**WebService**: It is a piece of software(program) that is being called by another software/program

over the internet by using the XML based communication.

**API**: It acts as an interface between two programs or two systems. API can be web based, library based,

System based.

A webservice is a kind of API but not all API’s need to be webservices.

**SoapWebservice**:

Soap is platform, language and transport independent

Soap is a standard. It is based on WSDL definitions.

Soap uses XML based communication.

Soap has inbuilt security ws security.

It is heavyweight choice.

**REST APIs:**

REST is language and platform independent. It depends on http protocol.

REST is architecture. It is designed by using http methods for CRUD operations.

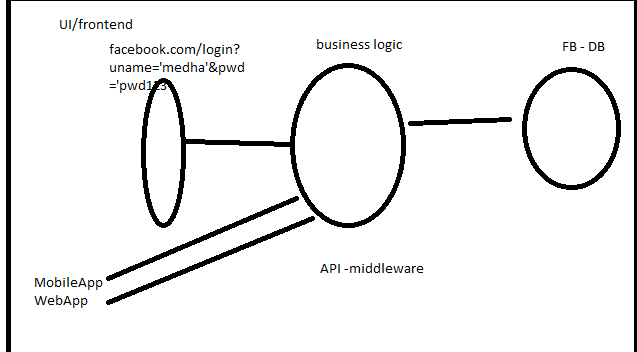
It can communicate through either Xml/JSON/csv/plain text.

It is a light weight choice.

It is easy to learn and understand.

Eg: <http://api.talentscreen.io/candidates?id=456>

Response: candidate name, phone, mail, subject, assignment.



**How to test webservices:**

Tools like **SoapUI**

Plugins like Postman-chrome,RestClient-Mozilla

Frameworks using HttpClient, RestAssured

SoapUI – It is a tool to test webservices. We can test both Soap and Rest webservices using SoapUI tool.

CRUD

Create

Retrieve

Update

Delete

Online Shopping Cart:

Create order id with order details given by customer

Retrieve/get the order details by using orderid

Update the details of exisiting order

Delete the order with order id

StudentPortal:

Create student details with id

Get

Update

Delete

Soap webservice:

Cretae new soap project

Give wsdl url or actual wsdl file

WSDL- web service definition language, it is an xml file with information about webservice.

Types

Messages

Porttype

binding

service – details about endpoint

<soapenv:envelope xmlns:ns1=<http://predic8.com/wsdl/shop/1/>>

<soapenv:header></soapenv:header>

<soapenv:body>

<ns:addToCart>

<cartId>

<articleId>

<quantity>

</ns:addToCart>

</soapenv:body>

</soapenv:envelope>

Response:

Header – statuscode, response message

Content type

Content length

Server

X access level

Payload/Body:

What to test in a webservice:

Functionality Testing:

Test important header information like statuscode, response message, content type, content length

Test for schema compliance(whether response is as per the WSDL standard)

Test for actual response body/payload:

Element present

Element value

Data count

Data order

Data types

EndPoint Testing

SecurityTesting:

AuthenticationTesting

SQL Injection attacks

XSS attacks

PerformanceTesting

TestData: Valid ,invalid, boundary values…

Success – get- 200

Bad request – get- 400

Authorisation-401- unauthorized

Server side- 500- internal server error

SoapUItool assertions:

Contains

NotContains

SchemaCompliance

Valid http status codes

inValid http status codes

xpath assertion

json path assertions

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Xpath- xml path language which helps to locate elements inside xml by traversing through nodes of xml.

What information is needed before you start testing webservices:

1.Specification Documents/wiki links which describe the functionality of webserivce

This is given by test lead/development team

2. Go through the document to see what are required parameters/ optional.

3. Get more information about datatypes and request /response details by observing WSDL and also Specification documents

4. Have a plan on testcases – expected data – execute DB queries to know expected data in some cases

5. If possible get sample request and response from team if specifications do not have one which helps for assertions.

SoapUITool steps to automate:

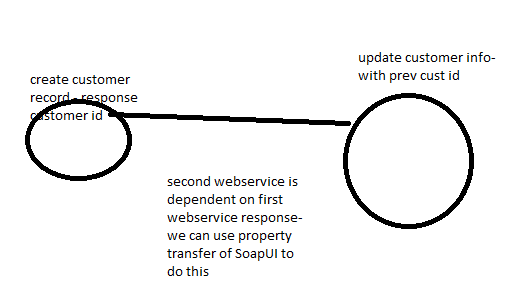
TestSuite->TestCases->TestSteps

1. Request(Soap/Rest/JDBC)
2. Properties – to load external file properties
3. Property Transfer – to transfer data from one step to other step

eg: transfer data from properties step to request step

4.ManualTestStep- to test any manual testcases and set pass or fail in tool

5.ConditionalGoto- like if condition- used when we want to execute any testcase based on Previous result.



We can also transfer data from one response to other request in case of dependent webservices.

Eg: first execute login request and get session id in reponse, which should be used for next request

First create an order with order details and use the order id from response to get the orderdetails

First create(post) a customer with customer details and use the customer id from response to get/update/delete the customer details.

Array – group of similar data

Eg: list of customers

List of student id’s

count(//ns2:getAllResponse/customer) – count helps to know number of elements when its array

Index for array of elements is starting from 1.

Exists function helps to check if a specific element is present or not

Groovy is a scripting language which SoapUI tool supports internally to perform any

further actions which we could not do using in built test steps.

Get response from previous teststep and extract data using groovy

https://github.com/SmartBear/soapui-groovy-examples

http://www.soapui.org/apidocs/index.html

def sessionId = testRunner.testCase.getTestStepByName( “loginRequest” )

If(sessionId!=null){

testRunner.runTestStepByName(“checkout”);

}else{

Log.info(“login is not proper”);

testRunner.runTestStepByName(“loginRequest”);

}

def userIdStr = userIdInStep.getPropertyValue( “loginId” );

userIdInStep.setPropertyValue(“loginId”, “StringValueAsInput”)

create order id

verify the database if details are properly inserted as per request

JDBC- helps to connect to database programitically

MockService: creating Dummy response using the tool and proceed with further testing.

When the webservice is not fully functional in TDD

Helps to figure out issues in early stages

Webservice is down

LoadTest

SecurityTest:

BoubdaryScan- id- 1-100

Date-