1. Describe REST Assured in detail.

With the help of the Java package REST Assured, developers can create robust, maintainable tests for RESTful APIs. It is frequently employed to test XML- and JSON-based web applications. Also supported are all methods, such as GET,, DELETE, PATCH, POST, and PUT.

|  |
| --- |
| ***If you want to enrich your career and become a professional in REST Assured, then enroll in "***[***REST Assured Training***](https://mindmajix.com/rest-assured-training)***". This course will help you to achieve excellence in this domain.*** |

2. Describe REST.

Representational state transfer, sometimes known as REST, is an architectural style or design pattern for APIs. A RESTful web service discloses resource information about itself.

|  |
| --- |
| **Related Article:**[**Rest Assured Tutorial**](https://mindmajix.com/rest-assured-tutorial) |

3. What exactly is JSON?

It is a message standard that describes structured data that is based on the object syntax of JavaScript. In web applications, JSON is often used to convey data to client and the server.

4. What protocol is utilized by RESTful Web Services?

The HTTP protocol is used by RESTful web services to transmit data between both the server and the client.

5. What exactly is "client-server architecture"?

How a server distributes services and resources to one or more clients is outlined in the client-server architecture model. Mail servers, data centers and Web servers are a few instances of servers. Therefore, when a client requests a resource, the server fulfills the request.

6. RESTfully define a resource.

Any material is treated as a resource in the REST framework. This material consists of dynamic business information, HTML pages, text files, photos, and videos. These resources are accessible to users through a REST server, which can also modify them. Each resource is identified by a unique URI or global IDEST

7. Explain REST Assured method chaining

Method chaining is a common syntax for calling any number of methods in the setting of object-oriented programming languages. Every function compares an object, allowing for the chaining of several calls in a single line. This trait implies that variables are not required to maintain interim outcomes.

8. Why would a programmer choose REST Assured over Postman to automate RESTful services?

Postman cannot accomplish this since REST Assured has the ability to customize reports. Additionally, REST Assured Java client allows code reuse, whereas Postman does not. Finally, Postman has a limit of one data file, whereas REST Assured has no constraints on data file submissions for collections.

9. What are the details of the request?

In REST Assured, request specifications are used to group similar request specifications and transform them into a specific object. The base URL, header, baseline path, as well as other parameters can all be defined using this interface. To get a connection for the Request specification, you must utilize the provided() feature of the REST Assured class.

10. How does REST Assured start a request specification?

**syntax**

RequestSpecification reqSpec = RestAssured.given();

reqSpec.baseUri("http://localhost:8080")

reqSpec.basePath("/employees");

Top Rest Assured Interview Questions for Experienced

11. How is chaining carried out in REST Assured?

Method chaining is a technique used in [object-oriented programming](https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/) languages to call numerous methods at once. Each function compares an object, enabling the chaining of calls into a single phrase without the need for variables to store intermediate results.

12. Create a piece of code that uses REST Assured to test REST API.

import org.testng.annotations.Test;

import io.restassured.RestAssured;

import io.restassured.http.Method;

import io.restassured.response.Response;

import io.restassured.specification.RequestSpecification;

public class EmployeesTest

{

@Test

public void GetAllEmoloyees()

{

// base URL to call

RestAssured.baseURI = "http://localhost:8080/employees/get";

//Provide HTTP method type - GET, and URL to get all employees

//This will give response

Response employ eesResponse = RestAssured.given().request(Method.GET, "/all");

// Print the response in string format

System.out.println(employeesResponse.getBody().asString());

}

}

|  |
| --- |
| **Related Article:**[**REST API Tutorial**](https://mindmajix.com/rest-api-tutorial) |

13. Within the framework of REST Assured, what does a JSON path mean?

Without using XPath, it is simple to retrieve values from an Object document using a JsonPath. In order to obtain an object from a document, it adheres to the Groovy XPath syntax. Think of it as a JSON-specific implementation of XPath.

14. What is a static import, and why would REST Assured utilize one?

A feature of the Java programming language called static import enables the use of members such as fields and methods zoomed as public static inside their container class without naming the field's defining class.

15. In the context of Java, what do serialization and deserialization mean?

The process of converting an organism's state into a stream of bytes is known as serialization. Deserialization, in contrast, hand, is the procedure that uses a byte stream to recreate the Java class in memory. The item is kept alive using this strategy.

