Register | Login | Logout | Contact Us

Java-Success.com

Industrial strength Java/JEE Career Companion to open more doors

search here … Go

Home | Java FAQs | 600+ Java Q&As | Career | Tutorials | Member | Why?

Can u Debug? | Java 8 ready? | Top X | Productivity Tools | Judging Experience?

Home › member-paid › 04: 5 JAXB interview Questions & Answers

04: 5 JAXB interview Questions & Answers

Posted on August 20, 2014 by Arulkumaran Kumaraswamipillai

JAXB interview Questions & Answers Links:

6 Java RESTful Web services Interview | 11 SOAP Web service interview | Java Web Services interview Questions & Answers

Q1. What does JAXB stand for? What is an XML Binding?
A1. JAXB means Java API for Xml Binding. XML binding maps an XML to in-memory Java objects. The principle advantage of using JAXB when marshalling and demarshalling XML is that is simplifies the programming model by allowing us to simply annotate a few POJOs and use the JAXB API's and you can serialise to XML and

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

open all | close all

- ⊞ Ice Breaker Interview
- ⊞ Core Java Interview (
- ⊟ JEE Interview Q&A (3
  - ⊞ JEE Overview (2)
  - ⊞ Web basics (8)
  - ⊟ WebService (11)
    - 01: ♥♦ 40+ Java
    - 02: ♦ 6 Java RES
    - 03: ♥ JAX-RS hc
    - 04: 5 JAXB inter
    - 05: RESTFul We
    - 06: RESTful Wel
    - 07: HATEOAS R
    - 08: REST constr
    - 09: 11 SOAP We
    - 10: SOAP Web S
    - 11: ♥ JAX-WS hc
  - ⊞ JPA (2)
  - ⊞ JTA (1)
  - ⊞ JDBC (4)
  - ⊞ JMS (5)

deserialise from XML very easily. This makes it much simpler than alternatives such as DOM and SAX.

Q2. Why use JAXB?

A2. JAXB is a reference interface for which you can have a number of different implementations like default reference implementation shipped JDK6, JaxMeAPI, MOXy, Metro, etc. JAXB is available in JDK6 onwards, so it doesn't require any external library and it doesn't require a XML schema to work. XML schema is optional. Easy to use as you can use annotations on your model classes. It is supported by RESful frameworks like Jersey, Spring MVC, RESTEasy, etc.

Here is an example of Unmarshalling XML to Java object:

Step 1: Sample XML to unmarshall

```xml
<employee>
    <id>123</id>
    <name>Peter</name>
</employee>

```

Step 2: POJO with JAXB annotations.

```java
import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement
public class Employee
{

    private Integer id;
    private String name;

    //JAXB requires a default constructor.
    private Employee(){}

    public Employee(Integer id, String name)
    {
        super();
        this.id = id;
        this.name = name;
    }

    public Integer getId()
    {
        return id;
    }

    public void setId(Integer id)
```

⊞ JMX (3)
⊞ JNDI and LDAP (1
⊞ Pressed for time? Jav
⊞ SQL, XML, UML, JSC
⊞ Hadoop & BigData Int
⊞ Java Architecture Inte
⊞ Scala Interview Q&As
⊞ Spring, Hibernate, & I
⊞ Testing & Profiling/Sa
⊞ Other Interview Q&A f
⊞ ▶ Free Java Interviev

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 In
⊞ Low Latency (7)
⊞ Memory Managemen
⊞ Performance (13)
⊞ QoS (8)
⊞ Scalability (4)
⊞ SDLC (6)
⊞ Security (13)
⊞ Transaction Managen

80+ step by step Java Tutorials

open all | close all
⊞ Setting up Tutorial (6)

```
    {
        this.id = id;
    }

    public String getName()
    {
        return name;
    }

    public void setName(String name)
    {
        this.name = name;
    }

    @Override
    public String toString()
    {
        return "Person [id=" + id + ", name=" +
    }
}

```

Step 3: The JAXB unmrshaller utility class that can be used to

```
import java.io.StringReader;
import javax.xml.bind.JAXBContext;
import javax.xml.bind.JAXBException;
import javax.xml.bind.Unmarshaller;
import javax.xml.transform.stream.StreamSource;

public class JAXBUnMarshaller
{
    public Object unmarshalObject(Class<?> class
    {
        Object object = new Object();
        try
        {
            StringReader stringReader = new Stri
            StreamSource streamSource = new Stre

            JAXBContext jaxbContext = JAXBContex

            Unmarshaller unMarshaller = jaxbCont
            object = unMarshaller.unmarshal(stre
            return object;
        }
        catch (JAXBException e)
        {
            e.printStackTrace();
            throw new RuntimeException("Error un
        }
    }
}

```

Step 4: The main class that unmarshalls XML string to Employee object.

