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 › XSD reuse interview questions and answers

XSD reuse interview questions and answers

Posted on April 14, 2015 by Arulkumaran Kumaraswamipillai

#1. Reuse by defining a "ComplexType". EmployeeType is reused.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.w3.org/2001/XMLSch
    xmlns:tns="http://www.myapp.com/Person" xmln
    elementFormDefault="qualified">

    <!-- XML root element -->
    <xsd:element name="CompanyRequest" type="tns

    <xsd:complexType name="CompanyType">
        <xsd:sequence>
            <xsd:element name="PermanentStaff" t
            <xsd:element name="ContractStaff" ty
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="EmployeeType">
        <xsd:sequence>
            <xsd:element name="firstName" type="
            <xsd:element name="lastName" type="s
```

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
⊞ Pressed for time? Jav
⊟ SQL, XML, UML, JSC
⊞ ERD (1)
⊞ NoSQL (2)
⊞ Regex (2)
⊞ SQL (7)
⊞ UML (1)
⊞ JSON (2)
⊞ XML (2)
⊟ XSD (2)
11 FAQ XSD inte
XSD reuse interv
⊞ YAML (2)
⊞ Hadoop & BigData Int
⊞ Java Architecture Inte
⊞ Scala Interview Q&As
⊞ Spring, Hibernate, & I
⊞ Testing & Profiling/Sa

```
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>

```

Once JAXB annotated classes are created

```
public void process() throws XmlMappingException
        CompanyRequest req = new CompanyRequest(

        EmployeeType ps = new EmployeeType();
        ps.setFirstName("Peter");
        ps.setLastName("Smith");


        EmployeeType cs = new EmployeeType();
        cs.setFirstName("John");
        cs.setLastName("Lewis");

        req.setPermanentStaff(ps);
        req.setContractStaff(cs);

        StringResult result = new StringResult()
        marshaler.marshal(req, result);
        System.out.println(result.toString());
}

```

XML output:

```
<?xml version="1.0" encoding="UTF-8" standalone=
<CompanyRequest xmlns="http://www.myapp.com/Pers
   <PermanentStaff>
       <firstName>Peter</firstName>
       <lastName>Smith</lastName>
   </PermanentStaff>
   <ContractStaff>
       <firstName>John</firstName>
       <lastName>Lewis</lastName>
   </ContractStaff>
</CompanyRequest>

```

#2. Inheritance reuse where sub types can define additional elements like “salary”, “contractRate”, etc.

⊞ 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 Ir

⊞ 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)

⊞ Tutorial - Diagnosis (2

⊞ Akka Tutorial (9)

⊞ Core Java Tutorials (2

⊞ Hadoop & Spark Tuto

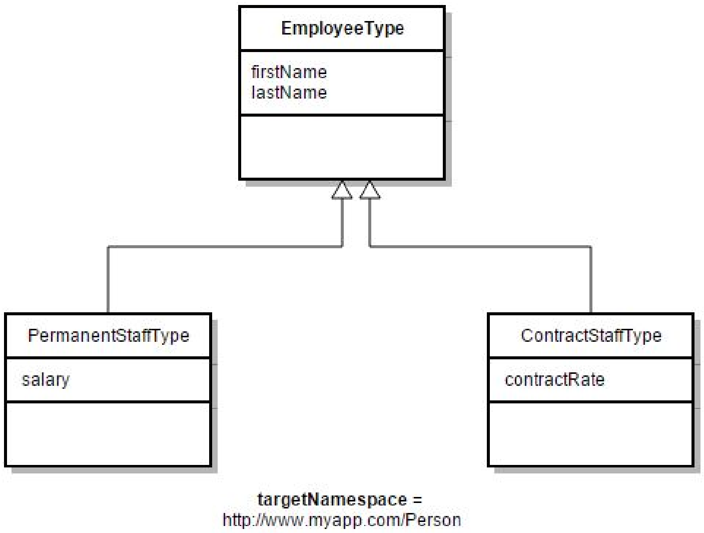
⊞ JEE Tutorials (19)

⊞ Scala Tutorials (1)

⊞ Spring & HIbernate Tu

⊞ Tools Tutorials (19)

⊞ Other Tutorials (45)



```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.w3.org/2001/XMLSch
    xmlns:tns="http://www.myapp.com/Person" xmln
    elementFormDefault="qualified">

    <!-- XML root element -->
    <xsd:element name="CompanyRequest" type="tns

    <xsd:complexType name="CompanyType">
        <xsd:sequence>
            <xsd:element name="PermanentStaff" t
            <xsd:element name="ContractStaff" ty
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="EmployeeType">
        <xsd:sequence>
            <xsd:element name="firstName" type="
            <xsd:element name="lastName" type="s
        </xsd:sequence>
    </xsd:complexType>


