## REST VS SOAP

An Application Programming Interface (API) is a set of rules to facilitate communication between different programs. [1] APIs let the product or service communicate with other products and services without having to know how they are implemented.[2]

### REST

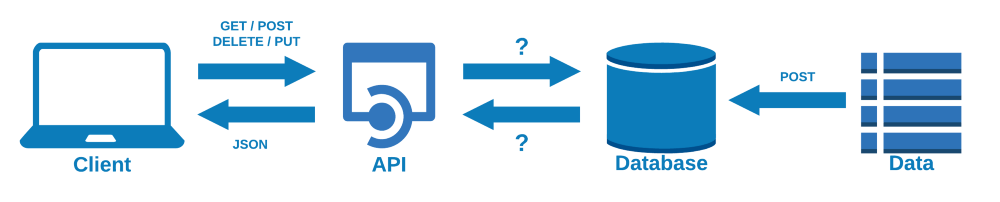
**Re**presentational **S**tate **T**ransfer (REST) is an architectural Style for designing networked applications. It is the most popular style for building web APIs. REST determines specifications of the API through a set of rules that are followed when a REST API is created.

APIs are RESTful as long as they comply with the 6 guiding constraints of a RESTful system [2]:

* **Client-server architecture:** REST architecture is composed of clients, servers, and resources, and it handles requests through HTTP.
* **Statelessness:** No client content is stored on the server between requests. Information about the session state is, instead, held with the client.
* **Cacheability:** Caching can eliminate the need for some client-server interactions.
* **Layered system:** Client-server interactions can be mediated by additional layers. These layers could offer additional features like load balancing, shared caches, or security.
* **Code on demand (optional):** Servers can extend the functionality of a client by transferring executable code.
* **Uniform interface:** This constraint is core to the design of RESTful APIs and includes 4 facets:
* **Resource identification in requests**: Resources are identified in requests and are separate from the representations returned to the client.
* **Resource manipulation through representations**: Clients receive files that represent resources. These representations must have enough information to allow modification or deletion.
* **Self-descriptive messages**: Each message returned to a client contains enough information to describe how the client should process the information.
* **Hypermedia as the engine of application state**: After accessing a resource, the REST client should be able to discover through hyperlinks all other actions that are currently available.

REST treats any data as a *resource*that the client can fetch/edit/delete. REST mandates that a client should be able to perform the appropriate operation by accessing a specific URL and sending a *request*. The server then sends an appropriate *response*.

REST is stateless, which implies that each request from the client must have all the necessary information for the server to understand it. For example, the client cannot assume that the server remembers what they had asked for earlier. With REST APIs, we usually send HTTP requests such as GET , PUT or POST[1]



*Picture show the main framework of REST API in which the client-side of the API allows for data to be added and removed using GET, POST, DELETE and PUT methods.[3]*

### SOAP

Simple Object Access Protocol, more known as SOAP APIs designed with SOAP use XML for their message format and receives requests through HTTP or SMTP. SOAP makes it easier for apps running in different environments or written in different languages to share information.[2]

SOAP is a W3C standard whose main purpose was to replace more specific communication protocols (RPC), the use of which may be restricted by firewalls or other security measures.

Example of the SOAP document [6]:

POST /InStock HTTP/1.1

Host: www.example.org

Content-Type: application/soap+xml; charset=utf-8

Content-Length: nnn

<?xml version="1.0"?>

**<soap:Envelope**

xmlns:soap="http://www.w3.org/2001/12/soap-envelope"

soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding"**>**

**<soap:Body** xmlns:m="http://www.example.org/stock"**>**

**<m:GetStockPrice>**

**<m:StockName>**IBM**</m:StockName>**

**</m:GetStockPrice>**

**</soap:Body>**

**</soap:Envelope>**

A SOAP message is an ordinary XML document containing the following elements [5]:

* An Envelope element that identifies the XML document as a SOAP message
* A Header element that contains header information
* A Body element that contains call and response information
* A Fault element containing errors and status information

All the elements above are declared in the default namespace for the SOAP envelope:

<http://www.w3.org/2003/05/soap-envelope/>

and the default namespace for SOAP encoding and data types is:

<http://www.w3.org/2003/05/soap-encoding>

Here are some important syntax rules:

* A SOAP message MUST be encoded using XML
* A SOAP message MUST use the SOAP Envelope namespace
* A SOAP message must NOT contain a DTD reference
* A SOAP message must NOT contain XML Processing Instructions

### REST vs SOAP

**Soap Advantages**  
SOAP provides the following advantages when compared to REST:  
•    Language, platform, and transport independent (REST requires use of HTTP)  
•    Works well in distributed enterprise environments (REST assumes direct point-to-point communication)  
•    Standardized  
•    Provides significant pre-build extensibility in the form of the WS\* standards  
•    Built-in error handling  
•    Automation when used with certain language products

**REST Advantages**  
REST is easier to use for the most part and is more flexible. It has the following advantages over SOAP:  
•    No expensive tools require to interact with the web service  
•    Smaller learning curve  
•    Efficient (SOAP uses XML for all messages, REST can use smaller message formats)  
•    Fast (no extensive processing required)  
•    Closer to other web technologies in design philosophy

## REFERENCES:

[1] Introduction to REST APIs; <https://towardsdatascience.com/introduction-to-rest-apis-90b5d9676004>; Dec 13, 2020;

[2] What is an API?; <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>

# [3] Launch your own REST API using Flask & Python in 7 minutes; <https://towardsdatascience.com/launch-your-own-rest-api-using-flask-python-in-7-minutes-c4373eb34239>; Jun 15, 2020

# [4] REST vs SOAP; <https://www.redhat.com/en/topics/integration/whats-the-difference-between-soap-rest>

# [5] XML Soap; <https://www.w3schools.com/xml/xml_soap.asp>

# [6] SOAP; https://pl.wikipedia.org/wiki/SOAP

## [7] Deciding Between SOAP and REST https://smartbear.com/blog/test-and-monitor/soap-vs-rest-whats-the-difference/