What is the difference between SOAP and REST?

SOAP and REST are Internet Data Exchange Mechanisms. For example, imagine your internal accounting system shares data with a customer's accounting system in order to automate billing tasks. The two applications share data using an application programming interface (API) that defines communication rules. SOAP and REST are two different approaches to designing APIs. SOAP is very structured and uses XML data formatting. REST, on the other hand, is more flexible and allows applications to exchange data in multiple formats.

What are the Similarities Between SOAP and REST?

To create apps, you can use many different programming languages, designs, and platforms. It is difficult to share data between these various technologies because they have different data formats. SOAP and REST both came about in an attempt to solve this problem.

You can use SOAP and REST to create APIs, or connection points, between various applications. The term web service and the term API are used interchangeably. However, APIs are the broadest category. Web services are a special kind of API.

Here are other similarities between SOAP and REST:

• Both describe the rules and standards for how applications create, process, and respond to requests for data from other applications

• Both use HTTP, the Internet Protocol standard, to exchange information

• Both support SSL/TLS protocol for a secure and encrypted connection

You can use SOAP or REST to create secure, scalable, and fault-tolerant distributed systems.

How do SOAP APIs and REST APIs work?

SOAP is an old technology that requires strict communication between systems. New web service standards have been added over time to accommodate technological changes, but result in additional overheads. The development of REST came after the development of SOAP and thus it inherently solves many of its shortcomings. REST web services are also called RESTful web services.

SOAP API interfaces

SOAP is a protocol that defines strict rules for communication. It has many associated standards that control every aspect of data exchange. For example, here are some of the standards that SOAP uses:

• Web Services Security (WS-Security) defines security measures such as the use of unique identifiers called tokens

• Web Services (WS-Addressing) processing requires routing information to be included as metadata

• Reliable Messaging between Web Services (WS-ReliableMessaging) standardizes error handling in SOAP messages

• Web Services Description Language (WSDL) describes the scope and functionality of Web services that implement SOAP

When sending a request to a SOAP API, you must put the HTTP request in a SOAP envelope. It is a data structure that modifies the underlying HTTP content through the requirements of a SOAP request. Because of the envelope, you can also send requests to web services that implement SOAP with other transport protocols, such as TCP or Internet Control Message Protocol (ICMP). However, SOAP APIs and web services that implement the SOAP protocol always return XML documents in their responses.

REST APIs

REST is a software design style that imposes six conditions on how APIs should work. These are the six principles REST APIs adhere to:

1. Client-server architecture. The sender and the receiver are independent of each other in terms of technology, platform, programming language, etc.

2. Multi-layered. A server can have several brokers working together to complete clients' orders, but the brokers are not visible to the client.

3. Unified Interface. The API returns data in a complete, fully usable, standard format.

4. Stateless. The API completes each new request independently of previous requests.

5. Spoolable. All API responses are cacheable.

6. Code on demand. The API response can include a piece of code if needed.

You can send REST requests using HTTP verbs such as GET and POST. Rest API responses are usually in JSON format but can also be in other data format.