Interview Questions

API Questions:

Q1) What is API?

Application Programming Interface (API) is a software interface that enables two applications to interact with each other.

**Explanation**: - API is defined as a interface that helps two different software communicate and exchange data with each other. It is a combination of endpoint + resource + input.

**Example**: The weather app on your phone “talks” to this system via APIs and shows you daily weather updates on your phone.

Q2) What are the main differences between API and Web Service?

|  |  |
| --- | --- |
| API | Web Services |
| APIs acts as an interface between two different applications for interoperability. | Web Services provide interaction between two machines over a network |
| It Supports HTTP/HTTPS Protocol | It Support only HTTP protocol |
| All API are not Webservices because webservice would not perform all operations that an API would do. | All Webservice is an API. |

Q3) What are the advantages of API?

* When using an API managed by computers, less human effort is required and workflows can be easily updated to become faster and more productive.
* APIs empower developers to be more productive by reusing code in complex but repetitive processes.
* It is language independent and time effective.

Q4) What is Web API?

API stands for **A**pplication **P**rogramming **I**nterface. A Web API is an application programming interface for the Web. A Browser API can extend the functionality of a web browser. A Server API can extend the functionality of a web server.

**Q**5) What is URI?

URI: **Uniform Resource Identifier** is used for identifying each resource of the REST architecture.

URI are of two types:

1)**URN**: Uniform Resource Name identifies the resource by means of a name that is both unique and persistent.

2)**URL**: Uniform Resource Locator has the information regarding fetching of a resource from its location. Examples: <http://abc.com/samplePage.html>.

URLs start with a protocol (http) and they have the information of the network hostname (sampleServer.com) and the path to the document(/samplePage.html). It can also have query parameters.

Q6) What is Endpoint?

An Endpoint is a Location where an API receives requests about a specific resource on its server. In APIs, an endpoint is typically a uniform resource locator (URL) that provides the location of a resource on the server.

Q7) What is Resource?

Resources are data sets that an API allows you to work with, and which are accessible via endpoints. A resource is focused on data set that is returned by a request.

**Rest API Questions:**

Q8) What is Rest API?

REST stands for Representational State Transfer. REST API treats all data as resource and each one is represented by unique uniform resource identifier.

Example: Airlines use APIs to expose the flight times and prices to allow travel and ticketing sites for businesses.

Q9) What are the advantages of RESTAPI?

* Easy to Learn
* Wide range of data transfer like JSON, XML.
* Statelessness i.e allowing simple client experience.
* Scalability- independent in nature.

Q10) What are the disadvantages of RESTAPI?

* Lack of built in security.
* Consistency in URI’s difficulty to maintain complex projects.

Q11) What do you mean by RESTful web services?

REST API is also known as RESTful web services that follow the REST architecture.

Q12) What are the types of parameters?

* Header Parameter: Parameters can be passed in the request header. Usually header parameters are used for passing sensitive data that shouldn’t be cached like a Bearer Token or an API key. Header parameters are most often used in API security and authentication.
* Query Parameter: Query Parameters are added to the end of a URL after a question mark. The question mark is followed by a parameter name, and the value.
* Example: `/users?role=admin`.
* Request Body Parameter: Request body parameters, which look similar to query parameters, are most often used in POST requests to send values in the request body.
* Template Parameter: It is used to parameterize the endpoint or resource.

Q13) What are Types of HTTP Methods?

* GET– Get method retrieve the data from database.
* POST– POST is used to send data to a server to create a record in database.
* PUT– Put Create a new entity or update an existing one.
* DELETE– Delete the existing record from the database.
* PATCH – Patch is a method of modifying resources where the client sends partial data that is to be updated without modifying the entire data.

Q14) What are the main characteristics of Rest API?

* Rest API uses HTTP protocol for communication.
* Rest API is Stateless, the server does not store information about past communication with clients.
* Rest API uses GET to retrieve data from server.

Q15) How do you keep REST APIs secure?

REST APIs can be kept secure with the help of safety measures such as Authentication and authorization, and sensitive information such as username, password, or authentication token should not be visible in URI’s.

Q16) Differentiate POST and PUT methods.

|  |  |
| --- | --- |
| POST Method | PUT Method |
| POST can create a data on the server | PUT is used to replace a data at a specific URI with another resource. |
| POST is not idempotent. So if we retry the request N times, we will end up having N resources with N different URIs created on the server. | PUT is idempotent that it will only result in one resource even after calling it multiple times. |
| POST responses are cacheable | PUT responses are not. |

### Q17) Explain ‘Addressing’ in RESTful web services.

The process of locating various types of resources with the help of a URL on the REST server is known as ‘addressing’ in RESTful web services. Usually, single or multiple resources are addressed by resources.

### Q18) What are Idempotent methods?

Idempotent methods ensure that the responses to a request if called once or ten times or more than that remain the same.  GET, PUT, DELETE, HEAD, OPTIONS, and TRACE are the idempotent HTTP methods.

## Q19) What is API authentication and how does it work?

API authentication is the process of verifying that an API user has permission to access the data and resources they are requesting. It works by denying or blocking access to servers for unauthorized users when an identity mismatch occurs. For example, if you were to provide the wrong username, password, or another form of credential, then the API would be unable to verify your right to access data within its server.

Q20) What is Authentication?

* Authentication is the process of identifying someone's identity by assuring that the person is the same as what he is claiming for.
* It is used by both server and client. The server uses authentication when someone wants to access the information, and the server needs to know who is accessing the information. The client uses it when he wants to know that it is the same server that it claims to be.

Q21) What is Authorization?

