**SOA -Service Oriented Architecture**

Every information is provided as a Service.

What is Service?

Web Application : GUI. allows interaction between user and computer program.

Web Service : allows interaction between Application program to another Application program.

xml(Xtensible Markup Language) -based web Services interact between applications in any kind of programming application.

Html **: Hyper Text Markup Language**

**a presentation language**

**predefined tags or elements**

Html is static, it can't support the dynamic content i.e. it fails when it comes to the data transfer.

XML **: Data Transfer Language**

**No pre-defined tags – user’s own tags or elements**

XML is used for web services communication between data.

**independent platform**

**follows a standard format**

**Components of SOA**

**1. Web Services Provider:** returns response to the client (web services consumer)

**2. Web Services Consumer** - an application, web service consuming application- it may be a GUI or a non GUI application

**3. UDDI -Universal Description Discovery and Integration.**

**UDDI:** registry, an information about who provides which Web Service

**4. WSDL** -Web Services Description Language

written in Xml.

probably provided as a file with extension as .wsdl or provided as an http link.

WSDL Contents are Web Services:

Type

Message (operations performed by this web service)

port type

Bindings (will be SOAP protocol)

5. **SOAP**- Simple Object Access Protocol

SOAP - protocol, a set of rules followed by web service communication.

SOAP provides a standard format and rules for exchanging information through web services.

Structure of SOAP message/ SOAP format

-Envelope

Header

-Body

Message

Fault (optional)

Flow of User interaction with Web Services provider:

-User uses some web service consuming application (Web App)

-Web Services consumer sends a request to UDDI to locate who is providing the web service

-UDDI passes on the request to Web Services providers.

-Web Services provider returns the response to the client (Web Services consumer)

What are the minimum things required to start testing web services with SOAPUI tool?

Ans: web services specification document or a valid WSDL file or a Http link.

I have tested Airline Reservation web service

-SearchTicket

-BookTicket

-BuyTicket

-ModifyTicket

-CancelTicket

Assertion: validate web services response using assertions

What were the messages or operations that you tested in that web service?

Common Assertion:

SOAP Response

SOAP Fault.

Not SOAP Fault.

Schema Compliance.

Contains.

Not contains.

Xpath.

**Xpath: express / navigate the value of a particular element/ node of Xml**

Property transfer Scenario

-----------------------------------

TC1 ( Schedule payment)

Req 1 -- > Schedule this amt $1000 as payment

Resp 1 -- > ScheduleID of particular scheduled transaction-scheduledID(4653330)

TC2 -- > (Modify scheduled payment )

Req (get scheduleID(4653330) from Resp1)

Types of Property Transfer

--------------------------------

TR to another TR ( can't be done directly)

TR to TC

TC to another TC (can't be done directly)

TC to TS

TS to another TS(can't be done directly) ( TS --- > TC --- > another TS)

TS to TC

TS to TR

TR to TS

Property Expansion Formula/ property expansion expression:

${StepName#PropertyName}

e.g, ${DataSource#From} ${DataSource#To}

Why SOAP protocol is used for web services but not Http?

Ans: SOAP protocol is a standardized protocol which has some special wrappers that are not present in Http. SOAP messages are not easily blocked by firewalls.

6. REST - REpresentational State Transfer

It is a kind of architectural pattern/design.

Everything is represented in the form of URL.

College

-departments

-Chemistry

-Chem100

-Chem102

-Physics

-Ph22

-Ph33

-Ph44

URL will be College/departments/Chemistry/Physics/Ph22

REST

Logical steps that we need to follow to perform RESTFUL web services / API testing

**Testing of RESTFUL APIs**

1. Create a REST service (u can do that as u will get an API link from developer)

2. Create a resource

3. Locate the resource

4. Perform operation on resource using GET, POST, PUT or DELETE

RESTFUL response can be in multiple format- XML, JSON

**JSON is preferred format as it is lighter than XML - lighter in the sense that it does not bring back the unnecessary elements like header etc, which we might not need.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Diff-SOAP-REST**

REST

**REST is exposing a public API over the internet to handle CRUD operations on data.**

REST is focused on accessing named resources through a single consistent interface.

Restful Web services uses HTTP protocol

**HTTP uses 4 methods:- GET, POST, PUT, DELETE**

HTTP CRUD Equivalent

==============================

GET read

POST create,update,delete

PUT create,update

DELETE delete



**Http Method**

|  |  |
| --- | --- |
| **S.N.** | **Method and Description** |
| 1 | **GET** The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data. |
| 2 | **HEAD** Same as GET, but only transfer the status line and header section. |
| 3 | **POST** A POST request is used to send data to the server, for example customer information, file upload etc using HTML forms. |
| 4 | **PUT** Replace all current representations of the target resource with the uploaded content. |
| 5 | **DELETE** Remove all current representations of the target resource given by URI. |
| 6 | **CONNECT** Establish a tunnel to the server identified by a given URI. |
| 7 | **OPTIONS** Describe the communication options for the target resource. |
| 8 | **TRACE** Perform a message loop-back test along the path to the target resource. |

**\_\_\_\_\_\_\_\_\_\_\_\_\_**

SOAP

SOAP brings its own protocol and focuses on exposing pieces of application logic (not data) as services.

SOAP exposes operations.

SOAP is focused on accessing named operations, each implement some business logic through different interfaces.

SOAP-based web services uses SOAP Protocol.

------------------

SOAP and RESTful web services have a very different philosophy from each other.

SOAP is really a protocol for XML-based distributed computing, whereas REST adheres much more closely to a bare metal, web-based design. SOAP by itself is not that complex; it can get complex, however, when it is used with its numerous extensions (guilt by association).

To summarize their strengths and weaknesses:

\*\*\* SOAP \*\*\*

Pros:

Language, platform, and transport agnostic

Designed to handle distributed computing environments

Is the prevailing standard for web services, and hence has better support from other standards (WSDL, WS-\*) and tooling from vendors

Built-in error handling (faults)

Extensibility

Cons:

Conceptually more difficult, more "heavy-weight" than REST

More verbose

Harder to develop, requires tools

\*\*\* REST \*\*\*

Pros:

Language and platform agnostic

Much simpler to develop than SOAP

Small learning curve, less reliance on tools

Concise, no need for additional messaging layer

Closer in design and philosophy to the Web

REST does not require XML parsing and does not require a message header to and from a service provider. This ultimately uses less bandwidth

Cons:

Assumes a point-to-point communication model--not usable for distributed computing environment

where message may go through one or more intermediaries

Lack of standards support for security, policy, reliable messaging, etc.,

so services that have more sophisticated requirements are harder to develop ("roll your own")

Tied to the HTTP transport model

\_\_\_\_\_\_\_\_\_\_\_\_\_