16. Name the essential elements of an HTTP response.

There are five components to an HTTP request

* event such as DELETE, GET, and POST. This component displays the HTTP method.
* A unique resource identification number (URI). The service on the server is identified by this element.
* The version over HTTP.
* An application header The information for the message is contained in this element. The message body format, client or browser type, cache settings, and other factors could all be included in the metadata.
* Message body

17. Can a GET query be made in place of a PUT to create a resource?

Because GET has view-only rights, use it. A resource should never be created using the POST and PUT methods.

18. What distinguishes path parameters from query parameters, and how do you use them?

While query parameters are employed to sort/filter resources, path parameters are used to adding relevant on the server. Path parameters occur before the question mark in the URL, while query parameters follow it and look like key-value pairs.

19. Describe the Rest Assured request specification and response.

Rest certain that both the Request Specification and the Response are interfaces. Request specification allows for the specification of the request's appearance. Methods for defining base URL, base paths, headers, bodies, etc. are available in the Request Specification Interface. When returning a response to a request, the Response interface extends the ResponseBody and ResponseOptions interfaces.

20. URI: What is it? What is the primary function of REST-based web services, and how are they structured?

Uniform Resource Identifier is what URI stands for. It is a group of characters created for the URI scheme's extensibility and clear resource identification. A URI's function is to identify a resource or resources on the server that hosts the web service.

Frequently Asked Rest Assured Interview Questions

**21. What is a RESTFul Web service's payload?**

The information you want to send over is called the "payload." This is distinct from components like HTTP/S Request/Response headers, authentication, etc. that package data for transport.

**22. What is the maximum payload size that can be sent using the POST method?**

The service URL is updated by the GET command. However, it shouldn't be larger than allowed by the URL length. However, there is no such restriction for POST.

**23. What technique does caching use?**

Simply said, caching is the act of temporarily storing information and either deliberately or implicitly retrieving it from a high-performance repository, typically memory. By keeping a copy of the object you ordered and later retrieving the cached copy rather than the original, a caching mechanism can speed **up delivery.**

**24. Why does Rest Assured use static import?**

A feature of the [Java](https://www.javatpoint.com/java-tutorial) programming language called static import enables the use of members such as fields and methods that were scoped as public static in their container class without naming the class in which they were defined.

**25. What is Client-server architecture?**

A server provides services and resources to one or even more clients in accordance with the client-server model. Examples of servers include mail servers, file servers, and web servers. As a result, the Client makes a request, and the Server grants it.

**26. Describe REST.**

The term "REpresentational State Transfer" is referred to by its acronym. It implies that the server will convey to the client a depiction of the condition of the particular request whenever a RESTful API is called. The HTTP method you use to specify the action you wish the server to do on that resource.

**27. Describe JSON.**

Based on JavaScript object syntax, JSON or JavaScript Object Notation is a text-based standard for representing organized data. To communicate data to the server and client, it is frequently used in web applications.

**1. What is an API?**

An API (Application Programming Interface) is a software intermediary that enables two applications to communicate with each other. It comprises a number of subroutine definitions, logs, and tools for creating application software.

In an API testing interview, you could be asked to give some [API examples](https://katalon.com/resources-center/blog/api-examples), here are the well-known ones: Google Maps API, Amazon Advertising API, Twitter API, YouTube API, etc.

**2. What are the main differences between API and Web Service?**

* All Web services are APIs but not all APIs are Web services.
* [Web services](https://katalon.com/web-testing)might not contain all the specifications and cannot perform all the tasks that APIs would perform.
* A Web service uses only three styles of use: SOAP, REST, and XML-RPC for communication whereas API may be exposed in multiple ways.
* A Web service always needs a network to operate while APIs don’t need a network for operation.

**3. What are the Limits of API Usage?**

Many APIs have a certain limit set up by the provider. Thus, try to estimate your usage and understand how that will impact the overall cost of the offering. Whether this will be a problem depends in large part on how data is leveraged. Getting caught by a quota and effectively cut off because of budget limitations will render the service (and any system or process depending on it) virtually useless.

