Cisco

(200-901)

DevNet Associate (DEVASC)

Total: **560 Questions**

Link:

Question: 1

Which two concepts describe test-driven development? (Choose two.)

A. User acceptance testers develop the test requirements.
B. It enables code refactoring.
C. Tests are created when code is ready for release.
D. Implementation is driven by incremental testing of release candidates.
E. Write a test before writing code.

Answer: BE

Explanation:

The correct answers are B and E. Test-Driven Development (TDD) is a software development approach centered around writing tests before writing the actual code. Option E accurately captures this core concept; the tests are written first, specifying the desired functionality, and then the code is written to satisfy these tests. Option B, "It enables code refactoring," is also correct because TDD facilitates safer refactoring. With comprehensive tests in place, developers can confidently modify code knowing that any regressions will be quickly identified by failing tests. This allows for continuous improvement and simplification of code without introducing bugs.

Option A, "User acceptance testers develop the test requirements," is incorrect as TDD focuses on developer-level testing, not user acceptance. Option C, "Tests are created when code is ready for release," contradicts the fundamental principle of TDD where tests precede code. Option D, "Implementation is driven by incremental testing of release candidates," while related to testing, is not a core tenant of TDD; rather TDD focuses on short, incremental cycles of test creation followed by code implementation. The primary focus of TDD is on creating a robust test suite which drives the code's functionality, thereby enhancing code quality and reducing bugs.

For further research on Test-Driven Development, consider these resources:

Martin Fowler's Article on TDD:https://martinfowler.com/bliki/TestDrivenDevelopment.html
Wikipedia on TDD:https://en.wikipedia.org/wiki/Test-driven_development
Agile Alliance on TDD:https://www.agilealliance.org/glossary/tdd/

Question: 2

Which task is performed because the test-driven development approach is being used?

A. creating test scenarios based on continuous development
B. writing code without committing any coding violations
C. refactoring code that is covered by existing tests
D. testing existing software before developing new code

Answer: C

Explanation:

The correct answer is **C. refactoring code that is covered by existing tests.**

Test-Driven Development (TDD) is a software development approach where tests are written before the actual code. This process typically follows a red-green-refactor cycle: write a failing test (red), write the minimal code to pass the test (green), and then refactor the code while ensuring all tests still pass. Therefore,

refactoring, or improving the internal structure of the code without changing its external behavior, becomes crucial in TDD. The existence of robust tests, created before the code itself, allows developers to confidently refactor. If a refactoring breaks any functionality, the pre-existing tests will immediately fail, alerting the developer. Options A and D describe processes that may be linked to agile and general testing practices, but are not core tenets of the TDD cycle. Option B relates to coding standards, not a direct consequence of adopting a TDD methodology. It is possible to have code without violations in non-TDD methodologies. The emphasis in TDD lies on the feedback loop of failing tests and confirming functionality through passing tests, and then refactoring as necessary to refine the codebase.

Supporting Links:

Agile Alliance - Test Driven Development:https://www.agilealliance.org/glossary/tdd **Wikipedia - Test-driven development:**https://en.wikipedia.org/wiki/Test-driven_development **Martin Fowler - Refactoring:**https://martinfowler.com/refactoring/

Question: 3

What is the first development task in test-driven development?

A. Write code that implements a desired function.
B. Write a failing test case for a desired function.
C. Reverse engineer the code for a desired function.
D. Write a passing test case for existing code.

Answer: B

Explanation:

The core principle of Test-Driven Development (TDD) revolves around writing tests before writing the actual code. Therefore, the first step is to define the expected behavior of the desired function through a test case that initially fails. This failing test serves as a concrete specification of what the function should achieve. Only after creating this failing test, do you then proceed to write the minimal amount of code necessary to make the test pass. This iterative approach – test-fail-code-pass-refactor – ensures that the implemented code meets the defined requirements and minimizes the risk of unexpected behaviors. Options A, C, and D do not align with this fundamental TDD methodology. Option A jumps directly to code implementation, bypassing the test-first requirement. Option C is unrelated to TDD. Option D, writing a passing test for existing code, is a retrospective test, not the initial step in TDD. The failing test acts as a guide, driving development towards fulfilling a specific requirement rather than writing code based on assumption. By initiating with a failing test, the developer is forced to clarify their intentions before diving into implementation. This prevents unnecessary code complexity and keeps the process focused. This practice helps ensure that all implemented features are backed by a demonstrable test case. This approach also promotes refactoring with confidence because a regression test suite has already been defined.

Here are some authoritative links for further research on Test-Driven Development (TDD):

Martin Fowler on TDD:https://martinfowler.com/articles/tdd.html
Wikipedia on TDD:https://en.wikipedia.org/wiki/Test-driven_development
Agile Alliance on TDD:https://www.agilealliance.org/glossary/tdd/

Question: 4

In test-driven development, what are two of the green bar patterns? (Choose two.)

A. another test
B. break
C. triangulate
D. starter test
E. fake it

Answer: CE

Explanation:

The correct answers are **C. triangulate** and **E. fake it**. In test-driven development (TDD), the "green bar" refers to the state where all tests pass. This is the desired outcome of each cycle of the red-green-refactor loop. The patterns contributing to the "green bar" are strategies used during the "green" phase – making the tests pass.

Triangulation (C) is a crucial green bar pattern. It involves writing tests that drive out more general code behavior. When the existing code is too specific to the initial test, you add more tests, forcing the code to become more generalized to accommodate the various test cases. This process refines the logic towards the intended result, ensuring the code isn't over-engineered.

Fake It (E) is another useful green bar pattern. When you initially struggle to write the actual code to make a test pass, you might choose to temporarily return a hardcoded value that satisfies the test. This approach provides immediate movement to the "green" state, making the test pass, allowing you to subsequently focus on the more intricate or complex parts of the functionality after the initial test passes. It acts as a placeholder to get to the green bar quickly.

Option **A. another test** is incorrect as writing a test is in the "red" phase of the loop. Option **B. break** is an anti-pattern; the goal is to make tests pass, not break them. Option **D. starter test** isn't a recognized green bar pattern, though it's generally good practice to start with a simple test to get the ball rolling.

In summary, triangulation and fake it are both green bar patterns employed to achieve a passing test suite, leading to an incremental development process.

Authoritative links:

Refactoring.guru on Test-Driven Development
Agile Alliance on Test Driven Development
Martin Fowler's article on TDD

Question: 5

In the test-driven development model, what is changed after a test fails?

A. schedule
B. project requirements
C. code
D. test

Answer: C

Explanation:

The correct answer is **C. code**. In Test-Driven Development (TDD), the process revolves around a tight loop of writing a test, watching it fail, and then writing just enough code to make the test pass. This iterative cycle emphasizes testing first, shaping the development process. When a test fails, it signifies a discrepancy

between the expected behavior (defined by the test) and the actual behavior (implemented by the code). Therefore, the primary action after a test failure is to modify the **code** to align it with the test's expectations.

Changing the schedule (A) or project requirements (B) at this point would deviate from the core principle of TDD, which is to ensure the code behaves correctly according to defined specifications. Modifying the test (D) might be necessary if the test was flawed initially, but normally the code, not the test is the problem at this phase. The cycle continues by re-running the test after code changes to verify the fix and ensure no regression occurs. This approach promotes cleaner, more reliable code that's easier to maintain and integrate. It also aids in early detection of defects and contributes to better design practices.

Authoritative Links:

Martin Fowler on Test-Driven Development:https://martinfowler.com/bliki/TestDrivenDevelopment.html
Agile Alliance on TDD:https://www.agilealliance.org/glossary/tdd/
Wikipedia on Test-Driven Development:https://en.wikipedia.org/wiki/Test-driven_development

Question: 6

What is a benefit of test-driven development?

A. strict adherence to product requirements
B. faster releases that have minimal features
C. early customer involvement
D. increased code quality

Answer: D

Explanation:

The correct answer is D, "increased code quality." Test-Driven Development (TDD) is a software development practice where tests are written before the code itself. This approach forces developers to think about the desired behavior of a function or module before they implement it. Writing tests first leads to more focused and modular code, as the code is specifically written to pass the defined tests. When following the TDD approach, developers begin with a failing test, then write the minimal amount of code to make the test pass and then refactor for better readability and performance. This cycle, often referred to as "Red-Green-Refactor," results in code with better coverage and less bugs. This is because every line of code that is written to satisfy the requirements will have associated test cases. TDD leads to improved code design and readability, and reduces the likelihood of defects, hence increases code quality. Options A, B and C, are incorrect. Strict adherence to product requirements (A) is a goal of software development in general, not specific to TDD, though TDD may help in achieving that goal. Faster releases with minimal features (B) is not a direct result of TDD, but a potential consequence of choosing to implement a minimal viable product, and early customer involvement (C) is also not a characteristic of TDD itself, but is a broader requirement.

Here are some resources for further research:

Agile Alliance:https://www.agilealliance.org/ (General information on agile methodologies, including TDD) **Martin Fowler's Blog:**https://martinfowler.com/ (Articles on software development, including TDD) **Wikipedia on Test-Driven Development:**https://en.wikipedia.org/wiki/Test-driven_development
(Comprehensive overview of TDD)

Question: 7

Which two statements describe the advantages of using a version control system? (Choose two.)

A. It allows for branching and merging so that different tasks are worked on in isolation before they are merged into a feature or master branch.

B. It provides tooling to automate application builds and infrastructure provisioning.

C. It allows multiple engineers to work against the same code and configuration files and manage differences and conflicts.

D. It provides a system to track User Stories and allocate to backlogs.

E. It allows developers to write effective unit tests.

Answer: AC

Explanation:

Justification:

Version control systems (VCS) like Git are foundational for collaborative software development. Option A accurately describes a core advantage: branching and merging. This feature enables developers to work on separate features or bug fixes concurrently without interfering with each other's code. These isolated branches can then be merged back into a main branch (e.g., master or main) when complete, providing a structured approach to development. This isolates changes, thus enabling individual developers to create different features at the same time, without introducing conflicts.

Option C correctly highlights another key advantage: collaborative code management. VCS allows multiple developers to simultaneously work on the same codebase, tracking and managing differences and conflicts through a process such as merging. Without this, teamwork on the same project is impossible without overwriting each other's files. This greatly aids developers in collaborative code creation and configuration, by ensuring that all changes and alterations are tracked and easily referenced.

Option B describes the function of CI/CD (Continuous Integration/Continuous Deployment) tools, not version control. CI/CD automates building, testing, and deploying applications, whereas version control focuses on code tracking and collaboration.

Option D relates to project management tools and task tracking. While version control might integrate with these tools, it doesn't provide the functionality of user story tracking. Similarly, Option E describes testing functions, which is not a core function of a version control system. While version control may be used in tandem with testing, it doesn't actually provide the tooling or ability to write tests. Therefore, Options B, D and E are inaccurate.

Authoritative Links:

Git Documentation:https://git-scm.com/doc - Official Git documentation, a popular version control system, providing in-depth information on its features.

Atlassian Version Control Tutorial:https://www.atlassian.com/git/tutorials - A resource from Atlassian on using Git with examples and guides.

Understanding Version Control:https://martinfowler.com/articles/versionControl.html - Martin Fowler's article on the principles and importance of version control.

Question: 8

What are two advantages of version control software? (Choose two.)

A. It supports tracking and comparison of changes in binary format files.

B. It allows new team members to access the current code and history.

C. It supports comparisons between revisions of source code files.

D. It provides wiki collaboration software for documentation.

E. It allows old versions of packaged applications to be hosted on the Internet.

Answer: BC

Explanation:

Version control systems (VCS), like Git, are fundamental tools for collaborative software development. Option B is correct because VCS maintains a complete history of all changes, enabling new team members to quickly understand the project's evolution and access the current codebase. They can readily clone the repository and see the full history of modifications, facilitating onboarding and project understanding. Option C is also correct; VCS excels at tracking and comparing changes to text-based files, especially source code.

Developers can easily see exactly what lines were added, deleted, or modified between different versions, facilitating debugging, code review, and revert operations. This granular comparison is crucial for maintaining code integrity and collaboration efficiency. Option A is incorrect because while some VCS can store binary files, they are generally not very effective at tracking changes within them or offering meaningful
comparisons of binary file changes, as it is a byte-by-byte comparison. Option D is incorrect as wikis are distinct collaboration tools for documentation, not part of VCS functionalities. Option E is incorrect; version control does not involve application hosting. Instead, it manages the source code of applications.

Justifications with authoritative links:

Version Control Systems Basics: For an in-depth understanding of the core functionalities of VCS, refer to the official Git documentation: https://git-scm.com/doc
Benefits of Version Control: Read more about the advantages of using version control systems in software development here: https://www.atlassian.com/git/tutorials/why-git
Binary Files and VCS: An explanation of limitations with binary file tracking can be found here: https://stackoverflow.com/questions/11706250/version-control-for-binary-files

Question: 9

Before which process is code review performed when version control is used?

A. checkout of code
B. merge of code
C. committing code
D. branching code

Answer: B

Explanation:

The correct answer is **B. merge of code**. Code review, a critical practice in software development, typically occurs before code is integrated into a shared branch, such as the main or master branch. This integration process is known as merging. When developers work on separate features or bug fixes, they usually create branches. After completing their work, they submit a merge request or pull request. Before the code is actually merged into the target branch, a code review is performed by peers. This process helps identify potential bugs, logic errors, security vulnerabilities, and code style issues. Code review ensures that the code meets the project's standards and guidelines and is ready for integration with the existing codebase.

Reviewing before the merge prevents the introduction of problematic or flawed code into the main project codebase. Performing code review before merging is a core aspect of a robust DevOps CI/CD pipeline, fostering a culture of shared responsibility and code quality. Options A (checkout), C (committing), and D (branching) all occur before the point at which code review is typically performed in the process.

Further Research:

Atlassian - Code Review Best Practices:https://www.atlassian.com/agile/code-reviews/best-practices
SmartBear - What is Code Review?https://smartbear.com/learn/code-review/what-is-code-review/ **Microsoft - What is code review?**https://learn.microsoft.com/en-us/azure/devops/develop/code-review?view=azure-devops

Question: 10

What is an advantage of a version control system?

A. facilitates resolving conflicts when merging code
B. ensures that unit tests are written
C. prevents over-writing code or configuration files
D. forces the practice of trunk-based development

Answer: A

Explanation:

The correct answer is **A. facilitates resolving conflicts when merging code.**

Version control systems (VCS), like Git, are designed to track changes to files over time, enabling multiple developers to collaborate on the same codebase. A primary advantage is their ability to manage and resolve conflicts that arise when different developers modify the same lines of code concurrently. While VCS does not automatically resolve every conflict, they provide robust tools that highlight conflicting changes, making it much easier for developers to identify and merge them. This is achieved by presenting a clear view of diverging modifications, allowing users to choose which changes to incorporate or combine.

Options B, C, and D are incorrect. While unit tests are essential, VCS does not enforce their creation. Similarly, VCS does not prevent overwriting code, instead, it facilitates recovery from such mistakes through its revision history. And while trunk-based development is a popular approach used in conjunction with VCS, VCS itself doesn't force this practice. The core strength of VCS lies in its ability to track changes and support
collaborative code development by providing mechanisms to manage diverging branches and their eventual merge.

For deeper understanding, explore these resources:

Atlassian's Git Tutorials:https://www.atlassian.com/git (Provides comprehensive guides and explanations on Git concepts, including branching and merging).

Git Documentation:https://git-scm.com/doc (Official documentation for Git, offering detailed information on all aspects of the version control system).

Microsoft Learn version control:https://learn.microsoft.com/en-us/devops/develop/git/
GitHub Docs:https://docs.github.com/en (Documentation related to GitHub features and workflow, demonstrating the practical usage of VCS in a collaborative environment).

Question: 11

Refer to the exhibit.

```
json_string = """
{
        "researcher": {
            "name": "Ford Perfect",
            "species": "Betelgeusian",
            "relatives": [
                {
                    "name": "Zaphod Beeblebrox",
                    "species": "Betelgeusian"
                }
            ]
        }
}
"""

my_json = json.loads(json_string)
```

Which Python data structure does my_json contain?

A.map
B.list
C.json
D.dict

Answer: D

Explanation:

The right answer is D.Dictionaries are analogous to "hashes" in some other languages. A dictionary is a data structure that stores simple key-value pairs. They make it very easy and fast to store and retrieve a value by supplying a key to be used as an index for that value. Values: The values in a Python dictionary can be anything, and like lists the types don't have to be consistent. Keys: A dictionary's keys have an important restriction: Whatever you want to use as a key has to be immutable and hash-able. This means that you could use a tuple as a key (immutable), but not a list (mutable). Also note that all of the basic data types that you learned about previously (int, float, bool, str, bytes) are immutable and can be used as dictionary keys. You create a dictionary by using curly braces] , separating a key from its value with a colon : , and separating the key-value pairs with commas. You can access and update elements using indexing; however, instead of using numerical sequential index numbers, you use the key as the index. You can add new elements simply by assigning a value to a new key.

Reference:

https://stackoverflow.com/questions/34764979/loading-python-string-with-u-as-json?noredirect=1&lq=1

Question: 12

When a Cisco IOS XE networking device is configured using RESTCONF, what is the default data-encoding

method?

