Understanding and Using APIs Summary

4.9.1

What Did I Learn in this Module?

Introducing APIs

An Application Programming Interface (API) defines the ways users, developers, and other applications can interact with an application's components. An API can use common web-based interactions or communication protocols, and it can also use its own proprietary standards. Some use cases are automation tasks, data integration, and functionality. Most modern APIs are designed into the product. These APIs are usually thoroughly tested, are reliable, and are sometimes even used by the product itself.

API Design Styles

APIs can be delivered synchronously or asynchronously. Synchronous APIs respond to a request directly, usually providing data (or another appropriate response) immediately. Asynchronous APIs provide a response to signify that the request has been received, but that response doesn't have any actual data. The server processes the request, which may take time, and sends a notification (or triggers a callback) with the data after the request has been processed. The client can then act on that returned data.

API Architectural Styles

The three most popular types of API architectural styles are RPC, SOAP, and REST.

Remote Procedure Call (RPC) is a request-response model that lets an application (acting as a client) make a procedure call to another application (acting as a server). The "server" application is typically located on another system within the network.

Simple Object Access Protocol (SOAP) is a messaging protocol. It is used for communicating between applications that may be on different platforms or built with different programming languages. It is an XML-based protocol and is commonly used with HyperText Transfer Protocol (HTTP) transport, but can be applied to other protocols. SOAP is independent, extensible, and neutral.

REpresentational State Transfer (REST) is an architectural style that was created as a hybrid style, derived from several network-based architectural styles that are combined with additional constraints that define a uniform connector interface. There are six constraints applied to elements within the architecture: Client-

Server, Stateless, Cache, Uniform Interface, Layered System, and Code-On-Demand. These six constraints can be applied to any protocol, and when they are applied, you will often hear that it is *RESTful*.

Introduction to REST APIs

A REST web service API (REST API) is a programming interface that communicates over HTTP while adhering to the principles of the REST architectural style. REST API requests are essentially HTTP requests that follow the REST principles. These requests are a way for an application (client) to ask the server to perform a function. REST API responses are essentially HTTP responses. These responses communicate the results of a client's HTTP request. The response may contain the data that was requested, signify that the server has received its request, or even inform the client that there was a problem with their request. Sequence diagrams are used to explain a sequence of exchanges or events. They provide a scenario of an ordered set of events. They are also referred to as event diagrams.

Authenticating to a REST API

For security reasons, most REST APIs require authentication so that random users cannot create, update or delete information incorrectly or maliciously, or access information that shouldn't be public. Without authentication, a REST API permits anyone to access the features and system services that have been exposed through the interface. Authentication is the act of verifying the user's identity. The user is proving that they are who they say they are. Authorization is the user proving that they have the permissions to perform the requested action on that resource. Common types of authentication mechanisms include Basic, Bearer, and API Key. Open Authorization, also known as OAuth, combines authentication with authorization.

API Rate Limits

Using an API rate limit is a way for a web service to control the number of requests a user or application can make per defined unit of time. Common rate limit algorithms include: Leaky bucket, Token bucket, Fixed window counter, and Sliding window counter. Many rate limiting APIs add details about the rate limit in the response's header. Because there is no standard, the key-value pair used in the header may differ between APIs. Some commonly used keys include:

- **X-RateLimit-Limit**: The maximum number of requests that can be made in a specified unit of time

- **X-RateLimit-Remaining**: The number of requests remaining that the requester can make in the current rate limit window

- **X-RateLimit-Reset**: The time the rate limit window will reset

When the rate limit has been exceeded, the server automatically rejects the request and sends back an HTTP response informing the user. In addition, it is common for the response containing the "rate limit exceeded" error to also include a meaningful HTTP status code.

Working with Webhooks

A webhook is an HTTP callback, or an HTTP POST, to a specified URL that notifies your application when a particular activity or “event” has occurred in one of your resources on the platform. Webhooks enable applications to get real-time data, because they are triggered by particular activities or events. With

webhooks, applications are more efficient because they no longer need to have a polling mechanism. In order to receive a notification from a webhook provider, the application must be running at all times to receive HTTP POST requests, and it must register a URI on the webhook provider so that the provider knows where to send a notification when target events occur. In addition to these two requirements, the application must handle the incoming notifications from the webhook server.

Troubleshooting API Calls

Before you troubleshoot, make sure you have the API reference guide and API authentication information handy.

