**REST API Interview Questions and Answers:**

1. What is REST?

**Answer**: REST stands for Representational State Transfer, and is an architectural style based on the Hypertext Transfer Protocol (HTTP) for developing web-based applications.

REST outlines several guidelines that web services must follow to be considered RESTful. These guidelines ensure that requests and resources are sent easily and efficiently between client and server using standardized HTTP methods.

2. What is a REST API?

**Answer**: An application programming interface is a software-to-software interface that allows otherwise separate applications to interact and share data. For example, a news website could leverage the Twitter API to automatically find relevant tweets and include them in news articles.

A REST API, also called a RESTful API, is an API that follows REST principles. In a REST API, all data are treated as resources, each one represented by a unique uniform resource identifier (URI). For example, the Twitter API makes each tweet an available resource that can be retrieved by clients. Clients can also use Twitter’s API to post tweets and perform other actions on the site.

3. What are the principles of REST?

**Answer**: REST APIs must adhere to five requirements:

**Client-server decoupling**: The client and server can only interact in a series of requests and responses. Only clients can make requests, and only servers can send responses. This simple principle allows both parties to operate independently of each other.

**Uniform interface**: All communications between the client and server must follow the same protocol. For REST, this protocol is HTTP. A uniform interface simplifies integrations because every application is using the same language to request and send data.

**Stateless**: In stateless communication, the server does not store any information about past requests/responses. Each request and response contains all information needed to complete the interaction. Stateless communication reduces server load, saves memory, and improves performance. It also eliminates the possibility of a failed request caused by missing data.

**Layered system**: Layers are servers that sit between the client and API server. These additional servers perform various functions, like identifying spam and improving performance (See also: What Is a CDN?). In REST, layers are modular and can be added and removed without affecting the messages between the client and the API server.

**Cacheable**: Server responses indicate whether or not the resource is cacheable, so that clients can cache any resources to improve performance.

Additionally, REST includes one optional condition:

**Code on demand**: An API can send executable computer code to clients in its response. This lets the client application run the code in its own back end.

4. What does it mean for an API to be stateless?

**Answer**: Statelessness is one of the key principles of REST architecture. In stateless communication, the server does not store any information about previous communications. In other words, the client and server do not know each other’s state. Every request and response is a new interaction, and each request includes everything the server needs to give a successful response.

Statelessness simplifies client-server interactions because the server does not rely on past requests to process future requests, and thus does not need to consume space and resources storing data from these requests.

5. Which protocol do REST APIs use?

**Answer**: REST APIs use the HTTP protocol to communicate with clients. This allows REST APIs to be easily deployed over the internet, since HTTP is the same protocol that is used to deliver web pages to client browsers.

6. Which markup languages are primarily used to represent resources in REST APIs?

**Answer**: In REST APIs, XML (extensible markup language) and JSON (JavaScript Object Notation) are the two most common languages for representing resources.

7. Which HTTP request methods are supported by REST?

**Answer**: An HTTP request method indicates which action the client wants the API to perform on a resource. The four primary HTTP request methods in REST are:

**GET**: Requests a resource from the server. (Note that GET cannot modify server resources, as it is a read-only method.)

**POST**: Creates a new resource on the server.

**PUT**: Updates an existing resource on the server.

**DELETE**: Removes a resource from the server.

Additionally, two less common HTTP requests you should also know are:

**HEAD**: Requests meta-information about a resource. This request is similar to GET, but the response does not include a response body.

**OPTIONS**: Retrieves a list of possible methods for a resource.

8. What is the difference between the POST method and the PUT method?

**Answer**: POST and PUT are similar, but not exactly the same. POST is for creating a resource on the server, whereas PUT is for replacing a resource at a specific URI with another resource. If you use PUT at a URI that already has an associated resource, PUT will replace that resource. If there is no resource at the specified URI, PUT creates one.

Additionally, PUT is idempotent, which means that calling it multiple times will only result in one resource. This is because each call replaces the existing resource (or creates a new one if there is nothing to replace).

POST is not idempotent. If you call POST 10 times, you’ll end up with 10 different resources on the server, each with its own URI. This also means that POST responses are cacheable, whereas PUT responses are not.

9. What is CRUD?

**Answer**: CRUD stands for “Create, Read, Update, Delete.” These are the four basic actions that can be performed on databases through a REST API. Each action corresponds to an HTTP request method:

Create = POST

Read = GET

Update = PUT

Delete = DELETE

It’s not the most elegant of acronyms, but it works.

10. What is caching?

**Answer**: Caching is the method of temporarily storing a copy of a server response in a location (like computer memory) in order to retrieve it more quickly in the future.

When working with REST APIs, Caching improves server performance by reducing the work the server has to do to fulfill the request. Caching also makes applications that use the API run faster since they don’t need to send a new request every time they need a resource.

Cache duration of a resource (i.e., how long the resource can be cached by the client before the resource is retrieved again) is specified in the HTTP response header with the Cache-Control field.