A. JSON
B. YAML
C. XML
D. x-form-encoding

Answer: C

Explanation:

The correct answer is C, XML. RESTCONF, a protocol designed for managing network devices via a RESTful API, relies on data encoding formats for transmitting configurations and operational data between the client and the device. While REST APIs can utilize various data encoding methods, including JSON, XML, and YAML, Cisco IOS XE devices, by default, employ XML when interacting through RESTCONF. This default behavior stems from the early adoption and standardized nature of XML within networking protocols, predating the widespread use of JSON and YAML in web development. Though JSON is gaining traction for its simplicity and human readability, many networking vendors and APIs retain XML as the foundational encoding. When a client interacts with a Cisco IOS XE device via RESTCONF and doesn't explicitly specify a different content-type header, the device will default to sending and receiving data encoded in XML format. Specifying
"application/json" or "application/yang" in the HTTP header allows the use of JSON or YANG (with JSON payload) respectively. However, without any specified header, Cisco IOS XE will always assume and uses the XML format, as this was the early standard implementation. This makes the default choice XML, even though RESTCONF supports other encodings. Understanding this default is essential for developers writing scripts or programs that interact with Cisco devices via RESTCONF. Developers must adjust their code to account for data being sent and received in XML unless they actively override this default.

Supporting Links:

Cisco DevNet RESTCONF:https://developer.cisco.com/docs/ios-xe/#!restconf (Look for sections on data encoding or content-type)
IETF RFC 8040 - RESTCONF Protocol:https://datatracker.ietf.org/doc/html/rfc8040 (While this RFC specifies the protocol, you'll find information on content-types and encoding there. It allows for many encodings but doesn't specify defaults per device vendor.)

Question: 14

FILL BLANK -
Fill in the blanks to complete the Python script to request a service ticket using the APIC-EM REST API for the user `devnetuser`. import requests import json controller = 'devnetapi.cisco.com/sandbox/apic_em' url = `https://` + controller + `api/va/ticket` payload = 'username': '_____', 'password': '370940885' header = 'Content-type': 'application.json' response = _____.post(url, data=json.dumps(payload), \ headers= _____, verify=False) r_json = response.json() print(r_json) ticket = r_json[`response`]
[`serviceTicket`] print(ticket)

Answer:

See explanation below.

Explanation:
devnetuser
requests
header

Reference:

https://developer.cisco.com/docs/apic-em/#!hello-world

Question: 15

Which two statements about JSON and XML are true? (Choose two.)

A. The syntax of JSON contains tags, elements, and attributes.
B. XML objects are collections of key-value pairs.
C. JSON objects are collections of key-value pairs.
D. JSON arrays are an unordered set of key-value pairs.
E. The syntax of XML contains tags, elements, and attributes.

Answer: CE

Explanation:

The correct answer is **C and E**. Let's break down why:

C. JSON objects are collections of key-value pairs. This statement is accurate. JSON (JavaScript Object Notation) is fundamentally built around key-value pairs. These pairs represent data where the key (a string) acts as an identifier and the value can be a string, number, boolean, another JSON object, an array, or null. This structure makes it highly suitable for transmitting structured data over networks, particularly in APIs. Think of it like a dictionary where you look up values using the keys.

E. The syntax of XML contains tags, elements, and attributes. This statement is also true. XML (Extensible Markup Language) uses a hierarchical, tag-based structure. Elements are defined using opening and closing tags, like <element>content</element>. Tags can also have attributes that provide additional information about the element, such as <element attribute="value">content</element>. This markup-based approach is useful for representing complex, nested data structures. It's like describing information in a structured document.

Why the other options are incorrect:

A. The syntax of JSON contains tags, elements, and attributes. This is incorrect because tags, elements, and attributes are characteristic of XML, not JSON. JSON uses key-value pairs and arrays.

B. XML objects are collections of key-value pairs. This is incorrect because XML data is structured using tags, elements, and attributes. Key-value pairs are the core structure of JSON, not XML.

D. JSON arrays are an unordered set of key-value pairs. This is incorrect. While JSON objects use key-value pairs, JSON arrays are ordered lists of values. An array can hold numbers, strings, booleans, other JSON objects, or even other arrays, but they maintain their sequence.

In essence, JSON and XML are both data serialization formats, but they differ in syntax and application. JSON is lightweight and better suited for web APIs due to its simplicity, while XML is more verbose and suitable for complex data that needs extensive metadata.

Authoritative Links:

JSON Official Website:https://www.json.org/ - Provides official documentation and resources on JSON.

XML Official Specification:https://www.w3.org/XML/ - From the World Wide Web Consortium, this offers detailed information on the XML standard.

Mozilla Developer Network (MDN) - JSON:https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON - Excellent explanations of JSON, its syntax, and usage in JavaScript.

Mozilla Developer Network (MDN) - XML:https://developer.mozilla.org/en-US/docs/Web/XML - In-depth information on the use of XML in web development.

Question: 16

FILL BLANK -
Fill in the blanks to complete the Python script to request a service ticket using the APIC-EM REST API for the user `devnetuser`. import requests import json controller = `~devnetapi.cisco.com/sandbox/apic_em' url = `https://` + controller + `api/va/ticket` payload = `~username': `~_____', `~password': `~370940885' header = `~Content-type': `~application.json' response = _____.post(url, data=json.dumps(payload), \ headers= _____, verify=False) r_json = response.json() print(r_json) ticket = r_json[`response`] [`serviceTicket`] print(ticket)

Answer:

See explanation below.

Explanation:
devnetuser
requests
header

Reference:
https://developer.cisco.com/docs/apic-em/#!hello-world

Question: 17

Which platform is run directly using a hypervisor?

A.bare metal systems
B.virtual machines
C.containers
D.applications

Answer: B

Explanation:

The correct answer is B, virtual machines (VMs). Hypervisors are fundamental components of virtualization, acting as a software layer that creates and manages VMs. Unlike bare metal systems, which run directly on physical hardware, VMs operate on top of a hypervisor. The hypervisor abstracts the underlying hardware, enabling multiple VMs, each with its own operating system, to run simultaneously on a single physical host. Containers, such as those managed by Docker, do not directly interact with the hypervisor; instead, they share the host operating system's kernel, making them lighter and more resource-efficient than VMs. Applications, on the other hand, represent the software running within these environments (either VMs or containers) or directly on the operating system; they are not directly associated with the hypervisor's operational domain.

The hypervisor, therefore, is what directly interacts with and manages the VMs, making them the entity run through a hypervisor's operations. This virtualization layer allows for resource allocation, isolation, and portability of virtualized environments.

For further research, refer to the following authoritative sources:

1. **VMware's documentation on Hypervisors:** https://www.vmware.com/topics/glossary/content/hypervisor
2. **Microsoft's Azure Documentation on Virtual Machines:**https://learn.microsoft.com/en-us/azure/virtual-machines/
3. **Red Hat's Learning Resources on Virtualization:**https://www.redhat.com/en/topics/virtualization

Question: 18

What is a benefit of organizing code into modules?

A. reduces the length of code
B. enables code to be multifunctional
C. enables the reuse of code
D. improves overall performance

Answer: C

Explanation:

The correct answer is C, "enables the reuse of code." Organizing code into modules, a fundamental principle in software development and relevant to cloud-native applications, promotes code reusability. Modules encapsulate specific functionalities, meaning these self-contained units can be incorporated into different parts of the same project or even entirely separate projects without modification. This contrasts with monolithic codebases where logic is intertwined, hindering reuse.

Reusing modules leads to faster development cycles as developers can leverage existing, tested code instead of rewriting it, saving time and resources. It also enhances maintainability because changes to a module impact only that module, not the entire codebase, thus simplifying debugging and modifications. Moreover, code reuse decreases overall project size by preventing redundant implementations of the same functionality.

Options A and B are less direct benefits. While modular code might seem shorter due to logical grouping, its primary goal isn't length reduction but rather organization and reusability. While modules can perform multiple related functions, their benefit is not in "multifunctionality" itself but in the structured grouping of related actions. Option D, performance, can be affected by modularization, both positively (due to better organization and potentially parallel processing in some scenarios) and negatively (potentially due to increased overhead from module interactions). However, it is not the primary benefit. The primary intent behind creating modules is definitely not performance enhancements, but code reusability and organization.

Therefore, the key advantage of modularization is improved code reusability, enabling the development of robust, maintainable and consistent cloud applications.

Further reading:

Software Modularity:https://en.wikipedia.org/wiki/Modular_programming
Code Reusability:https://www.geeksforgeeks.org/code-reusability/
Modular Design:https://www.tutorialspoint.com/software_engineering/software_design_methodologies.htm

Question: 19

What is a benefit of organizing code into modules?

A.enables the code to be broken down into layers
B.improves collaboration of the development team
C.makes it easier to deal with large and complex systems
D.enables the inclusion of more programming languages in the code

Answer: B

Explanation:

The correct answer is **B. improves collaboration of the development team.** Here's why:

Modularizing code involves breaking down a large program into smaller, self-contained units called modules. Each module ideally handles a specific functionality. This has significant benefits for collaborative development. When code is organized into modules, different developers can work on different modules simultaneously without interfering with each other's work. Each module acts as an independent unit, which allows for easier code management and testing in isolation. This reduced complexity and clearer separation of concerns leads to fewer merge conflicts and improves the overall team's productivity. Furthermore, when using version control systems, committing changes to different modules becomes more organized and straightforward, facilitating parallel development and faster iteration cycles.

While other options have their merit in certain scenarios, they don't directly and primarily address the question of what benefits modular code provides to a development team. Option A (enables the code to be broken down into layers) is related to architectural design rather than team collaboration directly. Option C (makes it easier to deal with large and complex systems) is a general benefit of modularization but doesn't highlight the specific collaboration aspect. And option D (enables the inclusion of more programming languages in the code) is less likely the core reason behind modularization for collaborative purposes, it might be a consequence in some cases.

Modular code promotes team collaboration by allowing developers to focus on specific areas of responsibility, improving concurrent development, and reducing the complexity involved in maintaining a large codebase.

Further reading:

Software Engineering Body of Knowledge (SWEBOK): https://www.computer.org/education/bodies-of-knowledge/software-engineering (Search for sections on modularity and team software engineering). **Martin Fowler's website** on modularity and code design: https://martinfowler.com/ (Search his articles and books for related concepts).
Wikipedia's article on modular programming:https://en.wikipedia.org/wiki/Modular_programming

Question: 20

What is the Git command to delete a local branch named `experiment` without a warning?

A.git branch "rm experiment
B.git branch "n experiment
C.git branch "f experiment
D.git branch "D experiment

Answer: D

Explanation:

The correct Git command to forcefully delete a local branch named experiment without a warning is git branch-D experiment. Let's break down why. Git's branch command is used to manage branches. When deleting a branch, Git normally tries to prevent accidental data loss. If the branch you're trying to delete hasn't been merged, Git will issue a warning. However, sometimes, you explicitly want to remove the branch, even if it hasn't been merged, and you don't want a prompt. The -D option achieves this. It's a shortcut for -d --force. The-d option tries to delete the branch but fails if it's not merged. The --force part overcomes this by deleting the branch regardless of its merge status. Option A, git branch -rm experiment, is not a valid Git command. Option B, git branch -n experiment, also doesn't exist for branch deletion. Option C, git branch -f experiment, while it does have the -f option which is "force" its intended use is for renaming a branch not deletion. Therefore, git branch -D experiment is the only command that will forcefully delete a local branch without any warning or

errors. This provides efficiency in version control workflow by allowing developers to remove unnecessary branches quickly.

For further research, consider these authoritative sources:

Git Branching - Basic Branching and Merging:https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging
git-branch Documentation:https://git-scm.com/docs/git-branch
Atlassian Git Tutorials - Working with Branches:https://www.atlassian.com/git/tutorials/using-branches

Question: 21

What is the outcome of executing this command?
git clone ssh://path/to/my-project.git

A.creates a local copy of a repository called my-project
B.initiates a new Git repository called my-project
C.creates a copy of a branch called my-project
D.creates a new branch called my-project

Answer: A

Explanation:

The command git clone ssh://path/to/my-project.git utilizes the git clone command to create a local copy of a remote Git repository. The ssh: prefix indicates that the repository will be accessed via SSH protocol, ensuring secure communication and authentication. The URL /path/to/my-project.git specifies the location of the remote repository, including the user, server address, and the project path. Consequently, executing this command will result in a full replica of the repository named "my-project" being downloaded to the user's local machine in a directory also called "my-project." This local copy includes all branches, commits, and the entire project history. Option B is incorrect because git clone doesn't initialize a new repository, it copies an existing one. Options C and D are incorrect as they refer to branches which are not the focus of git clone; they operate on an existing repository rather than copying it. The clone command fetches all content from the remote repository, allowing developers to work on it locally. This facilitates code modification, collaboration, and version control. The core function of git clone is to mirror an existing Git repository from a remote location to the local environment.

Further research:

Git documentation on cloning:https://git-scm.com/docs/git-clone
Understanding Git URLs:https://git-scm.com/docs/git-clone#_git_urls_a_id_urls_a **SSH and Git:**https://docs.github.com/en/authentication/connecting-to-github-with-ssh

Question: 22

Which action does the Git command git merge allow the developer to perform?

A.Combine multiple sequences of commits into one unified history
B.Push changes to the remote repository
C.Create, list, rename, and delete branches
D.Switch between branches

Answer: A

Explanation:

The correct answer is **A. Combine multiple sequences of commits into one unified history.**

git merge is a fundamental Git command designed to integrate changes from one branch into another. It achieves this by combining the commit history of the source branch into the target branch. This process results in a single, unified commit history reflecting the incorporated changes. Merging is crucial for collaborative development, allowing developers to integrate their work from separate branches. For instance, feature branches are often merged into the main development branch after a feature is completed,
integrating their changes. While merging can sometimes lead to conflicts, these conflicts highlight differing modifications and require manual resolution before the merge can be finalized. It is important to note that git merge doesn't alter the original branch; it creates a new merge commit in the target branch to record the incorporation of changes. It is a critical process in maintaining a cohesive and traceable project history. Option B is incorrect as git push is the command to send changes to the remote repository. Option C describes actions of the git branch command, and option D corresponds to actions performed by the git checkout command.

For further information on git merge, refer to the official Git documentation:https://git-scm.com/docs/git-mergeand explore resources like:https://www.atlassian.com/git/tutorials/using-branches/git-merge.These resources provide a deeper understanding of the mechanics of git merge and its applications in version control.

Question: 23

How does a developer create and switch to a new branch called `my-bug-fix` to develop a product fix?

A.git checkout -b my-bug-fix
B.git branch -b my-bug-fix
C.git branch my-bug-fix
D.git checkout my-bug-fix

Answer: A

Explanation:

The correct command to create a new branch and immediately switch to it in Git is git checkout -b <branch_name>. In this case, the branch name is my-bug-fix, making git checkout -b my-bug-fix the correct answer. The git checkout command, when used with the -b flag, serves a dual purpose: it creates a new branch with the specified name and then immediately switches the working directory to that newly created branch. This is a common workflow when starting a new feature or, in this case, a bug fix.

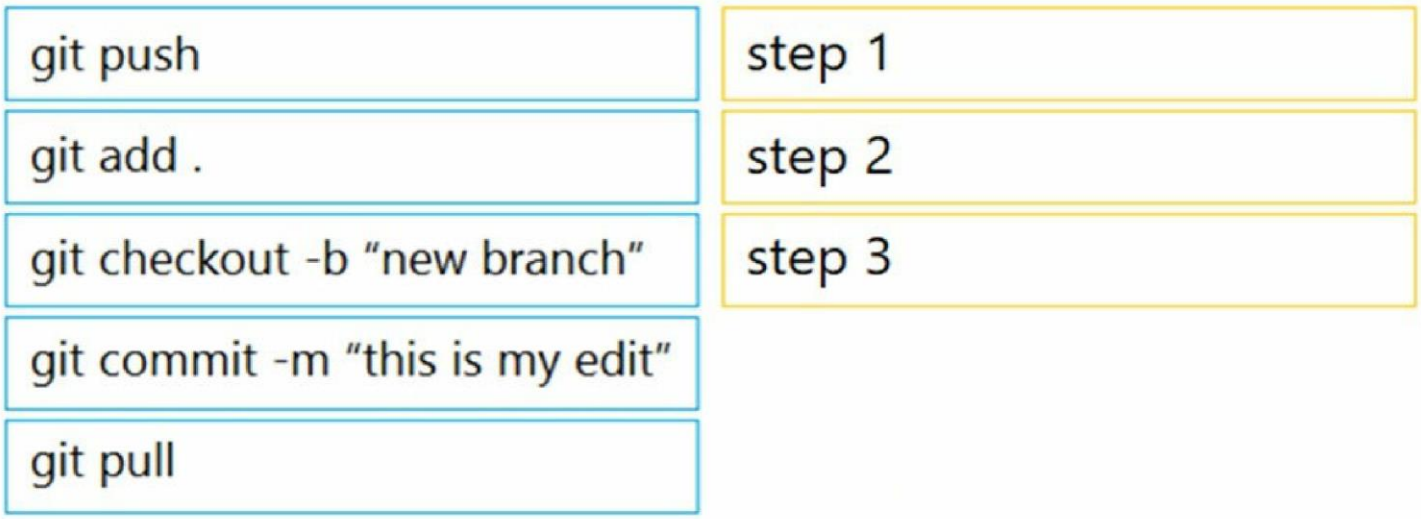
Option B, git branch -b my-bug-fix, is incorrect because while the -b flag can be used with git branch in some older versions of git, its behavior is different. It is not standard and could result in errors or unexpected behaviour. In modern git, -b is specifically associated with git checkout for creating and switching. Option C, git branch my-bug-fix, only creates the new branch but does not switch to it. After executing this command, the developer would still be on the original branch and would need a separate command (git checkout my-bug-fix) to switch. Option D, git checkout my-bug-fix, would only switch to an existing branch named my-bug-fix; if the branch doesn't exist, it will throw an error.