**Creating an API (Common Web API Testing interview questions)**

**4. What are some architectural styles for creating a Web API?**

This is one of the fundamental Web API interview questions. Bellows are four common Web API architectural styles:

* HTTP for client-server communication
* XML/JSON as formatting language
* Simple URI as the address for the services
* Stateless communication

**5. Who can use a Web API?**

Web API can be consumed by any clients which support HTTP verbs such as GET, PUT, DELETE, and POST. Since Web API services do not require configuration, they can be easily used by any client. In fact, even portable devices such as mobile devices can easily use Web API, which is undoubtedly the biggest advantage of this technology.

**Testing an API – Top Web API Testing interview questions & answers**

**6. What is API Testing?**

[API testing](https://katalon.com/api-testing/) is a kind of [software testing](https://katalon.com/resources-center/blog/software-testing) that determines if the developed APIs meet expectations regarding the functionality, reliability, performance, and security of the application.

**7. What are the advantages of API Testing?**

In an API interview, they are likely to ask about the advantages of API testing. So be prepared with the significant ones such as:

* ***Test for Core Functionality:***API testing provides access to the application without a user interface. The core and code-level of functionalities of the application will be tested and evaluated early before the GUI tests. This will help detect minor issues which can become bigger during the GUI testing.
* ***Time Effective:***API testing usually is less time-consuming than functional GUI testing. The web elements in GUI testing must be polled, which makes the testing process slower. Particularly, API test automation requires less code so it can provide better and faster test coverage compared to GUI test automation. These will result in cost saving for the testing project.
* ***Language-Independent:*** In API testing, data is exchanged using XML or JSON. These transfer modes are completely language-independent, allowing users to select any coding language when adopting automation testing services for the project.
* ***Easy Integration with GUI:*** API tests enable highly integrable tests, which is particularly useful if you want to perform functional GUI tests after API testing. For instance, simple integration would allow new user accounts to be created within the application before a GUI test started.

**8. Some common protocols used in API testing?**

Many protocols are now available to be used in API testing, such as JMS, REST, HTTP, UDDI and SOAP.

**9. What is the test environment of API?**

Setting up the API’s test environment is not an easy task, so you should have a ready answer if your API testing interview is coming. The test environment of API is a bit complete and requires the configuration of the database and server, depending on the software requirements. No GUI (Graphical User Interface) is available in this test form.

When the installation process is complete, API is verified for proper operation. Throughout the process, the API called from the original environment is set up with different parameters to study the test results.

**10. What are the principles of an API test design?**

The five most important principles of an API test design are:

* Setup: Create objects, start services, initialize data, etc
* Execution: Steps to apply API or the scenario, including logging
* Verification: Oracles to evaluate the result of the execution
* Reporting: Pass, failed, or blocked
* Clean up: Pre-test state

**11. What are the common API testing types?**

While there are certainly specialty tests, and no list can be asked to be comprehensive in this realm, most tests fit broadly into the following nine categories that you should remember before attending an API testing interview.

1. Validation Testing
2. [Functional Testing](https://katalon.com/resources-center/blog/functional-testing)
3. [UI testing](https://katalon.com/resources-center/blog/ui-testing)
4. Load testing
5. Runtime/ Error Detection
6. Security testing
7. Penetration testing
8. Fuzz testing
9. Interoperability and WS Compliance testing

**12. What is the procedure to perform API testing?**

1. Choose the suite to add the API test case
2. Choose the test development mode
3. Demand the development of test cases for the required API methods
4. Configure the control parameters of the application and then test conditions
5. Configure method validation
6. Execute the API test
7. Check test reports and filter API test cases
8. Arrange all API test cases

**13. What must be checked when performing API testing?**

During the API testing process, a request is raised to the API with the known data. This way you can analyze the validation response. While testing an API, you should consider:

* Accuracy of data
* Schema validation
* HTTP status codes
* Data type, validations, order, and completeness
* Authorization checks
* Implementation of response timeout
* Error codes in case API returns, and
* Non-functional testing like performance and security testing

**14. What is the best approach method to perform API testing?**

The following factors should be considered when performing API testing:

* Defining the correct input parameters
* Verifying the calls of the mixture of two or more added value parameters
* Defining the basic functionality and scope of the API program
* Writing appropriate API test cases and making use of testing techniques such as equivalence class, boundary value, etc. to check the operability
* Testing case execution
* Comparing the test result with the expected result
* Verifying the API behavior under conditions such as connection to files and so on.

