Rest API Interview Questions

Q.1 What do you understand by REST?

A.1. REST stands for Representational State Transfer. It is a web standards based architecture and take advantage of ubiquity of HTTP protocol and leverages HTTP method for data communication and to define actions.

It completely revolves around resources where every component is a resource and a resource is accessed by a common interface using HTTP standard methods.

In the architecture of REST, a REST Server provides access to resources and REST client accesses and presents the resources. Each resource in REST is identified by URIs or global IDs. It uses various representations to represent different resources like text, JSON and XML. Now a days JSON is the popular format and being commonly used in web services.

——————————————————————————————————————

Q.2. Explain the RESTFUL webservice?

A.2. Web services that are written by applying REST architectural concept are known as RESTFUL web services. HTTP methods are used in these services to implement the concept of REST architecture. It mainly focuses on system resources and defines how state of resource should be transported over HTTP protocol to different clients written in different language.

They usually defines a URI that provides resource representation such as JSON and set of HTTP Methods. HTTP methods like GET, POST, PUT and DELETE can be used to perform CRUD operations in RESTFUL webservices.

—————————————————————————————————————-

Q.3. What is a Resource in REST and how to represent a Resource?

A.3. In REST architecture, every component is treated as a resource. These resources can be text files, images, videos, html pages or dynamic business data. REST Server provides access to resources and REST Client accesses and modifies those resources. Each resource in REST is identified by URIs or global IDs

Each resource in REST is identified by logical URLs or global IDs. Unlike SOAP web services, we view the product data as a resource and this resource should contain all the required information.

REST uses various representations like text, JSON and XML to represent a resource . XML and JSON are the most popular and commonly used representations of resources.

—————————————————————————————————————-

Q.4. Mention some key characteristics of REST?

A.4. Below are some of the key characteristics of REST:

Rest is defined as stateless, hence the server has no state or no session data.

The server can be restarted between two calls as every data is passed to the server. This is possible with a well-applied REST API.

Mostly Web Services uses POST method to make operations, but REST uses GET method to access the resources.

Q.5. What are the best practices to design a resource representation?

A.5. Below are some of the best practices that needs to be considered while designing a representation format of a resource in a RESTful web services −

Understandability: Both the Server and the Client should understand and utilize the representation format of the resource.

Completeness: The Format used should be able to represent a resource completely. It should be able to represent simple as well as complex structures of resources.

For eg., one resource can contain another resource.

Linkablity : One resource can have a linkage to another resource, hence a format should be able to handle such situations.

Q.6. What are the core components of a HTTP Request?

A.6. There are 5 parts of a HTTP Request−

Verb: It indicates HTTPS methods like GET, PUT, POST, DELETE.

URI: It is Uniform Resource Identifier that is used to identify the resource on Server.

HTTP Version: It indicates the version of HTTP, for eg. HTTP v1.1

Request Header: It contains the metadata for HTTP request message like key-value pairs. For eg.: client or browser type, format supported by client, format of message body, cache settings,etc.

Request Body: It contains Message content or Resource representation.

Q.7. What are the HTTP methods that are supported by REST?

A.7. Below are the HTTP methods that are supported by REST:

GET: It is used to request a resource at the request URL. It do not contain a request body.

POST: It is used to submit information to the service for further processing. It typically returns the modified or new resource.

PUT: It is used to update the resource at the request URL.

DELETE: It is used to remove the resource at the request URL.

OPTIONS: It is used to indicate the supported techniques.

HEAD: It returns the information about the request URL.

Q.8. What are the core components of HTTP Response?

A.8. There are 4 parts of a HTTP Response:

Status/Response Code: It indicates server status for the requested resource. For eg. Code 200 means response is ok and code 404 means resource is not found

HTTP Version: It indicates the version of HTTP, for eg. HTTP v1.1

Response Header: It contains the metadata for HTTP response message like key-value pairs. For eg.: content length, content type, response data, server type,etc.

Response Body: It contains Response Message content or Resource representation.

Q.9. What is URI and its purpose in REST based webservices? What is the format of a URI in REST architecture?

A.9. URI stands for Uniform Resource Identifier. In REST Architecture, each resource is identified by its URI.

The main purpose of URI is to locate a resource on the server that is hosting the webservice.