The efficient and most common approach to create a branch and switch to it in one step is using git checkout -b <branch_name>, which aligns with modern Git practices and avoids unnecessary extra commands. This

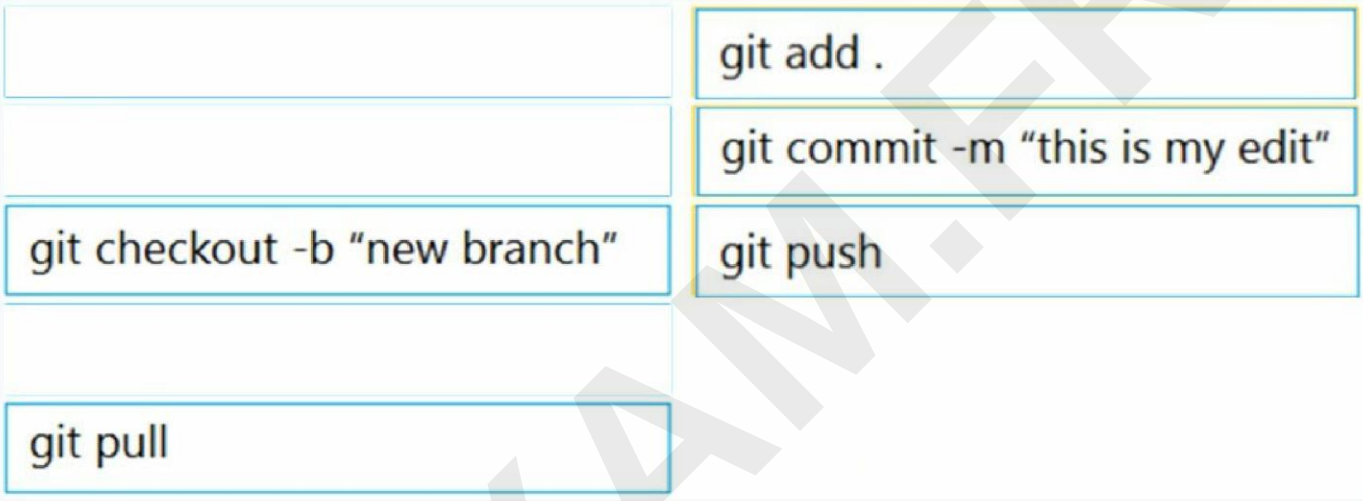
approach streamlines the developer's workflow. This reflects a core element of effective software development, focusing on efficiency and clarity within version control.https://git-scm.com/docs/git-checkouthttps://git-scm.com/docs/git-branch

Question: 24

DRAG DROP -
Drag and drop the Git commands from the left onto the right that add modified local files to a remote repository. Not all options are used.
Select and Place:



Answer:



Explanation:

Step 1: git add .Step 2: git commit -m "this is my edit"Step 3: git push <repository>(if the remote repository is not connect yet, we will need to use a Step 2.5 of "git remote add origin <url>"

Question: 25

A developer needs to prepare the file README.md in the working tree for the next commit operation using Git. Which command needs to be used to accomplish this?

A.git -a README.md
B.git add README.md

C.git add README.md staging

D.git commit README.md

Answer: B

Explanation:

The correct command to prepare a file for the next commit operation in Git is `git add README.md`. This command stages the specified file, `README.md`, meaning it moves the current version of the file from the working directory into the staging area. The staging area acts as an intermediary between the working directory and the Git repository. It is a snapshot of changes that will be included in the next commit. Option A, `git -a README.md`, is incorrect because the `-a` flag is used with `git commit` to stage all modified or deleted files, not `git add`. Option C, `git add README.md staging`, is incorrect because `staging` is not a valid parameter for `git add`. Option D, `git commit README.md`, is also incorrect; `git commit` is used to record changes to the repository, but only for files that have been staged. Therefore, it would not work if `README.md` is not in the staging area. To commit changes, you must first use `git add` to stage them, and then `git commit` to save the staged changes to the repository. This separation of staging and committing allows developers to selectively include changes in specific commits.

Relevant Links for further research:

1. **Git documentation on git add:**https://git-scm.com/docs/git-add
2. **Git documentation on the staging area:**https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository
3. **Atlassian tutorial on Git staging:**https://www.atlassian.com/git/tutorials/saving-changes/git-add

Question: 26

A developer is reviewing a code that was written by a colleague. It runs fine, but there are many lines of code to do a seemingly simple task repeatedly. Which action organizes the code?

A.Refactor the code by removing any unnecessary tests.

B.Reverse engineer and rewrite the code logic.

C.Using functions, rewrite any pieces of code that are repeated.

D.Modify the code to use loops.

Answer: C

Explanation:

The correct answer is **C. Using functions, rewrite any pieces of code that are repeated.** This addresses code redundancy directly. When a task is performed repeatedly with minor variations, encapsulating that logic within a function allows for a single definition to be reused. This promotes the "Don't Repeat Yourself" (DRY) principle, a fundamental concept in software engineering. Functions improve code readability, reduce the chances of introducing errors (because changes are made in one location), and simplify maintenance. Options A, B, and D, while potentially beneficial in other scenarios, do not specifically address the problem of duplicated code. Refactoring to remove unnecessary tests (A) focuses on streamlining logic but doesn't address redundancy. Reverse engineering and rewriting (B) is an extreme measure that shouldn't be needed for a functional (albeit repetitive) code, it's typically used for undocumented or very poorly constructed code, not for refactoring for simple redundancy. While loops (D) can help, they only assist if the code involves repetition based on numeric or list iteration, and are less applicable for logically repeating code blocks. Function abstraction is the most precise tool to address the described situation of repeatedly performing the same action. By converting a piece of code used more than once into a function, the developer then replaces

these code blocks with calls to the function. This keeps the code clean and more manageable. Functions are a fundamental concept in all programming languages, and key to good coding style.

Supporting Resources:

Code Duplication:https://www.perforce.com/blog/qac/what-code-duplication-and-why-it-matter/ (Explains why code duplication is bad)
Don't Repeat Yourself (DRY) Principle:https://en.wikipedia.org/wiki/Don%27t_repeat_yourself (Explains the DRY principle)
Functions in Programming:https://www.geeksforgeeks.org/functions-in-programming/ (Basic understanding of functions)

Question: 27

Which principle is a value from the manifesto for Agile software development?

A.processes and tools over teams and interactions
B.detailed documentation over working software
C.adhering to a plan over responding to requirements
D.customer collaboration over contract negotiation

Answer: D

Explanation:

The correct answer is **D. customer collaboration over contract negotiation**. The Agile Manifesto, a foundational document in modern software development, emphasizes valuing individuals and interactions, working software, customer collaboration, and responding to change. This selection highlights a core principle where actively involving the customer throughout the development process is more important than rigidly adhering to contractual agreements. Agile methodologies prioritize frequent feedback loops, enabling teams to adjust to evolving customer needs and deliver more valuable solutions. Options A, B, and C represent the "left" side of the manifesto's stated values. Option A goes against the value "individuals and interactions over processes and tools". Option B is the opposite of "working software over comprehensive documentation".

Option C similarly contrasts with the value "responding to change over following a plan". Agile's focus on flexibility and iterative development promotes building products that truly meet user demands. Focusing on customer collaboration allows for a more adaptive and successful development process. The prioritization ensures a mutual understanding of the project goals and ensures the end product is aligned with expectations. This collaborative approach avoids misinterpretations and fosters a higher degree of satisfaction.

For further reading and understanding, you can visit:

Agile Manifesto
12 Principles of Agile

Question: 28

Which advantage does the agile process offer compared to waterfall software development?

A.to add or update features with incremental delivery
B.to view the full scope of end-to-end work
C.to have each phase end before the next begins

D.to fix any issues at the end of the development cycle

Answer: A

Explanation:

Justification:

The correct answer is **A. to add or update features with incremental delivery**. Agile methodologies, unlike waterfall, prioritize iterative development. This means software is built and delivered in small, functional increments or "sprints." This approach allows for continuous feedback from stakeholders and customers, enabling teams to adapt to changing requirements and incorporate new features or modifications easily throughout the project lifecycle. Waterfall, on the other hand, follows a rigid, sequential process where each phase (requirements, design, implementation, testing, deployment) must be completed before the next begins. This makes it difficult and costly to accommodate changes after the initial requirements phase. Option B, viewing the full scope upfront, is a characteristic of waterfall, not agile. Option C, sequential phase completion, is also a waterfall principle. Option D, fixing issues only at the end, is problematic and inefficient, often associated with waterfall's late integration and testing phases, while agile encourages early and frequent testing. Agile's incremental approach drastically reduces the risk of building something that doesn't meet user needs, facilitating quicker time-to-market, higher customer satisfaction and making it a highly suitable choice for the dynamic nature of cloud-based software development. Agile's flexibility and responsiveness are critical in the constantly evolving cloud landscape, where rapid changes and updates are commonplace.

Authoritative Links:

1. **Agile Alliance:**https://www.agilealliance.org/agile101/ - Provides a comprehensive overview of agile principles and methodologies.
2. **Atlassian (Agile):**https://www.atlassian.com/agile - Offers detailed information and resources about agile project management.
3. **Scrum.org:**https://www.scrum.org/ - A key resource for understanding the Scrum framework, a popular agile methodology.
4. **TechTarget (Waterfall vs Agile):** https://www.techtarget.com/searchsoftwarequality/definition/Waterfall-model & https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development - Offers clear definitions and comparisons of waterfall and agile approaches.

Question: 29

How do XML and JSON compare regarding functionality?

A.XML provides more support for mapping data structures into host languages than JSON.
B.XML provides more human readability than JSON.
C.JSON provides less support for data types than XML.
D.JSON natively supports arrays and XML does not natively support arrays.

Answer: B

Explanation:

The correct answer is **B. XML provides more human readability than JSON.**

While both XML and JSON are used for data serialization and exchange, their structure and design differ, leading to varying strengths. XML utilizes a verbose, tag-based structure with explicit start and end tags,

resembling HTML. This explicit markup, while adding overhead, also enhances readability for humans. The hierarchical structure is clear, with nested tags providing a visual guide to the data organization. JSON, in contrast, employs a key-value pair structure, making it concise and compact. JSON's use of brackets and braces can be slightly less visually apparent to a human when compared to XML's explicit tags.

Although both formats support encoding data, the human-readable factor is one of the defining distinctions.

XML was designed with the human-readable aspect more consciously in its design, whilst JSON focused on efficient machine parsing. However, both XML and JSON provide robust support for mapping data to host languages. In modern software development, both formats are very widely used and mature, with very advanced tools that exist for both. While XML uses schema definitions for data types, JSON also does support various data types like strings, numbers, booleans, objects, and arrays natively. Both formats handle data in a structured manner, including the use of arrays.

Therefore, while both serve similar purposes, the distinct structural approach of XML makes it more inherently readable for human consumption when compared to JSON.

Authoritative Links for further research:

W3C XML:https://www.w3.org/XML/
JSON:https://www.json.org/
Differences between XML and JSON:https://www.tutorialspoint.com/xml/xml_vs_json.htm

Question: 30

What are two principles of an infrastructure as code environment? (Choose two.)

A.Components are coupled, and definitions must be deployed for the environment to function.
B.Redeployments cause varying environment definitions.
C.Environments must be provisioned consistently using the same inputs.
D.Service overlap is encouraged to cater for unique environment needs.
E.Complete complex systems must be able to be built from reusable infrastructure definitions.

Answer: CE

Explanation:

Here's the justification for why options C and E are the correct principles of Infrastructure as Code (IaC):

C. Environments must be provisioned consistently using the same inputs: This principle emphasizes the core benefit of IaC: **repeatability and consistency**. IaC tools use configuration files to define infrastructure, ensuring that identical environments can be created repeatedly from the same code. This eliminates the inconsistencies and errors that often arise from manual configuration, making deployments predictable and reliable. When changes are made to the infrastructure code, those changes can be consistently applied to existing or new environments by redeploying the code, ensuring uniformity across deployments.

E. Complete complex systems must be able to be built from reusable infrastructure definitions: IaC encourages the creation of modular and reusable infrastructure components. This allows developers and operations teams to assemble complex systems from pre-built building blocks. This promotes efficiency, reduces duplication, and makes maintenance and modification of infrastructure significantly easier.

Reusability also encourages a consistent approach across various projects and teams by standardizing infrastructure elements. For example, you can define a standard web server configuration as reusable code and consistently use it across various applications.

Options A, B, and D are incorrect due to the following reasons:

A. Components are coupled, and definitions must be deployed for the environment to function: While components in a system built by IaC are often coupled, the principle of IaC is based around decoupling the definitions from the application environment. You define your infrastructure separately from application code to manage it.

B. Redeployments cause varying environment definitions: The goal of IaC is the opposite; redeployments should result in identical environment configurations. Any differences in redeployment indicate issues, such as poorly defined IaC or configuration drifts.

D. Service overlap is encouraged to cater for unique environment needs: IaC promotes standardization, not overlapping services. Overlap leads to inefficiencies, cost increases, and unnecessary complexities.

Authoritative Links for Further Research:

1. **HashiCorp's Introduction to Infrastructure as Code:** https://www.hashicorp.com/resources/introduction-to-infrastructure-as-code
2. **AWS Documentation on Infrastructure as Code:**https://aws.amazon.com/what-is/infrastructure-as-code/
3. **Microsoft Azure documentation on IaC:**https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/considerations/infrastructure-as-code
4. **Google Cloud Platform's definition of IaC:**https://cloud.google.com/solutions/infrastructure-as-code-iac

Question: 31

Which two encoding formats do YANG interfaces support? (Choose two.)

A. XML
B. JSON
C. XHTML
D. BER
E. plain text

Answer: AB

Explanation:

YANG (Yet Another Next Generation) is a data modeling language used to define the structure of configuration and state data for network devices. This model is then used to programmatically manage and automate network devices using protocols like NETCONF and RESTCONF. The key to accessing and manipulating this data lies in the encoding format used to represent it during transmission.

XML (Extensible Markup Language) is a highly structured, self-describing markup language that excels at hierarchical data representation. It is a popular choice for representing complex data structures and is readily parsed and processed by various tools. JSON (JavaScript Object Notation) is a lightweight, human-readable format that uses key-value pairs. Its simplicity and widespread support in web technologies make it ideal for API interactions.

YANG interfaces specifically support these two encoding formats due to their suitability for structured data exchange and ease of integration with existing network management and automation tools. While XHTML is an XML-based markup language for representing web pages, it is not used as a general data encoding format for network configuration. BER (Basic Encoding Rules) is an encoding scheme for Abstract Syntax Notation One (ASN.1) and isn't used with YANG directly. Plain text, while simple, is not suitable for representing structured data in a way that is both machine-readable and easily parsed for complex configurations.

Therefore, XML and JSON are the two encoding formats supported by YANG interfaces.

For further research, consult the following resources:

RFC 7950 - The YANG 1.1 Data Modeling Language:https://datatracker.ietf.org/doc/html/rfc7950 (This is the defining document for the YANG language.)
RFC 8040 - RESTCONF Protocol:https://datatracker.ietf.org/doc/html/rfc8040 (This document describes how RESTCONF uses JSON and XML to access YANG data.)
Understanding YANG:https://www.cisco.com/c/en/us/solutions/enterprise-networks/automation-programmability/index.html (This Cisco page explains more about programmability and YANG usage within their devices.)

Question: 32

Refer to the exhibit.

```
{
  "items": [{
      "kind": "object#NetworkObj",
      "selfLink": "https://10.201.230.5/api/objects/networkobjects/db01",
      "name": "db01",
      "host": {
         "kind": "IPv4address",
         "value": "172.16.0.11"
      },
      "objectId": "db01"
  }]
}
```

The JSON data in the exhibit has been parsed and stored into a variable `data`. What returns the value `172.16.0.11`?

A.data['items']['host']['value']
B.data['items'][1]['host']['value']
C.data['items'][0]['host']['value']
D.data['items']['host'][1]

Answer: C

Explanation:

C is correct - ITEMS - is a dictionary with a list as its value. This list has only one entry though - hence you need the positional arg 0 to target the nested K/V pairs.

Question: 33

Refer to the exhibit.

```
def get_result()
        url = "https://sandboxdnac.cisco.com/dna/system/api/v1/auth/token"
        resp = requests.post(url, auth=HTTPBasicAuth(DNAC_USER, DNAC_PASSWORD))
        result = resp.json()['Token']
        return result
```

What does the Python function do?

A.It returns HTTP Basic Authentication.
B.It returns DNAC user and password.
C.It reads a token from a local JSON file and posts the token to the DNAC URL.
D.It returns an authorization token.