**15. What tools could be used for API testing?**

There is myriad different [API testing tools](https://katalon.com/resources-center/blog/top-5-free-api-testing-tools/) available. A few common tools are [Katalon Platform](https://katalon.com/katalon-platform), Postman, SoapUi Pro, Apigee, etc.  While doing Unit and API testing, both target source code. If an API method uses code based in .NET then another supporting tool must have .NET.**16. What are the differences between API Testing and Unit Testing?**

|  |  |
| --- | --- |
| **API Testing** | **Unit Testing** |
| Conducted by QA Team | Conducted by the development team |
| Mostly black-box testing | White box testing |
| Aimed to assess the full functionality of the system for it will be employed by the end-user (external developers who will use your API) | Used to verify whether each unit in isolation performs as expected or not |
| Often run after the build is ready and authors do not have access to the source code | Each of the code modules must be ensured to pass the unit test before being built by developers |

**Read More:**[What is Unit Testing? A Comprehensive Guide](https://katalon.com/resources-center/blog/unit-testing)

**17. What are the differences between API Testing and UI Testing?**

* API enables the communication between two separate software systems. A software system implementing an API contains functions or subroutines that can be executed by another software system.
* On the other hand, UI ( User Interface) testing refers to testing graphical interfaces such as how users interact with the applications, and testing application elements like fonts, images, layouts, etc. UI testing basically focuses on the look and feel of an application.

**18. What are the major challenges faced in API testing?**

If you can overcome the [challenges in API Testing](https://docs.katalon.com/docs/create-tests/introduction-to-test-creation/introduction-to-api-testing-in-katalon-studio#challenges-in-api-testing), you can be confident in the API testing interview too. They are:

* Parameter Selection
* Parameter Combination
* Call sequencing
* Output verification and validation
* Another important challenge is providing input values, which is very difficult as GUI is not available in this case.

**19. What are the testing methods that come under API testing?**

One of the most common Web API testing interview questions is about the testing methods. They are:

* Unit Testing and Functional testing
* Load testing to test the performance under load
* Discovery testing to list, create and delete the number of calls documented in the API
* Usability and Reliability testing to get consistent results
* Security and Penetration testing to validate all types of authentication
* Automation testing to create and run scripts that require regular API calls
* End to end Integration and Web UI testing
* API documentation testing to determine its efficiency and effectiveness

**20. Why is API testing considered as the most suitable form for Automation testing?**

API testing is now preferred over GUI testing and is considered as most suitable because:

* It verifies all the functional paths of the system under test very effectively.
* It provides the most stable interface.
* It is easier to maintain and provides fast feedback.

**21. What are common API errors that are often found?**

Not only API fundamental questions, the interviewer also determine your knowledge and experience by asking about the API errors in a Web API testing interview. So the most common ones are:

* Missing module errors
* Documentation errors
* Parameter validation errors
* And some standard error expectations as if the result is not so predicted then the occurrence of errors can be seen and the same warnings are specified in the form of a message. There can be one or more warnings within an individual module.

**22. What kinds of bugs that API testing would often find?**

* Missing or duplicate functionality
* Fails to handle error conditions gracefully
* Stress
* Reliability
* Security
* Unused flags
* Not implemented errors
* Inconsistent error handling
* Performance
* Multi-threading issues
* Improper errors

**Documenting the API (Common Web API Testing interview questions)**

**23. What is API documentation?**

The API documentation is a complete, accurate technical writing giving instructions on how to effectively use and integrate with an API. It is a compact reference manual that has all the information needed to work with the API and helps you answer all the API testing questions with details on functions, classes, return types, arguments, and also examples and tutorials.

**24. What are API documentation templates that are commonly used?**

There are several available API documentation templates that help to make the entire process simple and straightforward, which could be  answered in your API testing interview, such as:

* Swagger
* Miredot
* Slate
* FlatDoc
* API blueprint
* RestDoc
* Web service API specification

**25. When writing API document, what must be considered?**

* Source of the content
* Document plan or sketch
* Delivery layout
* Information needed for every function in the document
* Automatic document creation programs

**26. How often are the APIs changed and, more importantly, deprecated?**

APIs, especially modern RESTful APIs, is a nice creation that can certainly simplify and accelerate integration efforts, which makes it more likely you will benefit from them. But APIs can and do change for various reasons, sometimes abruptly, and hence REST APIs do not differ from traditional integration methods in this respect. If an API call is obsolete and disappears, your procedure will interrupt and it is important to understand how often the APIs you depend on change or are deprecated.

