Webservices Theory:
https://www.youtube.com/watch?v=fvG7FEU1Rt8

Webservices (ReST)

Webservices : (Integration of Applications)
 Linking two or more applications which are running in different
servers.
 (Applications can be in same server also).

Ex: BookMyShow ---Link ---- PaymentGateway
       (java)   (.net)
     ICICI Bank ---Link ---- SBI Bank

=> Webservices is implemented using HTTP Protocol and
   data in Global Format(XML/JSON)

=> Webservices is not for UI design, they are used for integration.
   For UI use : Thymeleaf, ReactJS, JSF ....etc

=> Webservices two types
  a. SOAP webservices (XML Concept - Simple Object Access Protocol)
  b. ReST Webservices (Representation State Transfer)

                     ReST Webservices
                (Representation State Transfer)
   State   = Data
   Transfer = Send/Receice
   Representation = With Protocol(HTTP) + Global Format(XML/JSON)

 ReST : Exchange data between applications using HTTP protocol
        and data in Global Format.

=> To develop web(any)application we use 3 layers
  a. PL (Presentation Layer)
  b. SL (Service Layer)
  c. DAL (Data Access Layer)

=> To create link(communication) we use 4th layer
  d. IL (Integration Layer)

=> This layer must present in two applications
   one application bahaves like Service Provider
   and another one Service Consumer.

Ex: BookMyShow is using service of PayTM
    BookMyShow is consumer and PayTm is Provider.

=> One Application can behaves like consumer for one integration
   and for another it might be producer even.

Spring ReST

=> To implement ReST Webservices, Spring F/w has provided one Web API
   ie called as Spring ReST (ReST API).

=> Sun has provided API called JAX-RS (rules and guidelines).
   Implementation with new Annotations API is given by Spring ReST.

=> By using Spring ReST we can implement Provider and Consumer
Applications
   Provider using : RestController
   Consumer using : RestTemplate

*) we can use Spring ReST in 3 ways

case#1 Both different applications(Boot Applications only)
         + Integrated using Spring Boot REST

   Consumer   Provider
   RestTemplate   RestController

case#2 one applications but designed as FrontEnd and backend

   Consumer   Provider
   Angular/   RestController
   React/
   Andriod

case#2 Both different applications(java and non-java)
         + Integrated using Spring Boot REST

   Consumer   Provider
   .net Client   RestController
   php client

   Consumer   Provider
   RestTemplate   .net /php/python

---Provider work flow Design------------------------------------
=> Client makes request to Provider, taken by FC(DispatcherServlet)
=> With the help of HandlerMapping, FC gets RestController details
   and executes code.

=> RestController gives output in String/Object/Collection Format
   ie called as Entity(Final Output) and placed in Response object
   along with Http Status (1xx--information, 2xx--success,3xx-
redirect,
      4xx--client side error, 5xx--server side error).

=> ResponseEntity is given back to FC, same is sent to client.

=> *** If Entity is non-String (object or Collection)
   ex: List<Employee> , Student, Product[] ..etc then
      data is converted into XML or JSON.

String is sent as text format text/plain as Response Body.