

Industrial strength Java/JEE Career Companion to open more doors

[Home](#)
[Java FAQs](#)
[600+ Java Q&As](#)
[Career](#)
[Tutorials](#)
[Member](#)
[Why?](#)
[Can u Debug?](#)
[Java 8 ready?](#)
[Top X](#)
[Productivity Tools](#)
[Judging Experience?](#)

[Home](#) › [Interview](#) › [Pressed for time? Java/JEE Interview FAQs](#) › [FAQ JEE Job Interview Q&A Essentials](#) › [JSON interview Q&A](#)

JSON interview Q&A

Posted on [September 14, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — No

[Comments](#) ↓



Q1. What is JSON and how does it differ from an XML?

A1. JSON (JavaScript Object Notation) is a lightweight, text-based, language-neutral like XML, but less verbose than XML data exchange format. JSON is used in Web services to exchange data between client and server. Client makes ajax requests to get snippets of data in JSON format from the server via RESTful web service calls. JSON is easy for humans and machines to read and write. JSON can be represented as objects and arrays.

XML

```
1 <Employee>
2   <name type="first">Peter</name>
3   <age>25</age>
4 </Employee>
```

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

[open all](#) | [close all](#)

- ✚ [Ice Breaker Interview](#)
- ✚ [Core Java Interview C](#)
- ✚ [JEE Interview Q&A \(3](#)
- ✚ [Pressed for time? Jav](#)
- ✚ [Job Interview Ice B](#)
- ✚ [FAQ Core Java Jot](#)
- ✚ [FAQ JEE Job Inter](#)
- ✚ [♦ 12 FAQ JDBC](#)
- ✚ [♦ 16 FAQ JMS ir](#)
- ✚ [♦ 8 Java EE \(ake](#)
- ✚ [♦ Q01-Q28: Top](#)
- ✚ [♦ Q29-Q53: Top](#)
- ✚ [01: ♦ 12 Web ba](#)
- ✚ [06: ♦ MVC0, MV](#)
- ✚ [JavaScript mista](#)
- ✚ [JavaScript Vs Ja](#)
- ✚ [JNDI and LDAP](#)
- ✚ [JSF interview Q&](#)
- ✚ [JSON interview](#)
- ✚ [FAQ Java Web Ser](#)
- ✚ [Java Application Ar](#)

5

JSON employee object:

```

1 {
2   "name": "Peter",
3   "nameType": "first",
4   "age": "25"
5 }
```

JSON with Name as a separate object

```

1 {
2   "name": {
3     "name": "Peter",
4     "type": "first"
5   }
6   "age": "25"
7 }
```

JSON Array:

```

1 {"employees": [
2   {"name": "Peter", nameType="first", "age": "25"
3   {"name": "John", nameType="first", "age": "52"}
4   {"name": "Simon", nameType="first", "age": "34"}
5 ]}
```

Q2. What is JSON and how does it differ from an XML?**A2.** When will you favor XML over JSON for data transfer, and when will you favor JSON over XML?**Favor XML over JSON**

When you need to validate your messages using XSDs, Schematron, etc

When you need to transform your messages using XSLT. For example, XML to HTML, etc

When you need to inter-operate with environments that don't support JSON

When you need to have a lots of strong mark-ups.

Favor JSON over XML

- [+ Hibernate Job Interview](#)
- [+ Spring Job Interview](#)
- [+ Java Key Area Essentials](#)
- [+ OOP & FP Essentials](#)
- [+ Code Quality Job Interview](#)
- [+ SQL, XML, UML, JSC](#)
- [+ Hadoop & BigData Interview](#)
- [+ Java Architecture Interview](#)
- [+ Scala Interview Q&As](#)
- [+ Spring, Hibernate, & JPA](#)
- [+ Testing & Profiling/SaaS](#)
- [+ Other Interview Q&A Interview](#)
- [+ !\[\]\(c25df6c57dccd81095d14ed16c5e6013_img.jpg\) Free Java Interview](#)