Answer: D

Explanation:

Option A, "It returns HTTP Basic Authentication," is incorrect. While the function does use HTTP Basic Authentication to authenticate the request, it does not return the authentication itself. Option B, "It returns DNAC user and password," is also incorrect. The function does not return the DNAC_USER and DNAC_PASSWORD variables. Option C, "It reads a token from a local JSON file and posts the token to the DNAC URL," is also incorrect. The function does not read a token from a local JSON file and does not post a token to the DNAC URL. Instead, it makes an HTTP POST request to the DNAC URL and parses the response from the server as JSON.

D.It does not POST anything, it returns a token

Question: 34

Package updates from a local server fail to download. However, the same updates work when a much slower external repository is used. Why are local updates failing?

A. The server is running out of disk space.
B. The Internet connection is too slow.
C. The Internet is down at the moment, which causes the local server to not be able to respond.
D. The update utility is trying to use a proxy to access the internal resource.

Answer: D

Explanation:

The most likely reason for local package update failures when external updates succeed, despite slower speeds, is that the update utility is configured to use a proxy to access the internal resource (Option D). This means that the utility first attempts to connect through the specified proxy, which might be improperly configured for the local network or simply not required for local communication. Consequently, the request to the local server fails because it's routed through a misconfigured or absent proxy. When using an external repository, the proxy configuration may be correct or the utility might be configured to bypass the proxy for external connections, hence the success. Options A, B, and C are less likely. While disk space issues (A) can

cause update problems, it would not explain why the same packages work from an external source. A slow internet connection (B) would affect external updates more, not local ones. Similarly, a complete internet outage (C) would stop the external updates, not the local ones; local update failures should not depend on general internet availability if the server is local. Essentially, the key difference is how the connection is routed. If the utility's proxy settings are aimed towards the internet, it will fail to access an internal server.

For further information on proxies, configurations and troubleshooting:

1. **Proxy Servers:** Check articles on proxy server functions and configurations in IT environments. https://www.cloudflare.com/learning/network-layer/what-is-a-proxy-server/
2. **Package Management:** Refer to documentation of the specific package manager (e.g. apt, yum, pip) for proxy configuration instructions.
3. **Network Troubleshooting:** Network troubleshooting guides can help diagnose routing issues related to proxies. https://support.microsoft.com/en-us/windows/troubleshooting-network-connection-problems-in-windows-586742f1-3e34-b2b9-96e8-8b937b09f234

Question: 35

What is a functionality of the Waterfall method as compared to the Agile method for software development?

A.Waterfall increases agility to implement faster while Agile promotes reliability.

B.A phase begins after the previous phase has ended in Waterfall while Agile phases run in parallel.

C.Customers get feedback during the process in Waterfall while they can see the result at the end in Agile.

D.Requirements can be updated in Waterfall while in Agile it should be gathered in the beginning.

Answer: B

Explanation:

The correct answer is **B. A phase begins after the previous phase has ended in Waterfall while Agile phases run in parallel.**

Here's a detailed justification:

The Waterfall methodology is a sequential, linear approach to software development. Each phase (requirements, design, implementation, testing, deployment, maintenance) must be completed before the next phase can begin. This rigid structure is its defining characteristic, meaning there's no overlap and each phase is a prerequisite for the following one. This contrasts sharply with Agile methodologies. Agile, on the other hand, is iterative and incremental. It operates in short cycles called sprints or iterations. These sprints involve overlapping activities from all phases of the development lifecycle – planning, design, development, testing, and deployment. This parallel execution allows for greater flexibility, adaptability, and faster feedback loops.

Option A is incorrect because Waterfall is known for its inflexibility and lack of agility, while Agile prioritizes rapid adaptation. Option C is incorrect because the Waterfall method primarily provides a final product to the customer after the whole process is over. Agile incorporates user feedback frequently during iterations. Option D is incorrect as the Waterfall methodology is designed to avoid changing requirements mid process and Agile is flexible to adapt to changing requirements.

Therefore, the fundamental difference lies in the sequential nature of Waterfall versus the iterative and parallel nature of Agile. Option B accurately captures this distinction.

Authoritative Links for further research:

Agile Alliance:https://www.agilealliance.org/agile101/ (Provides comprehensive resources on Agile methodologies)
Atlassian - Waterfall vs Agile:https://www.atlassian.com/agile/methodologies/waterfall (A comparison of Waterfall and Agile methodologies)
Project Management Institute (PMI):https://www.pmi.org/ (Offers project management resources covering various methodologies)

Question: 36

What is a benefit of using functions in the code for the development process?

A.better user experience in the end product
B.improves code performance
C.easier to compile the code
D.faster code development

Answer: D

Explanation:

The correct answer, **D. faster code development**, is a primary advantage of employing functions in software development. Functions, in essence, encapsulate reusable blocks of code designed to perform specific tasks.

This modular approach drastically reduces the need to write the same logic repeatedly, streamlining the coding process. By breaking down complex problems into smaller, manageable function-based units, developers can work on different parts of the system concurrently and with greater clarity. This leads to faster feature implementation and quicker iteration cycles. Furthermore, functions facilitate testing and debugging; since each function handles a specific task, it becomes easier to isolate and fix issues. A well-structured code base that utilizes functions promotes maintainability and readability, which benefits teams working on a large project. This speeds up future development and modification. While other options may have some indirect relation to coding, they are not core advantages of using functions. For example, option B, improves code performance, is only sometimes true with functions, they may also add a small overhead.

Functions primarily focus on organization and reusability, directly contributing to faster code development rather than necessarily impacting the end user experience (A) or compile process (C). Therefore, the use of functions primarily helps developers with speeding up the coding process by reducing redundant code and creating a more organised structure.

For further research, consider exploring these authoritative resources:

Python Functions:https://docs.python.org/3/tutorial/controlflow.html#defining-functions - Learn how functions are defined and used in Python, a widely adopted programming language.
Code Reusability:https://en.wikipedia.org/wiki/Code_reuse - This Wikipedia page provides information on the general concept of code reusability, which functions directly enable.
Modular Programming:https://www.geeksforgeeks.org/modular-programming/ - A detailed introduction to modular programming, a concept closely linked to the advantages of using functions.

Question: 37

What is the difference between YAML and JSON data structure?

A. YAML uses spaces; JSON uses parentheses
B. YAML uses indentation; JSON uses brackets and braces
C. YAML uses brackets and braces; JSON uses indentation

D. YAML uses parentheses; JSON uses spaces

Answer: B

Explanation:

The correct answer is **B. YAML uses indentation; JSON uses brackets and braces.**

YAML (YAML Ain't Markup Language) and JSON (JavaScript Object Notation) are both popular data serialization languages, but they employ different syntaxes to represent structured data. YAML prioritizes human readability, using indentation to define hierarchical relationships between elements. This indentation, typically with spaces, dictates the nesting of data, making it visually intuitive to grasp the structure. Consider a list within a dictionary: in YAML, this is achieved through indented lines under the dictionary key.

In contrast, JSON uses a more rigid syntax based on brackets ([]) for arrays and braces {} for objects. Data structure is clearly defined using these delimiters, along with commas to separate elements and colons to denote key-value pairs. JSON's compact syntax is well-suited for machine parsing and is widely used for data transmission over the web due to its efficiency and ubiquitous support. The lack of reliance on whitespace for structure makes it less human-friendly than YAML but very fast for computer processing. JSON doesn't allow comments, while YAML does, making YAML preferable when human-written documentation is necessary alongside data.

In essence, YAML aims for human-readability through indentation, whereas JSON prioritizes machine parsing with its brace and bracket syntax. Both achieve the same goal of representing structured data, but their approach is fundamentally different. These differences are critical when choosing a data serialization format for various cloud computing tasks, such as configuration management, API communication, or data storage.

The choice between YAML and JSON depends on the specific use case and whether human readability or efficient machine processing is paramount.

Authoritative Links for Further Research:

YAML:https://yaml.org/
JSON:https://www.json.org/
Comparison of YAML and JSON:https://www.tutorialspoint.com/yaml/yaml_vs_json.htm

Question: 38

A developer is working on a feature for a new application. The changes in the existing branch named 'feat00304' must be integrated into a single commit with the current working primary branch named 'prodapp411926287'. Which git command must be used?

A.git rebase --merge feat00304
B.git merge --squash feat00304
C.git push --rebase feat00304
D.git checkout --squash feat00304

Answer: B

Explanation:

The correct answer is **B. git merge --squash feat00304**. Here's why:

The developer needs to combine changes from the feat00304 branch into a single commit on the prodapp411926287 branch. The --squash option in git merge accomplishes this. It takes all changes from the feat00304 branch and stages them as changes within the prodapp411926287 branch without actually merging

the branch history. This means the commit history remains linear on prodapp411926287, and a single commit will be created containing all changes.

Option A, git rebase --merge feat00304, is incorrect as rebase primarily rewrites history, and --merge isn't a standard option for rebase. It's used to move or combine commits to make a cleaner history, and doesn't achieve the single commit requirement. Option C, git push --rebase feat00304, is incorrect because push is for sending commits to a remote repository, and --rebase here is used when you need to rebase your branch against the remote. Option D, git checkout --squash feat00304, is incorrect because checkout is used for switching branches or restoring files; the --squash option is not valid for git checkout. Instead, the combination of git merge with the --squash flag allows us to incorporate the changes in a single commit, which fits the single commit requirement.

The git merge --squash approach helps keep the primary branch history cleaner and more concise by avoiding numerous commits from feature branches. The developer then has the opportunity to add a clear and concise commit message for this single aggregated change. This strategy is a best practice for managing feature integrations.

For further research on git merge --squash, you can refer to:

Git Documentation:https://git-scm.com/docs/git-merge
Atlassian Tutorial:https://www.atlassian.com/git/tutorials/merging-vs-rebasing (While focusing on rebasing vs. merging, it touches upon the benefits of squashing merges)
GitHub Docs:https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/incorporating-changes-from-a-pull-request/about-squash-merging (Explains squash merging in the context of pull requests, similar to git merge --squash)

Question: 39

A file that already exists in a local repository is updated. Which command must be executed to ensure that the changes in the file are included in the next Git commit?

A.git update
B.git merge
C.git add
D.git rebase

Answer: C

Explanation:

The correct answer is **C. git add**. When a file in a Git repository is modified, Git doesn't automatically stage these changes for a commit. It keeps track of the modifications but requires explicit instruction to include them in the next commit. The git add command is the tool that performs this action, moving modified files from the working directory to the staging area. Think of the staging area as a preparation zone for your next commit; it allows you to selectively choose which changes to include. git update is not a standard Git command. git merge combines branches, and git rebase rewrites commit history, neither of which is used for staging modified files. Therefore, the git add command, specifically with either a filename (e.g. git add filename.txt) or the command git add . to stage all changed files, is essential to get a changed file ready for committing. This process of staging is a fundamental component of the Git workflow, allowing for granular control over commit content and thereby enabling more accurate and descriptive project histories. The changes are not tracked in a commit until git add stages them. After git add, git commit will then package these staged modifications into a new commit.

Here are some authoritative resources for further research:

Git Documentation on git add:https://git-scm.com/docs/git-add
Atlassian Tutorial on Staging:https://www.atlassian.com/git/tutorials/saving-changes/git-add
GitHub Docs on Adding Changes:https://docs.github.com/en/get-started/using-git/adding-and-committing-changes

Question: 40

Into which type of Python data structure should common data formats be parsed?

A. sequence
B. set
C. dictionary
D. list

Answer: C

Explanation:

The correct answer is C, a dictionary. Common data formats like JSON and YAML, which are prevalent in cloud computing for configuration files and API responses, are best represented in Python using dictionaries. These formats inherently involve key-value pairs. JSON objects and YAML mappings directly translate to Python dictionaries, where each key is a string (typically) and the associated value can be a variety of data types, such as strings, numbers, lists, or even nested dictionaries. Parsing these formats into a dictionary structure provides immediate access to specific data elements using their corresponding keys, aligning with how the data is structured in the original format. This direct mapping simplifies data manipulation and retrieval. Using lists or sets would require additional steps to establish the key-value relationships inherent in the data.

Sequences (which encompass lists, tuples, and strings) are general purpose containers, but lack the explicit key access offered by dictionaries. A dictionary's organization directly mirrors the hierarchical or key-based access patterns needed when working with JSON or YAML, allowing programmers to immediately leverage the data. Furthermore, Python's built-in json and yaml modules conveniently decode these formats directly into dictionaries, making this the most efficient and intuitive approach. This is standard practice in software development. Therefore, dictionaries provide the natural and most convenient representation for parsed data from common cloud computing configuration and data exchange formats.

Further research on Python dictionaries and data format parsing can be found at these authoritative links:

Python Dictionaries:https://docs.python.org/3/tutorial/datastructures.html#dictionaries
Python JSON Module:https://docs.python.org/3/library/json.html
Python YAML Module (PyYAML):https://pyyaml.org/wiki/PyYAMLDocumentation

Question: 41

Several teams at a company are developing a new CRM solution to track customer interactions with a goal of improving customer satisfaction and driving higher revenue. The proposed solution contains these components: * MySQL database that stores data about customers
* HTML5 and JavaScript UI that runs on Apache
* REST API written in Python
What are two advantages of applying the MVC design pattern to the development of the solution? (Choose two.)

A. to enable multiple views of the same data to be presented to different groups of users
B. to provide separation between the view and the model by ensuring that all logic is separated out into the

controller

C. to ensure data consistency, which requires that changes to the view are also made to the model

D. to ensure that only one instance of the data model can be created

E. to provide only a single view of the data to ensure consistency

Answer: AB

Explanation:

The correct answer is **A and B**. The Model-View-Controller (MVC) design pattern offers significant advantages in developing complex applications like the CRM solution described. Option A, enabling multiple views of the same data, is a core strength of MVC. The model stores the data (e.g., customer information from the MySQL database). The view is responsible for presenting this data to the user, and because MVC separates them, you can have different views (e.g., a customer service view, a sales view) that show the same data in tailored formats. Option B, separation of view and model by isolating logic in the controller, is another crucial aspect of MVC. The controller acts as an intermediary, receiving user input, updating the model, and selecting the appropriate view. This separation enhances maintainability, allowing developers to modify components independently. Option C is incorrect because the MVC pattern does not guarantee automatic data
synchronization. Data consistency needs to be handled explicitly through the controller. Option D is also incorrect. MVC is not designed to restrict data model instantiation. Option E is incorrect as MVC's goal is to allow multiple views on the same data for different purposes. The use of MVC makes the system more scalable and maintainable as a result.

https://en.wikipedia.org/wiki/Model–view–controllerhttps://www.geeksforgeeks.org/model-view-controller-mvc-architecture/

Question: 42

```
module ietf-interface {
  namespace "urn:ietf:params:xml:ns:yang:ietf-interfaces";
  prefix if;
  import ietf-yang-types {
    prefix yang;
  }
  container interfaces-state {
    list interface {
      key "name";
      leaf name {
        type string;
      }
      leaf admin-status {
        type enumeration {
          enum up {
            value 1;
          }
          enum down {
            value 2;
          }
          enum testing {
            value 3;
          }
        }
      }
      leaf if-index {
        type int32 {
          range "1..2147483647"
        }
      }
      container statistics {
        leaf in-octets {
          type yang:counter64;
        }
        leaf in-unicast-pkts {
          type yang:counter64;
        }
      }
    }
  }
}
```

Refer to the exhibit. Which XML snippet has interface information that conforms to the YANG model? A.

```
<interfaces-state
xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces">
    <interface name="GigabitEthernet1">
        <admin-status>1</admin-status>
        <if-index>1</if-index>
        <statistics>
            <in-octets>408164820</in-octets>
            <in-unicast-pkts>728061</in-unicast-pkts>
        </statistics>
    </interface>
</interfaces-state>
```

B.

```
<interfaces-state
xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces">
    <interface>
        <name>GigabitEthernet1</name>
        <admin-status>up</admin-status>
        <if-index>1</if-index>
        <in-octets>408164820</in-octets>
        <in-unicast-pkts>728061</in-unicast-pkts>
    </interface>
</interfaces-state>
```

C.

```
<interfaces-state
xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces">
    <interface>
        <name>GigabitEthernet1</name>
        <admin-status>up</admin-status>
        <if-index>1</if-index>
        <statistics>
            <in-octets>408164820</in-octets>
            <in-unicast-pkts>728061</in-unicast-pkts>
        </statistics>
    </interface>
</interfaces-state>
```

D.

```
<interfaces-state
xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces"
xmlns:if="urn:ietf:params:xml:ns:yang:ietf-interfaces">
    <interface>
        <name>GigabitEthernet1</name>
        <admin-status>1</admin-status>>
        <if-index>1</if-index>
        <statistics>
            <in-octets>408164820</in-octets>
            <in-unicast-pkts>728061</in-unicast-pkts>
        </statistics>
    </interface>
</interfaces-state>
```

Answer: C

Explanation:

Answer is C. Watch "Getting NETCONF Data with NCCLIENT" video on CBT Nuggets.

Question: 43

What is a benefit of version control?