**HTTP Methods Interview Questions and Answers:**

11) What is HTTP?

**Answer**: HTTP stands for Hypertext Transfer Protocol. It is a set of rule which is used for transferring the files like, audio, video, graphic image, text and other multimedia files on the WWW (World Wide Web). HTTP is a protocol that is used to transfer the hypertext from the client end to the server end, but HTTP does not have any security. Whenever a user opens their Web Browser, that means the user indirectly uses HTTP.

12) What are HTTP Request Messages?

**Answer**: HTTP Requests are messages which are sent by the client or user to initiate an action on the server.

It consists of various things:

**a**. Request Line: The Request-Line starts with a method token, which is followed by the Request-URI, the protocol version, and ending with CRLF. Using the SP characters, the elements are separated.

Syntax

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

**b**. The Resource Identified by a Request:

**c**. Request Header Fields: The request-header fields are used to allow the client to pass additional information to the server like the request and the client itself. The request header fields act as request modifiers, with semantics equivalent to the parameters on a programming language method invocation.

13) What are HTTP Request Methods?

**Answer**:

**GET**: This method retrieves information from the given server using a given URI. GET request can retrieve the data. It cannot apply other effects on the data.

**HEAD**: The HEAD method is the same as the GET method. It is used to transfer the status line and header section only.

**POST**: The POST request sends the data to the server. For example, file upload, customer information, etc. using the HTML forms.

**PUT**: The PUT method is used to replace all the current representations of the target resource with the uploaded content.

**DELETE**: The DELETE method is used to remove all the current representations of the target resource, which is given by URI.

**CONNECT**: The CONNECT method is used to establish a tunnel to the server, which is identified by a given URI.

14) What is the Status Code?

**Answer**: The Server issues an HTTP Status Code in response to a request of the client made to the server. Status code is a 3-digit integer. The first digit of status code is used to specify one of five standard classes of responses. The last two digits of status code do not have any categorization role.

15) What is HTTPS?

**Answer**: HTTPS stands for Hypertext Transfer Protocol Secure. HTTPS has a secure transfer. HTTPS is used to encrypt or decrypt user HTTP page or HTTP page requests that are returned by the webserver.

16) What is Content Negotiation in HTTP?

**Answer**: Most of the responses of HTTP include an entity which contains the information for interpretation by a user. Naturally, it is used to supply the user with the best available entity corresponding to the request. Unfortunately for cache and server, not all users have the same preferences for what is best. That's why HTTP has provisions for several mechanisms for "content negotiation", when there are multiple representations available, the process of selecting the best representation for a given response.

17) What is HTTP Message?

**Answer**: HTTP Message is used to show how data is exchanged between the client and the server. It is based on a client-server architecture. An HTTP client is a program that establishes a connection to a server to send one or more HTTP request messages. An HTTP server is a program that accepts connections to serve HTTP requests by sending an HTTP response messages.

18) What is HTTP CURL?

**Answer**: HTTP CURL is a command-line tool. It is available on all major operating systems

19) What is HTTP Response?

**Answer**: HTTP Response sent by a server to the client. The response is used to provide the client with the resource it requested. It is also used to inform the client that the action requested has been carried out. It can also inform the client that an error occurred in processing its request.

An HTTP response contains the following things:

**1**. Status Line

**2**. Response Header Fields or a series of HTTP headers

**3**. Message Body

20) What is HTTP Security?

**Answer**: HTTP is used to communicate over the internet, so users, information providers, and application developers should be aware of the limitations of security in HTTP/1.1. There are two methods to establish a secure HTTP connection: https URI scheme and the HTTP/1.1 Upgrade header.

**JSON Interview Questions and Answers:**

21. What exactly is JSON?

**Answer**: JSON is a faster and more accessible data exchange format. JSON (JavaScript Object Notation) is a platform and language-independent data format. JSON is a wire protocol that specifies a data format for data exchange/communication between processes/applications.

22. What exactly are JSON objects?

**Answer**: A collection of name/value pairs is referred to as an object. An object in JSON starts with a left brace and ends with a right brace. A colon follows each name, and the name/value pairs are separated by a comma (comma).

23. How do you convert a JSON text to a JavaScript object?

**Answer**: JSON is widely used to collect JSON data from a web server in a file or an HTTP request, convert the JSON data to JavaScript, and display the data on a web page.

24. Why is it necessary to utilize JSON instead of XML?

**Answer**: JSON objects are typed JSON types: Number, Array, Boolean, and String are all sorts of data. It is faster and lighter than XML since XML data is type less over the wire.

In XML, everything is a string.

Data is readily available as a JSON object in your JavaScript.

You have to read from an object property in your JavaScript code to get values.

