**1) Explain what is REST and RESTFUL?**

REST represents REpresentational State Transfer; it is a relatively new aspect of writing web API.

RESTFUL is referred for web services written by applying REST architectural concept are called RESTful services, it focuses on system resources and how state of resource should be transported over HTTP protocol to different clients written in different language. In RESTFUL web service HTTP methods like GET, POST, PUT and DELETE can be used to perform CRUD operations.

**2) Explain the architectural style for creating web API?**

The architectural style for creating web api are

* HTTP for client server communication
* XML/JSON as formatting language
* Simple URI as the address for the services
* Stateless communication

**3) Mention what tools are required to test your web API?**

SOAPUI tool for SOAP WS and Firefox "poster" plugin for RESTFUL services.

**4) Mention what are the HTTP methods supported by REST?**

HTTP methods supported by REST are:

* **GET:**It requests a resource at the request URL. It should not contain a request body as it will be discarded. Maybe it can be cached locally or on the server.
* **POST:**It submits information to the service for processing; it should typically return the modified or new resource
* **PUT:**At the request URL it update the resource
* **DELETE:**At the request URL it removes the resource
* **OPTIONS:**It indicates which techniques are supported
* **HEAD:**About the request URL it returns meta information

**5) Mention whether you can use GET request instead of PUT to create a resource?**

No, you are not supposed to use POST or GET. GET operations should only have view rights

**6) Mention what are resources in a REST architecture?**

Resources are identified by logical URLs; it is the key element of a RESTful design. Unlike, SOAP web services in REST, you view the product data as a resource and this resource should contain all the required information.

**7) Mention what is the difference between AJAX and REST?**

|  |  |
| --- | --- |
| **AJAX** | **REST** |
| * In Ajax, the request are sent to the server by using XMLHttpRequest objects. The response is used by the JavaScript code to dynamically alter the current page * Ajax is a set of technology; it is a technique of dynamically updating parts of UI without having to reload the page * Ajax eliminates the interaction between the customer and server asynchronously * REST requires the interaction between the customer and server | * REST have a URL structure and a request/response pattern the revolve around the use of resources * REST is a type of software architecture and a method for users to request data or information from servers * REST requires the interaction between the customer and server |

**7) Mention some key characteristics of REST?**

Some key characteristics of REST includes

* REST is stateless, therefore the SERVER has no state (or session data)
* With a well-applied REST API, the server could be restarted between two calls as every data is passed to the server
* Web service mostly uses POST method to make operations, whereas REST uses GET to access resources

**8) Mention what are the different application integration styles?**

The different integration styles include

* Shared database
* Batch file transfer
* Invoking remote procedure (RPC)
* Swapping asynchronous messages over a message oriented middle-ware (MOM)

**9) Explain how JAXB related to RESTful web API?**

JAXB stands for java arch for XML binding.

**10) Mention what is the difference between PUT and POST?**

"PUT" puts a file or resource at a particular URI and exactly at that URI. If there is already a file or resource at that URI, PUT changes that file or resource. If there is no resource or file there, PUT makes one

POST sends data to a particular URI and expects the resource at that URI to deal with the request. The web server at this point can decide what to do with the data in the context of specified resource

PUT is idempotent meaning, invoking it any number of times will not have an impact on resources.

However, POST is not idempotent, meaning if you invoke POST multiple times it keeps creating more resources

**11) Mention which markup language can be used in restful web api?**

JSON and XML are the two markup language that can be used in restful web api

**12) Mention what is the difference between RPC or document style web services? How you determine to which one to choose?**

In document style web services, we can transport an XML message as part of SOAP request which is not possible in RPC style web service. Document style web service is most appropriate in some application where XML message behaves as document and content of that document can alter and intention of web service does not rely on the content of XML message.

**13) Mention what is JAX-WS and JAX-RS?**

Both JAX-WS and JAX-RS are libraries (APIs) for doing communication in various ways in Java. JAX-WS is a library that can be used to do SOAP communication in JAVA, and JAX-RS lets you do the REST communication in JAVA.

**14) List out the tools or API for developing or testing web api?**

Testing tools for web services for REST APIs includes

* Spring REST web service using MVC
* Jersey API
* CFX
* Axis
* Restlet,

**15) Mention what is the difference between SOAP and REST?**

|  |  |
| --- | --- |
| **SOAP** | **REST** |
| * SOAP is a protocol through which two computer communicates by sharing XML document * SOAP permits only XML * SOAP based reads cannot be cached * SOAP is like custom desktop application, closely connected to the server * SOAP is slower than REST * It runs on HTTP but envelopes the message | * Rest is a service architecture and design for network-based software architectures * REST supports many different data formats * REST reads can be cached * A REST client is more like a browser; it knows how to standardized methods and an application has to fit inside it * REST is faster than SOAP * It uses the HTTP headers to hold meta information |

**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. It revolves around resource where every component is a resource and a resource is accessed by a common interface using HTTP standard methods. REST was first introduced by Roy Fielding in 2000.

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 resource(s) 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?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 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.