

APIs and API Design with Python

Course Summary

Description

Application Programming Interfaces (APIs) have become increasingly important as they provide developers with connectivity to everything from rich datasets in an array of formats (such as JSON), to exposing the configurability of software applications and network appliances. Lessons and labs focus on using Python to interact, design, and build APIs for the purposes of scripting automated solutions to complex tasks. Class is mostly live demonstrations and hands on labs.

Objective

Develop Python scripts that communicate with RESTful (and non-RESTful) APIs, as well as design RESTful API interfaces themselves. Use Python to open SSH sessions and pass commands to remote servers, move files via SFTP, parse and manipulate popular data structures (JSON, XML, CSV, and YAML), handle errors, interface with the operating system, create highly efficient regular expressions for parsing, and best practice techniques.

Audience

System administrators, network engineers, and developers will find this course compelling as they build and interact APIs that not only return highly parsable datasets, but also trigger scripted actions. Some previous experience with Python is ideal, although, coding experience in another language is also enough to find success within this course.

Topics

- Python Review
- OS Interfacing
- Web and RESTful APIs
- Python Protocol Clients
- JSON, YAML, XML, CSV and Excel
- Generating and Sending Emails
- Dates and Times
- Python Regular Expression (Regex)
- Code Review
- Web API Design with Flask
- SQLite
- Processes and Threads

Prerequisite

- Python Basics (5 days)
- Coding experience in another language serves as an adequate prerequisite

Duration

Five Days

APIs and API Design with Python

Course Outline

I. *Python Review*

- A. Lists
- B. Dictionaries
- C. Tuples
- D. Conditionals (if, elif, else)
- E. Loops (for and when)
- F. Functions
- G. Variable Scope - Review
- H. Converting boiler plate code to functional code
- I. Writing Functions (reusable code)
- J. Using pip
- K. Useful 3rd party modules
- L. Publishing a module
- M. Documenting modules

II. *OS Interfacing*

- A. Operating System interfacing with OS module
- B. OS module - listdir(), getcwd(), mkdir()
- C. Common sys Module Attributes & Methods
- D. Working With the os Module & Files/Directories
- E. Walking File Trees with os.walk()
- F. File, Path and Directory Examples
- G. Working with os.path
- H. os & os.path Module Examples
- I. Environmental Variables
- J. Running Shell Commands
- K. Compressing and archiving (gzip, tar, zip)

III. *Web and RESTful APIs*

- A. REST
- B. REST APIS and HTTP CRUD
- C. REST and OpenStack
- D. URI analysis and formation
- E. Wireshark capturing
- F. cURL
- G. EtcD keystore
- H. Creating a Python client to interact with API endpoints
- I. API dev keys
- J. SSH and Python
- K. Secure password retrieval
- L. Tokens and APIs

IV. *Python Protocol Clients*

- A. Scripting the browser
- B. Scripting with HTTP
- C. Creating an HTTP Client & Server
- D. Python and SSH
- E. Building an SFTP Client & Server
- F. Python and SFTP limitations
- G. Paramiko for SSH
- H. Netmiko and Major Network Vendors (Cisco, Juniper, Arista)

V. *JSON, YAML, XML, CSV and Excel*

- A. JSON RFC 7159
- B. JSON Formatting
- C. YAML intro
- D. YAML lists
- E. YAML dictionaries
- F. YAML line spanning
- G. Reading YAML is easy
- H. XML
- I. CSV
- J. Import json
- K. Import yaml
- L. Decoding json and yaml to use
- M. Using python to decode data structures like YAML, XML, CSV, and JSON
- N. Reading from Excel
- O. Writing to Excel

VI. *Generating and Sending Emails*

- A. Overview of email modules
- B. Creating simple emails
- C. Interfacing with your email account

VII. *Dates and Times*

- A. Python and Cron
- B. Import time and time.time()
- C. Suspending with sleep()
- D. Dealing with time
- E. Time formatting
- F. Time tuples
- G. Creating Calendars

APIs and API Design with Python

Course Outline (cont.)

VIII. *Python Regular Expression (Regex)*

- A. Metacharacter review
- B. Re modules
- C. search() and match()
- D. findall()
- E. Compiling regex search patterns
- F. Creating highly efficient searches
- G. Sorting data sets
- H. Complex sorts
- I. sort() vs sorted()
- J. Sorting with functions
- K. Applying Regex to file searches
- L. Applying Regex to API results

IX. *Code Review*

- A. Best practice
- B. Using pylint
- C. Conventions
- D. Underscore
- E. Double underscore
- F. Monkey Patching

X. *Web API Design with Flask*

- A. Flask Overview
- B. Decorators
- C. Building APIs with Python and Flask
- D. APIs returning Jinja2 templating
- E. Returning a 'cookie'
- F. Building Sessions
- G. Redirecting from URIs
- H. Build an API to accept a file upload

XI. *SQLite*

- A. Overview
- B. Connecting to Python
- C. Read / Write operations
- D. Other useful instructions
- E. Connecting APIs and SQLite
- F. Reading and Writing to Databases with APIs

XII. *Processes and Threads*

- A. Threading
- B. Context change
- C. Deadlock errors
- D. Thread starvation
- E. Racing conditions and racing specifics
- F. Working with Locks

Labs:

- Using vim
- Making a Github account
- Getting dir(obj) help() and pydoc
- Lists
- Dictionaries
- Python Data to JSON file
- Python Data to YAML file
- import time
- List and Dict Modeling
- try and except
- Construct a SimpleHTTPServer and HTTP Client
- RESTful Open APIs
- APIs and JSON Decode
- requests library
- APIs and Dev Keys
- RESTful APIs and Dev Keys
- getpass - Secure Password Retrieval
- Paramiko SFTP with UN and PW
- Paramiko SSH with RSA
- Scripting Commands over SSH
- Argument Parsing
- Making Excel Sheets
- Reading Excel Sheets
- Timestamping Data
- API Tokens and Subjects
- RegEx to Check IP Address
- Use RegEx to Search Text
- Search and Replace Data
- Compiling Search Objects
- Testing of a Match Exists
- Getting sorted()
- Sort Stability and Complex Sorts
- CSV data

APIs and API Design with Python

Course Outline (cont.)

- Unpacking Arguments
- XML Parsing with ElementTree
- Automating SMTP and Extended SMTP (Email)
- Archive with zipfile
- Building APIs with Python
- Flask APIs and Jinja2
- Flask APIs and Cookies
- Flask Sessions
- Flask Redirection and Errors
- Flask Uploading Files
- Connecting an API to a Database
- Learning sqlite3
- Tracking API Data with sqlite3
- Tracking Inventory with sqlite
- Working with Threads