**1. What are SOAP Web services?**

SOAP is defined as an XML-based protocol. It is known for designing and developing web services as well as enabling communication between applications developed on different platforms using various programming languages over the Internet. It is both platform and language independent.

**2. How does SOAP work?**

SOAP is used to provide a user interface that can be accessed by the client object, and the request that it sends goes to the server, which can be accessed using the server object. The user interface creates some files or methods consisting of server object and the name of the interface to the server object. It also contains other information such as the name of the interface and methods. It uses HTTP to send the XML to the server using the POST method, which analyzes the method and sends the result to the client. The server creates more XML consisting of responses to the request of user interface using HTTP. The client can use any approach to send the XML, like the SMTP server or POP3 protocol to pass the messages or reply to queries.

**3. When to use SOAP API?**

Use the SOAP API to create, retrieve, update, or delete records, like accounts, leads, and use-defined objects. With more than 20 different calls, you can also use the SOAP API to manage passwords, perform searches, etc. by using the SOAP API in any language that supports web services.

**4. How do users utilize the facilities provided by SOAP?**

* PutAddress(): It is used to enter an address in the webpage and has an address instance on the SOAP call.
* PutListing(): It is used to allow the insertion of a complete XML document into the web page. It receives the XML file as an argument and transports the XML file to XML parser liaison, which reads it and inserts it into the SOAP call as a parameter.
* GetAddress(): It is used to get a query name and gets the result that best matches a query. The name is sent to the SOAP call in the form of text character string.
* GetAllListing(): It is used to return the full list in an XML format.

**5. What is the major obstacle users faced when using SOAP?**

When using SOAP, users often see the firewall security mechanism as the biggest obstacle. This block all the ports leaving few like HTTP port 80 and the HTTP port used by SOAP that bypasses the firewall. The technical complaint against SOAP is that it mixes the specification for message transport with the specification for message structure.

**6. What are the various approaches available for developing SOAP-based web services?**

There are two different methods available for developing SOAP-based web services, which are explained below:

* Contract-first approach: the contract is first defined by XML and WSDL, and then Java classes are derived from the contract.
* Contract-last approach: Java classes are first defined, and then the contract is generated, which is normally the WSDL file from the Java class.

“Contract-first” method is the most popular approach.

**7. What are the elements of a SOAP message structure?**

It is a common XML document that contains the elements as a SOAP message

* Envelope: It is an obligatory root element that translates the XML document and defines the beginning and end of the message.
* Header: It is an optional item which contains information about the message being sent.
* Body: It contains the XML data comprising the message being sent.
* Fault: It provides the information on errors that occurred while during message processing.

**8. What are the syntax rules for a SOAP message?**

* Must use encoded XML
* Envelope namespace must be used
* Encoding namespace must be used
* Must not consist of a DTD reference
* Must not have XML processing instruction

**9. What is the transport method in SOAP?**

Application layer and transport layers of a network are used by SOAP; HTTP and SMTP are the valid protocol of the application layer used as the transport for SOAP. HTTP is more preferable, since it works well with the current Internet infrastructure, in particular with firewalls.

The SOAP requests can be sent using an HTTP GET method while the specification only contains details about HTTP POST.

**10. What are some important characteristics of a SOAP envelope element?**

* SOAP message has a root Envelope element
* Envelope is an obligatory part of the SOAP message.
* If an envelope includes a header element, it should not contain more than one.
* Envelop version will change if the SOAP version changes.
* The SOAP envelope is indicated by the prefix ENV and the envelope element.
* The optional SOAP encoding is also specified using a namespace and the optional encoding style element.

