RESTful Webservice Interview Questions

This post is compilation of various interview question available on internet with their answers with an intent to prepare and revise for interview on the topic.
What REST stands for?
REST stands for REpresentational State Transfer.

What is REST?
REST is web standards based architecture and uses HTTP Protocol for data communication. Here every component is a resource which are accessed using common interface using HTTP standard methods.
In REST architecture, a REST Server simply provides access to resources and REST client accesses and presents the resources. Here each resource is identified by URIs/ global IDs. REST uses various representations to represent a resource like text, JSON and XML. Now a days JSON is the most
popular format being used in web services.

Name some of the commonly used HTTP methods used in REST based architecture?
Following well known HTTP methods are commonly used in REST based architecture ?
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.

What are webservices?
A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like
the Internet in a manner similar to inter-process communication on a single computer.

What are RESTful webservices?
Web services based on REST Architecture are known as RESTful web services. These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, provides resource representation such as
JSON and set of HTTP Methods.

What is a Resource in REST?
REST architecture treats every content as a resource. These resources can be text files, html pages, images, videos or dynamic business data. REST Server simply provides access to resources and REST client accesses and modifies the resources. Here each resource is identified by URIs/
global IDs.

How to represent a resource in REST?
REST uses various representations to represent a resource where text, JSON, XML. XML and JSON are the most popular representations of resources.

What are the best practices to design a resource representation?
Following are important points to be considered while designing a representation format of a resource in a RESTful web services ?
Understandability ? Both Server and Client should be able to understand and utilize the representation format of the resource.
Completeness ? Format should be able to represent a resource completely. For example, a resource can contain another resource.
Format should be able to represent simple as well as complex structures of resources.

Linkablity ? A resource can have a linkage to another resource, a format should be able to handles such situations.

Which protocol is used by RESTful webservices?
RESTful web services make use of HTTP protocol as a medium of communication between client and server.

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.

What are the core components of a HTTP Request?
A HTTP Request has five major parts ?
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.

What are the core components of a HTTP response?
A HTTP Response has four major parts ?
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.

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.

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

What is purpose of a URI in REST based webservices?
Purpose of an URI is to locate a resources on the server hosting the web service.

What is format of a URI in REST architecture?
A URI is of following format ?
<protocol>://<service-name>/<ResourceType>/<ResourceID>

What is the purpose of HTTP Verb in REST based webservices?
VERB identifies the operation to be performed on the resource.

What are the best practices to create a standard URI for a web service?
Following are important points to be considered while designing a URI ?
Use Plural Noun ? Use plural noun to define resources. For example, we've used users to identify users as a resource.
Avoid using spaces ? Use underscore _ or hyphen ? when using a long resource name, for example, use authorized_users instead of authorized%20users.
Use lowercase letters ? Although URI is case-insensitive, it is good practice to keep url in lower case letters only.

Maintain Backward Compatibility ? As Web Service is a public service, a URI once made public should always be available. In case, URI gets updated, redirect the older URI to new URI using HTTP Status code, 300.
Use HTTP Verb ? Always use HTTP Verb like GET, PUT, and DELETE to do the operations on the resource. It is not good to use operations names in URI.

What is statelessness in RESTful Webservices?
As per REST architecture, a RESTful web service should not keep a client state on server. This restriction is called statelessness. It is responsibility of the client to pass its context to server and then server can store this context to process client's further request. For example, session maintained by server is identified by session identifier passed by the client.

What are the advantages of statelessness in RESTful Webservices?
Following are the benefits of statelessness in RESTful web services ?
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.

What are the disadvantages of statelessness in RESTful Webservices?
Following is the disadvantage of statelessness in RESTful web services ?
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.

What do you mean by idempotent operation?
Idempotent operations means their result will always same no matter how many times these operations are invoked.

Which type of Webservices methods are to be idempotent?
PUT and DELETE operations are idempotent.

Which type of Webservices methods are to be read only?
GET operations are read only and are safe.

What is the difference between PUT and POST operations?
PUT and POST operation are nearly same with the difference lying only in the result where PUT operation is idempotent and POST operation can cause different result.

What should be the purpose of OPTIONS method of RESTful web services?
It should list down the supported operations in a web service and should be read only.

What should be the purpose of HEAD method of RESTful web services?
It should return only HTTP Header, no Body and should be read only.

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.

Which header of HTTP response, provides the date and time of the resource when it was created?
Date header provides the date and time of the resource when it was created.