**REST (Common Web API Testing interview questions)**

**27. What is REST?**

REST (Representational State Transfer) is an architectural style for developing web services that exploit the ubiquity of HTTP protocol and uses the HTTP method to define actions. It revolves around resources where every component is a resource that can be accessed through a shared interface using standard HTTP methods.            
  
In REST architecture, a REST Server provides access to resources and REST client accesses and makes these resources available. Here, each resource is identified by URIs or global IDs, and REST uses multiple ways to represent a resource, such as text, JSON, and XML. XML and JSON are nowadays the most popular representations of resources.

**28. What is a RESTFul Web Service**

Mostly, there are two kinds of Web Services that should be remembered in your next API testing interview:

1. SOAP (Simple Object Access Protocol) – an XML-based method to expose web services.
2. Web services developed in the REST style are referred to as RESTful web services. These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, and provides resource representation like JSON and a set of HTTP methods.

**29. What is a “Resource” in REST?**

REST architecture treats any content as a resource, which can be either text files, HTML pages, images, videos, or dynamic business information.            
REST Server gives access to resources and modifies them, where each resource is identified by URIs/ global IDs.

**30. What is the most popular way to represent a resource in REST?**

REST uses different representations to define a resource like text, JSON, and XML.            
XML and JSON are the most popular representations of resources.

**31. Which protocol is used by RESTful Web services?**

RESTful web services use the HTTP protocol as a medium of communication between the client and the server.

**32. What are some key characteristics of REST?**

Key characteristics of REST are likely asked in a Web API Testing interview. So please get the answer ready in your mind with these 2 ones:

* REST is stateless, therefore the SERVER has no status (or session data)            
  With a well-applied REST API, the server could be restarted between two calls, since all data is transferred to the server
* Web service uses POST method primarily to perform operations, while REST uses GET for accessing resources.

**33. What is messaging in RESTful Web services?**

RESTful web services use the HTTP protocol as a communication tool between the client and the server. The technique that when the client sends a message in the form of an HTTP Request, the server sends back the HTTP reply is called Messaging. These messages comprise message data and metadata, that is, information on the message itself.

**34. What are the core components of an HTTP request?**

An HTTP request contains five key elements:

1. An action showing HTTP methods like GET, PUT, POST, and DELETE.
2. Uniform Resource Identifier (URI), which is the identifier for the resource on the server.
3. HTTP Version, which indicates HTTP version, for example-HTTP v1.1.
4. Request Header, which carries metadata (as key-value pairs) for the HTTP Request message. Metadata could be a client (or browser) type, format supported by the client, format of a message body format, cache settings, and so on.
5. Request Body, which indicates the message content or resource representation.

**35. What are the most commonly used HTTP methods supported by REST?**

* GET is only used to request data from a specified resource. Get requests can be cached and bookmarked. It remains in the browser history and haS length restrictions. GET requests should never be used when dealing with sensitive data.
* POST is used to send data to a server to create/update a resource. POST requests are never cached and bookmarked and do not remain in the browser history.
* PUT replaces all current representations of the target resource with the request payload.
* DELETE removes the specified resource.
* OPTIONS is used to describe the communication options for the target resource.
* HEAD asks for a response identical to that of a GET request, but without the response body.

**36. Can GET request be used instead of PUT to create a resource?**

The PUT or POST method should be used to create a resource. GET is only used to request data from a specified resource.

**37. Is there any difference between PUT and POST operations?**

PUT and POST operations are quite similar, except for the terms of the result generated by them.

PUT operation is idempotent, so you can cache the response while the responses to POST operation are not cacheable, and if you retry the request N times, you will end up having N resources with N different URIs created on the server.

In a Web API Testing interview, you should give a specific example for PUT and POST operations to make it crystal clear to the interviewer. Below is an example:

*Scenario:* *Let’s say we are designing a network application. Let’s list down a few URIs and their purpose to get to know when to use POST and when to use PUT operations.*

**Final Thoughts**

The aforementioned API Testing interview questions are quite common in interviews for software QA engineers and testers positions. To best prepare for your upcoming interview, make sure to read thoroughly those questions, and try to understand the reasoning and knowledge behind them instead of simply memorizing.

