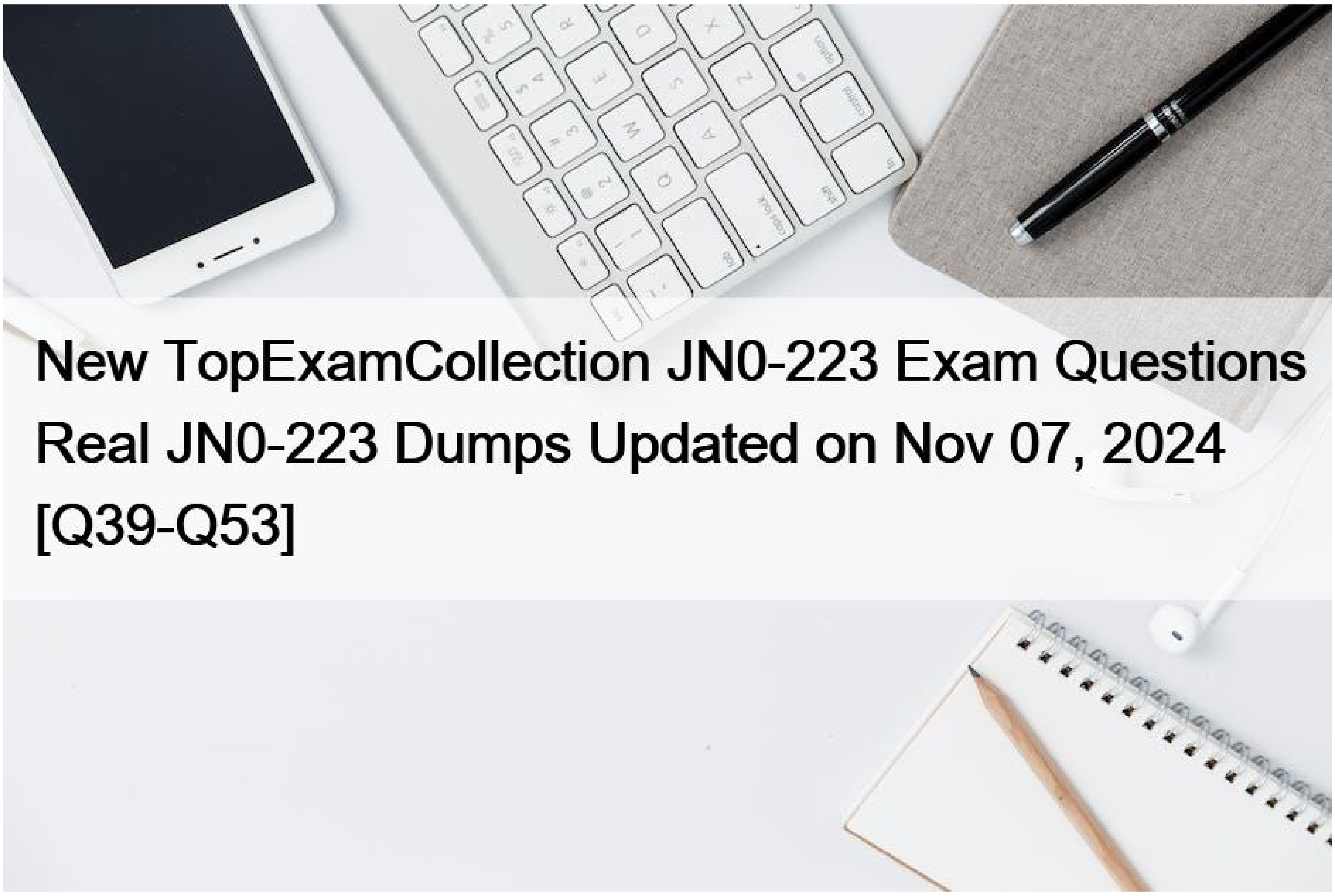
New TopExamCollection JN0-223 Exam Questions Real JN0-223 Dumps Updated on Nov 07, 2024 [Q39-Q53



New TopExamCollection JN0-223 Exam Questions| Real JN0-223 Dumps Updated on Nov 07, 2024
JN0-223 Braindumps &ndash; JN0-223 Questions to Get Better Grades

The Automation and DevOps, Associate (JNCIA-DevOps) certification exam covers a wide range of topics related to automation and DevOps, including the fundamental concepts of automation, scripting languages, and tools used in automation. JN0-223 exam also covers the basics of software development and deployment, DevOps practices, and cloud computing concepts.

