“API Interview Success: Key Questions and Expert Insights”

1. **How do you perform API testing?**

Ans: We perform API Testing using Tools like Postman. We will receive the API contract from the dev team and based on Test cases we will Pass the request payload and will validate the Response received. For eg. Status code and response body.

---

2. **What is API Testing?**

Ans: API testing is a type of software testing that focuses on verifying and validating the functionality, performance, and reliability of Application Programming Interfaces (APIs). An API is a set of rules and protocols that allows different software applications to communicate and interact with each other. API testing helps identify issues such as incorrect data formats, incorrect response codes, security vulnerabilities, performance bottlenecks, and functional errors.

---

3. What are the different methods in API Testing?

Ans:

1. GET: Get is used to retrieve data stored on the server. GET method does not have a request body.

2. POST: Post method is used when you want to create a new record on the server. Post method used a request body.

3. PUT: PUT is used to update the existing resource server, It modifies the existing resource. If the resource does not exist then it creates the resource. In PUT we need to pass the whole body in the request

4. DELETE: Delete is used to delete any record from the server.

5. PATCH: PATCh is also used to partially modify the existing resource. For eg, If you want to update only the address of the resource Then we use PATCH. In the Patch request body, we will pass only those parameters which you want to modify.

---

4. When do we use PUT and POST?

Ans: PUT is used to replace the entire resource on the server. For eg. In the below Eg. If you want to update all the details except Email then you can use PUT and in the request body, we will pass everything except Email.

{

id: 1,

email: diksha@gmail.com ,

city: Dubai,

phone: +971873947490

}

Now If you want to update only the email address then we can use PATCH and will pass only the email address parameter in the request body.

{

email: diksha123@gmail.com

}

---

5. What are the different response codes in API Testing?

Ans: List of commonly used Status codes

200—OK, When the request is successful.

201: Created, The request has been fulfilled, resulting in the creation of a new resource

202: Accepted, The request has been accepted for processing, but the processing has not been completed yet.

400—Bad Request, These are Client-side errors. And the server cannot process the request because the request is not formatted properly.

401=Unauthorized, This status code is received when authentication is required and the user has failed or has not yet been provided yet.

403- Forbidden, This error is received when the user is authorized but is not authenticated for the requested resource.

404- Not Found, This code is received when the requested resource is not found

405- Method Not Allowed, This code is received when the request method is not supported for the requested resource;

408 Request Timeout. The server timed out waiting for the request.

500—Internal Server error, These are Server side errors. This is received when an unexpected condition has encountered

502- Bad Gateway. This is received when the server which was acting as a gateway has received an invalid response from the upstream server.

503- Service Unavailable. This is received when the server is not available to process the request.

504- Gateway Timeout. When the server did not receive a timely response from the upstream server.

---

6. What are the challenges we see while doing API Testing?

Ans: Limited documentation: Inadequate or incomplete documentation can make it challenging to understand the API's behavior, available endpoints, supported parameters, and expected responses. A lack of proper documentation can slow down the testing process and lead to ambiguities.

2. Versioning and Compatibility: APIs undergo updates, version changes, or deprecations over time. Testing across different API versions and ensuring backward compatibility can be challenging, especially when multiple versions coexist.

3. Testing Environment Setup: Creating and managing the required testing environments, including stubs, and mocks can be time-consuming. Configuring the test environment to

mimic real-world scenarios or simulate specific responses or error conditions can be challenging.

4. Automation and Maintenance: Automating API tests for regression testing and continuous integration can be challenging due to the complexity of APIs, dynamic data, and dependencies. Maintaining and updating test scripts when API changes occur can be challenging.

5. Dependency on External Systems: APIs may depend on external systems, such as databases, third-party services, or other APIs. Testing APIs with dependencies can be challenging due to the need for proper integration, coordination, and availability of these external systems during testing.

6. Test Data Management: APIs often rely on dynamic data, such as unique identifiers, or changing data sources. Managing such dynamic data and ensuring consistent test data setup and cleanup can be complex.

---

7. Which information is exposed in the web developer tool?

Ans: Request Header, Request payload, Response body and cookies, Status codes, etc are exposed in the developer tool.

---

8. What is Postman collection?

Ans: In Postman, the collection is a structured format used to organize and group API requests, along with their associated data, tests, and documentation. a collection can be thought of as a folder or container that holds multiple API requests. It provides a centralized place to store and organize all the endpoints related to a particular API or project.

