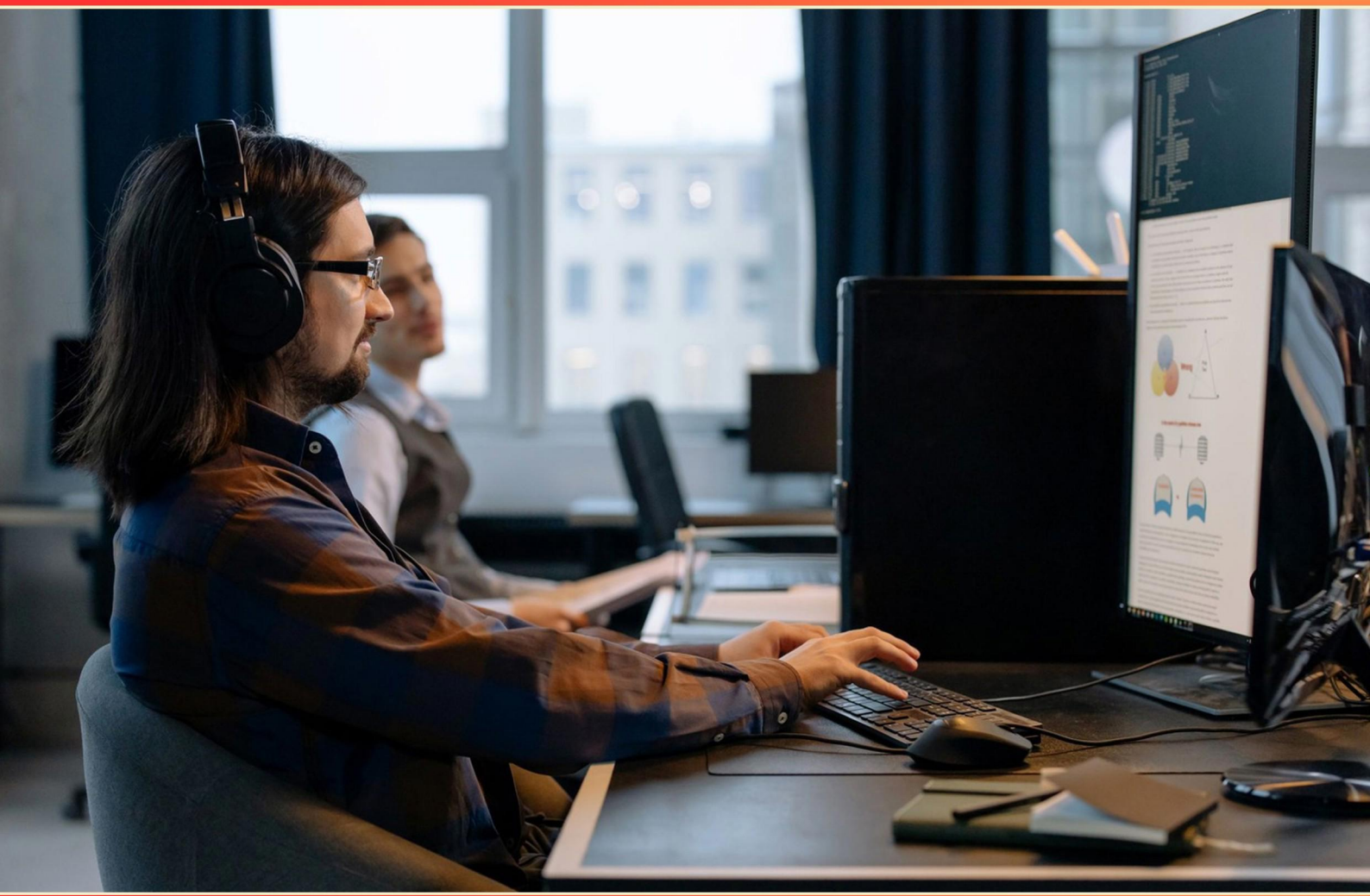
REST ASSURED QUALITY ASSURANCE (QA) PRACTICE TEST (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE

