01: ♥♦ 40+ Java Web Services Architecture & Basics Interview Q&A – Q01 – Q12

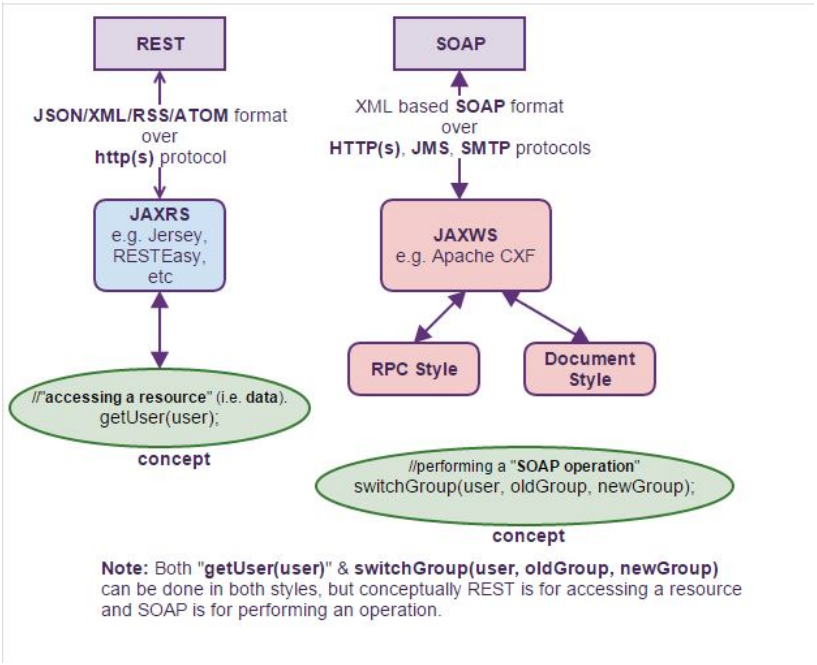
Posted on August 19, 2014 by Arulkumaran Kumaraswamipillai

As an enterprise Java developer, you will be spending more time integrating systems via web services & messaging. Java Web Services interview questions are very popular.

Q1. What are the different styles of Web Services used for application integration? and What are the differences between both approaches?

A1. SOAP WS and RESTful Web Service. Web services are very popular and widely used to integrate similar (i.e. Java applications) and disparate systems (i.e. legacy applications

and applications written in .Net etc) as they are **language neutral**.



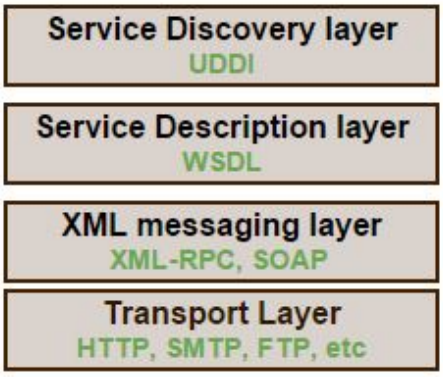
Java Web Service styles comparison

SOAP vs. REST comparison

SOAP Web service	RESTf
SOAP (Simple Object Access Protocol) is a standard **communication protocol** on top of transport protocols such as HTTP, SMTP, Messaging, TCP, UDP, etc.	REST can be as HT Each u resour and yo Object “DELE

16 Technical Key Areas

80+ step by step Java Tutorials



SOAP Layers

SOAP uses its own protocol and focuses on exposing pieces of **application logic (not data)** as services. SOAP exposes operations. SOAP is focused on accessing named operations, which implement some business logic through different interfaces.