**11. What are the major functionalities provided by the SOAP protocol class?**

The SOAP protocol is used to provide simple access methods for all the applications available on the Internet, providing the following functionalities:

* Call: A class which provides the main functionality for a remote method for which a call is needed. It is used to create the call() and to specify the encoding style of the registry that will be assigned when if necessary. This call() function is used by the RPC call, which represents the options of the call object.
* Deployment Descriptor: A class used to provide the information about the SOAP services. It enables easy deployment without the need for other approaches.
* DOM2 Writer: A class that serializes and uses DOM node as XML string to provide more functionalities.
* RPC Message: A class used as the base class that calls and replies to the request submitted to the server.
* Service Manager: A class that provides, lists and then outputs all SOAP services.

**12. What are the web relation functionalities provided by SOAP protocol?**

* HTTPUtils: This provides the functionality of the POST method to safely meet the requirements.
* Parameter: It is an argument for a RPC call used by both the client and the server.
* Response: It is an object that represents an RPC reply from both client and server, but the result will not be displayed until after the method call.
* TCPTunnel: It is an object that provides the ability to listen on a specific port and to forward all the host and port names.
* TypeConverter: It helps to convert an object of one type into another type and this is called using the class in the form object.

**13. How does the message security model allow the creation of SOAP more secure to use?**

The security model includes the given security tokens. These tokens comprise digital signatures for protection and authentication of SOAP messages. Security tokens can be used to provide the bond between authentication secrets or keys and security identities. Security token uses the authentication protocols and an X.509 certificate to define the relationship between the public key and identity key. The signatures are used to verify the messages and their origin, generate knowledge to confirm the security tokens to bind the identity of a person to the identity of the originator. Security model prevents different attacks and can be used to protect the SOAP architecture.

**14. What is the difference between the top-down and bottom-up approaches in SOAP Web services?**

* Top-down SOAP Web services include creating WSDL document to create a contract between the web service and the client, with a required code as an option. This is also known as the Contract-first approach. The top-down approach is difficult to implement because classes must be written to confirm the contract defined in WSDL. One of the benefits of this method is that both the client and server code can be written in parallel.
* Bottom-up SOAP web services require the code to be written first and then WSDL is generated. It is also known as the Contract-last approach. Since WSDL is created based on the code, the bottom-up approach is easy to implement and client codes must wait for WSDL from the server side to start working.

**15. What are the advantages of SOAP?**

* SOAP is both platform and language independent.
* SOAP separates the encoding and communications protocol from the runtime environment.
* Web service can retrieve or receive a SOAP user data from a remote service, and the source’s platform information is completely independent of each other.
* Everything can generate XML, from Perl scripts through C++ code to J2EE app servers.
* It uses XML to send and receive messages.
* It uses standard internet HTTP protocol.
* SOAP runs over HTTP; it eliminates firewall problems. When protocol HTTP is used as the protocol binding, an RPC call will be automatically assigned to an HTTP request, and the RPC response will be assigned to an HTTP reply.
* Compared to RMI, CORBA, and DCOM, SOAP is very easy to use.
* SOAP acts as a protocol to move information in a distributed and decentralized environment.
* SOAP is independent of the transport protocol and can be used to coordinate different protocols.

**16. What are the disadvantages of SOAP?**

SOAP is typically significantly slower than other types of [middleware](https://searchmicroservices.techtarget.com/definition/middleware) standards, including CORBA, because SOAP uses a detailed XML format. A complete understanding of the performance limitations before building applications around SOAP is hence required.

SOAP is usually limited to pooling and not to event notifications when HTTP is used for the transport. In addition, only one [client](https://searchenterprisedesktop.techtarget.com/definition/client) can use the services of one server in typical situations.

If HTTP is used as the transport protocol, firewall latency usually occurs since the firewall analyzes the HTTP transport. This is because HTTP is also leveraged for Web browsing, and so many firewalls do not understand the difference between using HTTP within a web browser and using HTTP within SOAP.

SOAP has different support levels, depending on the supported programming language. For instance, SOAP supported in [Python](https://whatis.techtarget.com/definition/Python) and PHP is not as powerful as it is in [Java](https://www.theserverside.com/definition/Java) and [.NET](https://searchwindevelopment.techtarget.com/definition/NET).

