WebServices

==========

=>WebServices is a mechanism of linking two apps as consumer and producer apps using the protocol http.

=> WebSerevices platform independent and archicture independent and language independent..

A producer developed in "Java" can be used/consumed in

.net, java, php, phyton, Java script and etc.. (Even reverse possible)

=> WebServices says Develop/produce any where and consume any where.

=> Webservices makes the consumer and producer Apps exchanging data either in XML or in JSON (Global formats)

=> Java and languages have provided multiple apis/Technologies and frameworks to implement web service logics as skelton logics (Integration Logics Producer) and as stub logics (Integration logics of consumer)

=>WebServices support http request and http response based method invocation i.e as http request the cosumer App invokes method of producer App and the results method execution goes back to consumer App as http response.

Two ways of implementing web services

a) SOAP WebServices

b) RESTfull WebServices

SOAP :: Simple object Access protocol (protocol) REST :: REpresentational STatefull

web services (It is not a protocol

it is kind of architecture/ mechanism)

a) SOAP WebServices

on

=> This runs based 3 component priciple a) Serivce Provider(producer /Server) b) Sevice Consumer (Consumer/Client)

c) Service Registry (where services (Provider App) will be registered to

The Registries in SOAP based webServices env..

is UDDI (Universal Description Discovery and Integration)

SOAP based web services

fall under SOA Archicteture

(Service Orienteted Architecture)

=>SOA is the mechanism of linking two (find)

Apps

contains

=> SOA service producer,

Service consumer and

Service registry (4)

as shown in the diagram

Service

Registry (UDDI) (WSDL docs)

(2) (publish) register

lookup (3)

wsdl doc

K SOAP over http request Producer (skelton) (stub) (5) Bind/Invoke (Service Provider) SOAP over http response

Cosumer (Serivce consumer)

is

SOAP message strictly typed or complex XML messages which will be conveted java obejcts and reverse using Jaxb api

jaxb :: Java archicture for xml binding

(nothing but converting java object data to xml content and vice-versa)

=>The consumer sends inputs to =>The Poducer sends outputs to

to expose)

For every SOAP web service

that is registered in service Registry there will be one WSDL doc (xml document) having multiple details about web service like how to consume, the services names and etc (endpoint details)

WSDL :: WebService Description Language

=> The http request carries xml baased SOAP 'message but as request body nothing inputs

=> The http response carries xml baased SOAP message as response body nothing but outputs

producer as SOAP message(xml) in http request cosumer as SOAP message(xml) in http response

=>The http request and http response contains two parts

a) HEAD part

|---> contains two sections

b) Body part

a) Initial Line

b) headers

Structure of http request

======

======

Initial line

http method path protocol & version

request headers

user-agent: chrome

http methods or modes are (9)

GET

POST

HEAD

TRACE

HEAD

accept: text/html, text/plain,... accept-language: en-US

PUT

Body

HEAD

>>>> blank line >>>> payload/body]

query params or SOAP message

Requests

```
POST/Wish
HTTP/1.1
Host: localhost:8000
```

DELETE

OPTIONS

CONNECT (reserved for future)

PATCH (for partial Updates)

In Normal http requeust, the request body will be query String having request param names and values like sno=101&sname=raja&sadd=hyd In soap over http request, the request body will be xml based SOAP message (xml tags)

POST mode

(Normal http request)

```
User-Agent: Mozilla/5.0 (Macintosh;) Firefox/51.0 Accept:
text/html,application/xhtml+xml,,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

initial line/request line

request headers

BODY

