SpringRest ====> SpringMVC++

WebApplication ====> Customer 2 Buisness
DistrubutedApplication => Buisness 2 Buisness
            eg:: Flipkart,Myntra,Amazon,ineuron,....

Technologies available to build Distrubuted applications are
  a. CORBA
  b. RMI
  c. EJB
  d. Webservices
  e. RESTFul services(RepresentationStatefull Services)

a. RMI
     1. It stands for Remote Method Invocation,It is a part of JDK s/w only given by SUNMS.
     2. It doesnot support Transaction Management,Security,....
     3. It is outdated because of EJB.
     4. It is language dependent,platform dependent and Architecture dependent.
      5. Here the data will be exchanges in the form of binary b/w producer and consumer apps.

b. EJB
     1. It stands for Enterprise Java Beans.
     2. It is given by SUNMS
     3. Enchanced version of RMI
     4. It provides container which will take care of 3rd party services
     5. It is langauge dependent(coded only using java)
         communication of applications can happen only if the applications are coded in java only.
     6. It fails Interoparability(langague dependancy)
      7. Container are heavy weight.
      8. They are complex to learn and use.
     9. Here the data will be exchanges in the form of binary b/w producer and consumer apps.

c. CORBA
     1. It stands for Common Object Request Broker Architecture.
     2. It is a specification which is implemented using IDL(Interface Defnition Language)
     3. Corba is complex to learn and implement.
      4. It is architecture and language indepent.
      5. Corba looks greate conceptwise,but gives problem towards the implementation.

To support Interopability Sun MicroSystem released one api called "JAX-RPC".
JAX-RPC(Java API for Xml - Remote Procedural Call) api is built on the specification of B.P.1.0(WSI organisation)
WSI -> WebServices Interoperability(ideas taken from many organisation and given the implementation)
B.P.1.0 is successfull, WS-I released B.P.1.1 specification
Note: B.P(stands for Basic Profile)

W.r.t B.P.1.1 specification SUN Microsystem released JAX-WS api(Java api for XML - WebServices)

We can devleop webservices in 2 ways
a. JAX-RPC(it is outdated becox of xml)
b. JAX-WS(It is good becoz of annotation approach)

WebServices
===========
WebServices in java can be developed in 2 ways

a. JAX-RPC(Java API for XML - Remote Procedure calls)

Implementation are
a. Sun Implementation(SI)
b. Apache Axis
c. IBM Websphere
d. Oracle Weblogic etc....

JAX-RPC api follows rules and guideliness provided by B.P.1.0 specification(WS-I => Webservices Interoperability)

b. JAX-WS(Java API For XML - WebServices)

Implementation are
a. Reference Implemenation (RI)
b. Apache Axis2
c. Apache CXF
d. Oracle weblogic
e. IBM Websphere etc...

JAX-WS api follows rules and guideliness provided by B.P.1.1 specification(WS-I => Webservices Interoperability)

If we devleop a webservice using JAX-RPC or JAX-WS then that webservice is called as "SOAP Based WebService".
SOAP(Simple Object Access Protocol) WebService is called as "Big Services".

RoyFielding identified few problems in B.P.1.1 specification, so SunMicroSystem released one api called "JAX-RS" api
JAX-RS (Java API For -Restful Services)
a. Jersey implementation(Sun)
b. Rest Easy implementation(JBoss)

WebServices
==========
It is a distrubuted technology which is used to develop distrubuted applications with Interoperability.

What is Interoperablitiy?
Irrespective of the platform and the Irrespective of the programming language, if two applications are communicating then those applications are called as "Interoperable applications".

```
eg: java   -------> python
    python -------> .net
    .net   -------> python
```

WebServices Architecture
========================
In the world of webservices we have 2 parties

a. Provider => Application which will be providing buisness services to other applications is called as "Provider application".

b. Consumer => Application which is consuming the services from the other application is called as "Consumer application".