It is not true that every request that is sent to the REST API server will return a response with a status code. While one might expect to always receive a HTTP status code such as 2xx, 4xx and 5xx, there are many cases where the API server cannot be reached or fails to respond. In those scenarios, you will not receive an HTTP status code because the API server is unable to send a response.

While a non-responsive server can be a big issue, the root cause of the unresponsiveness can be simple to identify. You can usually identify what went wrong from the error messages received as a result of the request. It could be client side error, user error, wrong URI, wrong domain, a connectivity issue, an invalid certificate, a server side error, or a communication problem between the server and the client. How do you narrow it down? By correctly interpreting status codes. 4xx codes are Client side error and 5xx codes are Server side errors.

4.9.2

Lab – Integrating a REST API with Python

In this lab, you will create an application that retrieves JSON data from the MapQuest Directions API, parses the data, and formats it for output to the user. You will use the GET Route request from the MapQuest Directions API.

You will complete the following objectives:

- Part 1: Launch the DEVASC VM
- Part 2: Demonstrate the MapQuest Directions Application
- Part 3: Get a MapQuest API Key
- Part 4: Build the Basic MapQuest Direction Application
- Part 5: Upgrade the MapQuest Directions Application with More Features
- Part 6: Test Full Application Functionality

Integrate a REST API in a Python Application

4.9.3

Module 4: Understanding and Using APIs Quiz

1. What is REST?

Topic 4.3.0 – REST is not a protocol or service, but rather a style of software architecture for designing web service applications.

- It is a human readable data structure that is used by applications for storing, transforming, and reading data.
- It is a way to store and interchange data in a structured format.
- It is a protocol that allows administrators to manage nodes on an IP network.
- **It is an architecture style for designing web service applications.**

2. Which HTTP response status code indicates that the request to update the database is completed?

Topic 4.4.0 – The most common HTTP status codes include the following:
- 200 – **OK** (using GET or POST to exchange data with an API successfully)
- 201 – **Created** (creating resources by using a REST API call successfully)
- 400 – **Bad Request** (The request from the client is failed due to client-side issue.)
- 401 – **Unauthorized** (The client is not authenticated to access site or API call.)
- 403 – **Forbidden** (The access request is not granted based on the supplied credentials.)
- 404 – **Not Found** (The page requested at HTTP URL location does not exist or is hidden.)

- 400
- 200
- 404
- 401
- **201**
- 403

3. How many elements does a SOAP message contain?

Topic 4.3.0 – A SOAP message is an XML document that can contain four elements, namely, Envelope, Header, Body, and Fault.

- 2

- 5
- 4
- 3

4.

```
import requests
uri = "sandboxdnac.cisco.com/dna/intent/api/v1/network-device"
resp = requests.get(uri, verify = False)
```

```
Traceback message:

....requests.exceptions.MissingSchema: Invalid URL
'sandboxdnac.cisco.com/dna/intent/api/v1/network-device'
```

Refer to the exhibit. A network administrator is using a Python script to test a REST API request. The traceback message indicates there is an error in the URI. What is the error?

✓ Topic 4.8.0 - The error message indicates “MissingSchema”. A REST API request URI needs the URI to use the proper HTTP protocol, HTTP or HTTPS.

- The destination web host address is invalid.
- The protocol is missing.
- The query component is missing.
- The path in the URI is invalid.

5. Which type of encoding is used in basic authentication for REST APIs?

✓ Topic 4.5.0 - REST APIs use Base64 encoding for the basic authentication. Base64 is a binary-to-text encoding scheme.

- UTF-8
- Base32
- Base64
- UTF-16

6. What is an architectural constraint to which a true RESTful API web service must adhere?

Topic 4.3.0 – Conforming to the constraints of the REST architecture is generally referred to as being “RESTful”. An API can be considered “RESTful” if it has the following features:
- **Client/server** – The client handles the front end and the server handles the back end.
- **Stateless** – No client data is stored on the server between requests. The session state is stored on the client.
- **Cacheable** – Clients can cache responses locally to improve performance.

- ( ) It uses HTTPS to transport data.
- ( ) It must support the XML data format.
- ( ) It operates as a cloud service.
- (•) It runs as client/server model.

7. What is a reason that a network engineer would use APIs?

Topic 4.1.0 – Application programming interfaces or APIs interact with one or more applications. APIs are commonly used to automate tasks, to facilitate data integration, and to increase the functionality of another application. Non-software engineers can now use APIs to automate configuration or data collection from network devices.

