands-on Labs

### Course Overview

This course provides a comprehensive guide to building and interacting with APIs using Python, equipping participants with the skills needed to create, manage, and deploy RESTful APIs effectively. Beginning with foundational Python programming concepts, the course progresses to cover HTTP principles, JSON data handling, and interacting with public APIs using tools like the `requests` library. Participants will learn to design and build their own APIs using Flask, incorporating best practices for routing, session management, and error handling. The course also explores integrating Jinja2 templates, tracking data with `sqlite3`, and deploying APIs in production environments with `Waitress` and `Docker`. By the end, attendees will have the confidence and expertise to create scalable, dynamic API-driven applications.

Review this course online at <https://www.alta3.com/courses/py201>

### Who Should Attend

- Developers and Programmers looking to build or integrate APIs into their applications.
- System Administrators aiming to automate workflows and enhance system interactions via APIs.
- IT Professionals seeking to understand API design and interaction fundamentals.
- Data Analysts who need to retrieve and process data from web APIs.
- Web Developers interested in creating backend services using Flask.
- DevOps Engineers looking to deploy and manage Flask APIs in production environments.
- Software Testers tasked with validating API functionality and performance.
- Technical Architects planning API-driven application architectures.
- Students and Beginners eager to start working with APIs and Python.
- Anyone Interested in APIs wanting to enhance their technical skills in API development and integration.

### What You'll Learn

- Master Python Fundamentals: Build a strong foundation in Python programming, including functions, data structures, and control flow, to support API development.

- Understand HTTP and RESTful APIs: Learn the basics of HTTP protocols, methods, and RESTful API principles to effectively design and interact with APIs.
- Work with JSON Data in Python: Explore JSON data handling in Python, including serialization, deserialization, and integration with local files.
- Send API Requests with Python: Use Python's `requests` library to perform GET and POST requests and interact with open APIs securely using API keys.
- Leverage Swagger for API Documentation: Understand and use Swagger to explore, document, and test APIs efficiently.
- Develop RESTful APIs with Flask: Build scalable and efficient RESTful APIs using Flask, incorporating best practices for routing, error handling, and API management.
- Manage Sessions and Cookies in Flask: Learn to implement and manage user sessions and cookies to enhance API functionality and security.
- Integrate Flask with Jinja2 for Dynamic Content: Use Jinja2 templates to create dynamic and user-friendly API responses and interfaces.
- Store and Track API Data with sqlite3: Use sqlite3 to integrate lightweight database solutions for tracking and managing API data.
- Deploy Flask Applications: Deploy Flask APIs with tools like Waitress and Docker, ensuring production-ready scalability and performance.

## Outline

### Day 1- Foundational Python









- 📖 Lecture + Lab: Built-in Functions
- 📖 Lecture + Lab: Custom Functions
- 📖 Lecture + Lab: Objects and Methods
- 🗣️ Lecture: Python Lists
- 📖 Lecture + Lab: Python Lists
- 🗣️ Lecture: Python Dictionaries
- 📖 Lecture + Lab: Python Dictionaries
- 🗣️ Lecture: Conditionals
- 📖 Lecture + Lab: If, Elif, and Else Conditions
- 📖 Lecture + Lab: While Loops

### Day 2- Foundational Python (Continued)








- 📖 Lecture + Lab: For Loops
- 🗣️ Lecture: Reading and Writing to Files
- 📖 Lecture + Lab: Reading Files
- 📖 Lecture + Lab: Using Modules
- 📖 Lecture + Lab: PIP and Third Party Libraries
- 📖 Lecture + Lab: Try and Except
- 📖 Lecture + Lab: Python Classes & Inheritance

### Day 3- RESTful APIs and Client Access








- 📖 Lecture + Lab: Your First API Request

-  Lecture: Python Data sets vs JSON
-  Lecture + Lab: Python Data to JSON file
-  Lecture: Introduction to HTTP
-  Lecture + Lab: Standard vs. Third Party Libraries and Open APIs
-  Lecture + Lab: requests library - Open APIs
-  Lecture + Lab: requests library - GET vs POST to REST APIs
-  Lecture + Lab: APIs and Dev Keys
-  Lecture + Lab: Swagger

#### Day 4- Building RESTful APIs with Flask

-  Lecture: RESTful API Best Practices
-  Lecture: Intro to Flask
-  Lecture + Lab: Building APIs with Python
-  Lecture + Lab: Flask APIs and Cookies
-  Lecture + Lab: Flask Sessions
-  Lecture + Lab: Flask Redirection, Errors, and API Limiting
-  Lecture + Lab: Flask Uploading and Downloading Files






#### Day 5- Building RESTful APIs with Flask (Continued)

-  Lecture: Introduction to Jinja
-  Lecture + Lab: Flask APIs and Jinja2
-  Lecture: Learning sqlite3
-  Lecture + Lab: Tracking API Data with sqlite3
-  Lecture + Lab: Tracking Inventory with sqlite3
-  Lecture + Lab: Flask and waitress
-  Lecture + Lab: Running Flask in a Docker Container

#### Additional Labs and Tools

-  Lecture: Introduction to Threads
-  Lecture + Lab: Working With Threads
-  Lecture + Lab: Threads and API requests
-  Lecture + Lab: Introduction to Asynchronous Programming with AsyncIO

#### Optional- PCEP Certification Guide

-  Lecture: Introduction to the PCEP Exam
-  Lecture + Lab: Advanced Numbers and Operators
-  Lecture + Lab: Pythonic Loops and Iteration
-  Lecture + Lab: Advanced Lists and Tuples
-  Lecture + Lab: Advanced Functionality and Error Handling

#### Prerequisites

- Basic Keyboard Proficiency: Ability to efficiently navigate and use a keyboard, including typing, copy-pasting, and basic text editing in terminal and/or text editors.

#### Next Courses

- Python 202: Network Automation with Python (<https://alta3.com/courses/py202>)
- Python 301: Data Sciences with Python (<https://alta3.com/courses/py301>)
- Git and GitHub (2 days) (<https://alta3.com/courses/github>)

- Git and GitLab CI/CD (2 days) (<https://alta3.com/courses/gitlab>)