SOAP only permits XML data formats.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <ns2:CreateCaseRequest xmlns:ns2="http://www.mydomain.com/svc/v1">
      <AuditHeader>
        <Message>
          <CaseId>123456</CaseId>
          <OperationName>createEmptyCase</OperationName>
          <CreationDate>2015-04-21</CreationDate>
        </Message>
      </AuditHeader>
    </ns2:CreateCaseRequest>
  </soap:Body>
</soap:Envelope>
```

SOAP

REST
interne
and De
on acc
consist

REST
XML, J
etc. JS
better

URL:
http://l

or

Tutorial - Diagnosis (2
Akka Tutorial (9)
Core Java Tutorials (2
Hadoop & Spark Tuto
JEE Tutorials (19)
Scala Tutorials (1)
Spring & HIbernate Tu
Tools Tutorials (19)
Other Tutorials (45)

100+ Java pre-interview coding tests

open all | close all
Can you write code? (
♦ Complete the given
Converting from A to I
Designing your classe
Java Data Structures
Passing the unit tests
What is wrong with th
Writing Code Home A
Written Test Core Jav
Written Test JEE (1)

How good are your .....?

open all | close all
Career Making Know-
Job Hunting & Resum

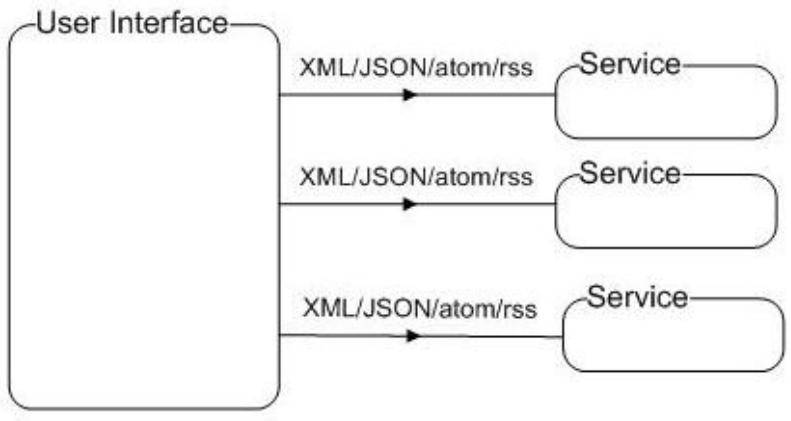
SOAP based reads **cannot be cached**. The application that uses SOAP needs to provide cacheing.	REST and sc
Supports both **SSL security** and **WS-security**, which adds some enterprise security features. Supports identity through intermediaries, not just point to point SSL. — WS-Security maintains its encryption right up to the point where the request is being processed. — WS-Security allows you to secure parts (e.g. only credit card details) of the message that needs to be secured. Given that encryption/decryption is not a cheap operation, this can be a performance boost for larger messages. — It is also possible with WS-Security to secure different parts of the message using different keys or encryption algorithms. This allows separate parts of the message to be read by different people without exposing other, unneeded information. — SSL security can only be used with HTTP. WS-Security can be used with other protocols like UDP, SMTP, etc.	Suppo — The client e retriev receive presen is only client a proxy v from th — The whethe
Has comprehensive support for both **ACID** based transaction management for short-lived transactions and **compensation** based transaction management for long-running transactions. It also supports two-phase commit across distributed resources.	REST **ACID** **comm** resour

SOAP has success or retry logic built in and provides end-to-end reliability even through SOAP intermediaries.	REST system to deal

Which one to favor? In general, a REST based web service is preferred due to its simplicity, performance, scalability, and support for multiple data formats. SOAP is favored where service requires comprehensive support for security and transactional reliability.

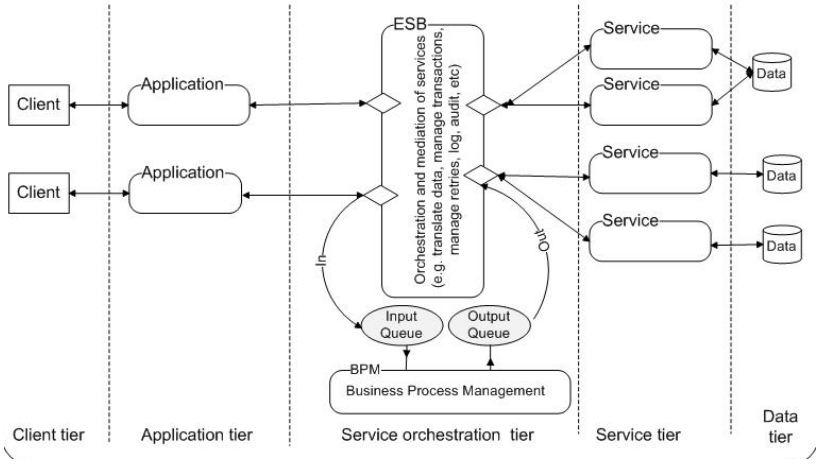
SOA done right is more about RESTFul + JSON, favoring lighter weight approaches to moving messages around than the heavyweight ESBs using WSDL+XML that gave SOA a bad name.

Q2. Differentiate between SOA (Service Oriented Architecture) versus WOA (Web Oriented Architecture)?
A2. **WOA extends SOA** to be a light-weight architecture using technologies such as REST and POX (Plain Old XML). POX compliments REST. JSON is a variant for data returned by REST Web Services. It consumes less bandwidth and is easily handled by web developers mastering the Javascript language



WOA – RESTFul Service Calls via AJAX to populate different sections of a UI

SOA and WOA differ in terms of the layers of abstraction. **SOA is a system-level architectural style** that tries to expose business capabilities so that they can be consumed by many applications. **WOA is an interface-level architectural style** that focuses on the means by which these service capabilities are exposed to consumers. You can start out with a WOA and then grow into SOA.



SOA (when SOAP WS in service tier) or WOA (when REST WS in service tier)

According to Nick Gall, "**WOA** = **SOA** + **REST** + **WWW**". In the above diagram from the **Service Orchestration tier**, which is responsible for loosely coupling services,

For the **SOA** => you will be making **SOAP** style web services in the "**Service Tier**".

For the **WOA** => you will be making more lighter **REST** style web services in the "**Service Tier**".

Q3. How would you decide what style of Web Service to use? SOAP WS or REST?
A3. In general, a REST based Web service is preferred due to its simplicity, performance, scalability, and support for multiple data formats. SOAP is favored where service requires comprehensive support for security and transactional reliability.

The answer really depends on the functional and non-functional requirements. Asking the questions listed below will help you choose.

1) Does the service expose data or business logic? (REST is a better choice for exposing data, SOAP WS might be a better choice for logic).

2) Do consumers and the service providers require a formal contract? (SOAP has a formal contract via WSDL)

3) Do we need to support multiple data formats?

4) Do we need to make AJAX calls? (REST can use the XMLHttpRequest)

5) Is the call synchronous or asynchronous?

6) Is the call stateful or stateless? (REST is suited for statless CRUD operations)