A. prevents two users from working on the same file
B. keeps track of all changes to the files
C. prevents the sharing of files
D. keeps the list of data types used in the files

Answer: B

Explanation:

The correct answer is B: keeps track of all changes to the files. Version control systems (VCS) like Git are fundamentally designed to meticulously record modifications made to files over time. This functionality allows developers to revert to previous states of a project, compare changes across different versions, and understand the evolution of the codebase. Option A, while sometimes facilitated by version control through branching and merging strategies, is not the core purpose; it is more of a workflow consideration. Option C is incorrect; version control actually encourages and simplifies file sharing, collaboration, and contribution among team members, especially in distributed teams, and doesn't inherently limit sharing. Option D is also incorrect; version control does not manage data types. Instead, it focuses on tracking the changes made within the file itself, regardless of the data type present. The primary benefit of a VCS is to provide an auditable history of a project's development, enabling efficient collaboration, experimentation, and recovery from errors. Branching, merging, and conflict resolution are features built upon this core functionality. For further information, you can refer to the Git documentation: https://git-scm.com/doc or resources explaining the principles of version control, such as those found on Atlassian's website : https://www.atlassian.com/git/tutorials/what-is-version-control.

Question: 44

```
git clone git://git.kernel.org/.../git.git my.git
cd my.git
git branch -d -r origin/todo origin/html origin/man   (1)
git branch -D test                                    (2)
```

Refer to the exhibit. What does the command marked (2) do when it is run?

A. It duplicates the test branch.
B. It deletes the test branch only if a new branch is created.
C. It deletes the test branch.
D. It does not delete the branch until it is merged.

Answer: C

Explanation:

C.git branch -D <branchname> = force delete branch name

Question: 45

What is a comparison of YAML and JSON?

A.YAML has a more consistent approach to representing data compared to JSON.
B.JSON does not support comments and YAML does.
C.YAML is a more verbose data structure compared to JSON.
D.JSON has more common usage in configuration management tools compared to YAML.

Answer: B

Explanation:

The correct answer is **B. JSON does not support comments and YAML does.**

JSON (JavaScript Object Notation) is a lightweight data-interchange format known for its simplicity and ease of parsing. However, a key limitation of JSON is its lack of support for comments. This can make it harder to understand complex JSON structures, especially for humans, and necessitates external documentation. YAML (YAML Ain't Markup Language), on the other hand, explicitly allows comments using the '#' symbol. This feature enhances readability and allows developers to include explanatory notes directly within the data files, improving maintainability and collaboration. While both are human-readable, YAML's comment capability provides a distinct advantage in terms of documenting configuration and data. Option A is incorrect as both are consistent in representing data based on structures like key-value pairs and lists. Option C is incorrect because YAML is generally considered less verbose than JSON due to its reliance on indentation and fewer structural symbols. Option D is also incorrect as YAML is increasingly prevalent in configuration management tools, such as Ansible, due to its human-readable nature. JSON tends to be more widely used in web-based APIs due to its direct compatibility with JavaScript.

For further research, refer to:

JSON official documentation:https://www.json.org/

YAML official website:https://yaml.org/
Comparison of YAML and JSON on GeeksforGeeks:https://www.geeksforgeeks.org/difference-between-json-and-yaml/

Question: 46

Which status code is used by a REST API to indicate that the submitted payload is incorrect?

A. 400
B. 403
C. 405
D. 429

Answer: A

Explanation:

The correct HTTP status code to indicate that a submitted payload is incorrect in a REST API is 400 Bad Request. This code signals to the client that the server cannot or will not process the request due to something perceived as a client error. Specifically, a 400 status signifies that the request syntax, framing, or routing is invalid, meaning the data within the request, like a JSON payload, doesn't conform to the API's expectations. For example, it could indicate a missing required field, incorrect data type, or invalid format.

This differs from other 4xx codes. A 403 Forbidden implies the client lacks the authorization to access the requested resource, not a problem with the payload. A 405 Method Not Allowed indicates that the requested method (e.g., GET, POST) is not supported for that resource. Finally, a 429 Too Many Requests is triggered when the client has exceeded its rate limit. These codes deal with authorization, method constraints, and rate limitations, not with malformed data within the request. The 400 code is the industry standard for flagging errors directly caused by the submitted client data. It is the correct code to respond with when the data in the JSON payload is incorrect. Using the correct status codes ensures that developers properly handle API errors and implement robust error handling. The consistent usage of these codes facilitates clear communication between clients and servers within the API interaction.

Further reading:

MDN Web Docs on HTTP status codes:https://developer.mozilla.org/en-US/docs/Web/HTTP/Status - This provides comprehensive details about various HTTP status codes including 400.

RFC 7231 Section 6.5.1 - 400 Bad Request:https://datatracker.ietf.org/doc/html/rfc7231#section-6.5.1 - This is the official specification for the 400 status code.

Question: 47

DRAG DROP -
Refer to the exhibits.

HTTP REQUEST

`POST /organizations/ {organizationId}/networks`

PARAMETERS

name
The name of the new network.

type
The type of the new network. Valid types are wireless, appliance, switch, systemsManager, camera, cellularGateway, or a space-separated list of those for a combined network.

tags
A space-separated list of tags to be applied to the network.

timeZone
The timezone of the network. For a list of allowed timezones, please see the 'TZ' column in the table in this article.

copyFromNetworkId
The ID of the network to copy configuration from. Other provided parameters will override the copied configuration, except type which must match this network's type exactly.

disableMyMerakiCom
Disables the local device status pages (my.meraki.com, ap.meraki.com, switch.meraki.com, wired.meraki.com). Optional (defaults to false).

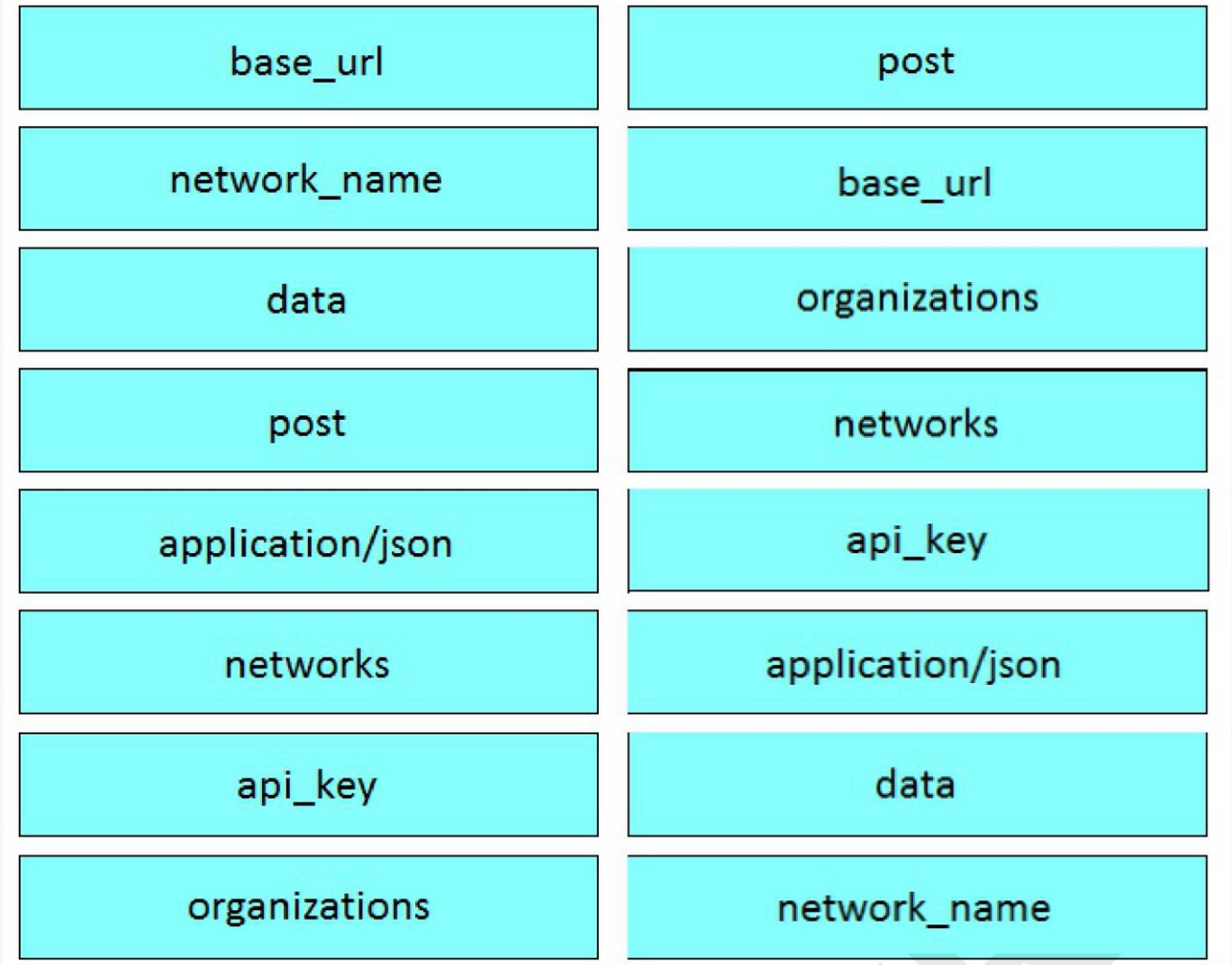
disableRemoteStatusPage
Disables access to the device status page (http://[device's LAN IP]). Optional. Can only be set if disableMyMerakiCom is set to false.

```
base_url = https://api.meraki.com/api/v0
org_id = 123456789
api_key = 1098hadpfsiapsf8ahf8ohp
network_name = "New Network"
requests.<item 1> (
    <item 2> + "/<item 3>/" + org_id + "/<item 4>",
    headers = {
        "X-Cisco-Meraki-API-Key": "<item 5>",
        "Content-Type": "<item 6>"
    },
    <item 7>=json.dumps ({ "name": <item 8>,
                           "type": "wireless switch"
                        }))
```

Drag and drop the code from the left onto the item numbers on the right to complete the Meraki Python script shown in the exhibit.
Select and Place:

base_url	<item 1>
network_name	<item 2>
data	<item 3>
post	<item 4>
application/json	<item 5>
networks	<item 6>
api_key	<item 7>
organizations	<item 8>

Answer:



Explanation:

the answer is :requests.post (base_url+"/organizations/" + org_id +"/networks/", headers = "X-Cisco-Meraki-API-Key": "api_key", "Content-Type" : "application/json" ,data = json.dumps("name": network_name,"type" :"wireless switch"))

Question: 48

An application calls a REST API and expects a result set of more than 550 records, but each time the call is made, only 25 are returned. Which feature limits the amount of data that is returned by the API?

A. pagination
B. payload limit
C. service timeouts
D. rate limiting

Answer: A

Explanation:

The correct answer is **A. pagination**.

Pagination is a technique used in API design to divide large result sets into smaller, manageable chunks, or "pages." Instead of sending all 550+ records at once, which can strain network resources and overwhelm the client, the API returns only a subset of the data (in this case, 25 records). The client then needs to request

subsequent pages to retrieve the complete dataset. This approach significantly improves performance, especially when dealing with large datasets, by reducing the amount of data transferred in a single request. It also makes the client application more responsive and less prone to errors caused by timeouts or excessive memory consumption. The API will typically provide metadata along with each page, indicating the presence of further pages and how to request them (e.g., using links or page numbers).

Pagination is a common best practice for API design and is widely implemented in RESTful services to handle large volumes of data. Options B, C, and D do not directly limit the amount of data returned in the way the question describes. Payload limits (B) impose a maximum size on the entire response but wouldn't dictate specific quantities of records returned. Service timeouts (C) refer to the duration an API call can take before being aborted, and Rate limiting (D) restricts the number of requests within a specific period, not the data amount in a single response.

Further Research:

RESTful API Pagination:https://restfulapi.net/pagination/
API Pagination Best Practices:https://nordicapis.com/10-best-practices-for-api-pagination/
Wikipedia - Pagination:https://en.wikipedia.org/wiki/Pagination

Question: 49

A developer is writing an application that uses a REST API and the application requires a valid response from the API. Which element of the response is used in the conditional check?

A. body
B. headers
C. link
D. URL
E. status code

Answer: E

Explanation:

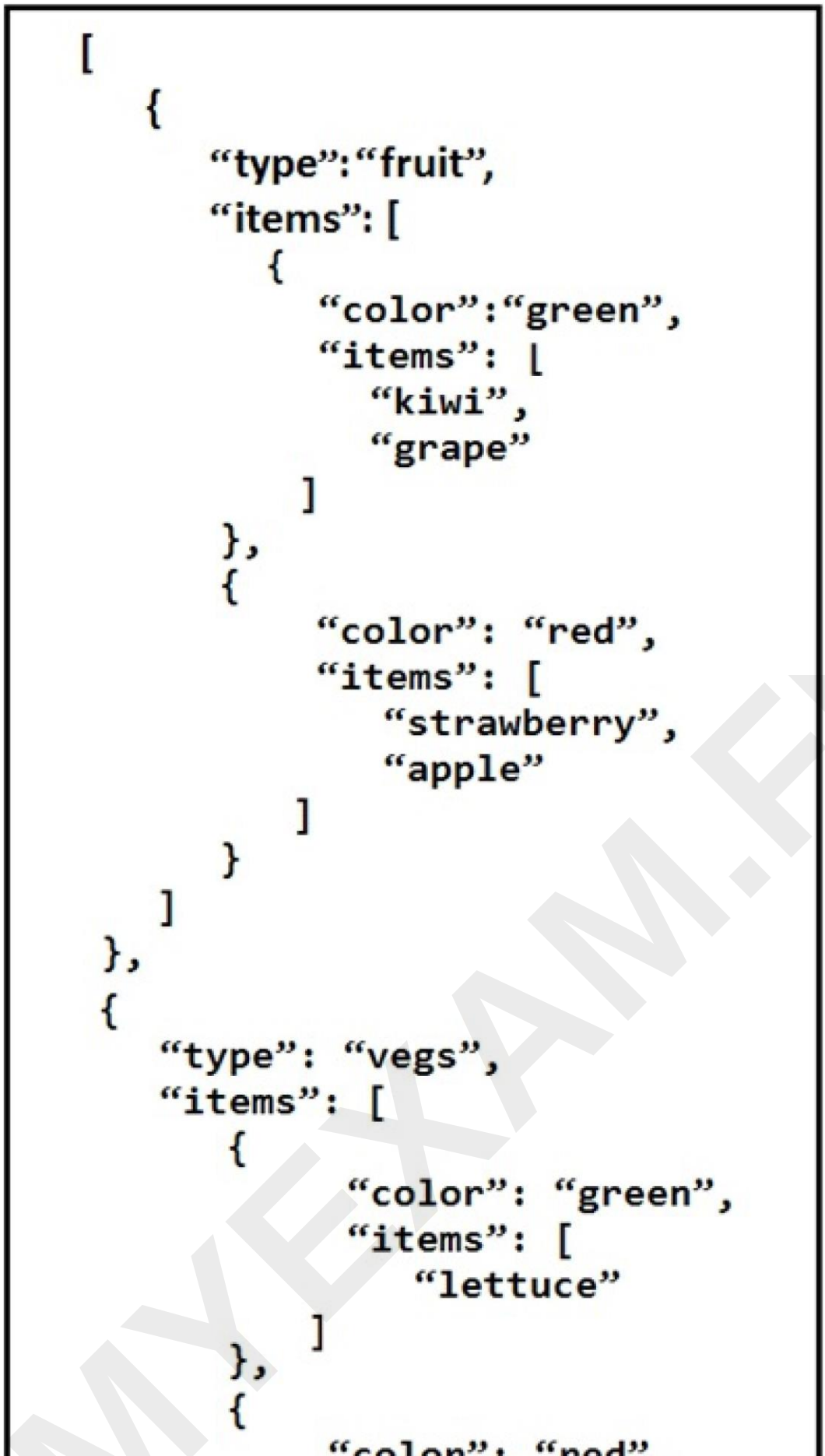
The correct answer is E, the status code. When a client application makes a request to a REST API, the server responds with various pieces of information. A crucial component is the HTTP status code, a three-digit number that indicates the outcome of the request. These codes fall into categories like 2xx (success), 4xx (client error), and 5xx (server error). A developer, when building an application that relies on a valid API response, needs to first determine if the request was successful before parsing the returned data. This determination is achieved by examining the status code. For instance, a 200 OK code signifies a successful request, whereas a 404 Not Found or 500 Internal Server Error indicates a problem. The response body, headers, link, and URL, although important, are not the immediate indicators of the request's success or failure. They carry supplementary information and are typically processed after validating the status code. Therefore, before parsing the response body or using any links, a conditional check based on the status code is paramount for proper error handling and program flow within an application utilizing a REST API. Failing to check the status code could lead to application errors or unexpected behavior by trying to process a failed request.

