# SOAP – Simple Object Access Protocol

SOAP is an acronym for Simple Object Access Protocol. It is an XML-based messaging

Protocol for exchanging information among computers. SOAP is an application of the XML specification.

SOAP defines the XML-based message format that Web service-enabled applications use to communicate and inter-operate with each other over the Web. The heterogeneous environment of the Web demands that applications support a common data encoding protocol and message format. SOAP is a standard for encoding messages in XML that invoke functions in other applications.

* SOAP is a communication protocol designed to communicate via Internet.
* SOAP can extend HTTP for XML messaging.
* SOAP provides date transport for Web services.
* SOAP can exchange complete documents or call a remote procedure.
* SOAP can be used for broadcasting a message.
* SOAP is platform and language independent.
* SOAP is the XML way of defining what information is sent and how.
* SOAP enables client applications to easily connect to remote services and invoke remote methods.

Although SOAP can be used in a variety of messaging systems and can be delivered via a variety of transport protocols, the initial focus of SOAP is remote procedure calls transported via HTTP.

Other frameworks including CORBA, DCOM, and Java RMI provide similar functionality to SOAP, but SOAP messages are written entirely in XML and are therefore uniquely platform- and language-independent.

# REST - **Re**presentational **S**tate **T**ransfer

REST stands for Representational State Transfer. REST is a web standards based architecture and uses HTTP Protocol for data communication. It revolves around resources 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 year 2000.

In REST architecture, a REST Server simply provides access to resources and the 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. JSON is now the most popular format being used in Web Services.

# Difference between SOAP and REST

SOAP-

* It is a XML based message Protocol.
* Uses WSDL for communication between consumer and provider.
* Invokes services by calling RPC method.
* Does not return Human readable result.
* Transfer over HTTP. Also uses other protocols like SMTP, FTP etc.
* JavaScript can call SOAP, but it is difficult to implement.
* Performance is not great compare to REST.

REST-

* An Architectural style protocol.
* Uses XML or JSON to send and receive data.
* Simply call services via URL Path.
* Result is readable, which is in just plain XML or JSON
* Transfer over HTTP only.
* Easy to call from JavaScript.
* Performance is much better compare to SOAP – less CPU intensive, leaner code etc.

# WSDL – Web Services Description Language.

It is the standard format for describing a web service. WSDL was developed jointly by Microsoft and IBM. WSDL is an XML-based protocol for information exchange in decentralized and distributed environments.

WSDL is often used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. Any special datatypes used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL.

# Different Set of Parameters under SOAPUI

* QUERY

Query is the most common parameter type. It is specified at the end of the URL after the question mark (?).

* HEADER

Header parameters are added as part of the HTTP header of a request.

* MATRIX

Matrix Parameters are added to the resource path before the query string.

* PLAIN

Plain parameters are defined in a request and can be accessed in Ready API, but are omitted when the request is sent.

* TEMPLATE

Template parameters are used to parametrize the request path.

# Types of Request

* [GET](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/GET)

The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.

* [HEAD](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/HEAD)

The HEAD method asks for a response identical to that of a GET request, but without the response body.

* [POST](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/POST)

The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server

* [PUT](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/PUT)

The PUT method replaces all current representations of the target resource with the request payload.

* [DELETE](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/DELETE)

The DELETE method deletes the specified resource.

* [CONNECT](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/CONNECT)

The CONNECT method establishes a tunnel to the server identified by the target resource.

* [OPTIONS](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/OPTIONS)

The OPTIONS method is used to describe the communication options for the target resource.

* [TRACE](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/TRACE)

The TRACE method performs a message loop-back test along the path to the target resource.

* [PATCH](https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/PATCH)

The PATCH method is used to apply partial modifications to a resource