⊞ 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

```
public class Test
{

    private static final String XML_STRING = "<e
            "    <id>123</id>\r\n" +
            "    <name>Peter</name>\r\n" +
            "</employee>";

    public static void main(String[] args)
    {
        JAXBUnMarshaller unmashaller = new JAXBU
        Employee unmarshalObject = (Employee) un
        System.out.println(unmarshalObject);
    }
}

```

Similar approach can be used for marshaling.

Q3. Can you create your Java objects from XSDs?
A3. Yes, you can by binding a schema. Binding a schema means generating a set of Java classes that represents the schema. All JAXB implementations provide a tool called a binding compiler to bind a schema (the way the binding compiler is invoked can be implementation-specific).

```
xjc.sh -p myapp13.model employee.xsd -d work
```

The -p option identifies a package for the generated classes, and the -d option identifies a target directory. So for this command, the classes are packaged in myapp13.model within the work directory. In response, the binding compiler generates a set of interfaces and a set of classes that implement the interfaces. The ObjectFactory.java is generated conatining methods for generating instances of the interfaces.

Q4. Why would be the motivating factor to use MOXy implementation of JAXB as opposed to the default implementation provided by the JDK6 implementation?
A4. Using the @XmlPath annotation and other extensions provided by MOXy will make your implementation cleaner.

Step 1: You need to bring in the **MOXy** implementation via maven pom.xml file

```
 <dependency>
    <groupid>org.eclipse.persistence</groupId>
    <artifactid>eclipselink</artifactId>
    <version>2.3.2</version>
</dependency>

```

Step 2: Tell JAXB to use MOXy via **jaxb.properties** file and place it on where the POJOs like Employee is packaged.

```
javax.xml.bind.context.factory=org.eclipse.persis
```

Step 3: You will only need Employee.java and Department.java will no longer required. Thanks to the power of annotation @XmlPath("faculty/name/text( )"). Here is the revised Employee.java unmarshalling the same XML file with within employee.

```
 import javax.xml.bind.annotation.XmlAccessType;
import javax.xml.bind.annotation.XmlAccessorType
import javax.xml.bind.annotation.XmlRootElement;
import org.eclipse.persistence.oxm.annotations.X

@XmlRootElement
@XmlAccessorType(XmlAccessType.FIELD)
public class Employee
{

    private Integer id;
    private String name;

    @XmlPath("faculty/name/text()")
    private String departmentName;

    public Integer getId()
    {
        return id;
    }

    public void setId(Integer id)
    {
        this.id = id;
    }

    public String getName()
    {
        return name;
    }

    public void setName(String name)
    {
        this.name = name;
    }

 protected String getDepartmentName() {
  return departmentName;
 }

 @Override
 public String toString() {
  return "Employee [id=" + id + ", name=" + name
 }
}

```

Q5. Can you use JAXB to marshal and unmarshal JSON objects?

A5. Yes. If You are using MOXy library, then you can specify a JSON mrshaling and unmarshling library like **Jackson** or **Jettison**. Here is an example with the Jettison library.

Step 1: JSON string. XML equivalent JSON is more concise.

```
{"Employee": {"id":"123", "name":"Peter"}
```

Step 2: Jettison with JAXB library.

```
import javax.xml.bind.*;
import javax.xml.stream.XMLStreamReader;
import org.codehaus.jettison.json.JSONObject;
import org.codehaus.jettison.mapped.*;

public class UnmarshalDemo {

    public static void main(String[] args) throw
        JAXBContext jc = JAXBContext.newInstance

        JSONObject obj = new JSONObject("{\"Empl
        Configuration config = new Configuration
        MappedNamespaceConvention con = new Mapp
        XMLStreamReader xmlStreamReader = new Ma

        Unmarshaller unmarshaller = jc.createUnm
        Employee employee = (Employee) unmarshal

        Marshaller marshaller = jc.createMarshal
        marshaller.setProperty(Marshaller.JAXB_F
        marshaller.marshal(employee, System.out)
    }
}

```

JAXB interview Questions & Answers Links:

6 Java RESTful Web services Interview | 11 SOAP Web service interview | Java Web Services interview Questions & Answers

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**

‹ 02: ♦ 6 Java RESTful Web services Interview Q&A you must know

♦ 12 UML interview Questions & Answers ›

Posted in member-paid**,** WebService

Tags: Java/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.