- (•) to automate configuration or data collection tasks
- ( ) to provide upper management with information about network performance
- ( ) to facilitate features across network devices that operate at different layers
- ( ) to provide a more robust security architecture

8. What is a webhook for REST APIs?

Topic 4.7.0 – A webhook is an HTTP callback, or an HTTP POST, to a specified URL that notifies the client application when a particular activity or event has occurred in one of the resources on the platform.

- ( ) It is an HTTP PUT message to update information on a website.
- ( ) It is an HTTP redirect to forward the API request to another web service.
- ( ) It is an HTTP UPDATE message to notify the user that an API request is successfully completed.
- (•) It is an HTTP callback to a URL to notify the client application that an event has occurred.

9. As part of creating an API request using Python, the following commands are entered. What is the purpose of this step?
ipaddr = 10.1.50.1
interface = Ethernet1/1
hostname = R1

Topic 4.9.0 – When a script is being coded or created, a variable can be used to store information. The variable name is to the left and the value is to the right.

- ○ to create functions to test the code
- ○ to import Python modules
- ○ to request JSON data
- ◉ to create variables

10. A web service uses a rate limit algorithm that puts all incoming REST API requests into a queue in the order in which they arrive. The algorithm allows incoming requests to arrive at any rate, but the server processes the requests from the queue at a fixed rate. Which rate limit algorithm is used by the web service?

Topic 4.6.0 – Common rate limit algorithms include the following:
- Leaky bucket
- Token bucket
- Fixed window counter
- Sliding window counter

The leaky bucket algorithm puts all incoming requests into a request queue in the order in which they were received. The incoming requests can come in at any rate, but the server will process the requests from the queue at a fixed rate. If the request queue is full, the request is rejected.

- ○ token bucket
- ○ fixed window counter
- ○ sliding window counter
- ◉ leaky bucket

11. What is an application requirement in order to receive a notification from a webhook provider?

Topic 4.7.0 – In order to receive a notification from a webhook provider, the client application must meet certain requirements:
- The application must always be running to receive HTTP POST requests.
- The application must register a URI on the webhook provider, so that the provider knows where to send a notification when target events occur.
- The application must handle the incoming notifications from the webhook server.

- (●) The application must always be running to receive HTTP POST requests.
- ( ) The application must accept HTTP UPDATE messages from the webhook provider.
- ( ) The application must register a unique domain name from a certificate authority.
- ( ) The application must provide a syslog service.

12. Which RESTful operation corresponds to the HTTP GET method?

Topic 4.4.0 – RESTful operations correspond to the following HTTP methods (shown to the left with the RESTful operation on the right):
- POST > Create
- GET > Read
- PUT/PATCH > Update
- DELETE > Delete

- ( ) post
- (●) read
- ( ) update
- ( ) patch

Check | Show Me

Reset

4.9.4

Project Activity 3: Social Coding

In this activity, you will complete the following tasks:

- Select your project.
- Define the features that you commit to delivering in your application.

Refer to the **DEVASC Project Rubric** below to record your process and outcomes.

1. Select an Application Project

Your team has the choice to either make feature enhancements to an existing application, or to create a new application. Choose from the options described below.

Your team has been given a specific budget and a time constraint. The budget is managed by the product owner who has allocated funding for your team and the resources needed. Your manager has also scheduled a certain number of sprints to complete these features.

Option 1: Feature Enhancements – MapQuest

You have on your computer the code from "Integrate a REST API in a Python Application" (lab 4.9.2). Your manager has requested:

- Moving the code from your computer to a collaborative version control system
- Feature enhancements

Your manager has received a request from the marketing department to add new features to the MapQuest API application. You manager has given your team the task of making feature enhancements to this application. These feature enhancements can involve UI changes such as improving the formatting output (with colors, tables, etc.), providing the user with additional options such as data using the metric system or miles, or the type of data you provide to the user. Any other improvements that you think are appropriate will differentiate your work from the work of other teams.

Your manager would also like a list of other enhancements for a future revision. These are called backlog items. This backlog will be used for Project Activity 4.

Note: You will be using the final code from the lab, "Integrate a REST API in a Python Application". In this lab, you used the MapQuest directions API to retrieve JSON data, parsed the data and formatted it for the user.

Option 2: New Code – Create an IPv4/IPv6 Address Application

Your manager has received a request from the engineering department for a prototype of a new application that will provide IP addressing information to network technicians.

Search the internet for REST APIs that retrieve a user's current public IPv4 address and IPv6 address, such as ipapi.co or ipsstack.com. Using the public APIs, create a Python application that displays and formats the computer’s current public IP addressing information.

