

REST Assured Quality Assurance (QA) Practice Test (Sample)

Study Guide



Everything you need from our exam experts!

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.

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!

Questions

SAMPLE

- 1. Which REST frameworks are commonly used?**
 - A. Spring MVC and Django**
 - B. Flask and Yii**
 - C. Hibernate and Angular**
 - D. CSS and HTML**

- 2. In REST Assured, how is a request made to a specific API endpoint?**
 - A. By specifying the basePath.**
 - B. By defining the baseURI only.**
 - C. Through the request body only.**
 - D. By chaining methods after specifying the baseURI.**

- 3. Which method is used to assert that the response body contains a specific value?**
 - A. assertThat().containsValue()**
 - B. body().contains()**
 - C. body().jsonPath().get()**
 - D. body().matches()**

- 4. What should be included in API test cases for thorough verification?**
 - A. Input parameters only**
 - B. Basic functionality and scope**
 - C. User feedback**
 - D. Graphical interface design**

- 5. How can you handle authentication in REST Assured requests?**
 - A. By using the secure() method.**
 - B. By using the authentication() method.**
 - C. By using the auth() method with basic(), preemptive(), etc.**
 - D. By using the login() method.**

- 6. What does HTTP status code 204 signify during a request?**
- A. Success**
 - B. No content in the response body**
 - C. Resource not found**
 - D. Resource has been created**
- 7. Which type of testing focuses primarily on verifying individual units or code modules?**
- A. Integration Testing**
 - B. Unit Testing**
 - C. Functional Testing**
 - D. System Testing**
- 8. What is the primary use of the PUT method in RESTAPI?**
- A. To create a new resource**
 - B. To delete an existing resource**
 - C. To develop or overwrite a resource**
 - D. To read a resource**
- 9. When asserting a response status code in REST Assured, which code represents a successful response?**
- A. 404.**
 - B. 200.**
 - C. 301.**
 - D. 500.**
- 10. Which of these is a common response format that REST Assured can validate?**
- A. CSV.**
 - B. XML.**
 - C. HTML.**
 - D. Text files.**

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. Which REST frameworks are commonly used?

- A. Spring MVC and Django**
- B. Flask and Yii
- C. Hibernate and Angular
- D. CSS and HTML

The choice of Spring MVC and Django as commonly used REST frameworks is accurate because both are well-established frameworks designed to facilitate the development of RESTful web services. Spring MVC is a part of the larger Spring Framework, widely adopted for building enterprise-level applications in Java. It provides comprehensive capabilities for handling HTTP requests, mapping these to appropriate controller methods, and returning structured data in formats like JSON or XML. Its flexibility and robust feature set make it a popular choice for RESTful service development. Django, on the other hand, is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It includes a powerful toolkit for building REST APIs through Django REST Framework, which offers features like serializers, viewsets, and routing, significantly simplifying the process of creating RESTful services in Python. In contrast, the other options do not represent frameworks primarily focused on building RESTful services. For instance, while Flask is a lightweight Python framework used for web development, it isn't exclusively a REST framework and is less complex than Django. Hibernate is a Java ORM tool, and Angular is a front-end JavaScript framework for building single-page applications. CSS and HTML are technologies for web design and layout, not frameworks for developing REST services. Thus, Spring MVC

2. In REST Assured, how is a request made to a specific API endpoint?

- A. By specifying the basePath.
- B. By defining the baseURI only.
- C. Through the request body only.
- D. By chaining methods after specifying the baseURI.**

In REST Assured, making a request to a specific API endpoint involves using a combination of methods that are chained together after specifying the base URI. To construct a complete request, you typically start by defining the base URI, which indicates the root address of your API. Following that, you can chain various methods to specify HTTP verbs (such as GET, POST, etc.), the specific endpoint (like `/users``), any query parameters or headers, and the request body if necessary. This chaining approach allows for a clear and fluent programming style, which enhances readability and maintainability of the test scripts. By sequentially adding to the request, such as indicating the path and the desired parameters, it effectively constructs a full and valid request that aligns with the API's specifications. Other options do not adequately capture the process. Specifying only the base URI does not complete the request as it lacks the specific endpoint and HTTP method. While defining basePath can help organize the API endpoints, it is not sufficient by itself for making a request. Similarly, using the request body alone fails to address how the request reaches the API endpoint, as HTTP methods and the endpoint path must also be defined. Thus, chaining methods after the base URI is the most comprehensive and accurate representation

3. Which method is used to assert that the response body contains a specific value?

- A. `assertThat().containsValue()`
- B. `body().contains()`**
- C. `body().jsonPath().get()`
- D. `body().matches()`

The method used to assert that the response body contains a specific value is indeed focused on how REST Assured allows you to interact with the response and extract or validate data from it. In this case, calling `body().contains()` is a valid way to check for the presence of a specific value within the response body. When using REST Assured, the `body()` method is utilized to access the response body, and from there, various assertions can be made. While `contains()` implies that you want to check for a substring or specific content within the body of the response, it aligns with how REST Assured allows for efficient validations. This method is particularly useful when you want to confirm that a piece of data is indeed part of what's returned in the full response, which is a common scenario in testing RESTful APIs. The other choices represent methods that do not clearly focus on asserting the presence of a specific value within the body in the same straightforward manner as `body().contains()`. The other options might relate to JSON path retrieval or content matching, but they do not directly encompass the assertion of simply containing a value as succinctly.