In the world of webservices always development will begin from provider side[Producer]
Provider development can be done in 2 ways

```
      a. Contract First Approach
      b. Contract last Approach

      What is Contract?
 => Contract stands for WSDL
 => Web Services Description language
 => WSDL is a special XML which describes how provider is providing buisness 
services to consumers.
 => Contract First Approach means WSDL file will be created first then Development
will be started.
 => Contract Last Approach means we will prepare the service first and then WSDL 
file will created.

WSDL -> url,input for provider,output for provider,how to access the services will 
be documented in the file.

Once provider developement is completed,provider will share WSDL share to consumer
through email,sharepoint,
UDDI(Universal Description Discovery Integration)

Once Consumer gets the WSDL file, we can start consumer development.
Consumer can be developed in below ways
      a. Stub based consumer
            Based on WSDL if we write some java class we called "Stub Based 
consumer".
      b. Dynamic Proxy consumer
            Dynamically at the runtime some Proxy objects will be created through
we send the request.

Once Consumer development is completed,Consumer will send request to Provider.
Provider will process consumer request and send response to consumer.
                  refer: webservicesarchitecture.png


Note:Key players when we build webservices
  1. Provider
  2. WSDL : WebServices Description Language
  3. UDDI : Universal Discovery Description Language
  4. Consumer
  5. SOAP : Simple Object Access Protocol

SOAP based webservices are not really 100% interoperable and adoptable.
SOAP based webservices wont support for JSON/It supports only XML.
SOAP based webservices are not easily adoptable(dependent on XML).

The above mentioned problems are identified by a person called "Roy Fielding".
He compared SOAP webservices with Internet Services(www).

RoyFielding found 5 principles those are called as "REST Architecture principles".

Challenges of SOAP webservices are resolved in RESTFul Services, SOAP is 
specification based where as ResTful services are Architecture principles based.

REST Architecture
================
  2 Actors are involved here
a. Resource
      It provides buisness services to other applications.
b. Client
```

```
      It access buisness services from other applications.


REST COMPONENTS
===============
   1. Resource( REST Resource)
   2. WADL /Swagger
   3. XML/JSON/Text/Yml
   4. Client
   5. PostMan(To test our application)

=> RoyFielding provided ReST Architecture principle(same prinicple already using by
the internet).
=> SUN Microsystem also supported RoyFielding principle and they released on API
called "JAX-RS".(Java API for XML -RestFulServices)
=> During the invention of this api to market JSON was not available.
=> JSON/XML are language independent and platform independent.

Note: API's are always partial, they contains set of interface and abstract
classes, we can't use api directly to develop a project
      we need implementation for API to develop the project.
            eg: JDBC API =====> we need implementation jars for JDBC API from db
community like MySQL,Oracle,PostgreSQL,...

Implementation of JAX-RS is "Jersey(SUN),Rest Easy (Jboss)".
Both Jersey and Rest Easy supports for both Rest Resource Devleopment and ReSt
Client Development.
=> We can develop RestFul services  using Spring ReSt Module.
=> Spring MVC jars are sufficient to develop RestFul Services (additional jars are
not required).


ReSt Architecture Principles
===========================
 1. Unique Address
 2. Uniform Constraint interfaces
 3. Media Representation
 4. Communication Stateless
 5. Hateos


What is Unique Address?
 Every distrubuted component operation has a unique address.
      eg: Every controller will have a unique URL pattern.


What is uniform constratints interfaces?
 This principle is given to provide easy adaptablity(without documentation u can
access).
 If we use SOAP we need to understand WSDL, which can't be understood by layman,To
resolve this problem we use ReSTful Services.
 constraints means limitation(becoz HTTP methods are limited).
 To achieve this we need to bind our resource methods with HttpProtocol methods.
   eg: @GetMapping    ===> for GET request
       @PostMapping   ===> for POST request
       @PutMapping    ===> for PUT request
       @DeleteMapping ===> for DELETE request

Media Representation
```

```
====================
  The main aim of this priniciple is to provide interoperability(real 
interoperability but this is not there is SOAP).
  It supports mulitple format of data to exchange.(XML,JSON,...)
  RestFul Services will support muliple formats of data to exchange b/w client and 
resource.


Communication Stateless
=======================
  Client and Resource will interact with each other in stateless manner.
  Once the response is reached to the client, that client information will not be 
available w.r.t server.
  Server should treat every request as the first reqeust only.
  Session concpet which is available in webapp's is not required when we build 
restapi's.


HATEOS
======
 Hypermedia As an Engine for application state.
 Server should send response to clients using hyper links.
 RestEnd Point URL ::  http://www.ineuron/get/CID-100
      {
            course-id : 'CID-100',
            course-name : 'SB & MS',
            start-date  : '19-jul-2022',
            timings     : '7.30 to 8.30 AM',
            trainer     : 'Mr. NaveenReddy',
            course-content: 'www.ineuron.ai/get/CID-100/course-content'
      }

Note:
 1. RestFul Services has no specification, it has only architectural principles.
 2. HTTP Request calls => GET,POST,PUT,DELETE
 3. Client information should not be stored in RestAPI(B 2 B communication)

Building First  ReStful application using SpringBoot + Rest
===========================================================
 To build ReST applications 
      a. Spring WebMVC Module is sufficient no need of any jars
            eg: spring-boot-starter-web
                        a. web app developement using spring mvc
                        b. rest app development using spring mvc
                        c. embeded tomcat container

Annotation used here
      1. @RestController
                  It will make our class as "Distrubuted component".
                  @RestController = @Controller + @ResponseBody
                        Default value is always Raw Data.

                  @Controller = It always returns "viewName".

      2. write methods in Restcontroller and bind them to Http request method with 
unique url pattern.
                  @GetMapping     -> GET request
                  @PostMapping    -> POST request
                  @PutMapping     -> PUT request
                  @DeleteMapping  -> Delete request

Configuring below properties in application.properties or application.yml file
   server.port=9999 (embeded container will run on 8080 in spring boot)


WishController.java
===================
package in.ineuron.restcontroller;

import java.time.LocalDateTime;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/wish")
public class WishController {

      @GetMapping("/msg")
      public ResponseEntity<String> generateWishMessage() {

            LocalDateTime ldt = LocalDateTime.now();
            int hour = ldt.getHour();

            String body = null;
            if (hour < 12)
                  body = "Good Morning";
            else if (hour < 16)
                  body = "Good Afternoon";
            else if (hour < 20)
                  body = "Good Evening";
            else
                  body = "Good night";

            ResponseEntity<String> entity = new ResponseEntity<>(body,
HttpStatus.OK);
            return entity;
      }
}

EndPoint :: http://localhost:9999/wish/msg
Output   :: Good night

Download postman from the following link:: https://www.postman.com/downloads/
Adding postman as plugin for chrome ::
https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop?
hl=en
```