7) What level of security is required? (SOAP WS has better support for security)

8) What level of transaction support is required? (SOAP WS has better support for transaction management)

9) Do we have limited band width? (SOAP is more verbose)

10) What's best for the developers who will build clients for the service? (REST is easier to implement, test, and maintain)

Q4. What tools do you use to test your Web Services?
A4. SoapUI tool for SOAP WS & RESTFul web service testing and on the browser the Firefox "**poster**" plugin or Google Chrome "**Postman**" extension for RESTFul services.

Q5. Why not favor traditional style middle-ware such as RPC, CORBA, RMI and DCOM as opposed to Web services?

A5. The **traditional middle-wares tightly couple** connections to the applications. Tightly coupled applications are hard to maintain and less reusable. Generally do not support heterogeneity. Do not work across Internet and can be more expensive and hard to use.

Web Services support loosely coupled connections. The interface of the Web service provides a layer of abstraction between the client and the server. The loosely coupled applications reduce the cost of maintenance and increases re-usability. Web Services present a new form of middle-ware based on XML and Web. Web services are language and platform independent. You can develop a Web service using any language and deploy it on to any platform, from small device to the largest supercomputer. Web service uses language neutral protocols such as HTTP and communicates between disparate applications by passing XML or JSON messages to each other via a Web API. Do work across internet, less expensive and easier to use.

Q6. What is the difference between SOA and a Web service?
A6. **SOA** is a software design principle and an architectural pattern for implementing loosely coupled, reusable and coarse grained services. You can implement SOA using any protocols such as HTTP, HTTPS, JMS, SMTP, RMI, IIOP (i.e. EJB uses IIOP), RPC etc. Messages can be in XML or Data Transfer Objects (DTOs).

Web service is an implementation technology and one of the ways to implement SOA. You can build SOA based applications without using Web services – for example by using other traditional technologies like Java RMI, EJB, JMS based messaging, etc. But what Web services offer is the standards based and platform-independent service via HTTP, XML, SOAP, WSDL and UDDI, thus allowing interoperability between heterogeneous technologies such as J2EE and .NET.

Q7. What is a microservice architecture (aka **MSA**)?
A7. Martin Fowler defines Microservices as a **subset** of Service Oriented Architecture (SOA). In SOA, the services

can be of any size. In a microservice, the service performs a single function. Micro means small. The term micro is not well defined, but the questions you need to ask are – Are my services **small** enough?

1) can they be **reused** in many different business contexts as possible?

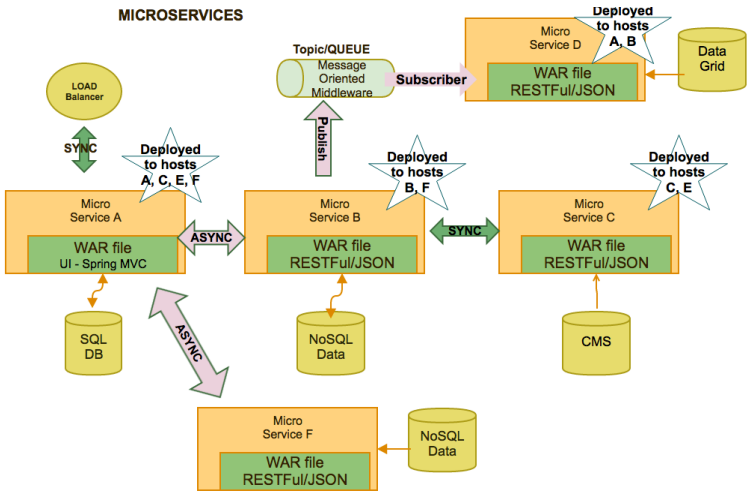
2) can they be individually deployed?

3) can they be individually scaled?

4) can they be individually monitored?

5) can they be individually developed by different small teams?

Q8. What are the key characteristics of microservices?
A8. Focuses more on “loose coupling & high cohesion” than reuse. The definition of a **service** is an individual execution unit, and the definition of “**micro**” is the size, so that the service can be autonomously developed, deployed, scaled & monitored. This means the service is full-stack and has control of all the components like UI, middleware, persistence, and transaction. The services can be invoked both synchronously and asynchronously.



Microservices deployed to multiple hosts (e.g. on the cloud)

Microservices

1) are **reusable**, but NOT ALWAYS. How do we ensure the highest possible reuse? Make all service so small that they can be reused in many different business contexts as possible. Whilst it is possible to have very small services with few or no dependencies on other services, microservices are not always reusable because **1)** unlike libraries that share only behavior, services generally share both **behavior and data**. **2)** By increasing reuse, you are increasing dependencies and coupling as explained below.

2) are **loosely coupled and highly cohesive**. All good software designs should strive for loose/low coupling & tight/high cohesion. Every time we reuse, we increase our coupling. How? If we wrote the code all in one place, there are no dependencies. By reusing code, you've created a dependency. The more you reuse, the more dependencies you have.

Coupling refers to how related are two or more classes/modules and how dependent they are on each other. Loose coupling would mean that changing something major in one class should not affect the other. Tight coupling would make your code difficult to make changes as making a change to one class/module will require a revamp to other tightly coupled classes/modules.

High cohesion means a class is focused on what it should be doing. Low cohesion means a class does a great variety of actions.

So, applications built from microservices aim to be as decoupled and as cohesive as possible by acting more like **filters** in UNIX:

```
1
2 ls | ws -c
3
```

Receiving a request -> applying logic as appropriate -> producing a response | Receiving a request -> applying logic as appropriate -> producing a response | …

3) require a **service registration** as multiple processes working together need to find each other. **Spring Cloud** is built on Spring Boot, and incorporates the registration service called “Eureka” built by Netflix.

Microservices allow large systems to be built from a number of collaborating components. The collaborating components can be web services, **messaging**, **event-driven APIs**, Web Sockets, and non-HTTP backed RPC mechanisms. So, a web service can be a microservice, but microservice might not be a web service.

Q09. How does SOA (Service Oriented Architecture) differ from MSA (i.e. MicroServices Architecture)?
A09.

SOA vs. MSA comparison

SOA (Service Oriented Architecture)	**MSA (MicroServices Architecture)**
SOA is coarse-grained.	MSA is fine-grained. Often behavior & data are encapsulated. Adheres to the Single Responsibility Principle(SRP).
SOA focuses more on **re-usability**. SOA has higher coupling, where services depend on other services to function.	MSA focuses more on **low coupling & high cohesion**. It also emphasizes on **autonomy**, where services can be individually developed, deployed, scaled & monitored to provide

	business value on its own with both **behavior** & **data**. Lower coupling allows the service to operate independently and high cohesion increases it's ability to add value on its own.

Q10. How do you achieve low latency in microservices ?
A10. A cloud based **scale out** model spins out more servers to improve microservices that don't perform well.

You can also **scale up** by

- Favoring in-memory services.
- Allocating more heap and tuning the JVM for GC as microservices tend to consume more memory.
- Making use of reactive programming techniques relying on asynchronous message passing. **Reactive Programming** (RP) in Java Interview Q&A
- Owning their own data without sharing with other services.
- Avoid caching (or use sparingly) and transactions where applicable.
- Using Web Sockets for better scalability of the real-time web applications. It provides bidirectional & full duplex communication over a single socket that is native to the browser without any additional overhead or complexity.

Q11. How do you version your RESTful web services?
A11. **1)** Via URL & **2)** Via HTTP headers.

1) A commonly used way to version your API is to add a version number in the **URL**.