For further research, consider these resources:

MDN Web Docs on HTTP status codes:https://developer.mozilla.org/en-US/docs/Web/HTTP/Status **Wikipedia on HTTP status codes:**https://en.wikipedia.org/wiki/List_of_HTTP_status_codes
REST API Best Practices: https://nordicapis.com/10-best-practices-for-rest-api-design/ (discusses status codes)

Question: 50

Refer to the exhibit.



```
[
   {
      "type":"fruit",
      "items": [
         {
            "color":"green",
            "items": [
               "kiwi",
               "grape"
            ]
         },
         {
            "color": "red",
            "items": [
               "strawberry",
               "apple"
            ]
         }
   ]
},
{
   "type": "vegs",
   "items": [
      {
         "color": "green",
         "items": [
            "lettuce"
         ]
      },
      {
         "color": "red",
```

A REST API returns this JSON output for a GET HTTP request, which has been assigned to a variable called

`vegetables`. Using Python, which output is the result of this command? print(filter(lambda 1: 1['type'] == 'fruit', vegetables) [0]['items'][0]['items'][0])

A. 'color': 'green', 'items': ['kiwi', 'grape']
B. ['kiwi', 'grape']
C. lettuce
D. kiwi

Answer: D

Explanation:

it is kiwi but that function is very syntactically wrong1) lambda functions can't be called integers, so it should be something like "lambda <name>: <name>["type"] == ..."2) filter() returns an iterator and is not directly subscriptable, meaning that you need to actually iterate over it:for item in filter(lambda name: name["type"] == "fruit", data): print(item)this returns: 'type': 'fruit', 'items': ['color': 'green', 'items': ['kiwi', 'grape'] , 'color': 'red', 'items': ['strawberry', 'apple']] 3) [0]['items'][0]['items'][0] is also syntactically incorrect since the outer-most object is a dictionary not a list, however:for item in filter(lambda name: name["type"] == "fruit", data): print(item['items'][0]['items'][0])does actually return "kiwi"another quality question from cisco

Question: 51

DRAG DROP -
Drag and drop the Python code from the left onto the correct step on the right to call a REST API.
Select and Place:

response = requests.post(url, data=json.dumps(payload), \ headers=header, verify=False)	Step 1 – Import the correct libraries.
ticket = r_json[“Token”] print (ticket)	Step 2 – Declare the necessary variable.
(_json = response.json() print(r_json)	Step 3 – Send the HTTP Request.
import requests import json	Step 4 – Format and display the response in JSON readable format.
controller = 'devnetapi.cisco.com/sandbox/dnacenter' url = 'https://' + controller + '/dna/system/api/v1/auth/token' payload = { 'username': 'devnetuser', 'password': 'pa55word'} header = { 'Content-type': 'application/json' }	Step 5 – Parse the response to display the Service Ticket.

Answer:

```
response = requests.post(url, data=json.dumps(payload), \
headers=header, verify=False)
```

```
import requests
import json
```

```
ticket = r_json["Token"]
print (ticket)
```

```
controller = 'devnetapi.cisco.com/sandbox/dnacenter'
url = 'https://' + controller + '/dna/system/api/v1/auth/token'
payload = { 'username': 'devnetuser', 'password': 'pa55word'}
header = { 'Content-type': 'application/json' }
```

```
(_json = response.json()
print(r_json)
```

```
response = requests.post(url, data=json.dumps(payload), \
headers=header, verify=False)
```

```
import requests
import json
```

```
(_json = response.json()
print(r_json)
```

```
controller = 'devnetapi.cisco.com/sandbox/dnacenter'
url = 'https://' + controller + '/dna/system/api/v1/auth/token'
payload = { 'username': 'devnetuser', 'password': 'pa55word'}
header = { 'Content-type': 'application/json' }
```

```
ticket = r_json["Token"]
print (ticket)
```

Question: 52

A 401 HTTP response code is returned when calling a REST API. What is the error state identified by this response code?

A.The server cannot process the request as it has detected an issue in the request syntax or body.

B.The request has not been accepted because it requires authentication.

C.The sever accepted the request but the client is not authorized for this content.

D.The server cannot find the requested resource because the path specified is incorrect.

Answer: B

Explanation:

The correct answer is **B. The request has not been accepted because it requires authentication.**

HTTP status codes are standardized responses from a server to a client, conveying the outcome of a request. A 4xx series code indicates a client-side error, meaning the problem likely originates from the user's request.

Specifically, a 401 Unauthorized error signals that the client attempted to access a protected resource without providing valid credentials. This means the server requires the client to authenticate itself before granting access. The client needs to present authentication information, such as a username/password combination or a token, before the request can be successfully processed.

Option A, describing a 400 Bad Request, is incorrect because 400 codes flag issues with the syntax or body of the request itself, not authentication. Option C describes a 403 Forbidden error, which means the server understands the request and the client's identity, but the user is not authorized to access the requested resource, even with valid authentication. Option D, corresponding to a 404 Not Found error, indicates the server could not locate the requested resource at the given path, which is a different issue altogether from lacking authentication.

In the context of REST APIs, authentication is a common security measure. A 401 response explicitly states that the API endpoint requires proper authentication before providing data or performing actions. Therefore, the client must include valid credentials in subsequent requests to access the resource. It's crucial to differentiate between authentication and authorization. Authentication verifies who the user is, while authorization determines what they can access. A 401 response means that the client failed the initial

authentication test and should provide credentials.

For more in-depth information, you can refer to these resources:

MDN Web Docs on HTTP response status codes:https://developer.mozilla.org/en-US/docs/Web/HTTP/Status **RFC 7231, section 6.5.1 (401 Unauthorized):**https://datatracker.ietf.org/doc/html/rfc7231#section-6.5.1 **REST API Documentation on HTTP status codes:** A general search for "REST API http status codes" will also provide a wealth of additional documentation from various platforms.

Question: 53 A

developer is creating a script to interact with a REST API service which requires basic authentication. The credentials are "devnet:391665405" and the Base64 encoding of the credentials is "GV2bmV0dXNlcjpDaXNj=". Which payload and header combination must be used for authentication?

A.

```
payload = {
    'Authorization' : 'Basic GV2bmV0dXNlcjpDaXNj=',
    'Content-Type' : 'application/json'
}
```

B.

```
headers= {}

payload = {}
headers= {
    'Authorization' : 'Basic GV2bmV0dXNlcjpDaXNj=',
    'Content-Type' : 'application/json'
}
```

C.

```
payload = {
    'Authorization' : 'Bearer GV2bmV0dXNlcjpDaXNj=',
    'Content-Type' : 'application/json'
}
```

D.

```
headers= {}

payload = {}
headers= {
    'Authorization' : 'Bearer GV2bmV0dXNlcjpDaXNj=',
    'Content-Type' : 'application/json'
}
```

Answer: B

Explanation:

B is correct, Basic authentication uses Base64 converting the string 'username:password'.

Bearer <token here> is for OAuth

Question: 54

A developer needs a list of clients connected to a specific device in a Meraki network. After making a REST API call, the developer receives an unfamiliar response code. Which Cisco DevNet resource should be used to identify the meaning of the response code?

A.API documentation
B.Code Exchange
C.Learning Labs
D.Sandbox

Answer: A

Explanation:

The correct answer is **A. API documentation.**

Here's why: When a developer encounters an unfamiliar response code from an API call, the primary and most authoritative source for understanding its meaning is the API's documentation. API documentation meticulously outlines all possible request formats, response codes, and their corresponding interpretations. This documentation acts as a contract between the API and the developer, ensuring proper usage and error handling. Response codes, like HTTP status codes (e.g., 200 OK, 404 Not Found, 500 Internal Server Error) and those specific to the Meraki API, are defined in this documentation to communicate the outcome of a request. Cisco's DevNet platform provides comprehensive API documentation for its various services, including the Meraki API.

Code Exchange (B) is a platform for sharing and accessing code samples and solutions, while Learning Labs (C) offers hands-on tutorials and exercises. The Sandbox (D) provides an environment for testing APIs. Although useful for development and learning, these resources are not the definitive source for interpreting API response codes. Therefore, when troubleshooting API-related issues, referencing the API documentation is the essential first step for a developer. Specifically, for the Meraki API, this documentation would contain a list and explanation of each response code that the API might send back.

Authoritative Links:

Cisco Meraki API Documentation:https://developer.cisco.com/meraki/ (This is the primary place to find Meraki API documentation)
HTTP Status Codes:https://developer.mozilla.org/en-US/docs/Web/HTTP/Status (A general reference for standard HTTP status codes that might also be relevant)

Question: 55

A developer is trying to retrieve data over a REST API. The API server responds with an HTTP client error response code. After investigating the response, the developer realizes the response has a Retry-After header. What is the root cause of this error?

A.An appliance limited the rate of requests to the transport layer.
B.The REST service is unreachable at the time of the REST request.
C.Too many requests were sent to the REST service in a given amount of time.
D.An appliance limited the rate of requests to the application layer.

Answer: C

Explanation:

The correct answer is **C. Too many requests were sent to the REST service in a given amount of time.**

Here's a detailed justification:

HTTP client error response codes, specifically the 429 Too Many Requests status code, indicate that a client has sent too many requests to a server in a given timeframe. The presence of a Retry-After header in the response further confirms this. This header instructs the client to wait a specific duration before making another request. It's a mechanism for rate limiting, a common practice to protect servers from being overwhelmed by excessive traffic.

Option A, "An appliance limited the rate of requests to the transport layer," is less likely because transport layer rate limiting typically results in connection-level errors (like TCP connection resets) rather than specific HTTP error codes like 429 with a Retry-After header.

Option B, "The REST service is unreachable at the time of the REST request," would typically lead to different errors, such as connection timeouts or 5xx server errors, not a specific 429.

Option D, "An appliance limited the rate of requests to the application layer," is plausible but less precise. While it's possible, the 429 and Retry-After combination specifically points to the application (REST service) itself enforcing rate limits rather than an intermediary appliance.

In essence, the server, upon receiving excessive requests, proactively signals the client to back off using the 429 status code and Retry-After header. This allows the server to maintain stability and prevent service degradation. The client needs to respect this guidance and delay subsequent requests as instructed.

Authoritative Links:

MDN Web Docs - HTTP response status codes:https://developer.mozilla.org/en-US/docs/Web/HTTP/Status -Search for "429 Too Many Requests" for specific information.

MDN Web Docs - Retry-After:https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Retry-After **RFC 6585 - Additional HTTP Status Codes:**https://datatracker.ietf.org/doc/html/rfc6585#section-4 -Specifically outlines the 429 status code

Question: 56

Refer to the exhibit.

Local File Attachments

To send local file attachments, simply post a message by including your access token in the Authorization header and the path to your local file with the files parameter. Optionally, you can also include a plain-text message with the attachment by using the text parameter. When uploading files directly from your local filesystem, your request will need to be a multipart/form-data request rather than JSON

```
curl --request POST \
  --form "files=@/home/desktop/example.png;type=image/png" \
  --form "roomId=Y2lzY2..." \
  --form "text=example attached" \
  https://webexapis.com/v1/messages
```

A developer needs to upload a local file by using the REST API. The developer gathers information according to the documentation and sends the request by using the cURL command in the exhibit but gets an error code. Which action should be followed to get a valid response?

A. change content-type as JSON
B. add the authorization header with the access token
C. and a username-password combination to request command D.
change request method as GET

Answer: B

Explanation:

Reference:
https://developer.webex.com/docs/api/basics

Question: 57

DRAG DROP -
Refer to the exhibit.

Drag and drop the descriptors from the left onto the corresponding parts of the API request and response on the right.
Select and Place:

Response Headers	A
HTTP Request	B
HTTP Response	C
Request Payload	D
Request Headers	E
Response Payload	F

Answer:

Response Headers	Request Headers
HTTP Request	Request Payload
HTTP Response	HTTP Request
Request Payload	HTTP Response
Request Headers	Response Headers
Response Payload	Response Payload

Question: 58

Which action resolves a 401 error in response to an HTTP GET that is issued to retrieve a configuration statement using RESTCONF on a CSR 1000V?

A.Change the HTTP method to PUT.
B.Check the MIME types in the HTTP headers.
C.Change the transport protocol to HTTPS.
D.Check the authentication credentials.

Answer: D

Explanation:

The correct answer is **D. Check the authentication credentials.** A 401 Unauthorized error in HTTP responses specifically indicates that the client's request lacks valid authentication credentials to access the requested resource. In the context of RESTCONF, this means the Cisco CSR 1000V device is refusing to provide the configuration because the supplied username and password, or token, are either incorrect or missing.

Changing the HTTP method to PUT (A) is inappropriate because PUT is used for updating, not retrieving, resources. Mismatched MIME types (B) typically lead to 415 Unsupported Media Type errors, not 401s. While switching to HTTPS (C) enhances security, it doesn't address the core authentication problem; a 401 error can still occur over HTTPS if credentials are faulty. Thus, the issue is directly linked to the client's failed authentication. To resolve this, one needs to verify the username, password, or any other authentication mechanisms used are valid against the device's configured users. Incorrect or expired tokens, if used, would also trigger this error. This aligns with fundamental security principles; the device requires proof of authorization before allowing access to sensitive configuration data.

For further reading, refer to:

1. **HTTP Status Codes - Wikipedia:**https://en.wikipedia.org/wiki/List_of_HTTP_status_codes (Specifically, look at the 4xx Client Error codes, notably 401 Unauthorized).

2. **RESTCONF Protocol - RFC 8040:**https://www.rfc-editor.org/rfc/rfc8040 (This explains RESTCONF's reliance on standard HTTP mechanisms, including status codes and authentication).

3. **Cisco CSR 1000V Configuration Guide:** (The specific document will depend on the CSR 1000V software version; consult Cisco's documentation for the authentication specifics relevant to your setup).

Question: 59

Refer to the exhibit.

An API call is constructed to retrieve the inventory in XML format by using the API. The response to the call is 401 Unauthorized. Which two headers must be added to the API call? (Choose two.)

A. Bearer-Token: dXNlcm5hbWU6cGFzc3dvcmQ=
B. Content-Type: application/xml
C. Authentication: Bearer dXNlcm5hbWU6cGFzc3dvcmQ=
D. Accept: application/xml
E. Authorization: Bearer dXNlcm5hbWU6cGFzc3dvcmQ=

Answer: DE

Explanation:

A. is Incorrect because there is no Bearer-Token header in HTTP

B. is Incorrect, because the image shows Method GET, and Content-Type header defines the type of data you use in the payload of a Method POST

C. is incorrect, because Authentication is not the correct HTTP header

D. is correct, because Accept header defines the expected response data format of a Method GET

E. is correct, because the header Authorization is the one used for OAuth and Basic Auth

Question: 60

Which HTTP response status code means `Forbidden`?

A.500
B.401
C.403
D.502

Answer: C

Explanation:

The correct answer is C, 403. HTTP status codes are three-digit numbers returned by a server in response to a client's request. They indicate the outcome of the request. A 403 "Forbidden" status code signifies that the server understands the client's request but refuses to fulfill it. This denial of access isn't due to invalid credentials, as in a 401 error, but rather because the client, even if authenticated, lacks the necessary permissions to access the requested resource. For example, a user might be logged in but not authorized to view a particular file. This contrasts with a 401 "Unauthorized" code, which implies that authentication is required first before access can be considered. A 500 "Internal Server Error" code indicates a generic server-side problem preventing it from fulfilling the request, not an issue with client permissions. A 502 "Bad Gateway" signifies an intermediary server received an invalid response from an upstream server. Therefore, 403 is the only option aligning with the definition of "Forbidden" access based on insufficient permissions. Understanding these distinct error codes is crucial for debugging and troubleshooting web applications and APIs. The 403 status code directly relates to access control and authorization mechanisms in cloud environments.

For further research, you can consult these authoritative links:

Mozilla Developer Network (MDN) on HTTP Status Codes:https://developer.mozilla.org/en-US/docs/Web/HTTP/Status
RFC 7231: HTTP/1.1 Semantics and Content:https://datatracker.ietf.org/doc/html/rfc7231#section-6.5These resources provide in-depth explanations of HTTP status codes, including their meanings and proper usage.

Question: 61

Which HTTP response code should be returned when the request is completed, but there is no content to send for the request?

A.100
B.204
C.302
D.402

Answer: B

Explanation:

The correct HTTP response code for a successful request with no content to return is **204 No Content**. This code signals to the client that the server has successfully processed the request, but there isn't any

information to send back in the response body. It's a common response for actions like successful deletions or updates where no new data needs to be communicated.

Option A, **100 Continue**, is an informational response code used in preliminary requests before the full request is sent. It indicates that the server is ready to receive the body of the request. It's not appropriate for successful operations where a response isn't needed. Option C, **302 Found**, signifies a temporary redirect, instructing the client to make a request to a different URI. This is unrelated to successful operations where no content is to be sent. Option D, **402 Payment Required**, is an uncommon code that indicates the request cannot be fulfilled without payment. This code is clearly not related to a successful operation with no content.

The 204 No Content response is vital for efficient API design, as it minimizes data transfer when a response body is unnecessary. It's a cornerstone of RESTful architecture, allowing clients to perform actions without needing to process superfluous data. This reduces bandwidth consumption and speeds up application performance.

Using the right status code is crucial for the client to interpret the outcome of its request properly. A 204 response allows the client to know the operation is complete without wasting time parsing an empty response body. It ensures that the client application can handle various server responses efficiently and accurately.

Here are some authoritative resources for further reading:

MDN Web Docs on HTTP Response Status Codes:https://developer.mozilla.org/en-US/docs/Web/HTTP/Status
IETF RFC 9110 (HTTP Semantics):https://www.rfc-editor.org/rfc/rfc9110These resources detail the proper usage of each HTTP status code, reinforcing that 204 is ideal when no content is provided after a successful operation. They provide the foundation for understanding HTTP protocol semantics and their role in web communications.