Which header of HTTP response, provides the date and time of the resource when it was last

modified?
Last Modified header provides the date and time of the resource when it was last modified.

Which header of HTTP response provides control over caching?
Cache-Control is the primary header to control caching.

Which header of HTTP response sets expiration date and time of caching?
Expires header sets expiration date and time of caching.

Which directive of Cache Control Header of HTTP response indicates that resource is cachable by any component?
Public directive indicates that resource is cachable by any component.

Which directive of Cache Control Header of HTTP response indicates that resource is cachable by only client and server, no intermediary can cache the resource?
Private directive indicates that resource is cachable by only client and server, no intermediary can cache the resource.

Which directive of Cache Control Header of HTTP response indicates that resource is not cachable?
no-cache/no-store directive indicates that resource is not cachable.

Which directive of Cache Control Header of HTTP response can set the time limit of caching?
max-age directive indicates that the caching is valid up to max-age in seconds. After this, client has to make another request.

Which directive of Cache Control Header of HTTP response provides indication to server to revalidate resource if max-age has passed?
must-revalidate directive provides indication to server to revalidate resource if max-age has passed.

What are the best practices for caching?
Always keep static contents like images, css, JavaScript cacheable, with expiration date of 2 to 3
days. Never keep expiry date too high.
Dynamic contents should be cached for few hours only.

What are the best practices to be followed while designing a secure RESTful web service?
As RESTful web services work with HTTP URLs Paths so it is very important to safeguard a RESTful web service in the same manner as a website is be secured. Following are the best practices to be followed while designing a RESTful web service ?
Validation ? Validate all inputs on the server. Protect your server against SQL or NoSQL injection attacks.
Session based authentication ? Use session based authentication to authenticate a user whenever a request is made to a Web Service method.
No sensitive data in URL ? Never use username, password or session token in URL , these values should be passed to Web Service via POST method.
Restriction on Method execution ? Allow restricted use of methods like GET, POST, DELETE. GET method should not be able to delete data.
Validate Malformed XML/JSON ? Check for well formed input passed to a web service method.
Throw generic Error Messages ? A web service method should use HTTP error messages like 403 to show access forbidden etc.

What is the purpose of HTTP Status Code?
HTTP Status code are standard codes and refers to predefined status of task done at server. For example, HTTP Status 404 states that requested resource is not present on server.

What HTTP Status Code 200 states?

It means, OK, shows success.

What HTTP Status Code 201 states?
It means, CREATED, when a resource is successful created using POST or PUT request. Return link to newly created resource using location header.

What HTTP Status Code 204 states?
It means, NO CONTENT, when response body is empty for example, a DELETE request.

What HTTP Status Code 304 states?
It means, NOT MODIFIED, used to reduce network bandwidth usage in case of conditional GET requests. Response body should be empty. Headers should have date, location etc.

What HTTP Status Code 400 states?
It means, BAD REQUEST, states that invalid input is provided e.g. validation error, missing data.

What HTTP Status Code 401 states?
It means, FORBIDDEN, states that user is not having access to method being used for example, delete access without admin rights.

What HTTP Status Code 404 states?
It means, NOT FOUND, states that method is not available.

What HTTP Status Code 409 states?
It means, CONFLICT, states conflict situation while executing the method for example, adding
duplicate entry.

What HTTP Status Code 500 states?
It means, INTERNAL SERVER ERROR, states that server has thrown some exception while executing
the method.

What is JAX-RS?
JAX-RS stands for JAVA API for RESTful Web Services. JAX-RS is a JAVA based programming language API and specification to provide support for created RESTful Webservices. Its 2.0 version was released in 24 May 2013. JAX-RS makes heavy use of annotations available from Java SE 5 to
simplify development of JAVA based web services creation and deployment. It also provides supports for creating clients for RESTful web services.

What is a REST Web Service?
There are a set of architectural constraints (we will discuss them shortly) called Rest Style Constraints. Any service which satisfies these constraints is called RESTful Web Service.

There are a lot of misconceptions about REST Web Services : They are over HTTP , based on JSON etc. Yes : More than 90% of RESTful Web Services are JSON over HTTP. But these are not necessary constraints. We can have RESTful Web Services which are not using JSON and which are not over HTTP.