Juniper JN0-223 certification exam is designed for individuals who are interested in developing expertise in automation and DevOps practices. Automation and DevOps, Associate (JNCIA-DevOps) certification is an associate-level exam that validates your skills in using Juniper Networks automation tools, protocols, and languages. It is an ideal certification for network engineers, automation engineers, and DevOps engineers who want to advance their careers in the field of network automation.

NO.39 What are two Junos PyEZ configuration object methods? (Choose two.)
* commie ()

- device ()
- lockO
- config ()

In Junos PyEZ, theConfigobject provides various methods for interacting with device configurations. Two of the key methods are:

* lock():This method locks the candidate configuration database to prevent other users or processes from making changes while you are modifying the configuration.

* config():This method is used to create aConfigobject that represents the configuration database, allowing you to load, modify, and commit configuration changes.

* Option C (lock)andOption D (config)are correct because they are valid methods provided by the PyEZConfigobject.

* Option A (commie)andOption B (device)are incorrect as they are not methods of theConfigobject.

Supporting References:

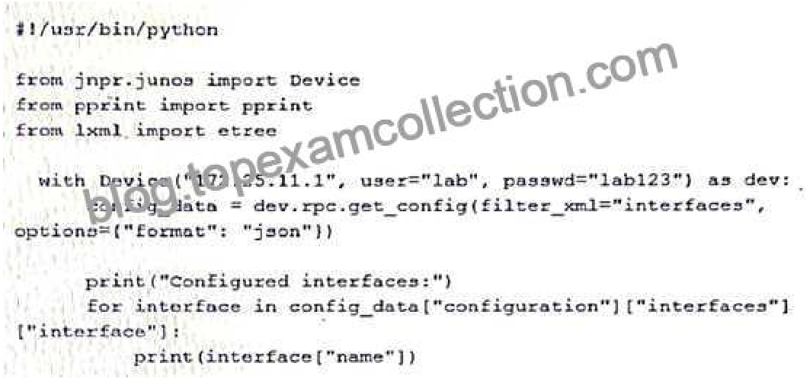
* Junos PyEZ Documentation:Details the methods available in theConfigobject, includinglock() andconfig().

NO.40 Which HTTP status code indicates a response to a successful request?

- 500
- 302
- 200
- 400

Reference: https://www.juniper.net/documentation/en_US/junos-space-sdk/13.1/apiref/com.juniper.junos_space.sdk.help/html/reference/Commonbehav.html

NO.41 Exhibit.



```
#!/usr/bin/python

from jnpr.junos import Device
from pprint import pprint
from lxml import etree

  with Device("172.25.11.1", user="lab", passwd="lab123") as dev:
      config_data = dev.rpc.get_config(filter_xml="interfaces",
options={"format": "json"})

      print("Configured interfaces:")
      for interface in config_data["configuration"]["interfaces"]
["interface"]:
          print(interface["name"])
```

Referring to the exhibit, which two statements about the script are correct? (Choose two.)

- The script prints the name of each configured interface.
- The script retrieves the interface configuration in XML
- The script prints interface information for each interface name.
- The script retrieves the interface configuration in JSON.

NO.42 Which Junos API supports direct modification of the Ephemeral database?

- JET

* WebSocket
* SOAP
* REST

In Junos, theJET (Junos Extension Toolkit)API supports direct modification of theEphemeral database.

The Ephemeral database is a temporary configuration database used in Junos OS, allowing for changes that do not persist after a reboot unless explicitly committed to the permanent configuration.

* JET API: Allows for high-performance interactions with Junos, including the ability to make changes to the Ephemeral database, which is useful for temporary configurations, dynamic policies, and other operational tasks.

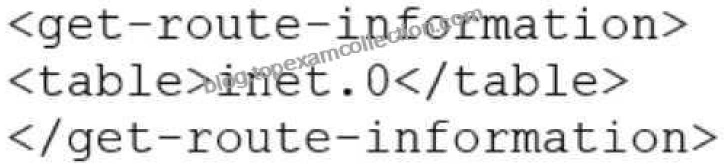
Other options like WebSocket, SOAP, and REST do not provide direct access to the Ephemeral database in Junos.

