Define Web Service?

A web service is a kind of software that is accessible on the Internet. It makes use of the XML messaging system and offers an easy to understand, interface for the end users.

UDDI, Universal description, discovery and integration

Simple Object Access Protocol(SOAP):

SOAP is a protocol specification for exchanging structured information in the implementation of Web services in computer networks. It relies on XML as its message format.

Web Service Description Language(WSDL):

Web Service Description Language. It is an XML file that describes the technical details of how to implement a web service, more specifically the URI,

port, method names, arguments, and data types. Since WSDL is XML, it is both human-readable and machine-consumable, which aids in the ability to call and bind to services dynamically.

Elements of WSDL are:

Description:

It is the root element of a WSDL 2.0 file. It usually contains a set of name space declarations which are used throughout the WSDL file.

Ques 7. What is WSDL?

Ans. The Web Services Description Language (WSDL) currently represents the service description layer within the Web service protocol stack.

In a nutshell, WSDL is an XML grammar for specifying a public interface for a Web service. This public interface can include the following:

Information on all publicly available functions. Data type information for all XML messages. Binding information about the specific transport protocol to be used.

Address information for locating the specified service.

types

It refers to the data type information. Each of the parameter has a type information associated with it.

For RPC style web services, the type information includes the standard XML data types like String, int, long etc.

But for DOCUMENT style web services, the types element refers to an XSD which can be used to describe complex data types by using XSD constructs.

message

For every invocation of SOAP based web services, there are messages being exchanged between the server and client. These messages include the request and response messages.

portType

This element defines the operations/transactions that can occur in a web service invocation

binding

This element defines the transport protocol to be used and the encoding style. For transport, mostly it is “HTTP” and style is “literal”

service

This element is used by the web service consumers to locate the web service. It consists of the endpoint URL and the binding name.

Contract last or Bottom up approach:

When using contract last approach,

you first write your java code then you create web service contract(WSDL).

There are various kinds of tools which can generate WSDL on the basis of java code.

Contract first or Top Down Approach :

It is reverse of contract first.Here you first define web service contract.

You define all the elements of WSDL first then after that you create your java logic.

JAX-WS Web Services

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

The root element “definitions”, consists of the five elements:

@WebService(serviceName = "MyWebService")

@SOAPBinding(style = Style.RPC, use = Use.LITERAL)

public class MyWebService {

// Dependency Injection (DI) via Spring

@Autowired

EmployeeService employeeservice;

@WebMethod(exclude = true)

public void setMyBObject(Employee emp) {

employeeservice.createEmployee(emp);

}

@WebMethod(operationName = "printMessage")

public String printMessage() {

return employeeservice.getEmployee(7521).toString();

}

JAX-RS Web Services

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

Diff in SOAP and REST service

which is better

The main advantage of SOAP-based SOA over ROA is the more mature tool support;

however, this could change over time. Another SOA advantages include the type-safety of XML requests (for responses, ROA can also use XML if the developers desire it).

The main advantage of ROA is ease of implementation, agility of the design, and the lightweight approach to things.

In a way, SOA and SOAP is for people in business suits; that's what you'll find used in the banking and finance industries

What are the major differences you can state while using SOAP or REST, in terms of

applicability as concern?

@webmethod & @Webserevice

<https://dzone.com/articles/apache-cxf-vs-apache-axis-vs>

Let's look at positives & concerns in nutshell of each of these frameworks:

|  |  |  |
| --- | --- | --- |
| **Framework** | **Key Positives** | **Key Concerns** |
| **Apache AXIS2** | ❶ Most Commonly Used, Matured & Stable Web Services Development Framework  ❷ Supports Multiple Languages (C++, Java)  ❸ Supports both Contract-first & Contract-last Approach  ❹ In context of Orchestration & Web Services Transaction (long-running transactions) it supports wide variety of related WS-\* specifications:  *WS-Atomic Transaction, WS-Business Activity, WS-Coordination, WS-Eventing, WS-Transfer*  *Compatible with Spring Framework* | ❶ Comparatively More Code Required/Generated w.r.t. Spring WS/CXF  ❷ Is being phased out gradually (mostly by Apache CXF)  ❸ It is not fully compliant for JAX-WS JAX-RS |
| **Apache CXF** | ❶ Most widely used Web Services Standard Now; Improvement over AXIS2, which is now gradually being replaced by Apache CXF  ❷ Intuitive & Easy to Use (less coding required as compared to AXIS2)  ❸ Clean separation of front-ends, like JAX-WS, from the core code  ❹ Fully compliant with JAX-WS, JAX-RS & others  ❺ Best Performance across all available framework with minimum computation overhead  ❻ Supports wide variety of front-end models  ❼ Supports both JAX-WS & JAX-RS (for Restful Services)  ❽ Supports JBI & SDO (not supported in AXIS2)  ❾ Compatible with Spring Framework | ❶ Does not support Orchestration & WS Transactions yet  ❷ Does not support WSDL 2.0 yet |
| **Spring WS** | ❶ Best in terms of supporting Contract-first Web Services Development Approach  ❷ Enforces Standards & Best Practices by Framework Constraints (no way out of it & hence limitation as well)  ❸ Supports Spring Annotations as well as JAX-WS  ❹ Least code from developer’s perspective  ❺ Best Aligned with Spring Technology Stack (also similar architectural stack as Spring MVC) including Spring Security | ❶ Least number of WS-\* Specifications supported (does not fully compliant with JAX-WS)  ❷ Spring offers itself as standard & hence other Java-compliant frameworks support better standards support  ❸ Only support Contract-first Web Services Development Model |