---

9. What is Postman Collection Runner?

Ans: Runner is when a group of API requests is run in a collection for multiple Iterations with different sets of data.

---

10. What type of bugs API finds?

Ans: **Functional Bugs:** API testing verifies if the API behaves as expected and adheres to the defined functionality. It helps identify bugs related to incorrect data validation, improper handling of input parameters, incorrect response formats, missing or inconsistent data, and unexpected behavior.

Integration Bugs: API testing ensures the smooth integration of different software components or systems. It can identify bugs related to incorrect API versioning, incompatible data formats, improper data exchange etc.

Performance Bugs: It helps identify performance-related bugs such as slow response times, high resource utilization, memory leaks, poor caching mechanisms, and bottlenecks under heavy loads

Security Vulnerabilities: API testing uncovers security vulnerabilities that could expose sensitive data or allow unauthorized access.

Error Handling Bugs: API testing helps uncover bugs related to improper error handling, inconsistent error codes, misleading error messages, and vulnerabilities that could potentially expose sensitive information.

---

11. What is the difference between HTTP and HTTPS?

Ans: The main difference between HTTP and HTTPS lies in the level of security they provide when communicating over the internet:

1. HTTP (Hypertext Transfer Protocol): HTTP is the protocol used for transmitting data between a web server and a web browser. It operates over the TCP/IP protocol and sends data as plain text. HTTP is not secure by default, meaning the data transferred between the client and the server is not encrypted. This lack of encryption makes it susceptible to eavesdropping, data tampering, and unauthorized access. It is commonly used for non-sensitive information transfer.
2. HTTPS (Hypertext Transfer Protocol Secure): HTTPS is the secure version of HTTP. It uses a combination of HTTP and SSL/TLS (Secure Sockets Layer/Transport Layer Security) protocols to provide encrypted communication over the Internet. HTTPS ensures the confidentiality and integrity of data by encrypting it before transmission and decrypting it upon receipt. This encryption prevents unauthorized interception and protects against data tampering. HTTPS is commonly used for secure online transactions, sensitive data transfer (e.g., login credentials, financial information), and protecting user privacy.

---

12. What is Authentication and Authorization?

Ans: Authentication: Authentication is the process of verifying the identity of a user, system, or entity. It ensures that the claimed identity is valid and trustworthy before granting access to protected resources. Authentication methods involve the presentation of credentials, such as usernames, passwords, digital certificates, biometrics, or other factors that prove the identity of the entity requesting access.

Authorization, on the other hand, is the process of granting or denying access to specific resources or functionalities based on the authenticated identity and the assigned access rights. Once a user or system is authenticated, authorization determines the level of access or permissions they have within the system.

---

13. What are different Authentication methods available?

Ans: Commonly used Authentication methods are:

Basic Authentication: Basic Authentication is an HTTP-based authentication scheme. The client includes a Base64-encoded username and password in the Authorization header of the request. The server verifies the credentials against a user database or other authentication mechanisms. However, Basic Authentication has the limitation of sending credentials in plain text, making it vulnerable to interception.

Bearer Token: Bearer token authentication involves the use of a token issued by an authentication server. The client includes the token in the Authorization header of the API request, typically as a bearer token prefixed with the word “Bearer.” The server validates the token to authenticate the client. Bearer tokens are commonly used in OAuth 2.0 and JWT (JSON Web Token) authentication

OAuth 2.0: OAuth 2.0 is an industry-standard authorization framework that allows third-party applications to access resources on behalf of a user. It involves the exchange of tokens between the client, the authorization server, and the resource server. OAuth 2.0 provides a secure and standardized way of delegating access and is commonly used for API authentication and authorization in scenarios like social login or granting API access to external applications.

---

14. What do you mean by an idempotent API request?

Ans: An idempotent API request is a request that can be repeated multiple times without producing different results. In other words, making the same request multiple times should have the same effect as making it only once.

Idempotent requests are commonly associated with HTTP methods, especially the GET, PUT, and DELETE methods:

- GET: The GET method is inherently idempotent. Retrieving the same resource multiple times should yield the same response, and it should not modify any data on the server.
- PUT: The PUT method is idempotent when used to update or replace a resource. Sending the same PUT request multiple times with the same payload should result in the same resource state.
- DELETE: The DELETE method is also idempotent. Deleting a resource multiple times should have the same effect, and subsequent DELETE requests should not produce any errors.

In the next article, I will post questions on API Automation Testing.