**Q-1. Explain REST?**

**Ans.** REST stands for Representational State Transfer. REST is an architectural style of developing web services which take advantage of the ubiquity of HTTP protocol and leverages HTTP method to define actions. It revolves around resource where every component is a resource which can be accessed by a common interface using HTTP standard methods.

In REST architecture, a REST Server provides access to resources and REST client accesses and presents those resources. Here each resource is identified by URIs or global IDs. REST uses different ways to represent a resource like text, JSON, and XML.XML and JSON are the most popular representations of resources these days.

**Q-2. Explain the RESTFul Web Service?**

**Ans.** Mostly, there are two kinds of Web Services which are quite popular.

**1.** SOAP (Simple Object Access Protocol) which is an XML-based way to expose web services.

**2.** Web services developed using REST style 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.

**Q-3. Explain what is a “Resource” in REST?**

**Ans.** REST architecture treats every content as a resource. These resources can be either text files, HTML pages, images, videos or dynamic business data.

REST Server provides access to resources and REST client accesses and modifies these resources. Here each resource is identified by URIs/ global IDs.

**Q-4. What is the most popular way to represent a resource in REST?**

**Ans.** REST uses different representations to define a resource like text, JSON, and XML.

XML and JSON are the most popular representations of resources.

**Q-5. Which protocol is used by RESTful web services?**

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

**Q-6. What is messaging in RESTful web services?**

**Ans.** RESTful web services make use of HTTP protocol as a medium of communication between client and server. The client sends a message in the form of an HTTP Request.

In response, the server transmits the HTTP Response. This technique is called Messaging. These messages contain message data and metadata, i.e., information about itself.

**Q-7. State the core components of an HTTP Request?**

**Ans.** Each HTTP request includes five key elements.

**1.** The Verb which indicates HTTP methods such as GET, PUT, POST, DELETE.  
**2.** URI stands for Uniform Resource Identifier (URI). It is the identifier for the resource on the server.  
**3.** HTTP Version which indicates HTTP version, for example-HTTP v1.1.  
**4.** Request Header carries metadata (as key-value pairs) for the HTTP Request message. Metadata could be a client (or browser) type, the format that the client supports, message body format, and cache settings.  
**5.** Request Body indicates the message content or resource representation.

**Q-8. State the core components of an HTTP response?**

**Ans.** Every HTTP response includes four key elements.

**1.** Status/Response Code – Indicates Server status for the resource present in the HTTP request. For example, 404 means resource not found, and 200 means response is ok.  
**2.** HTTP Version – Indicates HTTP version, for example-HTTP v1.1.  
**3.** Response Header – Contains metadata for the HTTP response message stored in the form of key-value pairs. For example, content length, content type, response date, and server type.  
**4.** Response Body – Indicates response message content or resource representation.

**Q-9. Name the most commonly used HTTP methods supported by REST?**

**Ans.** There are a few HTTP methods in REST which are more popular.

**1.** GET -It requests a resource at the request-URL. It should not contain a request body as it will get discarded. Maybe it can be cached locally or on the server.  
**2.** POST – It submits information to the service for processing; it should typically return the modified or new resource.  
**3.** PUT – At the request URL it updates the resource.  
**4.** DELETE – It removes the resource at the request-URL.  
**5.** OPTIONS -It indicates the supported techniques.  
**6.** HEAD – It returns meta information about the request URL.

**Q-10. Mention, whether you can use GET request instead of PUT, to create a resource?**

**Ans.** No, you shouldn’t use a PUT or POST method. Instead, apply the GET operation which has view-only rights.

**Q-11. Is there any difference between PUT and POST operations? Explain it.**

**Ans.** PUT and POST operation are almost the same. The only difference between the two is in terms of the result generated by them.

A PUT operation is idempotent while the POST operation can give a different result.

Let’s take an example.

**1.** PUT puts a file or resource at a particular URI and precisely at that URI. If the resource already exists, then PUT updates it. If it’s a first-time request, then PUT creates one.

