Spring Web Services

# Introduction to Web Services

## What is Web Service?

* A software system designed to support **interoperable** **machine to machine (or application to application)** **interaction** **over a network**.
* Three Keys Points –
  + A web service should support application to application interaction.
  + It should be interoperable (platform independent).
  + It should allow communication over the network.

## Hows?

### How does data exchange between applications take place?

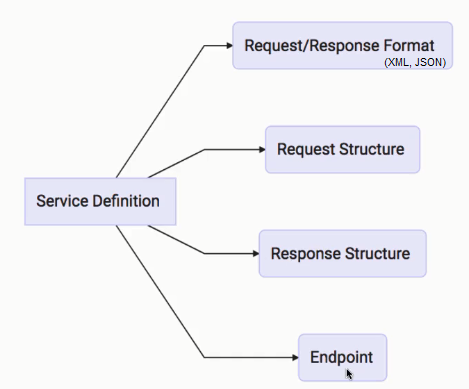
* The input to a web service is called a Request and the output from a web service is called a Response.

### How to make web services platform independent?

* The request and the response also should be platform independent.
* They should be in formats which are supported by all different kinds of platforms.
* E.g. XML, JSON

### How does an application know the format of request and response?

* Via Service definition. Every web service offers a service definition.

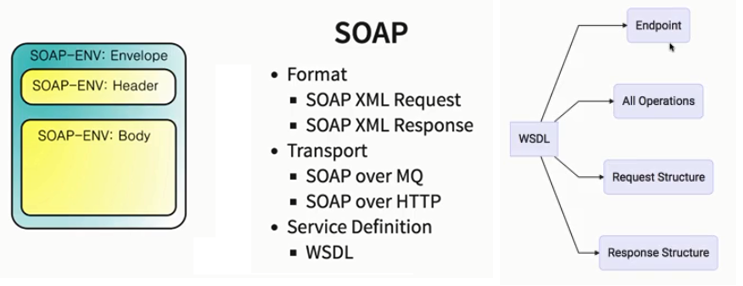


## Web Services – Key Terminologies

* Request is the input of our service and response is the output from a web service.
* Message exchange format is the format of the request and the response. E.g. XML, JSON.
* Service provider (Server) is the one which hosts the web service.
* Service consumer (Client) is the one which is consuming the web service.
* Service definition is the contract between the service provider and the service consumers.
* Transport defines how a service is called. E.g. HTTP, MQ.

## Introduction to SOAP Web Services

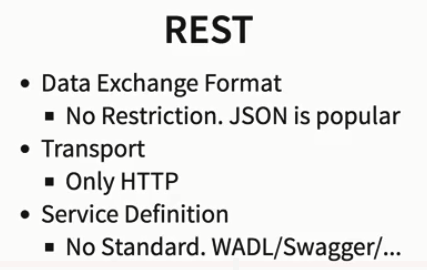
* Simple Object Access Protocol.
* In SOAP we use XML as the request exchange format.
* SOAP defines a specific XML request and response structure (SOAP XML format).
* If you're using SOAP then you have to use this structure. So you have to create a SOAP and envelope which contains a SOAP header and a SOAP body.
* The header contains meta information like authentication, authorization, signatures and things like that. SOAP Header can be empty.
* Body is where you really put the real content of your request or your response.



* In SOAP, the service definition is typically done using WSDL (Web service definition language).
* In summary, SOAP is all about adhering to the services XML structure. Adhering to the envelop header and the body.

## Introduction to RESTful Web Services

* REST stands for REpresentational State Transfer.
* It’s a term which is coined by Roy Fielding. Roy Fielding is the guy who also developed HTTP protocol.
* The key thing about REST services is the fact that they would want to make best use of his HTTP.
* RESTful web services try to define services using the different concepts that are already present in HTTP.
* The most important abstraction in the REST is something called a resource.
* A **resource** is anything that you'd want to expose to the outside world through your application.
* When we are talking about talking about RESTful services, we are always thinking about resources.
* **WADL** (web application definition language) is one of the formats in which you can specify your RESTful Web services. Another option is **swagger**.
* REST focuses on your resources and how do you perform actions on them making best use of HTTP.



## SOAP vs REST

* SOAP and REST are not really comparable. It's not really an apple to apple comparison.
* REST defines an architectural approach.
* Whereas SOAP poses restrictions on the format of XML which is exchanged between your service provider and the service consumer.
* RESTful services are typically easier to implement than SOAP.
* RESTful services are typically based on JSON which is an easy format to pass and do things with it and also with RESTful services, we don't really need to mandate really define a service definition. But with SOAP you have to define WSDL and there are a lot of complexities associated with parsing your XMLs as well.

# Introduction to Spring Framework

# Tips and Tricks