What are important constraints for a RESTful Web Service?
The five important constraints for RESTful Web Service are

Client - Server : There should be a service producer and a service consumer.
The interface (URL) is uniform and exposing resources. Interface uses nouns (not actions)

The service is stateless. Even if the service is called 10 times, the result must be the same.
The service result should be Cacheable. HTTP cache, for example.
Service should assume a Layered architecture. Client should not assume direct connection to server - it might be getting info from a middle layer - cache.

What is Richardson Maturity Model?
Richardson Maturity Model defines the maturity level of a Restful Web Service. Following are the different levels and their characteristics.
Level 0 : Expose SOAP web services in REST style. Expose action based services (http://server/getPosts, http://server/deletePosts, http://server/doThis, http://server/doThat etc) using REST.
Level 1 : Expose Resources with proper URI's (using nouns). Ex: http://server/accounts, http://server/accounts/10. However, HTTP Methods are not used.
Level 2 : Resources use proper URI's + HTTP Methods. For example, to update an account, you do a PUT to . The create an account, you do a POST to . Uri's look like posts/1/comments/5 and accounts/1/friends/1.
Level 3 : HATEOAS (Hypermedia as the engine of application state). You will tell not only about the information being requested but also about the next possible actions that the service consumer can do. When requesting information about a facebook user, a REST service can return user details along with information about how to get his recent posts, how to get his recent comments and how to retrieve his friend's list.

What are the best practices in designing RESTful APIs?
While designing any API, the most important thing is to think about the api consumer i.e. the client who is going to use the service. What are his needs? Does the service uri make sense to him? Does the request, response format make sense to him?
In Rest, we think Nouns (resources) and NOT Verbs (NOT actions). So, URI's should represent resources. URI's should be hierarchical and as self descriptive as possible. Prefer plurals.
Always use HTTP Methods. Best practices with respect to each HTTP method is described in the next question.

What are the best practices in using HTTP methods with Restful Web Services?
GET : Should not update anything. Should be idempotent (same result in multiple calls). Possible Return Codes 200 (OK) + 404 (NOT FOUND) +400 (BAD REQUEST)
POST : Should create new resource. Ideally return JSON with link to newly created resource. Same return codes as get possible. In addition : Return code 201 (CREATED) is possible.
PUT : Update a known resource. ex: update client details. Possible Return Codes : 200(OK)
DELETE : Used to delete a resource.

Can you explain a little bit about JAX-RS?
JAX-RS is the JEE Specification for Restful web services implemented by all JEE compliant web servers (and application servers).

Important Annotations:

@ApplicationPath("/"). @Path("users") : used on class and methods to define the url path.
@GET @POST : Used to define the HTTP method that invokes the method.
@Produces(MediaType.APPLICATION_JSON) : Defines the output format of Restful service.
@Path("/{id}") on method (and) @PathParam("id") on method parameter : This helps in defining a dynamic parameter in Rest URL.
@Path("{user_id}/followers/{follower_id}") is a more complicated example.
@QueryParam("page") : To define a method parameter ex: /users?page=10.
Useful methods:

Response.OK(jsonBuilder.build()).build() returns json response with status code.
Json.createObjectBuilder(). add("id",user.getId()); creates a user object.

What are the advantages of Restful web services?
Lightweight : Easy to consume from mobile devices also.
Easy to expose : Little or no restrictions on output format and communication protocol.
Most Restful services use HTTP protocol : Entire web is based on HTTP and is built for efficiency of HTTP. Things like HTTP caching enable Restful services to be effective.
High Performance : Less xml & soap overhead and More caching enable Restful services to be highly performant.

What is the difference between REST and SOAP Based Services?
First of all, REST is a set of architectural principles defining how a RESTful service should look look like. SOAP is a message exchange format. SOAP defines the structure of message to exchanged. How should the header be? How should the request content be? So, there is no real comparison between REST and SOAP.
Restful Sample Implementation : JSON over HTTP
SOAP Sample Implementation : XML over SOAP over HTTP
All comparison is between the Sample Restful and SOAP implementations described above.

REST is built over simple HTTP protocol. SOAP services are more complex to implement and more complex to consume.
REST has better performance and scalability. REST reads can be cached, SOAP based reads cannot be cached.
REST permits many different data formats (JSON is the most popular choice) where as SOAP only permits XML.
SOAP services have well defined structure and interface (WSDL).
SOAP is based on well defined standards (WS-Security, WS-AtomicTransaction and WS-ReliableMessaging).