For more information, you can have a look at the [What is API Testing](https://katalon.com/api-testing) article we wrote. If needed, you can download our Katalon Platform for free and practice API Testing on the platform to get some hands-on experience. We also have a [free course on API Testing](https://academy.katalon.com/courses/katalon-api-testing/) with Katalon Studio if you need a more visual and auditory tutorial.

### 1. Explain what REST Assured is.

REST Assured is a Java library that offers programmers a domain-specific language (DSL) to write maintainable, robust tests for RESTful APIs. It is widely used to test web applications based on [JSON](https://www.simplilearn.com/tutorials/python-tutorial/python-json-encoding-and-decoding) and [XML](https://www.simplilearn.com/tutorials/programming-tutorial/what-is-xml). Additionally, it supports all methods, including GET, DELETE, PUT, POST, and PATCH.

### 2. And what is REST?

REST is an acronym for "representational state transfer." It's a design pattern or architectural style for APIs. A RESTful web application reveals information about itself as resource information.

### 3. And what is JSON?

It is a text-based standard used to describe structured data based on [JavaScript](https://www.simplilearn.com/tutorials/javascript-tutorial/introduction-to-javascript) object syntax. JSON is frequently used in web applications to send data to clients and servers.

### 4. Which protocol does RESTful Web Services use?

RESTful web services use the HTTP protocol to communicate between the client and the server.

### 5. Define “client-server architecture.”

The client-server architectural model defines how a server allocates resources and services to one or more clients. Server examples include web servers, mail servers, and file servers. So, the server carries out the request when the client requests a resource.

### 6. Define a resource in REST.

The REST architecture treats any content as a resource. This content includes [HTML pages](https://www.simplilearn.com/tutorials/html-tutorial/what-is-html), text files, images, videos, or dynamic business information. A REST Server gives users access to these resources and modifies them, and URIs or global IDs identify each resource.

### 7. Explain REST Assured method chaining.

In the context of [object-oriented programming languages](https://www.simplilearn.com/tutorials/java-tutorial/oops-interview-questions), method chaining is an often-used syntax for invoking any number of method calls. Each method returns an object, so multiple calls can be chained together in a single line. This characteristic means that variables aren’t needed to hold interim results.

### 8. Why would a programmer use REST Assured to automate RESTful services instead of Postman?

Because REST Assured can customize reports, Postman can't do this. Additionally, since REST Assured is a Java client, you can reuse code, which [Postman](https://www.simplilearn.com/postman-interview-questions-answers-article) doesn’t allow. Finally, REST Assured has no restrictions on data file submission for collections, whereas Postman is limited to one data file.

### 9. What is the request specification?

Request specification in REST Assured is used to group common request specs and change them into a single object. This interface has the means to define the base URL, headers, base path, and other parameters. You must use the given() function of the REST Assured class to obtain a reference for the Request specification.

### 10. How do you initiate request specification in REST Assured?

Here is the syntax:

RequestSpecification reqSpec = RestAssured.given();

reqSpec.baseUri("http://localhost:8080")

reqSpec.basePath("/employees");

All codes courtesy of[Techgeek.](https://www.techgeeknext.com/rest-assured-interview-questions)

### 11. How do you perform chaining in REST Assured?

In the context of object-oriented programming languages, method chaining is used to invoke multiple method calls. Each method returns an object, which allows multiple calls to be chained into a single line that doesn’t require variables to hold interim results.

In REST Assured, it looks like this:

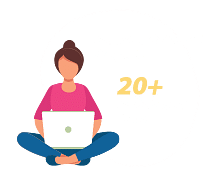
    given()

           .baseUri(baseUri)

           .queryParam(parameterName, parameterValues)

           .accept(contentType).

           .when()

           .then();

### 12. Write a code that tests REST API using REST Assured.

Here’s the solution:

import org.testng.annotations.Test;

import io.restassured.RestAssured;

import io.restassured.http.Method;

import io.restassured.response.Response;

import io.restassured.specification.RequestSpecification;

public class EmployeesTest {

    @Test

    public void GetAllEmoloyees()

    {

           // base URL to call

           RestAssured.baseURI = "http://localhost:8080/employees/get";

           //Provide HTTP method type - GET, and URL to get all employees

           //This will give respose

           Response employeesResponse = RestAssured.given().request(Method.GET, "/all");

           // Print the response in string format

    System.out.println(employeesResponse.getBody().asString());

    }

}

### 13. When using REST Assured, what’s the best method of keeping sensitive data out of the log?

Use a blacklist to prevent sensitive data from appearing in the log. Here’s how:

Set<String> headers = new HashSet<String>();

headers.add("X-REGION");

headers.add("content-type");

given().

baseUri("http://localhost:8080").

header("X-REGION", "NAM").