https://restassuredqa.examzify.com

Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright 1

Table of Contents 2

Introduction 3

How to Use This Guide 4

Questions 5

Answers 8

Explanations 10

Next Steps 16

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

SAMPLE

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

SAMPLE

Questions

SAMPLE

1. What is an essential feature of REST Assured?

A. It only supports Java programming.

B. It allows easy testing of HTTP requests and responses.

C. It requires an extensive learning curve.

D. It cannot handle complex payloads.

2. What does REST stand for in REST Assured?

A. Representational State Transfer

B. Resource Efficient Service Transfer

C. Remote Event Streaming Technology

D. Real-time Service Transfer

3. What does the statusCode() method do in REST Assured?

A. It returns the status code as an integer

B. It asserts the HTTP status code of the response

C. It sets the expected status code for a request

D. It logs the status code to the console

4. When should one consider using REST for an API development?

A. When security is a major concern

B. When resources are limited and ease of use is important

C. When complex operations are needed

D. When using SOAP is mandatory

5. Which of these is a common response format that REST Assured can validate?

A. CSV.

B. XML.

C. HTML.

D. Text files.

SAMPLE

6. What assertion can you use to check a specific JSON path value in the response?

A. body().value("jsonPath")

B. body().jsonPath().getString("jsonPath")

C. check().jsonPath().value("jsonPath")

D. response().jsonPath().get("jsonPath")

7. What are REST APIs generally preferred for?

A. Highly transactional applications

B. Simple data exposure and operations

C. Complex desktop applications

D. Extensive caching operations

8. How can you assert that a JSON response has a specific content type?

A. By using the type() method

B. By using the validateContent() method

C. By using the contentType() method

D. By using checkContent() method

9. Which HTTP methods are commonly used in REST APIs?

A. CONNECT, TRACE, PATCH

B. GET, POST, PUT, DELETE

C. OPTIONS, MERGE, LINK

D. HEAD, REPLACE, DELETE

10. Which of the following is NOT a challenge faced in API testing?

A. Output verification and validation

B. Parameter Selection

C. Testing graphical user interfaces

D. Call sequencing

SAMPLE

Answers

SAMPLE

1. B
2. A
3. B
4. B
5. B
6. B
7. B
8. C
9. B
10. C

SAMPLE

Explanations

SAMPLE

1. What is an essential feature of REST Assured?

A. It only supports Java programming.

B. It allows easy testing of HTTP requests and responses.

C. It requires an extensive learning curve.

D. It cannot handle complex payloads.

An essential feature of REST Assured is its ability to allow for easy testing of HTTP requests and responses. This tool is specifically designed to simplify the process of testing RESTful APIs by providing a domain-specific language (DSL) that makes it straightforward for testers to create requests and validate responses with minimal coding effort. This accessibility helps automate API testing and makes it seamless to perform assertions on various aspects of the response, such as status codes, headers, and body content. This focus on HTTP interactions makes REST Assured particularly valuable for Quality Assurance teams who need to ensure the reliability and functionality of web services. The handy methods and fluent API style provided by REST Assured reduce the complexity of writing tests, enabling testers to focus more on verifying the application logic rather than getting bogged down by the intricacies of HTTP communications.

2. What does REST stand for in REST Assured?

A. Representational State Transfer

B. Resource Efficient Service Transfer

C. Remote Event Streaming Technology

D. Real-time Service Transfer

The term REST stands for "Representational State Transfer," which is a crucial architectural style for designing networked applications. In the context of REST Assured, this represents a set of constraints and principles aimed at enhancing the performance, scalability, and simplicity of web services. REST leverages the existing capabilities of the HTTP protocol, utilizing stateless interactions and standard operations like GET, POST, PUT, and DELETE. This allows for resources to be manipulated independently through resource identifiers (URLs), facilitating easy integration and communication between different systems. Understanding REST as "Representational State Transfer" is fundamental for utilizing REST Assured effectively, as the library is built to test and validate RESTful services accurately. It provides a framework for verifying that the API adheres to the principles of REST, ensuring that the interactions and behaviors meet expectations based on specified resource representations. The other options do not correctly capture the architectural meaning or application of REST in the context of web services and testing, which is why they do not align with the correct definition.