Question: 62

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: nginx
Last-Modified: Wed, 22 Jan 2020 19:15:56 GMT
Content-Length: 88
Content-Type: application/json
Connection: keep-alive

{
    "id": "a123456bcde",
    "status": "online",
    "type": "switch"
}
```

Refer to the exhibit. Which data is specified in the response header?

A.Content-Type
B.HTTP/1.1 200 OK
C.type: switch
D. id: a123456bcde, HTTP/1.1 200 OK status: online, type: switch

Answer: A

Explanation:

A. Content-Type https://contenthub.netacad.com/devnet/4.4.3 One common entity header specifies the type of data being returned:KeyExample ValueDescriptionContent-Typeapplication/jsonSpecify the format of the data in the body

Question: 63

What are two use cases where webhooks are effective? (Choose two.)

A.Filter out information from a response to an API call.
B.Close a session with a web server after a specific amount of time.
C.Inform a previously defined chat channel after a deployment fails.
D.Send an email to a customer of an online store after payment is complete.
E.Change the response format or content type of an API call.

Answer: CD

Explanation:

The correct answers are C and D because they accurately represent effective use cases for webhooks.
Webhooks are a mechanism for real-time, server-to-server communication triggered by events. Option C, "Inform a previously defined chat channel after a deployment fails," exemplifies this. When a deployment fails, a system can send a webhook notification to a chat application, alerting relevant team members instantly. This eliminates the need for constant polling and provides immediate feedback on critical events.

Option D, "Send an email to a customer of an online store after payment is complete," also demonstrates a good use of webhooks. When a payment gateway completes a transaction, it can trigger a webhook to notify the online store's server. This allows the store to initiate follow-up actions, such as sending a confirmation email to the customer. This again, avoids constant polling for transaction status and provides timely updates based on real-time events.

Options A, B, and E are not ideal use cases for webhooks. Option A refers to filtering data within an API response, which is typically handled by the client making the API call, not webhooks. Option B, concerning session closure based on time, is usually managed directly by the web server through timeouts. Option E, regarding response format or content, falls under request header manipulation and API versioning, and is unrelated to webhook function.

In essence, webhooks are most effective in scenarios where a system needs to be passively notified about events as they happen, without requiring constant requests or polling. The selected options showcase webhook effectiveness in real-time communication for event-driven scenarios.

Further reading:

Webhooks explained: https://sendgrid.com/blog/webhooks-explained/
Understanding webhooks: https://pusher.com/tutorials/webhooks-guide/
Webhooks: What are they and how do they work?: https://www.integrations.io/blog/webhooks/

Question: 64

In which two ways is an application characterized when interacting with a webhook? (Choose two.)

A.receiver
B.transaction monitor
C.codec
D.processor
E.listener

Answer: AE

Explanation:

Here's a detailed justification for why the correct answer choices characterizing an application interacting with a webhook are **A. receiver** and **E. listener**:

Webhooks facilitate asynchronous communication between applications. They operate on a push mechanism, where one application (the source) sends a notification to another application (the target) when a specific event occurs. In this context, the application receiving the webhook is inherently a **receiver**. It actively waits for incoming HTTP requests that carry the webhook payload. This receiver application is also acting as a **listener**, as it is actively monitoring a specific URL or endpoint for these incoming webhook notifications. The source application "publishes" events, and the receiving application "subscribes" to those events by setting up the webhook.

Choices B (transaction monitor), C (codec), and D (processor) are not primary roles associated with webhook interaction. While an application might eventually process the data in the webhook payload, or the webhook may trigger a more extensive series of transactions, the fundamental role during webhook interaction is receiving the notification itself. Similarly, while encoding/decoding might be part of the overall process (particularly regarding data format such as JSON or XML), a codec is not a core characteristic defining an application when interacting with webhooks. Therefore, "receiver" and "listener" accurately and primarily describe the role of an application responding to a webhook.

Further research on webhooks and related concepts can be found on these authoritative resources:

Webhooks Guide by GitHub:https://docs.github.com/en/webhooks
MDN Web Docs on Webhooks:https://developer.mozilla.org/en-US/docs/Web/API/Webhooks_API
Atlassian Webhooks:https://developer.atlassian.com/cloud/confluence/webhooks/

Question: 65

How does requesting a synchronous API operation differ from requesting an asynchronous API operation?

A.clients receive responses with a task ID for further processing
B.clients subscribe to a webhook for operation results
C.clients poll for the status of the execution of operations
D.clients can access the results immediately

Answer: D

Explanation:

Here's a detailed justification for why option D is the correct answer, and why the others are incorrect, regarding synchronous and asynchronous API operations:

Synchronous API operations involve a client sending a request and waiting for an immediate response containing the result. This "request-response" model means the client's execution is blocked until the server completes the task and provides the outcome. This immediacy is characteristic of synchronous interactions.

Option D accurately captures this aspect by stating that clients can access results immediately.

In contrast, asynchronous API operations do not provide an immediate response. When a client sends a request asynchronously, the server typically acknowledges the request and returns something like a task ID or confirmation, freeing the client to continue with other tasks. The actual result is delivered later.

Option A describes the task ID scenario, which is indeed part of asynchronous workflows, where the task ID is used to later track progress. This is not associated with the synchronous API response.

Option B describes webhooks, where the server pushes updates to the client when ready. This is also an asynchronous pattern, not synchronous.

Option C, stating clients poll for operation status, also depicts an asynchronous mode. The client needs to repeatedly check whether the server has finished processing, indicating that the result is not immediately available.

Therefore, only option D accurately describes the immediate result retrieval characteristic of a synchronous API interaction, making it the correct response.

Further research links:

1. REST API Tutorial - Synchronous vs Asynchronous APIs - This link explains both concepts clearly. 2. Microsoft - Synchronous versus Asynchronous Operations - Explains from software architecture perspective about asynchronous and synchronous operations.
3. IBM - Understanding API calls: Synchronous and asynchronous - More insights into API calls and patterns.

Question: 66

DRAG DROP -
Drag and drop the HTTP status codes from the left onto the correct descriptions on the right.
Select and Place:

400	Unauthorized
401	Not Found
403	Bad Request
404	Forbidden

Answer:

400	401
401	404
403	400
404	403

Explanation:

Reference:
https://www.cisco.com/c/en/us/support/docs/security/web-security-appliance/118217-troubleshoot-wsa-00.ht ml

Question: 67

Which method should be used to safely store the API keys?

A.in an environment variable unique to the system database
B.encrypted in a configuration file that is separate from the code
C.plain text in the code as a constant
D.encrypted in the code in a function

Answer: A

Explanation:

The correct answer is **A. in an environment variable unique to the system database.** This is the most secure method for storing API keys among the given options. Environment variables are system-level settings that are not typically stored directly within application code or configuration files, reducing the risk of accidental exposure through version control or unauthorized access. They also allow for dynamic key management, enabling changes without needing to re-deploy the application. Storing API keys in the system database further isolates these sensitive credentials. Option B, while better than hardcoding, is still less secure.

Encrypted configuration files could potentially be decrypted if the encryption key is compromised or if the application logic is poorly implemented. Options C and D are fundamentally insecure and should never be used. Storing API keys as plain text in code (C) is extremely vulnerable to exposure, and while encrypting them within the code (D) might seem better, the decryption logic itself is also present in the code, which could be reverse-engineered or exploited. Utilizing environment variables allows for a segregation of sensitive information from application deployment, often supported by access controls at an OS level. This is aligned with the principle of least privilege within security best practices.https://12factor.net/confighttps://owasp.org/www-project-top-ten/

Question: 68

FILL BLANK -
Fill in the blanks to complete the statement.
Given a username of `devnet` and a password of `cisco123`, applications must create a base64 encoding of the string `_____` when sending HTTP requests to an API that uses _____ authentication.

Answer:

See explanation below.

Explanation:
YWRtaW46Y2lzY28xMjM -

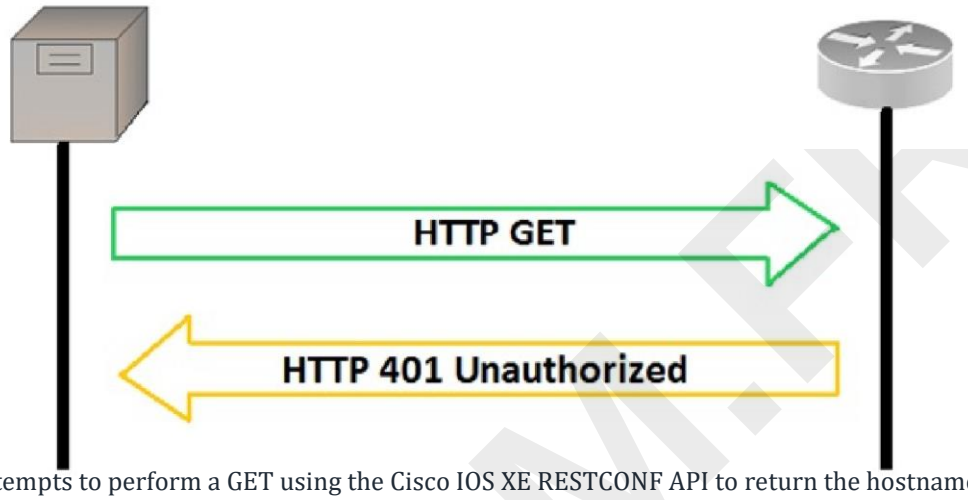
Basic -

Reference:
https://www.cisco.com/c/en/us/td/docs/net_mgmt/elastic_services_controller/4-1/install/guide/Cisco-Elastic-Services-Controller-Install-Upgrade-Guide-4-1/Cisco-Elastic-Services-Controller-Install-Upgrade-Guide-3-0_chapter_01011.html

Question: 69

Refer to the exhibit.

```
curl -i -k -X "GET" "https://172.18.110.64:443/ \
restconf/data/Cisco-IOS-XE-native:native/hostname" \
-H 'Accept: application/yang-data+json' \
```



An administrator attempts to perform a GET using the Cisco IOS XE RESTCONF API to return the hostname of a device. The sequence diagram illustrates the
HTTP messages observed. Which change to the API request resolves the issue?

A.Remove the "H 'Accept: application/yang-data+json' HTTP header because it is not required.

B.Add "'u cisco:cisco' in the end of the cURL command

C.Change the request method from "X GET to "X POST

D.Add Content-Type HTTP header with 'application/yang-data+json' using "H 'Content-Type: application/yang-data+json'

Answer: B

Explanation:

B is indeed correct. Basic authentication works for IOS XE RESTCONF

Question: 70

What are two benefits of managing network configuration via APIs? (Choose two.)

A.more security due to locking out manual device configuration
B.configuration on devices becomes less complex
C.eliminates the need of legacy management protocols like SNMP
D.reduction in network changes performed manually
E.increased scalability and consistency of network changes

Answer: DE

Explanation:

The correct answer is **D. reduction in network changes performed manually** and **E. increased scalability and consistency of network changes**.

Managing network configurations via APIs automates many tasks previously done manually. This automation significantly reduces human error, leading to more consistent and predictable configurations across the network (E). By scripting or coding changes via API calls, you eliminate the need to individually access each device and make modifications, which is prone to inconsistencies and is time-consuming. This approach enables repeatable and verifiable deployments. Furthermore, API driven network management facilitates orchestration allowing a single action to propagate across multiple devices, greatly improving scalability (E) and simplifying bulk updates.

Manual changes are often laborious and difficult to track, increasing the likelihood of configuration drift and errors. APIs facilitate audit trails, enabling better version control, and troubleshooting (D). While APIs certainly offer security advantages, locking out manual configuration alone does not guarantee more security, as improperly configured APIs can still pose a threat (A). API use doesn't simplify device configurations themselves (B); rather, it provides a more efficient way to manage them. Finally, APIs don't eliminate legacy protocols; they often work alongside them for various use cases and do not aim to replace protocols such as SNMP for monitoring (C).

Supporting Links:

Cisco DevNet Associate Exam Topics:https://developer.cisco.com/certification/exam-topic-devasc/ (Refer to Programmability and Automation topics)
Network Automation with APIs:https://www.networkcomputing.com/networking/network-automation-apis-driving-change
Benefits of Network Automation:https://www.sdxcentral.com/resources/network-automation/benefits-of-network-automation/

Question: 71

FILL BLANK -
Fill in the blanks to complete the cURL command that invokes a RESTful API to retrieve a resource in JSON format using OAuth. curl `"X _____ -H ` _____: application/json` \
-H ` _____: Bearer AbCdEf123456` https://localhost/api/myresource

Answer:

See explanation below.

Explanation:
GET -

Accept -

Authorization -

Reference:
https://webcache.googleusercontent.com/search?q=cache:Se6d2trvMsEJ:https://blogs.cisco.com/developer/ security-api-best-practices
+&cd=4&hl=en&ct=clnk&gl=pk&client=firefox-b-d

Question: 72

Which Cisco DevNet resource allows access to products in a development lab to explore, learn, and build applications that use Cisco APIs?

A. DevNet Code Exchange
B. DevNet Sandbox
C. DevNet Communities
D. DevNet Automation Exchange

Answer: B

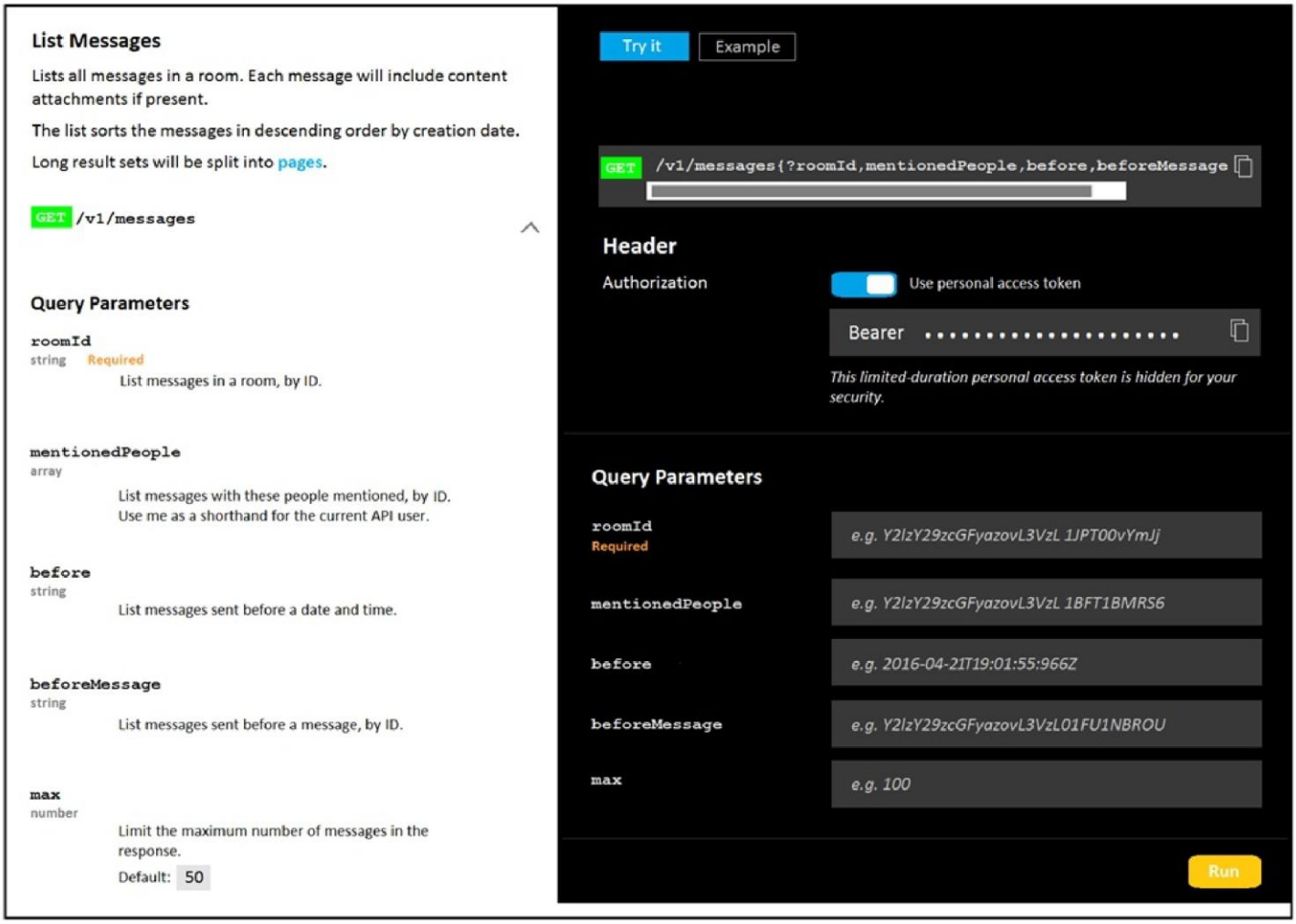
Explanation:

The correct answer is **B. DevNet Sandbox**. DevNet Sandbox provides pre-configured, readily accessible development environments featuring various Cisco hardware and software products. These sandboxes allow developers to interact with real Cisco APIs, experiment with configurations, and build applications without needing to own or manage physical infrastructure. This aligns with the cloud computing concept of "Infrastructure as a Service" (IaaS), where resources are provided on demand, enabling developers to focus solely on application development. DevNet Code Exchange primarily hosts code examples and solutions, while DevNet Communities is a forum for discussion and support. DevNet Automation Exchange focuses on sharing automation code and workflows. Thus, only DevNet Sandbox offers the interactive lab environment with access to Cisco products for development and exploration, fulfilling the question's requirement.

For further exploration, visit the official Cisco DevNet website: https://developer.cisco.com/ and specifically the sandbox resources page: https://developer.cisco.com/site/sandbox/ This will give you a complete picture of how these resources assist in Cisco API development.

Question: 73

Refer to the exhibits.