// blacklist headers

config(

config.logConfig(LogConfig.logConfig().blacklistHeaders(headers)))

// blacklist multiple headers

//config(config().logConfig(LogConfig.logConfig().blacklistHeader("Accept","set-cookie"))).

log().all().

when().

get("/employees").

then().

assertThat().

statusCode(200);

### 14. What is a jsonPath in the context of REST Assured?

A JsonPath (io.restassured.path.json.JsonPath) is an easy way to get values from an Object document without resorting to XPath. It conforms to the Groovy GPath syntax when it retrieves an object from a document. Consider it a JSON-specific version of XPath. Here’s an example:

{ "company": {

   "employee": [

{ "id": 1,

   "name": "TechGeekNextUser1",

   "role": "Admin"

},

{ "id": 2,

   "name": "TechGeekNextUser2",

   "role": "User"

},

{ "id": 3,

   "name": "TechGeekNextUser3",

   "role": "User"

}

  ]

  }

 }

Response employeesResponse = RestAssured.given().request(Method.GET, "/all");

JsonPath jsonPathObj = employeesResponse.jsonPath();

//get a list of all employees id:

List<String> employeeIds = jsonPathObj.get("company.employee.id");

//get the first employee name:

String empName = jsonPathObj.get("company.employee[0].name");

### 15. How do you log a request and response in case REST Assured fails a validation?

If the test validation fails, log().ifValidationFails() will log everything in a request and response.

/\*\*

  \* Log the request and response details if validation fails.

  \*/

@Test

public void testIfValidationFails() {

     given().

     baseUri("http://localhost:8080").

     header("X-REGION", "NAM").

     log().ifValidationFails().

     when().

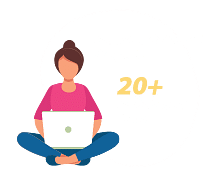
     get("/employees").

     then().

     log().ifValidationFails().

     assertThat().

     statusCode(200);

}

### 16. How do you use a REST Assured jsonPath to find all employee IDs between 15 and 300?

Like this:

Response employeesResponse = RestAssured.given().request(Method.GET, "/all");

JsonPath jsonPathObj = employeesResponse.jsonPath();

//get all employees id between 15 and 300

List<Map> employees = jsonPathObj.get("company.employee

                .findAll { employee -> employee.id >= 15 && employee.id <= 300 }");

### 17. What is static import and why would you use it in REST Assured?

Static import is a Java programming language function that lets members (e.g., fields and methods) scoped as public static within their container class to be employed in Java code without mentioning the field's defined class.

package com.techgeeknext.controller;

import org.testng.annotations.Test;

/\*\*

 \* this is static import to avoid writing

 \* into front of every method call of RestAssured

 \*/

import static io.restassured.RestAssured.\*;

public class EmpControllerTest {

    @Test

    public void testGetEmployees() {

           // with static import

           given();

           // without static import

           /\*\*

           \*  import io.restassured.RestAssured;

           \*  RestAssured.given();

           \*/

    }

}

### 18. What are serialization and deserialization in the context of Java?

Serialization is defined as the process of changing an object's state into a byte stream. On the other hand, deserialization is the process of recreating the Java object in memory using the byte stream. This approach keeps the object alive.

### 19. List the core components of an HTTP request.

An HTTP request consists of five elements:

* An action (DELETE, GET, POST). This element shows HTTP methods.
* A Uniform Resource Identifier (URI). This element identifies the resource on the server.
* The HTTP version.
* A request header. This element carries the metadata for the message. The metadata could be a format supported by the client, message body format, browser or client type, cache settings, etc.
* Request body. This element indicates the resource representation or message content.

### 20. Can a GET request be used to create a resource instead of PUT?

Use the GET option because it has view-only rights. The POST or PUT methods should never be used to create a resource.