3. What does the statusCode() method do in REST Assured?

A. It returns the status code as an integer

B. It asserts the HTTP status code of the response

C. It sets the expected status code for a request

D. It logs the status code to the console

The statusCode() method in REST Assured is designed to assert the HTTP status code of the response received from a server after sending a request. When this method is called, it compares the actual status code from the response with an expected status code specified in the assertion. If the status codes do not match, it throws an assertion error, which helps in validating that the API behaves as expected under various conditions. This method is crucial in API testing because different status codes indicate different outcomes of HTTP requests, such as success (200), redirection (300), client errors (400), and server errors (500). By asserting the status code, testers can confirm that the API responds correctly to requests, maintaining reliability and correctness in the application's functionalities. This is a fundamental aspect of validating RESTful services as it helps ensure that the APIs are responding as intended based on the design specifications.

SAMPLE

4. When should one consider using REST for an API development?

A. When security is a major concern

B. When resources are limited and ease of use is important

C. When complex operations are needed

D. When using SOAP is mandatory

Choosing REST for API development is particularly beneficial when resources are limited and ease of use is important. REST, or Representational State Transfer, is designed to be lightweight and simple. Its stateless nature allows clients and servers to communicate in a straightforward manner, which can lead to faster development cycles and reduced overhead. This simplicity makes RESTful APIs easier to consume, leading to better developer productivity and quicker onboarding for new team members. RESTful services use standard HTTP methods (GET, POST, PUT, DELETE, etc.), which are widely understood and supported, offering a more intuitive approach. When working in environments where quick iterations and minimal resource consumption are priorities, REST can provide an efficient solution that aligns well with these needs. In contrast to options that consider security, complexity, or mandatory protocols like SOAP, REST's design focuses on resource-oriented interactions, making it more approachable and manageable in scenarios where developer resources or technical expertise may be limited.

5. Which of these is a common response format that REST Assured can validate?

A. CSV.

B. XML.

C. HTML.

D. Text files.

The correct answer is XML, which is a common response format that REST Assured can easily validate. REST Assured is designed to interact with RESTful web services and has built-in support for various data formats, among which XML is prominent. XML, which stands for eXtensible Markup Language, is a markup language that defines rules for encoding documents in a format that is both human-readable and machine-readable. REST Assured enables you to parse XML responses using its powerful and intuitive testing capabilities. This includes using XPath to navigate through XML documents and validate specific elements and values, ensuring that the API provides the expected structure and content in its responses. While CSV, HTML, and text files can be part of other testing scenarios, they do not have the same level of direct support and functionality within REST Assured as XML does. CSV typically involves handling flat file structures where each line is a data record, which is outside the primary use case of REST Assured. HTML responses can be parsed, but they are often not structured data in the same way XML is, making validations more cumbersome. Additionally, text files are essentially unstructured and would not benefit from the specific XML validation capabilities that REST Assured provides. Therefore, XML stands out as the most

SAMPLE

6. What assertion can you use to check a specific JSON path value in the response?

A. body().value("jsonPath")

B. body().jsonPath().getString("jsonPath")

C. check().jsonPath().value("jsonPath")

D. response().jsonPath().get("jsonPath")

The assertion that allows you to check a specific JSON path value in the response is correctly identified as using the approach that involves accessing the JSON path directly through the response's jsonPath method. This method specifically allows you to retrieve a value based on the JSON path expression you provide. Using "body().jsonPath().getString("jsonPath")" is essential because it first navigates to the body of the response, then uses the jsonPath method to parse the JSON structure. By invoking the getString method, you can extract the value at the specified JSON path as a string, which is commonly needed when validating that the actual response matches expected criteria. This approach aligns with the necessary syntax and method chaining required by REST Assured for effective JSON data validation and is a key aspect of working with JSON assertions in API testing. It is both a clear and direct way to ascertain that specific values in the JSON response match what you expect based on your test cases.

