[**Top 20 Interview questions on REST API**](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/)

In this article,we will go through top 20 frequently asked interview questions on REST API

1. [What is REST?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#01)
2. [Name some of the commonly used HTTP methods used in REST based architecture?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#02)
3. [What is a Resource in REST?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#03)
4. [Explain REST design principles](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#04)
5. [Explain different REST Architectural Goals](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#05)
6. [Which protocol is used by RESTful webservices ?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#06)
7. [What is messaging in RESTful webservices?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#07)
8. [What are the core components of a HTTP Request and HTTP Response?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#08)
9. [What is addressing in RESTful webservices?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#09)
10. [What is URI?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#10)
11. [What is the purpose of HTTP Verb in REST based webservices?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#11)
12. [What is statelessness in RESTful Webservices?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#12)
13. [What are the advantages and disadvantages of statelessness in RESTful Webservices?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#13)
14. [What is the difference between PUT and POST operations?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#14)
15. [What should be the purpose of OPTIONS and HEAD method of RESTful web services?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#15)
16. [Explain different statemanagement principles in REST service](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#16)
17. [What is caching?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#17)
18. [What is the purpose of HTTP Status Code?](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#18)
19. [Explain the REST Constraints](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#19)
20. [Difference between SOAP and REST services](http://www.csharpstar.com/top-20-interview-questions-on-rest-api/#20)

1. What is REST?

Representational state transfer(REST) is an abstraction of architecture of world wide web. REST is an architectural style to design networked application.REST makes communication between remote computers easy by using the simple HTTP protocol which support for CRUD (Create, Read, Update, and Delete) operations on the server.

2. Name some of the commonly used HTTP methods used in REST based architecture?

Below are the commonly used HTTP methods used in REST.

* GET − Provides a read only access to a resource.
* PUT − Used to create a new resource.
* DELETE − Ued to remove a resource.
* POST − Used to update a existing resource or create a new resource.
* OPTIONS − Used to get the supported operations on a resource.

3. What is a Resource in REST?

* The service has a base URI that represents whole set of resources

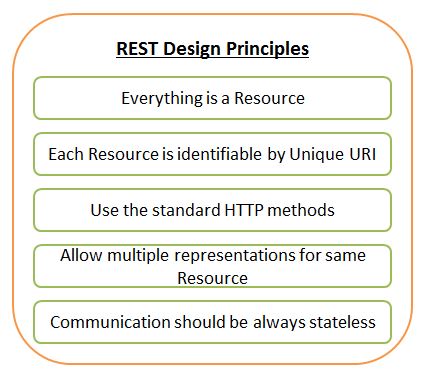
ex: CsharpStar.com/posts

* The base uri can be qualified with an identifier specifying an individual resource

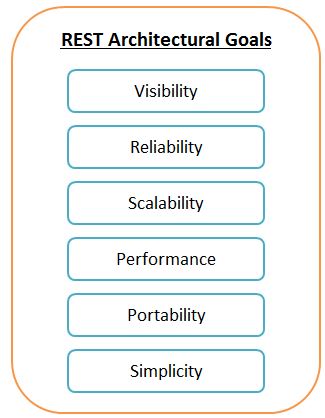
ex: CsharpStar.com/posts/1053

* RESTful services are hierarchical, resources offered by one service can contain more service
* REST uses various representations to represent a resource where text, JSON, XML. XML and JSON are the most popular representations of resources

4. Explain REST design principles

There are 5 design principles on REST.  
[](http://www.csharpstar.com/wp-content/uploads/2016/03/REST_Design_Principles.jpg)  
You can read more on this [here](http://www.csharpstar.com/rest-design-principles/).

5. Explain different REST Architectural Goals

[](http://www.csharpstar.com/wp-content/uploads/2016/03/REST_Architectural_Goals.jpg)

You can read detailed explanation on each goal [here](http://www.csharpstar.com/rest-architectural-goals/).

6. Which protocol is used by RESTful webservices ?

RESTful web services make use of HTTP protocol as a medium of communication between client and server.

7. What is messaging in RESTful webservices?

A client sends a message in form of a HTTP Request and server responds in form of a HTTP Response. This technique is termed as Messaging. These messages contain message data and metadata i.e. information about message itself.

8. What are the core components of a HTTP Request and HTTP Response?

There are 5 major components for HTTP Request.  
Verb − Indicate HTTP methods such as GET, POST, DELETE, PUT etc.  
URI − Uniform Resource Identifier (URI) to identify the resource on server.  
HTTP Version − Indicate HTTP version, for example HTTP v1.1 .  
Request Header − Contains metadata for the HTTP Request message as key-value pairs. For example, client ( or browser) type, format supported by client, format of message body, cache settings etc.  
Request Body − Message content or Resource representation.  
   
There are 4 major components for HTTP Response.  
Status/Response Code − Indicate Server status for the requested resource. For example 404 means resource not found and 200 means response is ok.  
HTTP Version − Indicate HTTP version, for example HTTP v1.1 .  
Response Header − Contains metadata for the HTTP Response message as key-value pairs. For example, content length, content type, response date, server type etc.  
Response Body − Response message content or Resource representation.

