**1. What is an API?**

***API (Application Programming Interface)***is a software intermediary that enables two applications to communicate with each other. HTTP-based API is often called a Web API since they are used to access Web Applications which are deployed to Servers accessible over the Internet or network. Applications that are accessed via HTTP APIs are often called Web Services.

Mobile Applications often use Web Services and REST APIs to communicate with servers to implement their functionality. The Mobile Application processes the message returned from the Web Service and displays it to the User in the application GUI. So again, the user is unaware that HTTP requests are being made or of the format of the requests and responses.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2. Define  API TESTING with its types ,examples and some common advantages?**

**API Testing**- It is a kind of software testing to determines and check if the developed APIs meet expectations regarding the functionality, reliability, performance, and security of the application or not

**Common protocols that are testing in API testing are** -HTTP, JMS, REST, SOAP, UDDI

**Types-**Validation Testing, Functional Testing, UI testing, Load testing, Runtime/ Error Detection, Security testing, Penetration testing, Fuzz testin, Interoperability testing

**Example of API Testing:**

**Twitter**: The API for Twitter is usually in two categories, one for accessing data and the other for interacting with the Twitter search.

**YouTube**: This API used for YouTube includes various functionalities including videos, live streaming, player, etc.

**Google Maps AP**I: These are designed mainly for mobile and desktop use with the help of a flash interface and JavaScript.

**Amazon Advertising API**: Amazon is known for their products and thus their advertising API accesses their product to discover their functionality and thus advertise accordingly

**ADVANTAGES of API Testing:**

**Compatibility and easy integration with GUI**: Simple integration would allow new user accounts to be created within the application before a GUI test started.

**Language-Independent**: In API testing, data is exchanged using XML or JSON. These transfer modes are completely language-independent that allows users to select any coding language while adopting automation testing services for the project.

**Time Effective**: In comparison to functional GUI testing API testing usually is less time-consuming. 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.

**Test for Core Functionality**: API testing provides the ability to 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 helps in detecting the minor issues which can become bigger during the GUI testing

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

**Main differences are given below:**  
a) All web services are APIs but not all APIs are web services.

b) A web service uses only three styles of use: SOAP, REST, and XML-RPC for communication whereas API may be exposed to in multiple ways.

c) A web service always needs a network to operate while APIs don’t need a network for operation.

d) Web services might not contain all the specifications and cannot perform all the tasks that APIs would perform.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4. 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 interface such as how users interact with the applications, testing application elements like fonts, images, layouts, etc. UI testing basically focuses on the look and feel of an application.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Basically, there are five most important principles of an API test design:

**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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

a) Create the suite to add the API test case

b) Create the test development mode

c) Demand the development of test cases for the required API methods

d) Configure the control parameters of the application and then test conditions

e) Configure method validation

f) Arrange all API test cases

g) Execute the API test

h) Check test reports

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

a) Accuracy of data

b) Non-functional testing like performance and security testing

c) Implementation of response timeout

d) Schema validation

f) HTTP status codes

g) Data type, validations, order, and completeness

h) Authorization checks

I) Error codes in case API returns

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**8. What are tools could be used for API testing?**

A few common tools are Katalon Studio, Postman, SoapUi Pro, Apigee, automated API testing using REST Assured and CURL etc.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**9. What are the types of Bugs API testing can find?**

a) Missing or duplicate functionality

b) Stress

c) Reliability

d) Security

e) Performance

f) Unused flags

g) Not implemented errors

h) Inconsistent error handling

I) Improper errors

j) Fails to handle error conditions gracefully

k) Multi-threading issues

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**10. Difference API and Unit Testing?**

API is owned by QA team where as  Unit testing is owned by the development team

API is mostly black box testing where as Unit testing is white box testing

Full functionality of the system is considered in API testing as it will be used by the end-user (external developers who will use your API )where as Unit testing is done to verify whether each unit in isolation performs as expected or not

API test are often run after the build is ready and authors do not have access to the source code where as For each of their module, the developers are expected to build unit tests for each of their code modules and have to ensure that each module pass unit test before the code is included in a build.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**11. What are the main challenges of API testing?**

1) Parameter Selection

2) Parameter Combination

3) Call sequencing

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

There are several available API documentation templates help to make the entire process simple and straightforward.