```
Content-Type: multipart/form-data; boundary=-12656974 Content-Length: 345
```

sno=101&sname=raja&sadd=hyd

blank line request body/ pay load (form data)

contain

=>POST mode reuqest contain request body representing query String where as GET Mode request does not request body becoz its carries inputs as query String appended to the request url.

HEAD

The HTTP Method

Normal http request GET mode

Path to the source

Parameters to the server

on Web Server

Protocol Version Browser supports

request url

stur

GET /RegisterStudent.asp?user=jhon&pass=java HTTP/1.1

Host: guru99.com

User-Agent: Mozilla/5.0

path with query String

Accept-text/xml,text/html,text/plain, image/jpeg

Accept-Language: en-us,en

The Request

Headers

Accept-Encoding: gzip,deflate

Accept-charset: ISO-8859-1, utf-8

Keep-Alive: 300

Connection: keep-alive

Http response structure

protocol

response line /initial line

& version

response

status code and message

response haders

server :: apache Tomcat

host:

...

contentType: ... contentLength:

>>> blank line >>>>>

[response body/payload

BODY

web comp results or SOAP message

soap over http request

=====

POST http://127.0.0.1:8088/mockServiceSoapBinding HTTP/1.1 request line

Accept-Encoding: gzip,deflate

request line

Only HEAD part is available in http request i.e no body part is available in GET mode request

Payload = body

Normal Http response

==========

HTTP/1.1 200 OK

(initial line/response line) Status Line

Date: Thu, 20 May 2004 21:12:58 GMT Connection: close

Server: Apache/1.3.27

Accept-Ranges: bytes

Content-Type: text/html

response headers

Content-Length: 170

General Headers Response Headers Entity Headers

Last-Modified: Tue, 18 May 2004 10:14:49 GMT

blank line

<html>

<head>

<title>Welcome to the Amazing Site!</title>

</head>

<body>

response

Message Body

This site is under construction. Please come

back later. Sorry!

body/pay load

</body>

</html>

Normal http response contains the web comp genertad ouput (generally html code

or plain text) as response body where as soap over http response contains

the soap message(xml) as the response body

Content-Type: text/xml; charset=UTF-8

HEAD

req

SOAPAction: "http://www.soapui.org/sample/login"

Content-Length:505

headers Host: 127.0.0.1:8088

Connection: Keep-Alive

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

| blank line <soapenv:Envelope xmlns:sam="http://www.soapui.org/sample/" xmlns:soap soapenv:Header> <wsse:Security xmlns:wsse="http://docs.oasis-open.org/v

BODY

soap

soapenv:Bo sam:login>

v:Body>

<username>Loginn0.196</username>

message password>Loginn123</password>

</sam:login> </soapenv:Body>

as

req bedysoapenv:Envelope>

SOAP over http response

HTTP/1.1 200 OK

response line

X-Powered-By: Servlet/2.5

HEADServer: Sun GlassFish Enterprise Server v2.1.1

Content-Type: text/xml; charset="utf-8"

Transfer-Encoding: chunked

Date: Wed, 05 May 2010 21:10:14 GMT

blank line

response

HTTP Headers

<?xml version="1.0"?>

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">

<S:Body>

<ns2:sayHelloResponse xmlns:ns2="http://soap/">

<return>Hello, Shawn!</return>

</ns2:sayHelloResponse>

</S:Body> </S:Envelope>

Http response status codes

1xx (100-199) :: Informational 2xx (200-299) :: Success

3xx (300-399) :: Redirection

4xx (400-499) :: Incomplete (Client side errors) 5xx (500-599) :: Server side errors

Message Body (soap message

as response body)

In java env... jax-ws is api or technology (part of JSE module) that is given

to developing SOAP based web services and the frameworks implementing SOAP based web services are a) apache axis b) apache cfx and etc..

Limitations of SOAP based web services

======

==========

(Service Oriented Architecture)

a) These are based on SOA where seperate registry is required

to publish/find web services..

to

b) The service registries in SOA are very heavy maintain (UDDI registries)

c) The SOAP messages in http request and http response are complex to build

and heavy to send over the network (SOAP based Xml messages are heavy)

d) Understanding about web service by reading WSDL document is not an easy process.

e) SOAP message use complex Schema .. So building SOAP message is costly give

very

f) Does not flexiblity to send data in different global formats.. should always use

the heavy SOAP message as xml messages. (Allows as Global Data formats

g) XML is becoming outdated day to day as the

and etc..

to exchange the data b/w the Apps)

Global Data format

h) The communication b/w consumer to provider App is uni-directional i.e only one way