Depending on the API you select, you may also obtain geolocation information, the provider (ISP), the ASN (Autonomous System Number) of the ISP, and country code.

Your manager would also like a list of other enhancements for a future revision. These are called backlog items. This backlog will be used for Project Activity 4.

Note: The objectives and specific tasks of your project will depend on the options provided by the IP information API you choose.

Rubric/Deliverable: The application that your team will work on and the reasons for your choice.

2. Team Roles

Now that you have selected your application project, you will begin to identify how each team member can contribute to the project. There may be some team members with more experience in certain areas than others. This can be an excellent opportunity for cross-training. Remember, in a successful team people are supportive, encouraging and willing to help other team members.

At this point, the Scrum Leader will facilitate the process of:

- Identifying any persons on the team who may have knowledge, experience, or be an authority in a particular area or topic that is relevant to the project.
- Developing a strategy to accomplish the tasks.
- Creating a project plan to complete the tasks on schedule.
- Identifying the tools needed to move forward.

Rubric/Deliverable: Document your team member roles, and any additional knowledge and skillsets that they can bring to this project.

3. Project Workload and Project Status

Discuss project workload and the team's time commitment. Team members have other responsibilities, and setting expectations can help avoid any future conflicts and help ensure completing the project on-time. Your manager will supply you with the amount of time that has been budgeted for your tasks.

Throughout this project, it will be important for the scrum leader to manage individual workloads and the status of each project. Many times, this role may be assigned to an individual who has expertise in this area and whose sole job is project management.

The scrum leader will need to facilitate the process of:

- Maintaining a list of specific tasks.
- Breaking down tasks into smaller tasks if possible.
- Prioritizing each task.
- Managing the status of each task and any effect it might have on the completion of other tasks and the project.

The scrum leader may assume the role of project manager, or have another team member perform these tasks. A simple cloud-based application such as Google Sheets or Google Docs may be used to for this purpose, allowing team members to continuously update the status of their individual tasks.

Rubric/Deliverable: Provide a brief description of your team's strategy for completing this project.

4. Set Up GitHub for Collaboration

As mentioned previously, your manager has requested that your team use GitHub to collaboratively work on your Python code.

Using GitHub.com:

- Create a repository
- Add collaborators
- Create branches for team members
- Work with pull requests, code review, merge, etc.

Note: Your instructor may request that you link to the GitHub repository.

Rubric/Deliverable: Record the link to your GitHub repository. Describe how GitHub was used to create branches, add team members, do pull requests, etc.

5. Application Development

In this part of the Project Activity, the application will be completed. Continuously refer to the processes and tools discussed previously. Continue to review these questions:

- Does the team continue to have open communications and positive attitudes?
- Are team members encouraging and supporting each other?
- Is the project being monitored with status updates of each task?
- Are there any conflicts or workload issues?
- Are the technical objectives achievable?

Your team is now ready to begin:

- Coding specific tasks
- Merging team code
- Verifying that the code works (quality assurance)

In addition, the scrum leader has the responsibility to ensure that the team is:

- Working collaboratively
- Completing tasks on schedule

6. Final Presentation

Create a presentation about the project you selected and your work completing the project.

Deliverable/Rubric:. Your presentation should include:

- Information about your application, covering what features your team included
- The reasons that your team decided on these specific features in your application
- Application code including comments and documentation. Your comments and documentation should be sufficient for any other team to be able to continue the project if required. Another team should be able to understand the application, your features and how to continue with the project.
- Demonstration of the application
- List of future enhancements (backlog)
- Reflection points – what issues have you faced while working on this activity, how did you find solutions, what have you learned, etc.

Team Activities and Reflection

Deliverable/Rubric: Your manager is interested in knowing how everyone worked together as a team. Here is a list of questions from your manager:

- What did you enjoy about working as a team? What worked well?
- What team problems did you encounter and how did you resolve them?
- What technical problems did you encounter and how did you resolve them?
- How was each team member held accountable individually and for the team as a whole?
- What was your team's decision-making process?
- Overall, how were the team dynamics and what were any lessons learned?

Final Deliverables

At this point, your team has completed Project Activity 3. Your team will present to your manager:

- Presentation
- Team activities and reflection
- Completed application code
- Link to GitHub repository for your work
- Completed rubric items

Project Activity 3 – Social Coding

4.8
Troubleshooting API Calls

5.0
Introduction