```
bash-3.2$ curl -H "Content-Type: application/json" -H "Authorization:
Fj2zzykEa091ic9GK2j8LtE1HklH6oRHPQdw1pAT60I7NDThhNwZl2B5pqMg14kK_b9ei59ISAClY7-
NarA-2n9H-tGgt-SxQ39iDejgcs" -i -X POST -d "{ \"roomId\":
\"Y2geK53sjEknosrC7SwQ5ZGL99pHgiuScB7DfNvUsr8Xx4wDKLiPORcEkryAnM3QmK9LQZsP0G4\"
, \"text\": \"test2\" }" https://api.ciscospark.com/v1/messages

HTTP/1.1 401 Unauthorized
Via: 1.1 linkerd
TrackingID: ROUTER_5E0FDC8C-EB6F-01BB-00EF-806BF1BD00EF
Date: Sat, 04 Jan 2020 00:30:04 GMT
Server: Redacted
Content-Length: 267
Content-Type: application/json
Strict-Transport-Security: max-age=63072000; includeSubDomains; preload

{"message":"The request requires a valid access token set in the Authorization request
header.","errors":[{"description":"The request requires a valid access token set in the
Authorization request
header."}],"trackingId":"ROUTER_5E0FDC8C-EB6F-01BB-00EF-806BF1BD00EF"}

bash-3.2$
```

A developer is troubleshooting an API with the given API documentation and cURL command. What is the cause of this problem?

A.The authorization header is missing or incomplete.

B.The request body is missing or incomplete.

C.The API token specified is expired.

D.The user is not allowed to post messages from their account.

Answer: A

Explanation:

A is correct. I had this exact problem when I was doing the Webex labs in the devnet course. I left "Bearer: " out of the token string and it returned a 401 error. When I added that to the token string, it worked.

Question: 74

Which two statements describe the traits of an asynchronous API call? (Choose two.)

A.Code execution blocks or waits for the call to an API to return.

B.A callback function typically is used to process the response from an API call.

C.A call to an API does not block the code, but rather it allows application processing to continue.

D.The end user can experience latency or performance lag while waiting for the API call to return.

E.The order in which API calls return can be guaranteed.

Answer: BC

Explanation:

Here's a detailed justification for why options B and C accurately describe the traits of an asynchronous API call, and why options A, D, and E are incorrect:

Asynchronous API calls are designed to prevent blocking the main thread of an application. Instead of waiting for a response before continuing, the application can proceed with other tasks. Option C directly captures this core characteristic: "A call to an API does not block the code, but rather it allows application processing to continue." This non-blocking behavior is essential for building responsive applications, especially in scenarios involving network requests that can be time-consuming.

To handle the eventual response from an asynchronous API call, a mechanism is needed to process the data when it arrives. Option B correctly identifies the common solution: "A callback function typically is used to process the response from an API call." After initiating the asynchronous request, the application specifies a callback function which the system invokes when the API response is received. This function then handles the data and performs any required actions.

Option A is incorrect because it describes synchronous behavior, where the application halts until the API returns a result. This is the opposite of asynchronous execution. Option D is also wrong; since asynchronous operations are non-blocking, they typically improve perceived latency and application responsiveness, not cause lags. Finally, option E is incorrect; the order in which asynchronous calls return cannot be guaranteed.

Their completion depends on factors like network latency, server load, and internal API processing time.

Therefore, assuming that API responses come in a predictable sequence is not realistic in asynchronous interactions.

Authoritative Links:

1. **MDN Web Docs on Asynchronous JavaScript:**https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Asynchronous - A great resource for understanding asynchronous programming concepts in a JavaScript context, which often applies to API interactions.
2. **Wikipedia on Asynchronous Communication:** https://en.wikipedia.org/wiki/Asynchronous_communication - Provides a broader definition of asynchronous interactions and their applications in computing.
3. **IBM Cloud Blog on Asynchronous APIs:**https://www.ibm.com/cloud/blog/what-are-asynchronous-apis-and-why-use-them - Explains the advantages of using asynchronous API calls within cloud environments.

Question: 75

Which mechanism is used to consume a RESTful API design when large amounts of data are returned?

A. data sets
B. scrolling
C. pagination
D. blobs

Answer: C

Explanation:

The correct answer is **C. pagination**. Pagination is a crucial technique for managing large datasets in RESTful APIs, particularly when dealing with web or mobile applications. Instead of sending the entire dataset in a single response, pagination divides the data into smaller, more manageable chunks or "pages". This approach is essential for several reasons. First, it reduces the initial load time of the response, improving user experience by providing data in a timely manner. Second, it minimizes network bandwidth usage by avoiding the transfer of potentially huge amounts of data that might not even be necessary for the user's current view.

Third, pagination helps to lessen resource consumption on the API server, preventing performance bottlenecks and potential crashes. It typically involves the use of parameters like "page" and "per_page" (or "limit" and "offset") in API requests to specify which page of data to return, as well as links in the response to navigate to the next, previous, first, and last pages. Options A, B, and D are not standard mechanisms for handling large API responses. Data sets are a generic term for structured data collections, not a specific technique. Scrolling, while a user interaction pattern, relies on underlying data loading mechanisms and is not a solution itself. Blobs are binary large objects, typically for media files, not structured data returned by APIs.

For further reading, explore the following resources:

REST API Pagination - Best Practices
Pagination - Wikipedia
Designing a RESTful API with Pagination - Medium

Question: 76

Refer to the exhibit.

URL: https://<ASA_IP>/api/access/global/rules
Method: POST
Payload:

```
{
   "destinationAddress": {
     "kind": "IPv4Address",
     "value": "10.1.1.1"
   },
   "remarks": [ ],
   "active": true
   "permit": true,
   "sourceAddress": {
     "kind": "IPv4Address",
     "value": "192.168.1.1"
   }
}
```

Response Status Code: 400 Bad Request

```
{"messages":[{"level":"Error","code":"JSON-PARSE-ERROR","details":
"Unexpected character ('\"' (code 34)):
```

What caused the error in this API request?

A.The API resource does not support the POST operation.
B.The submitted JSON payload has a formatting issue.
C.The API resource does not support JSON format payloads.
D.The submitted JSON payload includes a field that is not supported by the API resource.

Answer: B

Explanation:

The B is correct because it is no comma after "active": TrueThe correct form is:"active": True,

Question: 77

Which two use cases are supported by Meraki APIs? (Choose two.)

A.Build location-aware apps from Wi-Fi and LoRaWAN devices.
B.Build a custom Captive Portal for Mobile Apps.
C.Configure network devices via the Dashboard API.
D.Deploy applications onto the devices.

E.Retrieve live streams from a Meraki Camera.

Answer: CE

Explanation:

The correct answer is **C and E**. Let's break down why.

C. Configure network devices via the Dashboard API: Meraki's strength lies in its cloud-managed networking, and this cloud platform exposes an extensive API known as the Dashboard API. This API allows developers to programmatically configure Meraki devices, automate network settings, retrieve device information, and manage network operations. This enables automation and integration of Meraki networks with other systems. This is a core functionality of the API for network management.

E. Retrieve live streams from a Meraki Camera: Meraki's MV camera line offers API access to obtain live streaming video data and camera snapshots. This capability is critical for scenarios like video monitoring, building security systems and integrating video feeds with other applications. The API provides direct access to the camera's media feed, allowing developers to incorporate video streams into their custom solutions.

Options A, B and D are not primary use cases. Location data (A) may be indirectly attainable using data from APIs, but there is not specific built-in API for that. The captive portal (B) can be customized within the Dashboard, but there's no specific API for mobile app captive portal. Option D, deploying applications directly onto devices, isn't a typical Meraki use case; Meraki devices generally do not allow custom code execution like application deployments.

Justification Summary: The Meraki APIs are primarily designed to facilitate network management and monitoring, offering capabilities like device configuration and data extraction. The Dashboard API (C) caters to configuring the network, while the API's ability to fetch camera feeds (E) provides vital video data, confirming options C and E as the correct use cases.

Authoritative Links:

Meraki Dashboard API Documentation:https://developer.cisco.com/meraki/ (This is the main portal for all Meraki API information.)
Meraki API Python Library:https://github.com/meraki/dashboard-api-python (This provides a practical demonstration of API usage in Python)

Question: 78

Which API is used to obtain data about voicemail ports?

A. Webex Teams
B. Cisco Unified Communications Manager
C. Finesse Gadgets
D. Webex Devices

Answer: B

Explanation:

The correct answer is **B. Cisco Unified Communications Manager (CUCM)**. CUCM is the primary call control and management platform for Cisco's on-premises voice and video infrastructure. It maintains a comprehensive database of all telephony resources, including voicemail ports. Therefore, the API exposed by CUCM is the designated interface for obtaining information about these voicemail resources. Options A, C, and D are related to other Cisco platforms or applications with different focuses. Webex Teams (A) focuses on

team collaboration and messaging, not call control. Finesse Gadgets (C) are related to agent desktop software and contact center functionality, rather than voicemail port configuration. Webex Devices (D) encompasses a range of physical devices like desk phones and video endpoints, not directly the backend systems that manage voicemail. Since CUCM holds the centralized data on voicemail ports and provides the means to configure and monitor them through its API, it is the appropriate choice for retrieving this data.

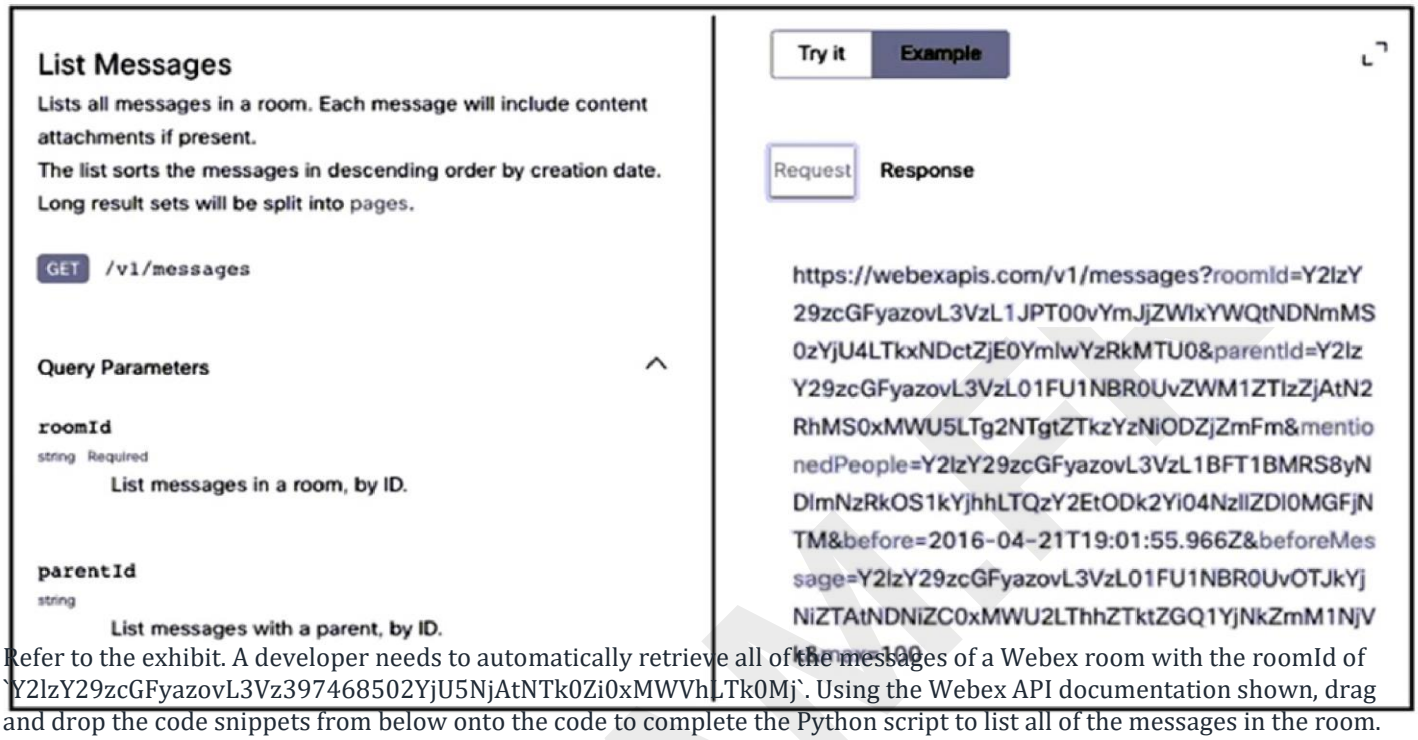
Cisco's official documentation consistently points to CUCM as the authoritative source for telephony configurations and resource management, including voicemail ports.

Authoritative Links for Further Research:

Cisco Unified Communications Manager API:https://developer.cisco.com/docs/collaboration/#!cucm-apis-dev-guide
Cisco Unified Communications Manager Documentation:https://www.cisco.com/c/en/us/support/unified-communications/unified-communications-manager-callmanager/tsd-products-support-series-home.html

Question: 79

DRAG DROP -



Refer to the exhibit. A developer needs to automatically retrieve all of the messages of a Webex room with the roomId of `Y2lzY29zcGFyazovL3Vz397468502YjU5NjAtNTk0Zi0xMWVhLTk0Mj`. Using the Webex API documentation shown, drag and drop the code snippets from below onto the code to complete the Python script to list all of the messages in the room. Not all options are used.
Select and Place:

Answer Area

```
import requests

webex_token = "NDA2OGV...f0-4434-a696-84fee4047e0a"

room_id = "Y2lzY29zcGF...jU5NjAtNTk0Zi0xMWVhLTk0Mj"

url = "https://webexapis.com/v1/ [          ] [          ] =" + [          ]

payload = {}
headers = {
   'Authorization': 'Bearer ' + [          ]
}

response = requests.request("GET", url, headers=headers,data = payload)

print(response.text.encode('utf8'))
```

room_id	room	messages?
webex_token	roomId	messages

Answer:

Answer Area

```
import requests

webex_token = "NDA2OGV...f0-4434-a696-84fee4047e0a"

room_id = "Y2lzY29zcGF...jU5NjAtNTk0Zi0xMWVhLTk0Mj"

url = "https://webexapis.com/v1/ [messages?] [roomId] =" + [room_id]

payload = {}
headers = {
   'Authorization': 'Bearer ' + [webex_token]
}

response = requests.request("GET", url, headers=headers,data = payload)

print(response.text.encode('utf8'))
```

room	
messages	

Question: 80

What is a difference between a synchronous API and an asynchronous API?

A.Synchronous API calls require an authentication header to be sent while asynchronous calls do not require authentication.

B.Synchronous API calls are returned immediately while asynchronous calls do not guarantee an immediate response.

C.An asynchronous API can make offline calls while synchronous APIs do not have this capability.

D.An asynchronous API can make a larger number of calls in a specified time period than a synchronous API.

Answer: D

Explanation:

The correct answer is **D: An asynchronous API can make a larger number of calls in a specified time period than a synchronous API.**

Here's why:

Synchronous APIs operate in a request-response cycle. When a client makes a request, it blocks (waits) until it receives a response before proceeding further. This blocking nature limits the number of requests a client can handle concurrently. If the server takes time to process the request, the client remains idle, reducing efficiency and throughput. In essence, each request occupies a thread until completion, a bottleneck in handling multiple requests.

Asynchronous APIs, on the other hand, do not require the client to wait for an immediate response. When a client sends a request, the API typically acknowledges the request and returns control to the client. The server processes the request separately and, upon completion, notifies the client via methods like callbacks, webhooks, or message queues. This non-blocking behavior enables the client to send multiple requests without waiting for responses, greatly increasing concurrency and throughput. The client can continue doing other tasks. Asynchronous calls are ideal for operations that might take a long time to complete.

Option A is incorrect as authentication requirements depend on the API design, not the synchronous/asynchronous nature. Option B is misleading because while synchronous responses are immediate in the sense they are sent during the same request/response lifecycle, asynchronous APIs can have fast responses for successful receipt of the request. The key distinction lies in the client's blocking nature, not response speed itself. Option C is incorrect because whether an API can operate offline depends on other design factors, such as caching and data sync mechanisms, not whether the API is synchronous or asynchronous.

For deeper understanding:

Microsoft's guide on asynchronous programming:https://learn.microsoft.com/en-us/dotnet/csharp/async **AWS API design best practices**: https://docs.aws.amazon.com/apigateway/latest/developerguide/api-design-best-practices.html
Difference between synchronous and asynchronous APIs:https://www.educative.io/answers/what-is-the-difference-between-synchronous-and-asynchronous-apis