The format of URI is:

<protocol>://<service-name>/<ResourceType>/<ResourceID>

Q.10.What are the best practices to create a standard URI for a web service?

A.10. Below are the best practices that should be followed while creating a standard URI:

Use Plural Noun: We should use plural noun to define the resources. For eg., To identify users as a resource ‘users’ should be used.

Avoid using spaces: Always use underscore(\_) or hyphen(-) while using long resource name. For eg. use ‘devlabs\_alliance’ instead of ‘devlabs alliance’

Use Lower case letters: Use lower case letter in the URI.

Maintain backward compatibility: Since WebService is a public service, so a URI once made public should always be available. If URI gets updated, then the older URI should be redirected to new URI using HTTP status code, 300.

Use HTTP Verb: Always use HTTP verbs like GET, PUT, POST or DELETE to do any operation on the resource.

Q.11. What is statelessness in RESTful Webservices?

A.11.Below are the advantages of statelessness in RESTFUL web services:

Each method request can be treated independently by Web services.

Web services are not required to maintain client’s previous interactions. It helps in simplifying application design.

RESTFUL Web services work seamlessly with HTTP protocol as HTTP is itself a statelessness protocol.

Q.12. What are the advantages of statelessness in RESTFUL Webservices?

A.12.Below are the advantages of statelessness in RESTFUL web services:

Each method request can be treated independently by Web services.

Web services are not required to maintain client’s previous interactions. It helps in simplifying application design.

RESTFUL Web services work seamlessly with HTTP protocol as HTTP is itself a statelessness protocol.

Q.13. What are the disadvantages of statelessness in RESTFUL Webservices?

A.13. Below is the major disadvantage of statelessness in RESTFUL web services:

Web services are required to get extra information in each request and then it needs to interpret that to get the client’s state, in case client interactions are to be taken care of.

Q.14. Explain the architectural style for creating web API.

A.14. Below is the architectural style for creating web API:

HTTP is used for client server communication

XML/JSON is used as formatting language

Simple URI is used as the address for the services

Stateless communication is there

Q.15. What is Caching in REST API?

A.15. Caching refers to a process when server stores response in client itself so that a client needs not to make server request for same resource again and again.

A server response should contain an information about how a caching is to be done. This helps a client to know whether it needs to cache some response for a period of time or caching the server response is not needed.

Q.16. What are the best practices for caching?

A.16. Below are the best practices to be followed while caching:

Always keep the expiry date of static contents like images, css, JavaScript cacheable, to 2-3 days. Never keep a high expiry date.

Always cached the Dynamic contents just for few hours only.

Q.17. What is JAX-RS?

A.17. JAX-RS stands for JAVA API for RESTFUL Web Services. It is a JAVA based programming language API and specification that is used to provide support for created RESTful Webservices.

JAX-RS makes heavy use of annotations that are available from Java SE 5 to simplify development of JAVA based web services and their deployment. It also provides supports to create clients for RESTful web services.

Q.18. List out the tools or API for developing or testing web api.

A.18. The testing tools for webservices for REST API are:

Spring REST web service using MVC

Jersey API

Axis

CFX

Restlet

Q.19. What are the best practices to be followed while designing a secure RESTFUL web service?

A.19. Below are the best practices to be followed while designing a RESTful web service −

Validation: Ensure validation of all inputs on the server. Always protect your server from SQL injection attacks.

Session based authentication: Always use session based authentication to authenticate a user whenever a request is made to a Web Service method.

No sensitive data in URL: Never keep username, password or session token in URL , these values should always be passed to Web Service via POST method.

Restriction on Method execution: Always allow restricted use of methods like GET, POST, DELETE.

Validate Malformed XML/JSON: Always check that well formed input is passed to a web service method.

Throw generic Error Messages: A web service method should always use HTTP error messages like 403 to show access forbidden etc.

Q.20. Mention what is the difference between SOAP and REST?

A.20. Below are the best practices to be followed while designing a RESTful web service −

SOAP

REST

SOAP is a protocol and is used for creating a communication between two computers by sharing XML document. REST is a service architecture and is designed for network-based software architectures.

Only XML format is supported in SOAP Many different data formats are supported by REST

SOAP based responses cannot be cached REST responses can be cached

It runs on HTTP but envelopes the message It uses the HTTP headers to hold meta information