    <xsd:complexType name="PermanentStaffType">
       <xsd:complexContent>
            <xsd:extension base="tns:EmployeeTyp
                <xsd:sequence>
                    <xsd:element name="salary"
                </xsd:sequence>
            </xsd:extension>
       </xsd:complexContent>
    </xsd:complexType>

     <xsd:complexType name="ContractStaffType">
       <xsd:complexContent>
            <xsd:extension base="tns:EmployeeTyp
                <xsd:sequence>
                    <xsd:element name="contract
                </xsd:sequence>
            </xsd:extension>
```

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

```
       </xsd:complexContent>
    </xsd:complexType>

</xsd:schema>

```

Once the JAXB annotated classes are created

```
public void process() throws XmlMappingException
        CompanyRequest req = new CompanyRequest(

        PermanentStaffType ps = new PermanentSta
        ps.setFirstName("Peter");
        ps.setLastName("Smith");
        ps.setSalary(300.00);

        ContractStaffType cs = new ContractStaff
        cs.setFirstName("John");
        cs.setLastName("Lewis");
        cs.setContractRate(25.00);

        req.setPermanentStaff(ps);
        req.setContractStaff(cs);

        StringResult result = new StringResult()
        marshaler.marshal(req, result);
        System.out.println(result.toString());
}

```

XML output:

```
<?xml version="1.0" encoding="UTF-8" standalone=
<CompanyRequest xmlns="http://www.myapp.com/Pers
  <PermanentStaff>
     <firstName>Peter</firstName>
     <lastName>Smith</lastName>
     <salary>300.0</salary>
  </PermanentStaff>
  <ContractStaff>
     <firstName>John</firstName>
     <lastName>Lewis</lastName>
     <contractRate>25.0</contractRate>
  </ContractStaff>
</CompanyRequest>

```

#3. using "**substitutionGroup**"

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.w3.org/2001/XMLSch
    xmlns:tns="http://www.myapp.com/Person" xmln
    elementFormDefault="qualified">

    <!-- XML root element -->
    <xsd:element name="CompanyRequest" type="tns
    
    <xsd:element name="Employee" type="tns:Emplo
    <xsd:element name="PermanentStaff" type="tns
    <xsd:element name="ContractStaff" type="tns:
    
    <xsd:complexType name="CompanyType">
        <xsd:sequence>
             <xsd:element ref="tns:Employee"  ma
        </xsd:sequence>
    </xsd:complexType>
    
    <xsd:complexType name="EmployeeType" abstrac
        <xsd:sequence>
            <xsd:element name="firstName" type="
            <xsd:element name="lastName" type="s
        </xsd:sequence>
    </xsd:complexType>
    
    
    <xsd:complexType name="PermanentStaffType">
        <xsd:complexContent>
            <xsd:extension base="tns:EmployeeTyp
                <xsd:sequence>
                    <xsd:element name="salary" t
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    
    <xsd:complexType name="ContractStaffType">
        <xsd:complexContent>
            <xsd:extension base="tns:EmployeeTyp
                <xsd:sequence>
                    <xsd:element name="contractR
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    
</xsd:schema>

```

and Java class that make use of the JAXB generated clasess

```
public void process() throws XmlMappingException
        CompanyRequest req = new CompanyRequest(

        PermanentStaffType ps = new PermanentSta
        ps.setFirstName("Peter");
        ps.setLastName("Smith");
        ps.setSalary(300.00);

        ContractStaffType cs = new ContractStaff
        cs.setFirstName("John");
        cs.setLastName("Lewis");
        cs.setContractRate(25.00);

        ObjectFactory of = new ObjectFactory();

        JAXBElement<PermanentStaffType> permStaf
        JAXBElement<ContractStaffType> contractS

        List<JAXBElement<? extends EmployeeType>
        employees.add(permStaff);
        employees.add(contractStaff);

        req.setEmployees(employees);


        StringResult result = new StringResult()
        marshaler.marshal(req, result);
        System.out.println(result.toString());
}

```

XML output:

```
<?xml version="1.0" encoding="UTF-8" standalone=
<CompanyRequest xmlns="http://www.myapp.com/Pers
    <PermanentStaff>
       <firstName>Peter</firstName>
       <lastName>Smith</lastName>
       <salary>300.0</salary>
    </PermanentStaff>
    <ContractStaff>
      <firstName>John</firstName>
      <lastName>Lewis</lastName>
      <contractRate>25.0</contractRate>
    </ContractStaff>
</CompanyRequest>

```

Q. How did we generate the JAXB annotated classes?

A. Refer to Basic JAXB tutorial with Maven & XSD – 1

Popular Posts

♦ 11 Spring boot interview questions & answers

828 views

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

768 views

18 Java scenarios based interview Questions and Answers

400 views

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

389 views

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

296 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

286 views

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

280 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

202 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**

‹ Debugging SSL issues in Java & tools to debug

5 Quotes that motivate me & leaders who inspire me ›

Posted in member-paid, XSD

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.