* Juniper Networks JET Documentation: Details how JET API interacts with the Ephemeral database.

* Junos Automation and DevOps Documentation: Discusses the use of JET for automation and dynamic configuration.

References:

NO.43 Exhibit.



```
<get-route-information>
<table>inet.0</table>
</get-route-information>
```

You need to translate the configuration shown in the exhibit into a PyEZ RPC call.

Which line shows the proper PyEZ syntax?

* dev.rpc.get_route_information(table=&#8221;inet.0&#8243;)
* dev.rpc.get-route-information(table=&#8221;inet.0&#8243;)
* dev.rpc.get_route_information:table=&#8221;inet.0&#8243;
* dev.rpc.get-route-information:table=&#8221;inet.0&#8243;

Reference: https://www.juniper.net/documentation/en_US/junos-pyez/topics/task/program/junos-pyez-rpcsexecuting.html

NO.44 You are creating a JSON data structure and must add a title called, The &#8220;Big&#8221; Router.

In this scenario, which example will produce the desired results?

* {&#8220;title&#8221;: &#8220;&#8221;The &#8220;Big&#8221; Router&#8221;&#8221;}
* {&#8220;title&#8221;: &#8220;The n&#8221;Bign&#8221; Router&#8221;}
* {&#8220;title&#8221;: &#8220;The &#8220;Big&#8221; Router&#8221;}
* {&#8220;title&#8221;: &#8220;The &#8220;&#8216;Big'&#8221; Router&#8221;}

NO.45 Which two statements about NETCONF are true? (Choose two.)

* It uses the operations layer to lock the configuration of a Junos device.
* It uses the messages layer to commit the configuration of a Junos device.
* It uses the messages layer to lock the configuration of a Junos device.

* It uses the operations layer to commit the configuration of a Junos device.
NETCONF (Network Configuration Protocol) operates through different layers, with the operations layer being particularly important for managing configurations:

* Operations Layer (A & D): This layer is responsible for actions like locking and committing the configuration on a Junos device. The lock operation prevents other sessions from modifying the configuration, and the commit operation applies the configuration changes to the device.

Options B and Care incorrect because the messages layer handles the communication aspects, such as exchanging data between the client and server, not performing configuration operations like locking and committing.

* IETF RFC 6241 (NETCONF): Describes the protocol layers and their functions, with a focus on the operations layer.

* Juniper Networks NETCONF Documentation: Provides insights into how NETCONF operations are managed in Junos
References:

NO.46 Which two statements are correct about a Python list data type? (Choose two.)
* The data contained in a list data type can be modified.
* The data contained in a list data type is sequenced and indexed starting from 0.
* The data contained in a list data type cannot be modified.
* The data contained in a list data type is not sequenced or indexed.
Python lists have the following characteristics:

* Modifiable Data (A): Lists are mutable, meaning you can change, add, or remove elements after the list has been created.

* Sequenced and Indexed (B): Lists maintain the order of their elements and are indexed starting from 0.

This means you can access elements by their position in the list.

Option C is incorrectbecause lists are mutable, allowing modifications.Option D is incorrectbecause lists are indeed sequenced and indexed, unlike dictionaries.

* Python Official Documentation: Covers the properties of lists, including mutability and indexing.

* Python Data Structures Guide: Explains list operations and how to manipulate them.

References:

NO.47 Which two PyEZ object methods are included by default when using a Python context manager? (Choose two.)
* lock () and unlock (>
* open() and close()
* |load() and commit ()
* pdiff() and diff()
When using a Python context manager with Junos PyEZ, two key methods are automatically included:

* open()andclose(): These methods are used to establish and terminate a connection to a Junos device.

When you use a context manager (thewithstatement),open()is called when entering the block, andclose() is called when exiting, ensuring the connection is properly managed.

* lock()andunlock(): These methods are used to lock the configuration database to prevent other users from making changes while you are working on it. When using a context manager,lock()is called at the start of the block, andunlock()is called at the end, ensuring safe configuration changes.