4. What should be included in API test cases for thorough verification?

- A. Input parameters only
- B. Basic functionality and scope**
- C. User feedback
- D. Graphical interface design

When developing API test cases for thorough verification, including basic functionality and scope is crucial. This approach ensures that the API performs its intended tasks correctly and meets the defined requirements. It involves checking that the various components of the API — such as endpoints, methods (GET, POST, PUT, DELETE), and response codes — are functioning as expected. Focusing on basic functionality allows testers to verify that the API responds correctly to various requests, returns appropriate status codes, and handles errors as specified. Covering the scope of the API involves understanding the limits of its capabilities, ensuring that all features are tested thoroughly. Test cases based on fundamentals provide a comprehensive foundation, guiding the development of more advanced tests later. Other options do not inherently provide a comprehensive framework for testing APIs effectively. Input parameters are important, but without understanding the broader functionality and requirements, they alone will not ensure thorough testing. User feedback can offer insights but is not a substitute for actual test cases that validate the API's operations and its specifications. Graphical interface design is irrelevant in the context of API testing, as APIs do not typically have a user interface to interact with; instead, they communicate via requests and responses over a network.

5. How can you handle authentication in REST Assured requests?

- A. By using the secure() method.**
- B. By using the authentication() method.**
- C. By using the auth() method with basic(), preemptive(), etc.**
- D. By using the login() method.**

Handling authentication in REST Assured requests is effectively achieved by using the `auth()` method, which provides various authentication strategies, including basic authentication, preemptive authentication, and others. This method allows you to specify different types of authentication schemes tailored to the needs of the API you are working with. The `basic()` and `preemptive()` options under the `auth()` method enable you to set up the necessary credentials in a straightforward and flexible way. With basic authentication, you provide a username and password, and REST Assured handles the preparation of the headers required for the request. Preemptive authentication is particularly useful where the server requires authentication before responding to the first request, ensuring that your request includes the necessary credentials right away. In contrast, the other methods listed are either not standard practices in REST Assured or do not exist. For instance, the `secure()` method and the `login()` method do not offer functionality for handling authentication in REST Assured, while the `authentication()` method is not the primary method for managing it. Thus, option C stands out as the most robust and accurate approach to achieving authentication in REST Assured.

6. What does HTTP status code 204 signify during a request?

- A. Success**
- B. No content in the response body**
- C. Resource not found**
- D. Resource has been created**

HTTP status code 204 signifies "No Content," which indicates that the server has successfully processed the request but is not returning any content in the response body. This is typically used in cases where the client does not need to update its view of the resource, such as after a successful deletion or when an operation does not require data to be sent back. The implication of a 204 status code is that the action requested by the client was completed correctly, but there is no additional information to convey, hence the absence of a message body. This differs from a successful status that typically includes a response body with additional details, such as in the case of a 200 status code. Other options may imply different states of a request or resource management. For example, an indication of success might lead one to think of a 200 status code, resource not found refers to a 404 status code, and resource creation corresponds to a 201 status code. Each of these carries its specific meaning and context in HTTP communications.

7. Which type of testing focuses primarily on verifying individual units or code modules?

- A. Integration Testing**
- B. Unit Testing**
- C. Functional Testing**
- D. System Testing**

Unit testing is a crucial stage in the software development process that aims specifically at verifying the functionality of individual units or components of code, often functions or methods. This type of testing is typically performed by developers during the development phase and focuses on ensuring that each piece of code behaves as expected under various conditions. Unit tests are designed to isolate each component or module, allowing developers to identify bugs at an early stage, thus facilitating easier debugging and a more efficient development process. By catching defects early, unit testing contributes to a more robust and reliable codebase. It sets a foundation upon which subsequent testing types, such as integration testing and system testing, can build, as they typically require a functioning set of modules or components to understand how they interact with one another. In this context, integration testing examines the interactions between multiple components rather than individual units. Functional testing assesses the software against functional requirements, focusing on user scenarios, while system testing evaluates the entire system's compliance with specified requirements and behaviors. Therefore, unit testing stands apart due to its specific focus on the functionality of single code units, making it essential for ensuring high code quality and reliable software development.

8. What is the primary use of the PUT method in RESTAPI?

- A. To create a new resource**
- B. To delete an existing resource**
- C. To develop or overwrite a resource**
- D. To read a resource**

The primary use of the PUT method in a REST API is to develop or overwrite a resource. PUT is typically employed for updates or the creation of a resource at a specific URI. When a client sends a PUT request, it usually includes the representation of the resource in the body of the request. If the resource already exists at the provided URI, the server will replace it with the new representation. If it does not exist, the server can create a new resource with that URI. This makes PUT an idempotent operation, meaning that multiple identical requests will result in the same state on the server, reinforcing the idea of developing or overwriting a resource as needed.

9. When asserting a response status code in REST Assured, which code represents a successful response?

- A. 404.
- B. 200.**
- C. 301.
- D. 500.

A status code of 200 in REST Assured signifies that the request was successful and that the server processed the request as expected. This status code is part of the HTTP standard, indicating that the response contains the requested information. It represents a successful outcome, typically associated with successful GET requests where the server returns the relevant data. In the context of web services and APIs, a successful response is critical as it confirms that the operation was performed correctly. Other status codes listed serve different purposes: 404 indicates that the requested resource was not found, 301 signifies that the resource has been permanently moved to a different URL, and 500 denotes an internal server error, meaning something has gone wrong on the server-side processing of the request. Understanding these codes helps in effectively diagnosing issues in web communication and managing server-client interactions.

10. 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

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