9. What is addressing in RESTful webservices?

Addressing refers to locating a resource or multiple resources lying on the server. It is analogous to locate a postal address of a person.

10. What is URI?

URI stands for Uniform Resource Identifier. Each resource in REST architecture is identified by its URI.

* Operations on the base URI affect the set of resources as a whole

1. GET lists them  
2. POST adds a new resource to the set  
3. DELETE deletes the whole set  
4. PUT replaces the set with a new set

* Operations on an ID-qualified URI affect an individual resource

1. GET retrieves it  
2. DELETE destroys it  
3. PUT replaces it or create if doesnot exists

11. What is the purpose of HTTP Verb in REST based webservices?

VERB identifies the operation to be performed on the resource.

12. What is statelessness in RESTful Webservices?

The communication between client and server must be stateless. This means that each request from a service consumer should contain all the necessary information for the service to understand the meaning of the request, and all session state data should then be returned to the service consumer at the end of each request

13. What are the advantages and disadvantages of statelessness in RESTful Webservices?

Advantages:

* Web services can treat each method request independently.
* Web services need not to maintain client’s previous interactions. It simplifies application design.
* As HTTP is itself a statelessness protocol, RESTful Web services work seamlessly with HTTP protocol

Disadvantages:

* Web services need to get extra information in each request and then interpret to get the client’s state in case client interactions are to be taken care of.

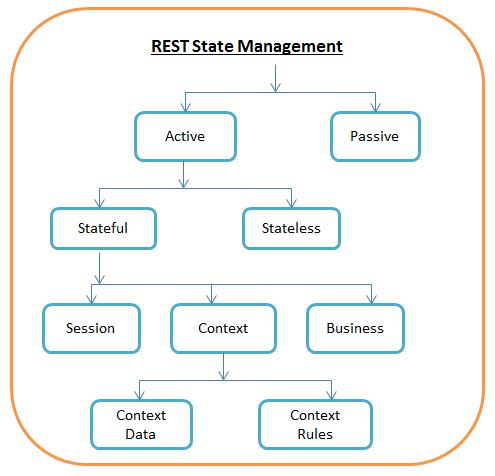
14. What is the difference between PUT and POST operations?

PUT − Used to create a new resource and POST − Used to update a existing resource or create a new resource.

15. What should be the purpose of OPTIONS and HEAD method of RESTful web services?

OPTIONS : list down the supported operations in a web service and should be read only.  
HEAD : return only HTTP Header, no Body and should be read only.

16. Explain different statemanagement principles in REST service?

[](http://www.csharpstar.com/wp-content/uploads/2016/03/REST_State_Management.jpg)  
You can read more on state management [here](http://www.csharpstar.com/rest-state-management/).

17. What is caching?

Caching refers to storing server response in client itself so that a client needs not to make server request for same resource again and again. A server response should have information about how a caching is to be done so that a client caches response for a period of time or never caches the server response.

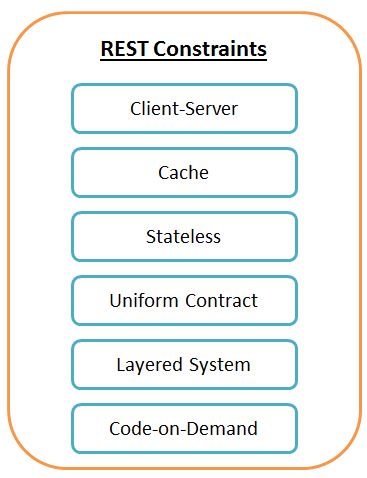
18. What is the purpose of HTTP Status Code?

HTTP Status code are standard codes and refers to predefined status of task done at server.

HTTP Status Code:

200 – OK, shows success.  
201 – CREATED, when a resource is successful created using POST or PUT request. Return link to newly created resource using location header.  
204 – NO CONTENT, when response body is empty  
304 – NOT MODIFIED, used to reduce network bandwidth usage in case of conditional GET requests  
400 – BAD REQUEST, states that invalid input is provided  
401 – FORBIDDEN, states that user is not having access to method being used  
404 – NOT FOUND, states that method is not available  
409 – CONFLICT, states conflict situation while executing the method  
500 – INTERNAL SERVER ERROR, states that server has thrown some exception while executing the method

19. Explain the REST Constraints

[](http://www.csharpstar.com/wp-content/uploads/2016/03/REST_Constraints.jpg)

You can read more on REST Constraints [here](http://www.csharpstar.com/rest-constraints/).

20. Difference between SOAP and REST services

* REST is simpler to program than SOAP
  + Retrieval consists of a URL(GET)
  + Creation consists of a URL(POST)
  + Update and Delete also simple (PUT,DELETE)
* SOAP semantics are opaque, so it bypasses the features of layered internet
* REST is lightweight on the client side
  + you need to write a little bit of script codeand you are off and running
  + Javascript frameworks make it browser independent and easy to integrate
* REST is lightweight in server side
  + you need to write a little bit of script codeand you are off and running
  + Frameworks support the web servers

Thanks for visiting !!