**2.** 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 the specified resource.

**Q-12. What purpose does the OPTIONS method serve for the RESTful web services?**

**Ans.** This method lists down all the operations a web service supports. It makes read-only requests to the server.

**Q-13. What is URI? Explain its purpose in REST-based web services. What is its format?**

**Ans.** URI stands for Uniform Resource Identifier. URI is the identifier for the resource in REST architecture.

The purpose of a URI is to locate a resource(s) on the server hosting the web service. A URI is of the following format-

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

**Q-14. What do you understand by payload in RESTFul web service?**

**Ans.** Request body of every HTTP message includes request data called as Payload. This part of the message is of interest to the recipient.

We can say that we send the payload in the POST method but not in <GET> and <DELTE> methods.

**Q-15. What is the upper limit for a payload to pass in the POST method?**

**Ans.** <GET> appends data to the service URL. But, its size shouldn’t exceed the maximum URL length. However, <POST> doesn’t have any such limit.

So, theoretically, a user can pass unlimited data as the payload to the POST method. But, if we consider a real use case, then sending a POST with large payload will consume more bandwidth. It’ll take more time, and present performance challenges to your server. Hence, a user should take action accordingly.

**Q-16. Explain the caching mechanism?**

**Ans.** Caching is a process of storing server response at the client end. It makes the server save significant time from serving the same resource again and again.

The server response holds information which leads a client to perform the caching. It helps the client to decide how long to archive the response or not to store it at all.

**Q-17. List the main differences between SOAP and REST?**

**Ans.**

|  |  |
| --- | --- |
| **SOAP** | **REST** |
| **1.** SOAP is a protocol through which two computer communicates by sharing the XML document. | **1.** Rest is a service architecture and design for network-based software architecture. |
| **2.** SOAP supports the only XML format. | **2.** It supports many different data formats. |
| **3.** SOAP does not support caching. | **3.** It supports caching. |
| **4.** SOAP is like a custom desktop application, closely connected to the server. | **4.** A REST client is just like a browser and uses standard methods. An application has to fit inside it. |
| **5.** SOAP is slower than the REST. | **5.** It is faster than SOAP. |
| **6.** It runs on HTTP but envelopes the message. | **6.** It uses the HTTP headers to hold meta information. |

**Q-18. What are the tools available for testing web services?**

**Ans.** Following tools can help in testing the SOAP and RESTful web services.

**1.** SOAP UI tool.  
**2.** Poster for Firefox browser.  
**3.** The Postman extension for Chrome.

**Q-19. Explain the factors that help to decide about the style of web service to use? SOAP or REST?**

**Ans.** In general, using REST-based web service is preferred due to its simplicity, performance, scalability, and support for multiple data formats.

However, SOAP is favorable to use where service requires an advanced level of security and transactional reliability.

But you can read the following facts before opting for any of the styles.

**1.** Does the service expose data or business logic? To expose data REST will be a better choice and SOAP for logic.  
**2.** If the consumer or the service providers require a formal contract, then SOAP can provide such a contract via WSDL.  
**3.** Need to support multiple data formats. REST supports this.  
**4.** Support for AJAX calls. REST can use the XMLHttpRequest.  
**5.** Synchronous and asynchronous calls – SOAP enables both synchronous/asynchronous operations whereas REST has built-in support for synchronous.  
**6.** Stateless or Stateful calls -REST is suited for stateless operations.

Here are some of the advanced-level facts that you can consider as well.

**1.** Security requirement – SOAP provides a high level of security.  
**2.** Transaction support – SOAP has good support for transaction management.  
**3.** Limited bandwidth – SOAP has a lot of overhead when sending/receiving packets since it’s XML based, requires a SOAP header. However, the REST requires less bandwidth to send requests to the server. Its messages are mostly built using JSON.  
**4.** Ease of use – It is easy to implement, test, and maintain REST-based application.