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**13. Explain what is an HTTP request?**

Hypertext Transfer Protocol is a way of sending messages to software on another computer over the internet or over a network.

**An HTTP request is sent to a specific URL and consists of:**

A) A VERB specifying the type of request e.g. GET, POST, PUT, DELETE

B) A set of HTTP Headers. The headers specify information such as the type of browser, type of content in the message, and what type of response is accepted in return.

C) A body, or payload in the request, representing the information sent to, or from, the Web Application. Not all HTTP messages can have payloads: POST and PUT can have payloads, GET and DELETE do not.

**14. What is a URL?**

URL is a Uniform Resource Locator and is the address we use to access websites and web applications. When working with APIs you will often see this referred to as a URI (Uniform Resource Identifier). We can think of a URI as the generic name for a URL. When we have to call an HTTP API we need the URL for the endpoint we want to call Example-

<https://www.thetravel.com/10-best-indian-foods/>

This is the locator that says “I want to call the “10 best indian foods” resource located at “thetravel.com” using the http protocol”.

**The above URL can be broken down into the form:**

**scheme://host/resource**

• **scheme**-HTTP

• **host**-thetravel.com

• **resource**-10-best-indian-foods

**A larger form for a URL is:**

scheme://host/resource?query#fragment

The query is a way of passing parameters in the URL to the endpoint e.g. Google uses query parameters to define the search term and the page:

<https://www.google.co.uk/?q=test&start=10#q=test>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**15. Describe what are HTTP verbs?**

A web browser will usually make **GET requests** and **POST requests.**

**GET**requests to ask to read information from the server e.g. clicking on a link.

**POST**requests supply information to the server e.g. submitting a form.

**GET**requests do not have a body, and just consist of the Verb, URL, and the Headers.

**POST**requests can have a payload body.

***When working with a Web Application or HTTP API the typical HTTP Verbs used are:***

**GET**, to read information.

**POST**, to create information.

**PUT**, to amend or create information.

**DELETE**, to delete information.

### 16.  What is an HTTP response?

When you issue an HTTP Request to the server you receive an HTTP Response. The response from the server tells you if your request was successful, or if there was a problem.

**Response contains:**

**status code of 200**, which means that the request was successful.

**content-Type** header of application/JSON means that the body is a JSON response

**body** that contains the actual payload response from the server.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 17. What is an HTTP status code?

Web Services and HTTP APIs use HTTP Status Codes to tell us what happened when the server processed the request.

**The simple grouping for HTTP Status Codes is:**

• **1xx**– Informational

•**2xx** – Success e.g. 200 Success

• **3xx** – Redirection e.g. 302 Temporary Redirect

• **4xx** – Client Error e.g. 400 Bad Request, 404 Not Found

• **5xx** – Server Error e.g. 500 Internal Server Error

The type of status code you receive depends on the application you are interacting with. Usually, a 4xx error means that you have done something wrong and a 5xx error means that something has gone wrong with the application server you are interacting with.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 18.  What is RESTFul Web Services?

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, provides resource representation like JSON and a set of HTTP methods.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 19. 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.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 20. Can you tell us which Java API helps in developing a RESTFul web service?

In JAVA there are many frameworks and libraries available that a developer can use to create RESTful web services. For example, the JAX-RS library is a standard way to develop a REST web service.

Also, Jersey is another most popular implementation of JAX-RS which offers more than what the specs recommend. There are others like RESTEasy, RESTlet, and Apache CFX and Play framework to develop RESTful web services.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 21. what is postman and Explain 3-tier architecture?

Postman is one of the most widely used **API testing tool**now.

All web application comes under ***3 tier architecture***. It consists of three layers.

**Client / Presentation**-  It is the presentation layer from which the user sends the request. Like Browsers.

**Business Logic**- As soon as the user sends the request from the client layer corresponding business logic gets executed. Business logic will pass the request to the database layer and vice versa.

**Database**- As per the request from the business logic database provide the set of data.

Business Logic acts as a mediator between the client and database, it contains business logic written in programming language collectively known as API. Business Logic takes input from the client and sends back the output to the client after interacting with the database.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 22. How we can add validation points in postman?