```
1
2 /api/v1/user/67
```

```
3
```

to move to another API with significant changes

```
1
2 /api/v2/user/67
3
```

2) Hypermedia way using the “Accept” **HTTP headers**.

```
1
2 GET /api/v2/user/67 HTTP/1.1
3 Accept: application/json; version=1.0
4
```

Q12. What are the different types of RESTful API changes that require newer **versioning**?
A12. There is no single standard approach, but here are some guidelines:

1) Simple format or cosmetic changes: Changing the output format from XML in v1 to JSON in v2. This requires JAXB annotation changes.

2) Modest schema changes. Make the required logic changes without having to create any new packages.

3) Major schema changes. May require a parallel set of packages and artifacts like controllers, services, entities, and possibly database tables.

Java Web Services interview questions & Answers Links:

6 Java RESTful Web services Interview | 11 SOAP Web service interview | SOAP Web Service Styles Interview Q&A | 5 JAXB interview Questions & Answers | 10 Java web services written test questions and answers | RESTful Web services and HATEOAS Q&A | 5 REST constraints (i.e. design rules) interview Q&A

Popular Posts

♦ 11 Spring boot interview questions & answers

825 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

766 views

18 Java scenarios based interview Questions and Answers

400 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

388 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

295 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

285 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

279 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

239 views

001B: ♦ Java architecture & design concepts interview questions & answers

201 views

Bio | Latest Posts

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in

2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ ♦ 14+ SQL interview Questions & Answers

09: 11 SOAP Web service interview questions and answers ›

Posted in FAQ Java Web Services Interview Q&A Essentials, Java Architecture Interview Q&A, WebService

Tags: Java/JEE FAQs, JEE FAQs

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.

↑