16 Technical Key Areas

[open all](#) | [close all](#)

- [+ Best Practice \(6\)](#)
- [+ Coding \(26\)](#)
- [+ Concurrency \(6\)](#)
- [+ Design Concepts \(7\)](#)
- [+ Design Patterns \(11\)](#)
- [+ Exception Handling \(3\)](#)
- [+ Java Debugging \(21\)](#)
- [+ Judging Experience Interview](#)
- [+ Low Latency \(7\)](#)
- [+ Memory Management](#)
- [+ Performance \(13\)](#)
- [+ QoS \(8\)](#)
- [+ Scalability \(4\)](#)
- [+ SDLC \(6\)](#)
- [+ Security \(13\)](#)
- [+ Transaction Management](#)

80+ step by step Java Tutorials

When messages don't need to be validated or transformed
 When messages predominantly have data and no marked up texts

When messaging end-points have JSON support. For example, JSON/HTTP, JSON/SMTP, etc

JSON is less verbose, simpler, and performs better.

Q3. What are some of the popular JSON libraries for Java?

A3. **Jackson**, **google-gson**, and **JSON-lib** to name a few.

Q4. How do you tell your RESTful web services to accept JSON data as opposed to XML?

A4. You tell it via the "**Accept**" HTTP header.

Accept=application/json

For example, in Spring MVC

```

1 @Controller
2 public class MyAppController {
3
4     @Resource(name = "myAppService")
5     private MyAppService myAppService;
6
7     @RequestMapping(value = "/addOrModifyAdjustment")
8     public @ResponseBody MyAppDetail addOrModifyAdjustment() {
9         MyAppDetail addOrModifyAdjustment = myAppService.addOrModifyAdjustment();
10        return addOrModifyAdjustment;
11    }
12 }
13 }
```

The JAX-RS annotation **@Consumes** and **@Produces**

```

1 @PUT
2 @Consumes("application/json")
3 @Produces("application/json")
4 @Path("/{accountId}")
5 public RestResponse<Account> update(Account account) {
6     ...
7 }
```

Q5. What is JSONP?

A5. **JSONP** is a simple way to overcome browser restrictions when sending JSON responses from different domains from the client.

[open all](#) | [close all](#)

- [Setting up Tutorial \(6\)](#)
- [Tutorial - Diagnosis \(2\)](#)
- [Akka Tutorial \(9\)](#)
- [Core Java Tutorials \(2\)](#)
- [Hadoop & Spark Tutorial \(1\)](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)
- [Spring & Hibernate Tutorials \(1\)](#)
- [Tools Tutorials \(19\)](#)
- [Other Tutorials \(45\)](#)

100+ Java pre-interview coding tests

[open all](#) | [close all](#)

- [Can you write code? \(1\)](#)
- [Complete the given code \(1\)](#)
- [Converting from A to B \(1\)](#)
- [Designing your class \(1\)](#)
- [Java Data Structures \(1\)](#)
- [Passing the unit tests \(1\)](#)
- [What is wrong with this code? \(1\)](#)
- [Writing Code Home Assignment \(1\)](#)
- [Written Test Core Java \(1\)](#)
- [Written Test JEE \(1\)](#)

How good are your?

[open all](#) | [close all](#)

- [Career Making Knowledge \(1\)](#)
- [Job Hunting & Resumes \(1\)](#)

