An API (Application Programming Interface) is the means by which third parties can write code that interfaces with other code.

A Web Service is a type of API, one that almost always operates over HTTP (though some, like SOAP, can use alternate transports, like SMTP).

The [official W3C definition](http://www.w3.org/TR/ws-gloss/)mentions that Web Services don't necessarily use HTTP, but this is almost always the case and is usually assumed unless mentioned otherwise.

For examples of web services specifically, see [SOAP](http://en.wikipedia.org/wiki/SOAP), [REST](http://en.wikipedia.org/wiki/REST), and [XML-RPC](http://en.wikipedia.org/wiki/XML-RPC).

For an example of another type of API, one written in C for use on a local machine, see the [Linux Kernel API](http://kernelbook.sourceforge.net/kernel-api.html/).

As far as the protocol goes, a Web service API almost always uses HTTP (hence the Web part), and definitely involves communication over a network. APIs in general can use any means of communication they wish. The Linux kernel API, for example, uses [Interrupts](http://en.wikipedia.org/wiki/Interrupt) to invoke the system calls that comprise its API for calls from user space.

Soap Vs Rest

SOAP is definitely the heavyweight choice for Web service access. It 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(ws-security)
* Built-in error handling
* Xml to transfer data…

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

* No expensive tools require to interact with the Web service
* Smaller learning curve
* Uses xml/json/plain text files…
* 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
* Rest uses http methods- crud operations- post,get, put, delete

Webservices

Soap

Rest

SoapUI- is a tool to test/automate testing of web services. We can automate both soap and rest webservices testing using soapui

Apple

Macys

Walmart

Google

Talentscreen.io

Webservices - Internal/external

What do you need to Test a webservice:

Specifications document:

Urls for webservices

Request parameters

Sample request / sample response

What type of protocol

What type request data

Webservices- soap vs rest

Order-management:

Api.macys.com/orders? ordereId=5678758

userLogin

createOrders

sendSMS/EmailToUser(orderDetails)

getOrders

deleteOrders

updateOrders

payment – 3rd party vendors- webservices – paypal…

shipping -3rd party vendors- webservices-ups…

What is endpoint -

What are operations

getCities

getWeather

**What to test:**

FunctionalTesting:

Different Operations

endpoint Testing :

<http://www.webservicex.com/globalweather.asmx>

dev,test,stage,prod

dev sit uat prod

state testing:

valid/invalid scenarios

Load/Performance Testing:

SecurityTesting:

How to test:

statusCodes – 200 – success related, 201- resource successfully Created, 400 - clientSide, 500-server-side

SchemaCompliance -

ContentType – json/xml/raw text file

specificElementIsPresent or not

Element Values

Count Of elements

OrderOfElement

Content-length

Header information

Groovy- is superset of java, it is used in soapUI tool to do logical verifications on your request or response….

Dependency webservice testing – one webservice output shud be input for next webservice/only if first webservice response is succes then only execute the second one

Create User

Verify db for the new data- data integrity/data mapping

sendSmsFordetails

saveDataToDataBase

Get Users

SoapMockService - mock response

Using the mock response:

Test cases

You can add your test steps

assertions

When to use:

TestDriven approach

When webservice is not fully functional-developers are still working on few operations but you have WSDL - operations&porttypes

Webservice is down- u want to do a smoke testing

Client based testing

LoadTesting/PerformanceTesting - concurrent requests /load to check the performance

Response time

Tps- threads per second

Bytespersecond

How many max threads application supports

SecurityTesting – make sure that application is preventing the vulnarabilities/extrenalattacks/hacking

Crosssitescripting

Sql injection

Malformed xml

Boundary scan – I> 2000

Thread- runs the tasks/processes parallely

10 – 0.5 ms

0.5 – 700ms

* 1. 1-500th ms-
  2. 1.5

200ms- delay , 0.5-random

Random\*testDelay

Delay=(TestDelay- any randomnumber(0,100))

TestDelay

Soap:

soap is a protocol

access request and response as per wsdl declarations.

<soap-envelope>

<soap-header>

<soap-bdy>

<ns:weatherservice>

<ns:cityName>

<ns:country>

Data length is large…

Always we need to use xml  
Soap uses http,ftp,smtp

Soap uses get&post

Always soap uses same endpoint url across multiple operations.

Soap header/ws-security are used for secure data…

REST:

REST is an architecture.

It only uses http protocol.

It can transfer data by using xml/json/plaintext/csv

For different operations- we use different http methods

CRUD operations:

Create - post

Update - put

Delete - delete

Retrieve – get

Rest is light weight

Rest uses – http headers for auth data

Rest can use any http methods- put/post/delete/get/head/trace…

Rest url has enpoint and resources/operations

<http://api.macys.com/order-management/createOrder>

<http://api.macys.com/order-management/deletOrder>

<http://api.macys.com/order-management/2345>

Rest – representational state transfer

Resource should be a noun not verb

UI pro:

DataDriven testing – dataSourceStep

dataSourceLoop