To overcome these problems use RestFull webServices.

b) RESTfull WebServices

=>GET is idempotent (safe to repeate the request)

to

=> POST is not idempotent (not safe repeat the request)

=> Idempotent means if we repeat the same request for multiple times then same result should come always (safe to repeat the request) => Non Idempotent means if we repeat the same request for multiple times then different times different results will come (not safe to repeat the request)

SOA:: Service Orienter Architecture

DELETE

=> GET,HEAD,OPTIONS,TRACE,PUT mode requests are idempotent => POST, PATCH mode are requests are not idempotent

note:: PUT looks like idem potent in most

of the scenarios becoz if try to update the item

that is already been updated then no updation takes place

=> For adding new web comp /replacing existing comp totally we use PUT mode request

=> For replacing existing web comp totally or partially we use PATCH mode request

communication is possible, the otherway is not possible (Consumer App can invoke the services

of Producer App)

=>REST is not a protocol.. REST is the mechasim or archicteture Representing the State of web comp in http request and in http response

So it is called RESTfull web services.

REST means REpresentationl State Tranfer (data)

=> Allows to send data in both global formats XML, JSON (directly as http request body and also as http response body)

XML:: eXtensible Markup language

(No protocol is involved while constructing this data) (schema)

To convert Xml tags data to java object (unmarshalling) and reverse (marshalling)

we use JaxB API (Java architecture for Xml binding)

but

JSON :: Java Scripot Object Notation (nothing Object in Java script)

The process of converting Java object to JSON content (key: value pairs)

is called Serialization and rerverse called DeSerialization

java data representation

String name="raja";

int age=30;

java script data representation

var/let name="raja";

Customer cust=new Customer();

var/let age=30;

var/let cust= { "cno": 1001,

cust.setCno(1001);

REstful web services is all about two operations

a) Identifying every services /method /operation of provider App with its url and consumer uses those urls to consume services of the Provider App..

No Service Registry is required (Every B.method/rest operation/endpoint of Restful Service /Rest API is identified

with 4 details a) url b) request path c) request mode d) params)

b)No SOAP protocol is involved becoz the consumer and provider sends and receive the as the XML/JSON data with out applying any porotocols

HEAD

BODY

cust.setCname("raja");

cust.setBillAmt(90.77);

(Java object)

"cname":"raja, "billAmt": 90.77

{...} is called document is JSON

[...] represents array in JSON

}

JSON doc means obj in Java script

To convert JSON content into Java Object and for reverse operation

we use JSON APIs

eg: JACKASON, JSON-B, GJSON and etc..

cust (java obj)

cno:1001 cname:raja billamt:90.77

serialization

K JACSON/ JSON-B GJSON API App

Deserialization

JSON Document

"cno": 1001, "cname":"raja, "billAmt": 90.77

=> JSON is ligght weight compare to XML to represent the data ..

=> In 10 streams the process of converting Java object data into bits and bytes i called Serialization and reverse called Deserialization For this Serialization, we need Serializable Java obj (The object whose class implements java.io.Serializable(I))

=> For the Serialization of JSON env.. we do not need any Serializable object JSON Serialization (nothin but converting Java object data into JSON document) any Normal Java Object is sufficient.

=>For IO streams serialization, we need Serializable object =>For JSON serialization,we can take any normal java object

xml doc

========

<customers>

<customer>

<cno>101</cno>

<cname>raja </cname>

<billAmt>90.77 </billAmt>

</customer>

and also to describe the data

JSON Doc

{ "cno": 1001,

"cname":"raja,

"billAmt": 90.77

}

note::10 Streams Serialization is totally different from the JSON Serialization

</customers>

=>here compare to data,.. the structure

is more like closing tag for every opening tag is bit extra

=>Gives structured data having relationship among the data memebers in bit heavy env..

webservices

=>Here data and structure both are

in good combination

=>Gives structured data having relationship among the data memebers in light env..

=> RESTfull is not based on SOA (SErvice oriented archicture) .. It is based on Consumer Producure Archicteture with FrontController support i.e no service registry is required here..

=>SOAP webservices archictecture is SOA (Service Oriented Architecture)