Supporting References:

* Junos PyEZ Documentation:The documentation explains how context managers work in PyEZ, including the automatic invocation ofopen(),close(),lock(), andunlock()methods.

NO.48 What are two important aspects of the DevOps culture? (Choose two.)
* communication
* separation of duties
* use of specific tools
* people
Two important aspects of the DevOps culture are:

* Communication (A): In a DevOps environment, continuous and effective communication between development, operations, and other stakeholders is crucial. It ensures that everyone is aligned, reduces misunderstandings, and facilitates faster and more efficient workflows.

* People (D): DevOps is as much about the people as it is about the processes and tools. The culture emphasizes collaboration, shared responsibility, and a mindset focused on continuous improvement.

Empowering people to work together across different disciplines is at the heart of DevOps.

Options B and C are not primary aspects of the DevOps culture. While tools are essential in implementing DevOps practices, the culture emphasizes communication and collaboration among people more than the use of specific tools or rigid separation of duties.

* DevOps Handbook: Discusses the cultural aspects of DevOps, with a focus on communication and people.

* DevOps Best Practices: Highlights the importance of fostering a culture that prioritizes collaboration and shared goals.

References:

NO.49 In the PyEZ utils module, which class would be used to begin the software upgrade process on a Junos device?
* SW
* FS
* SCP
* Config

NO.50 Junos PyEZ tables are formatted using which file type?
* SON
* YAML
* txt
* IXML
Junos PyEZ uses YAML (YAML Ain&#8217;t Markup Language) files to define the format for tables and views when working with operational and configuration data. YAML is a human-readable data format that is commonly used for configuration files, making it suitable for defining data structures in PyEZ.

* Option B (YAML)is correct because PyEZ tables are defined using YAML files.

* Options A (JSON), C (txt), and D (IXML)are incorrect in this context, as YAML is the standard format used.

Supporting References:

* Junos PyEZ Tables Documentation:Explains the use of YAML files for formatting tables and views in Junos PyEZ.

NO.51 Which two statements are correct about using the Junos REST API? (Choose two.)
* It supports data In CSV format.
* It must use SSH for a connection.
* NETCONF is not supported.
* It is a simple configuration.
A). It supports data in CSV format:The Junos REST API supports multiple data formats for transferring information between systems, including XML, JSON, andCSV (Comma Separated Values). This flexibility allows for easier data parsing, especially in environments where structured data (likeCSV) is a standard. CSV is often used for bulk data export or import and reporting purposes, making it an essential format for automation tasks involving external systems or large datasets.

Example Usage in REST API:When using the Junos REST API, a user can request configuration or operational data and specify the response format (XML, JSON, or CSV). CSV is particularly useful when integrating Junos devices with systems that require easily readable, tabular formats.

NO.52 Which two programming languages are used for Junos on-box scripting? (Choose two.)
* Perl
* Ruby
* SLAX
* XSLT
Junos on-box scripting supports the following programming languages:

* SLAX (C): SLAX (Structured Language for XML) is a scripting language designed specifically for Junos devices. It allows for easy manipulation of XML data, making it ideal for creating Junos scripts that interact with device configurations.

* XSLT (D): XSLT (Extensible Stylesheet Language Transformations) is another language used for transforming XML documents into other formats. It is commonly used in Junos for transforming XML data into different views or outputs.

Options A (Perl) and B (Ruby) are not used for Junos on-box scripting.While these languages are popular in other contexts, Junos scripting relies heavily on XML-based languages like SLAX and XSLT.

* Junos XML API and Scripting Guide: Describes the use of SLAX and XSLT for on-box scripting.

* Juniper Networks Automation Documentation: Provides examples and best practices for using SLAX and XSLT in Junos scripting.

References:

NO.53 Which cur1 command will successfully retrieve interface information from a Juniper device?
* cur1 -&#8221;lab:lab123&#8221; http://10.200.1.1:3000/rpc/get-interface-information -X GET
* cur1 -&#8221;lab:lab123&#8221; http://10.200.1.1:3443/rpc/get-interface-information -X POST
* cur1 -&#8221;lab:lab123&#8221; http://10.200.1.1:3443/rpc/get-interface-information -X POST
* cur1 -&#8221;lab:lab123&#8221; http://10.200.1.1:8443/rpc/get-interface-information -X GET