Once we will send a request we will get a response body and in the Test tab we can add the validation points like:

1- **Validating the Status Code:**

**tests[“Validating Status Code”] = responseCode.code == 200;**

2- **Next validating data in the response body**:

**tests[“Validating response body”] = responseBody.has(“data”);**

3- **Validating data in speficic field in the response body**:

***var response = JSON.parse(responseBody);***

***tests[“page no”] = response.page == 2;***

We have added 3 validation, even one validation failed the entire test will fail.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 23. What is the difference between the path parameter and query parameter?

***query parameter*** will filter the data.

**Example:** <https://reqres.in/api/users?page=2>

***path parameter*** will simply get the data based on the path we specify.

Example: <https://reqres.in/api/users/2>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 24. Define in detail about REST?

**REST**stands for ***REpresentational State Transfer***. REST is a web standards-based architecture and uses HTTP Protocol for data communication. It revolves around resources where every component is a resource and a resource is accessed by a common interface using HTTP standard methods. REST was first introduced by Roy Fielding in 2000.

**In REST architecture**, a REST Server simply provides access to resources, and a REST client accesses and presents the resources.

Here each resource is identified by URIs/ global IDs. REST uses various representations to represent a resource like text, JSON, and XML.

Nowadays JSON is the most popular format being used in web services.

**Following well-known HTTP methods are commonly used in REST-based architecture:**

**GET**− Provides read-only access to a resource.

**PUT**− Used to create a new resource.

**DELETE**− Used to remove a resource.

**POST**− Used to update an existing resource or create a new resource.

**OPTIONS**− Used to get the supported operations on a resource.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 25)Explain the RESTFul Web Service?

Mostly, there are two kinds of Web Services which are quite popular.

**1.** ***SOAP (Simple Object Access Protocol)*** which is an XML-based way to expose web services.

**2.** ***Web services developed using REST style are known 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, provides resource representation such as JSON and set of HTTP Methods.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### 26. 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.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### ****27. What are SOAP Web services and how does it work?****

The***SOAP (Simple Object Access Protocol)***is defined as an XML-based protocol. It is known for designing and developing web services as well as enabling communication between applications developed on different platforms using various programming languages over the Internet. It is both platform and language independent

**SOAP** is used to provide a user interface that can be accessed by the client object, and the request that it sends goes to the server, which can be accessed using the server object. The user interface creates some files or methods consisting of server object and the name of the interface to the server object.

 It also contains other information such as the name of the interface and methods. It uses HTTP to send the XML to the server using the POST method, which analyzes the method and sends the result to the client.

The server creates more XML consisting of responses to the request of user interface using HTTP. The client can use any approach to send the XML, like the SMTP server or POP3 protocol to pass the messages or reply to queries.

Use the SOAP API to create, retrieve, update or delete records, like accounts, leads, and user-defined objects. With more than 20 different calls, you can also use the SOAP API to manage passwords, perform searches, etc. by using the SOAP API in any language that supports web services.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### ****28. How users utilize the facilities provided by SOAP?****

**PutAddress():** It is used to enter an address in the webpage and has an address instance on the SOAP call.

**PutListing():** It is used to allow the insertion of a complete XML document into the web page. It receives the XML file as an argument and transports the XML file to XML parser liaison, which reads it and inserts it into the SOAP call as a parameter.

**GetAddress():** It is used to get a query name and gets the result that best matches a query. The name is sent to the SOAP call in the form of text character string.

**GetAllListing():** It is used to return the full list in an XML format.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### ****29. What is the major obstacle users faced when using SOAP?****

When using SOAP, users often see the firewall security mechanism as the biggest obstacle. This block all the ports leaving few like HTTP port 80 and the HTTP port used by SOAP that bypasses the firewall.

The technical complaint against SOAP is that it mixes the specification for message transport with the specification for message structure.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### ****30. What are the syntax rules for a SOAP message?****

A) Must use encoded XML.

B) Envelope namespace must be used.

C) Encoding namespace must be used.

D) Must not consist of a DTD reference.

E) Must not have XML processing instruction.