like

=> It is students getting training from trainer by accessing the trainer details from Inistitute

=> RESTfull web services is Consumer-Producer Architecture App and it is like Student and trainerfect interaction

global path

http request

service consumer (stub logics) [uses

with Xml/JSON data http response with XML/JSON data

=> every restfull service(producer) Service Producer

is a web comp and it is identified

request URL]

(skelton logics) with its request url

=> In RESTfull web services, we prefer JSON data alot compare to xml data..

=> The service Producer (Restfull comp) can be developed in any language like java,.net, php, phyton and etc..

=> The service consumer (Client) can be developed in any language like

taken

java,.net, php, phyton and etc.. also can be as any app like

Java app, andriod app, destktop app, php app, angular App,

React App, IOS App and etc..

Client App

(or)

Service Cosumer

RL/IL POST/report HTTP11

headers ..

http request

headers

22 bank line>>

...

Server App

HEAD

(or)

Service Provider

Body

XML/JSON data

http request with XML/JSON Data

The Provider application in Restfull webservices

is one MVC controller or advanced Controller class

having b.methods mapped with the request paths..

So the Consumer App invokes b.methods of producer App

@RequestMapping("/search-api") @RestController SearchOperationsController

by send http request based request url and request path of b.methods. (These b.method can be called as Rest operations/ EndPoints) Developer need not to convert the generated

results into XML or JSON becoz the spring boot rest

framework takes care of the work automatically

In Restful web services

=======

(Rest API/Rest Service)

Rest API/Rest service (Provider App)

--->method/report1") EndPoints

|--->method2 ("/report2")

(rest operation2)

=>Each b.method of the RestController is called RestOperation /Rest EndPoint

@RestController = @Controller + @ResponseBody

base

a) Provider App is Controller class which is identified with its url

(rest operation/endpoint).

b) B.mentods inside the controller class are indentified with their request paths

|-->linked with service class methods

note:: the Consumer App uses

proivider App/comp url and

b.methods request paths in order invoke the b.methods (Rest opertions/ EndPoints) of provider App

ed

This annotation makes the request method of @Restcontroller class to send the response directly to consumer app as the http response with out taking the support ViewREsolver and View Comp

note:: Every @RestController class is called Restful webservice/Rest API / Rest Provider App The methods in @RestController class that are mapped with request paths are called Rest operations/Endpoints/ b.method

http response with

XML/JSON Data

http response

RL/IL HTTP 1.1

headers

...

200 OK

headers

HEAD

ank tine>>

XML/JSON data

Body

Http request with JSON BODY

Method

URI

Version

POST http://localhost:8081/api/contacts HTTP/1.1 request

User-Agent: Fiddler

Host: localhost:8081

Both Consumer and Provider Apps use

XML APIS / JSON APIS to perform Marshalling/unmarshalling activities

or Serialization /DeSerialization activities

@RestController = @Controller + @ResponseBody

Http response with JSON Body

This makes the methods of Restful service/Rest API/ provider App directly sending the generated output to consumer app as response with out involving ViewResolver and view comps

HEAD

Headers

line /Initial line

HTTP Version

Status Code

HTTP/1.1 200 OK

response line /Initial line

Content-Type: application/json; charset=ut

Server: Kestrel

X-Powered-By: ASP.NET

Date: Sun, 11 Feb 2018 18:34:00 GMT

Content-Length: 69

<<<blank line >>>>

'name":"Product", 'category": "Appliances",

"subcategory": "Microwaves'

Headers Accept: application/json

Content-Type: application/json

11

Body

Content-Length: 121

>>>>> blank line >>>>>

"Name":"Jane User:,

"Address": "1 Any Street",

Body

"City":"Any city",

(Content)

"Zip":"00000",

"State": "WA",

"Email": "janeuser@example.com"

}

(producer App)

=>The service provider in Restfull App is java class and it can not take http requests directly.. So

we put one FrontController servlet like DispatcherServlet (spring), ActionServlet(struts) and etc..

=> The special servlet that acts as entry and exit point for all requests and reponses is called Frontroller Servlet

note: The provider app in restful web service is called Rest API/API/Restful service /Producer Service app and etc..

The methods is provider app (@RestController class) annotated with @XxxMapping annotations are called Rest operations / Rest end points /Rest methods and etc..