7. What are REST APIs generally preferred for?

A. Highly transactional applications

B. Simple data exposure and operations

C. Complex desktop applications

D. Extensive caching operations

REST APIs are generally preferred for simple data exposure and operations due to their lightweight nature and ease of use. They are designed to work over HTTP and typically use standard methods such as GET, POST, PUT, and DELETE to perform operations on resources. This simplicity makes REST APIs an excellent choice for applications that require straightforward CRUD (Create, Read, Update, Delete) operations, allowing developers to interact with data easily without the complexities often associated with other types of APIs. Additionally, REST APIs utilize JSON and XML for data interchange, making them accessible and easy to integrate with various platforms, especially in web and mobile applications. Their stateless nature and separation of client and server architecture further enhance this preference, allowing for scalability and flexibility in application design. In contrast, options like highly transactional applications, complex desktop applications, and extensive caching operations tend to be more suited to other types of APIs or architectures that may handle more complexity, state management, or performance optimizations specific to those use cases.

SAMPLE

8. How can you assert that a JSON response has a specific content type?

A. By using the type() method

B. By using the validateContent() method

C. By using the contentType() method

D. By using checkContent() method

The contentType() method is specifically designed for validating the content type of a response in REST Assured. When you are working with REST APIs that return JSON responses, it is crucial to ensure that the returned content matches the expected media type, such as "application/json". This method allows you to assert the exact content type, which helps in verifying that the API behaves as expected and that the response data can be properly processed. Using contentType() provides a clear and intuitive way to check the HTTP headers of the response. If the response does not match the specified content type, the assertion will fail, alerting you to a potential issue with the API. Other methods listed do not serve this specific purpose. For instance, the type() method is typically used for extracting the response type but does not focus on asserting content type. ValidateContent() and checkContent() methods are not recognized standard methods within REST Assured, which specifically undermines their applicability in this context. This highlights why contentType() is the correct and relevant choice for asserting the JSON response's content type.

9. Which HTTP methods are commonly used in REST APIs?

A. CONNECT, TRACE, PATCH

B. GET, POST, PUT, DELETE

C. OPTIONS, MERGE, LINK

D. HEAD, REPLACE, DELETE

The choice indicating GET, POST, PUT, and DELETE as commonly used HTTP methods in REST APIs is indeed the correct one due to their fundamental roles in the REST architecture. GET is utilized for fetching data from a server without modifying it, making it a primary method for retrieving resources. POST is employed for creating new resources on the server, allowing clients to submit data for processing. PUT is specifically used for updating existing resources or creating a resource if it does not exist, thus enabling full resource replacement. DELETE serves to remove resources from the server, effectively allowing clients to manage resource lifecycles. These methods align well with the principles of RESTful architecture, which emphasizes stateless communication and resource manipulation through standard HTTP methods. They are widely adopted due to their clear semantics, making it easier for developers to implement and understand interactions with RESTful services. The other options list HTTP methods that are either less common in standard RESTful APIs or are not typically recognized as action verbs in this context. For example, methods like CONNECT, TRACE, and MERGE do not play a central role in the operational paradigms of most REST APIs, focusing instead on debugging, tunneling, or other specialized functions. This makes them less relevant compared to the foundational CRUD operations encaps

SAMPLE

10. Which of the following is NOT a challenge faced in API testing?

A. Output verification and validation

B. Parameter Selection

C. Testing graphical user interfaces

D. Call sequencing

Testing graphical user interfaces (GUIs) is not typically a challenge encountered in API testing because API testing focuses on interactions at the network level, specifically dealing with requests and responses sent over protocols like HTTP. It assesses the functionality, reliability, performance, and security of APIs by examining the data exchanged rather than the visual components presented to users. When it comes to output verification and validation, parameter selection, and call sequencing, these are specific considerations essential in API testing. Output verification involves ensuring that the responses returned by the API are accurate and meet the expected results based on the input provided. Parameter selection is critical as it determines which data is sent in requests, impacting the results and the behavior of the API. Call sequencing refers to the correct order or timing in which API endpoints should be called, which is crucial for validating workflows or state-dependent actions. Thus, these elements present challenges that testers must address when evaluating the performance and correctness of APIs.

SAMPLE

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

https://restassuredqa.examzify.com

We wish you the very best on your exam journey. You've got this!

SAMPLE