* Authorization is the process of giving someone the ability to access a resource.
* Authorizing requests include authenticating the identity of the client who sends the request and verifies whether the client is allowed to access and conduct the endpoint operations.

Q22) Difference Between Authentication and Authorization?

|  |  |
| --- | --- |
| Authentication | Authorization |
| Authentication is the process of identifying a user to provide access to a system. | Authorization is the process of giving permission to access the resources. |
| It is usually performed before the authorization. | It is usually done once the user is successfully authenticated. |
| Data is provided through the Token Ids. | Data is provided through the access tokens. |
| **Example:** Entering Login details is necessary for the employees to authenticate themselves to access the organizational emails or software. | **Example:** After employees successfully authenticate themselves, they can access and work on certain functions only as per their roles and profiles. |

### Q23) What is the difference between REST and SOAP?

REST (Representational State Transfer)

* It is an architectural design pattern used to develop web services.
* It is faster in speed and more cacheable.
* It inherits only the security measures concerning the protocol that have been implemented.

SOAP (Simple Object Access Protocol)

* It is a strict protocol used to build secure APIs.
* It is slower in speed and not cacheable.
* It is able to define its own security measures.

POSTMAN Questions

Q24) What is Postman?

* Postman is an API (application programming interface) development tool which helps to build, test and modify APIs.
* The Postman tool also facilitates us to send HTTP/s requests to a service and get their responses. We can ensure that the service is up and running by using this.

Q25) What are the benefits using postman?

* Easy to create, share, test, and document APIs.
* Store information for running tests in different environments.
* Store data for use in other tests.
* Integrates with build systems.
* Easy to move tests and environments to code repositories.
* Good UI.

### Q26) What are the various authorization methods provided by Postman?

Postman provides the following API request authorization options:

* API Key
* Bearer Token
* Basic auth
* Digest auth
* Hawk Authentication

### Q27) What is the use of the collection in Postman?

### In Postman, a collection is used to group similar requests. It systematically arranges the requests into folders.

### Q28) What are the different Request Method types in Postman?

The different Request Method types in Postman are as follows:

* Post
* Get
* Put
* Delete
* Patch
* Head
* Options

### Q29) What is an HTTP request? Give an example of an HTTP request.

* The main aim of the HTTP request is to access a resource on the server.
* To make the HTTP request, the client uses components of a URL (Uniform Resource Locator), which includes the information needed to access the resource.

Q30) What are the core components of the HTTP request and HTTP response?

* **HTTP methods:** It is a set of request methods used to perform needed action for a given resource (GET, PUT, POST, and DELETE).
* **Uniform Resource Identifier (URI):** It is a kind of address that describes the resource.
* **HTTP Version:** It specifies the version of the HTTP. For example HTTP v1.1
* **Request Headers:** It specifies the content type and content length of the request. For example: **Content-type:** application/ JSON, Content-Length: 511
* **Payload:** It is used to specify the Request Body that includes message content.

Following are the four core components of an HTTP Response:

* **Status/Response Code**: The server generates the status or response codes when the client makes a request. For example, 404 means Page Not Found, and 200 means Response is OK.
* **HTTP Version**: It describes the HTTP version. For example: HTTP v1.1, HTTP v1.3 etc.
* **Response Header**: The Response Header includes the information for the HTTP response message. For example, Content-type, Content-length, date, status, and server type.
* **Response Body**: The Response Body contains the client's data requested from the server.

Q31) What is the meaning of the term environment in Postman?

* An environment is a set of variables you can use in your Postman requests. You can use environments to group related sets of values together and manage access to shared Postman data if you are working as part of a team.

Q32) Can we have two global scope variables with the same name in Postman?

The global variables are global, so we cannot set duplicate names for them without any environment as it creates confusion for the software. On the other hand, local variables can have the same name but in different environments.

Q33) What are the main limitations of Postman?

* Postman cannot process 1000+ API requests.
* In the case of huge projects, it isn't easy to manage the collections and requests.
* Postman is not suitable for managing the workspace in the form of code. This is because there would be a lot of code duplication for dynamic API requests.

Q34) What are the various variable scopes provided by Postman?

Following is a list of several variable scopes provided by Postman:

* **Local Scope:** This type of variable can be accessed only in the environment in which it was created.
* **Global Scope:** This type of variable can be accessed globally in any environment or no environment.

### Q35) Where are the query parameters stored in a GET request in Postman?

In the GET request, the query parameters are stored in the URL in Postman.

### Q36) What is the difference between Query Params and Path Variables?

Query Params or Query Parameters are used for sorting or filtering the resources. On the other hand, Path Variables are used for identifying specific resources.

Q37) What is a bearer token?

Bearer Token is a type of authentication used in API requests. It's a string of characters that represents authorization to access a specific resource.

### Q38) What are the most commonly seen status codes in Postman?

Following is the list of most commonly seen status codes in Postman:

* **200 (OK):** Status code 200 specifies that the request was correct.
* **201 (Created):** Status code 201 specifies that the value wrapped with the request has been created in the database. Here, it is obvious that the request was correct.
* **204 (No Content):** Status code 204 specifies that the request was correct and received, but there is no content to send. It means there is no response to send to the client by the server.
* **400 (Bad Request):** Status code 400 specifies a bad request. A bad request means that the syntax of the request was incorrect. It appears when we have sent the wrong parameters along with the request URL or in the request's body.
* **401 (Unauthorized request):** Status code 401 specifies an unauthorized request. This status code appears when we are not authorized to access the server or enter the wrong credentials.
* **404 (Not Found):** Status code 404 specifies that the server was connected, but it could not find what was requested. It simply means "request not found". This status code normally appears when we request a web page not available on the server.