# Course Abstract

***In this course, you will create a complete, secure API with documentation.*** This course is intended to explore RESTful API development with ample hands-on practice to better reinforce topics learned and to get practical exposure to building API services.

During this course, we examine RESTful design principles all while building our own API. Python is used as a language of choice for implementation while exploring various related topics. While this is intended as an introductory course, it still covers a breadth of topics to include key components of RESTful services, different tools used in the development of API systems, and Python-based frameworks for creating RESTful apps. Important coverage is given to API fundamentals including the structure and use of JSON data as a payload, best practices for API design, and techniques for defining API contracts. As students evolve their project, they will learn about API versioning, caching, and approaches for improving performance. A discussion of authentication and authorization will also occur allowing students to get exposure to techniques used in creating safe, secure APIs.

# Audience

This course is designed for software developers, engineers, consultants, and anyone interested in getting a better picture of how to create an API.

# Prerequisites

While knowledge of Python is helpful, it is not required. Basic programming fundamentals from any programming language will be adequate. The appendix within the student course manual contains a primer on Python language key concepts. Review of this content before the course should be sufficient.

# Duration

2 Days

# Learning Outcomes

Upon completion of this course, participants will be able to:

* Develop a RESTful API Implementation
* Create clients that can access the solutions
* Implement secure services

# Course Topics

**Day 1**

**Introduction and Overview**

Business Process Modeling

Service-Oriented Architectures

What is an API?

Purpose of an API

Key Components of a RESTful Service

API Lifecycle

RESTful Design Principles

Web Services vs RESTful APIs

RESTful API Examples

Different Tools and Approaches for API Development

Using Python for RESTful Development

RESTful Python Frameworks

*Environment Setup*

**Fundamentals**

HTTP/HTTPS

Delivery Formats: JSON, XML

JSON Data Format

Requests and Responses

Defining API Endpoints

*Exercise: Making Requests*

Consuming APIs

*Exercise: Retrieving Data*

HTTP Status Codes

Designing an API

Pagination

*Exercise: API Project Implementation*

Documenting APIs

Exposing Services

Defining the Contract

Contract Formats and OAS

*Exercise: API Project Implementation (continued)*

**Day 2**

*Exercise: Completion of API Project*

**Advanced Concepts**

Improving Performance

Caching

API Versioning

Unit Testing APIs

**Authorization and Authentication**

Securing RESTful APIs

Security Best Practices

Basic Auth

OAuth

*Exercise: Implementing OAuth*

**Appendix – Python Primer**

Python Data Types

Control Structures

Functions and Classes

Modules and Importing

Decorators

# Course Code

IN1803