**17. What are the differences between SOAP and REST?**

|  |  |
| --- | --- |
| **SOAP** | **REST** |
| **SOAP is a protocol. SOAP was designed with a specification.**  **It includes a WSDL file which has the required information on what the web service does in addition to the location of the web service.** | **REST is an Architectural style in which a web service can only be treated as a RESTful service if it follows the constraints of being:**   * **Client Server** * **Stateless** * **Cacheable** * **Layered System** * **Uniform Interface** |
| **SOAP cannot make use of REST since SOAP is a protocol and REST is an architectural pattern.** | **REST can make use of SOAP as the underlying protocol for web services, because in the end it is just an architectural pattern.** |
| **SOAP only permits the XML format.** | **REST permits many different data formats.** |
| **SOAP based reads cannot be cached.** | **REST reads can be cached.** |
| **SOAP is like custom desktop application, closely connected to the server.** | **A REST client is just like a browser and uses standard methods. An application has to fit inside it.** |
| **SOAP is slower than REST.** | **REST is faster than SOAP.** |
| **It runs on HTTP but envelopes the message.** | **It uses the HTTP headers to hold meta information.** |

**18. SOAP or Rest APIs, which method to use?**

SOAP is the heavyweight choice for Web service access. It provides the following advantages when compared to REST:

* SOAP is not very easy to implement and requires more bandwidth and resources.
* SOAP message request is processed slower as compared to REST and it does not use web caching mechanism.
* WS-Security: While SOAP supports SSL (just like REST) it also supports WS-Security which adds some enterprise security features.
* WS-AtomicTransaction: Need ACID Transactions over a service, you’re going to need SOAP.
* WS-ReliableMessaging: If your application needs Asynchronous processing and a guaranteed level of reliability and security. Rest doesn’t have a standard messaging system and expects clients to deal with communication failures by retrying.
* If the security is a major concern and the resources are not limited then we should use SOAP web services. Like if we are creating a web service for payment gateways, financial and telecommunication related work, then we should go with SOAP as here high security is needed.

REST is easier to use for the most part and is more flexible. It has the following advantages when compared to SOAP:

* Since REST uses standard HTTP, it is much simpler.
* REST is easier to implement, requires less bandwidth and resources.
* REST permits many different data formats whereas SOAP only permits XML.
* REST allows better support for browser clients due to its support for JSON.
* REST has better performance and scalability. REST reads can be cached, SOAP-based reads cannot be cached.
* If security is not a major concern and we have limited resources. Or we want to create an API that will be easily used by other developers publicly then we should go with REST.
* If we need Stateless CRUD operations then go with REST.
* REST is commonly used in social media, web chat, mobile services and Public APIs like Google Maps.
* RESTful service returns various MediaTypes for the same resource, depending on the request header parameter "Accept" as application/xml or application/json for POST and /user/1234.json or GET /user/1234.xml for GET.
* REST services are meant to be called by the client-side application and not the end user directly.
* ST in REST comes from State Transfer. You transfer the state around instead of having the server store it, this makes REST services scalable.

**19. What are the factors that help to decide which style of Web services — SOAP or REST — to use?**  
Generally, REST 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.

* Does the service expose data or business logic? REST is commonly used for exposing data while SOAP for logic.
* Requirement from clients or providers for a formal contract. SOAP can provide contract via WSDL.
* Support multiple data formats.
* Support for AJAX calls. REST can apply the XMLHttpRequest.
* Synchronous and asynchronous calls. SOAP enables both synchronous/ asynchronous operations whereas REST has built-in support for synchronous.
* Stateless or Stateful calls. REST is suited for stateless operations.
* Security. SOAP provides a high level of security.
* Transaction support. SOAP is good at transaction management.
* Limited bandwidth. SOAP has a lot of overhead when sending/receiving packets since it’s XML based, requires a SOAP header. However, REST requires less bandwidth to send requests to the server. Its messages are mostly built using JSON.
* Ease of use. REST-based application is easy to implement, test, and maintain.