25. Explain what JSON-RPC and JSON Parser are.

**Answer**: JSON RPC is a basic remote procedure call protocol similar to XML-RPC but uses the lightweight JSON format instead of XML. JSON Parser is a tool for parsing JSON data. JavaScript, PHP, and jQuery can parse it. Mentioning the JSON file extension JSON is a type of data format.

26. Mention what is JSON?

**Answer**: JSON is a simple data exchange format. JSON means JavaScript Object Notation; it is language and platform independent.

27. Mention what is the file extension of JSON?

**Answer**: File extension of JSON is .json

28. Mention which function is used to convert a JSON text into an object?

**Answer**: To convert JSON text into an object “eval” function is used.

29. Mention what are the data types supported by JSON?

**Answer**: Data types supported by JSON includes

1. Number
2. String
3. Boolean
4. Array
5. Object
6. Null

30. Mention what is the difference between JSON and JSONP?

**Answer**:

1. **JSON:** JSON is a simple data format used for communication medium between different systems
2. **JSONP:** It is a methodology for using that format with cross domain ajax requests while not being affected by same origin policy issue.

**Request – Response Interview Questions and Answers:**

31. What is HTTP Request?

**Answer**: HTTP requests are messages sent by the client to initiate an action on the server. Their start-line contain three elements: An HTTP method, a verb (like GET, PUT or POST) or a noun (like HEAD or OPTIONS), that describes the action to be performed

32. What is HTTP Response?

**Answer**: An HTTP response contains: A status line. A series of HTTP headers, or header fields. A message body, which is usually needed.

33. What is Request and Response model?

**Answer**: In request/response communication mode, one software module sends a request to a second software module and waits for a response. Because the first software module performs the role of the client, and the second, the role of the server, this mode is also referred to as client/server interaction.

34. What is Response Status code?

**Answer**: HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes: Informational responses (100 – 199) Successful responses (200 – 299)

35. What is HTTP?

**Answer**: HTTP is the protocol that governs communication between web servers and clients. It is the foundation of the World Wide Web.

36. How does the HTTP protocol work?

**Answer**: The HTTP protocol is a request-response protocol. A client, typically a web browser, sends a request to a server, and the server responds with the requested resource.

The HTTP protocol defines a set of methods, or verbs, that the client can use to request a resource from the server. The most common methods are GET, which requests a resource from the server, and POST, which submits data to the server.

37. What are 200 OK response codes in HTTP?

**Answer**: This response code is used to show that the request was successful.

38. What are 201 Created response codes in HTTP?

**Answer**: This response code shows that the request has been fulfilled, which results in the creation of a new resource.

39. What is the mean of 300 Multiple Choices response codes in HTTP?

**Answer**: This response code is used to indicate the multiple options for the resource from which the client may choose.

40. What are 400 Bad Request response codes in HTTP?

**Answer**: This code is used to indicate that the server did not understand the request due to invalid syntax.

method, a verb (like GET, PUT or POST) or a noun (like HEAD or OPTIONS), that describes the action to be performed

**Web Security Interview Questions and Answers:**

41. What REST API means?

**Answer**: **Representational State Transfer (REST)** is a software architecture that imposes conditions on how an API should work. REST was initially created as a guideline to manage communication on a complex network like the internet.

42. What is REST API in web?

**Answer**: A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding.

43. Why REST API security is important?

**Answer**: Rest API security is important because businesses use APIs to connect services and to transfer data, and so a hacked API can lead to a data breach. API abuse issues have roughly doubled over the past 4 years, according to the 2019 Application Security Risk Report by Micro Focus Fortify.

44. How does REST API handle security?

**Answer**: HTTPS and Transport Layer Security (TLS) offer a secured protocol to transfer encrypted data between web browsers and servers. Apart from other forms of information, HTTPS also helps to protect authentication credentials in transit.

45. How do I protect public REST API?

**Answer**: I can protect your API using strategies like generating SSL certificates, configuring a web application firewall, setting throttling targets, and only allowing access to your API from a Virtual Private Cloud (VPC).

46. What is JSON Web Token?

**Answer**: JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed.

47. How does JSON Web Tokens work?

**Answer**:

1. The application or client requests authorization to the authorization server.
2. When the authorization is granted, the authorization server returns an access token to the application.
3. The application uses the access token to access a protected resource (like an API).

48. What is rate limit throttling?

**Answer**: Rate Limiting and Throttling policies are designed to limit API access, but have different intentions: Rate limiting protects an API by applying a hard limit on its access. Throttling shapes API access by smoothing spikes in traffic.

49. What does request throttling mean?

**Answer**: Throttling is the process of limiting the number of requests you (or your authorized developer) can submit to a given operation in a given amount of time. A request can be when you submit an inventory feed or when you make an order report request.

50. What is Bearer Authentication/ Auth 2.0?

**Answer**: Bearer authentication (also called token authentication) is an HTTP authentication scheme that involves security tokens called bearer tokens, passes through request-response header. In General JSON Web Tokens JWT used for this purposes.