If you have a **GUI application** (i.e. a war) and a separate **RESTful service application as a separate application** running on two different domains, the you need JSONP for your Ajax to make cross domain calls. For example, if you have 2 domains local and dev. The initial “sum” page will be loaded from the “local” domain, and once you click on the “add” button, the Ajax call will be made to the “dev” domain to get the calculated sum via the RESTful web service call via jsonp callback.

```
1 <script type="text/javascript">
2
3 function add() {
4     var url = 'http://DEV:8080/aes-gui/simple/poc/
5
6     console.log("logging.....");
7
8     $.ajax({
9         type : 'GET',
10        url : url,
11        data : {
12            inputNumber1 : $("#inputNumber1").val(),
13            inputNumber2 : $("#inputNumber2").val()
14        },
15        async : false,
16        //contentType : "application/json",
17        dataType : 'jsonp',
18        //jsonp: "callback",
19        //jsonpCallback: processJSON(jsonData),
20        success : function(response, textStatus, jqXH
21            console.log("reached here");
22            // data contains the result
23            // Assign result to the sum id
24            $("#sum").replaceWith('<span id="sum">' + re
25            console.log(response);
26        },
27        error : function(jqXHR, textStatus, errorThro
28            console.log(errorThrown);
29        }
30    });
31 }
32 };
33
34 </script>
35
```

Q6. Is JSONP an industrial strength solution to overcome JavaScript cross domain restriction?

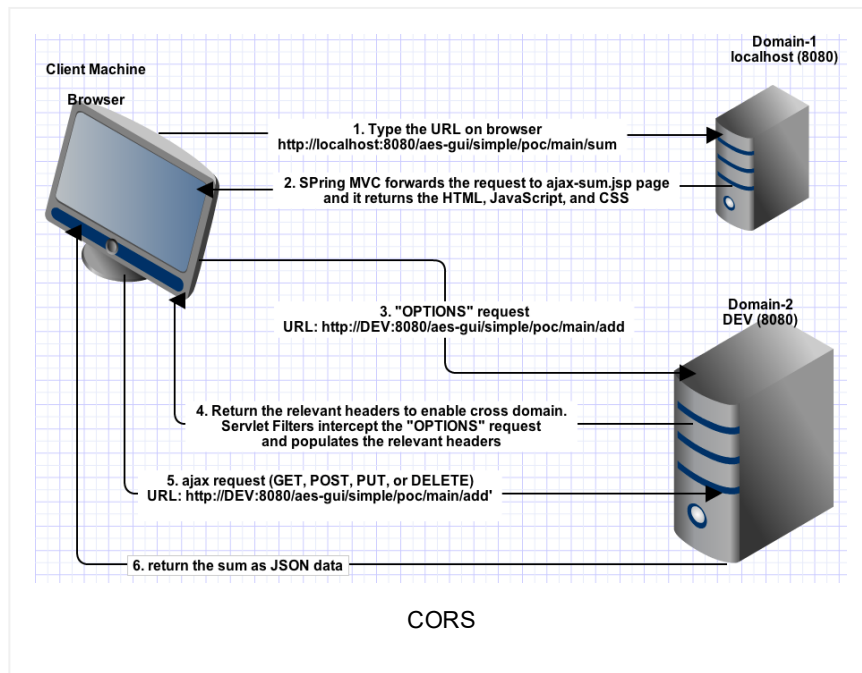
A6. No. **CORS** is the industrial strength solution for the cross domain Ajax calls.

JSONP has a number of limitations like, it supports only GET requests and not PUT, POST, DELETE, etc and it does not

also send headers across. CORS stands for **Cross Origin Resource Sharing**, which allows you to share GET, POST, PUT, and DELETE requests and CORS is supported by the modern browsers. The CORS make use of 2 requests.

Request 1: "OPTIONS" request as part of the handshake to determine if cross domain is allowed by the server.

Request 2: GET, POST, PUT, or DELETE request that performs the actual operation on the server.



Popular Posts

♦ 11 Spring boot interview questions & answers

825 views

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

767 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

240 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](https://www.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](https://www.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](#)

◀ XML Processing Interview Q&A JNDI and LDAP interview Q&A ▶

Posted in FAQ JEE Job Interview Q&A Essentials, JSON, member-paid

Tags: Architect FAQs, Java/JEE FAQs

Leave a Reply

Logged in as geethika. [Log out?](#)

Comment

Post Comment

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?](#)

© 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.