Like

(DispatcherServlet) Server Side

Front

(2) delegates [Server App

Controller

Servlet

to

Service Producer App

(3)

@RestController

In spring/spring boot applications

Client Side

(1)

Client app

http request with xml/

(Service consume)

JSON Data

[/]

(STub logics)

http respons with xml/

(url pattern

Response Entity

(skelton logics)

(4

(5)

obj

JSON Data

(result)

(java class acting as restController (provider))

(Can also be called as API Operations/API EndPoints)

this front controller is called "DispatcherServlet"

=>Restful webservice Provider App= API

=> Restult web services B.methods = API EndPoints

=>In Java, Jax-RS is technology/api to develop Restfull web services

where as Spring Rest, Rest easy, Restlet,Jersy and etc.. are the f/w to

support Restfull web services implementation.

Summary on webservices

==========

web services

=>Spring Rest is given on top

of spring MVC

Spring Rest =spring MVC ++

Spring Boot Rest spring boot MVC++

note:: Officially there is no

module called

Rest in spring/

spring boot frameworks..

we have only web module which

can be used in multiple ways

a) for developing web applications based on MVC archictecture

b) for developing restfull web services based on consumer-producer architecture

(No provision to develop SOAP based WebServices)

SOA::

SOAP Based

web serivces

(SOA) (Old/Legacy)

Desiging Models → RestFull webServices (Consumer-Producer with FrontController) (Modren /Relatively New) Jax-RS

Jax-WS

-Programming APIs (technology)

Service Oriented

Architecture

Apache Axis, <--Proramming Fraeworks ---> Apache CFX

SOAP

<------- protocol

over Http

Given based on

<----- Architectire ------

SOA

note:: Developing service providers in web services is also called as devleoping APIs.

note:: gathering and providing info that are required to develop service cosumer for invoking the services of spefic service provider is nothing

but providing endpoint letails

API development in webservices

in nothing but service provider/ producer app development

providing all the details of

API to Consumer app is

nothing but giving API

Endpoints to Consumer App

The details are like urls, request paths, request modes, parameters and etc..

Weather Report (API/webService Provider/Service Provider) (One @RestController class)

Spring Rest, Rest easy, Restlet, Jersy

Http with JSON/Xml data

API End Points

on

Given based Client-Server

or Consumer - Producer Architecture

|---> getTEmprature (Stirng city) (endpoint1) -->mapped with /wreport/temp --- GET ---- {city}

=> API

params

request url+path mode -->getExpectedTemparature(String city, int no.ofDays) (endpoint2) |--->mapped with /wreport/exptemp --- GET ---- {city}, {days}

API End Ponts (Details about API/

Every API EndPoint is indentified

Provider services)

with its request url, request mode and params

means

developed

nt WebService Provider App

either in SOAP or in RESTfull webSevice

XML

<--- Global data formats ------>

XML/JSON as the global

as global

data format

data format

=> API Consumer means WebService Consumer App development either in SOAP or in Restfull

UDDI

Registry is required <-----Registry

No UDDI Registry is required

The language/technology /framework API that comes in the form of jar file having packages with classes, interfaces,enums and annotations acting as the base

API

to develop software apps is called Library/API.. this API no way related webservices note :: JDBC API, servlet api and etc.. are technology APIs webservice note :: Provider Apps in the webservices are called APIs

Both the APIs are no way related

to each other

=> In software Domain, certain terminologies change its meaning context to context

eg1:: JDBC API, SErvlet API and etc.. called Programming APIS

=> The JSON Serialization is different from 10 streams Serialization

eg2:: SOAP/Restful webservice based provider App (Distributed App) is called WebService API

[Json Serialiation is all about converting JSON doc into Java class obj where as IO streams

Serialization is all about converting Java obj data into bits and bytes]

=> The JSON DeSerialization is different from IO Streams DeSerialization

[Json DeSerialization is all about converting Java object data into JSON doc where as IO streams

DeSerialization is all about creating object from the bits and bytes data]

=> The web service API is nothing but developing Provider App as the Distributed App

having services to consume Where as language/technology API is nothing but Library acting as base for software apps

development