Controlling Bindings with JAXB Annotations

Jesper de Jong
SOFTWARE ARCHITECT

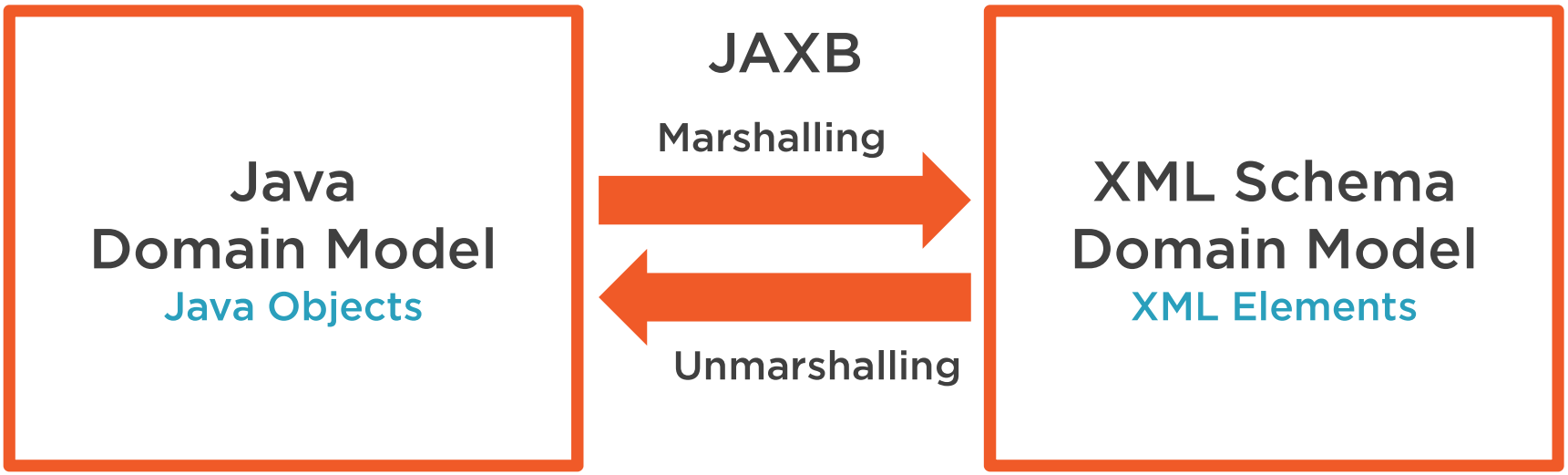
@jesperdj www.jesperdj.com

Overview

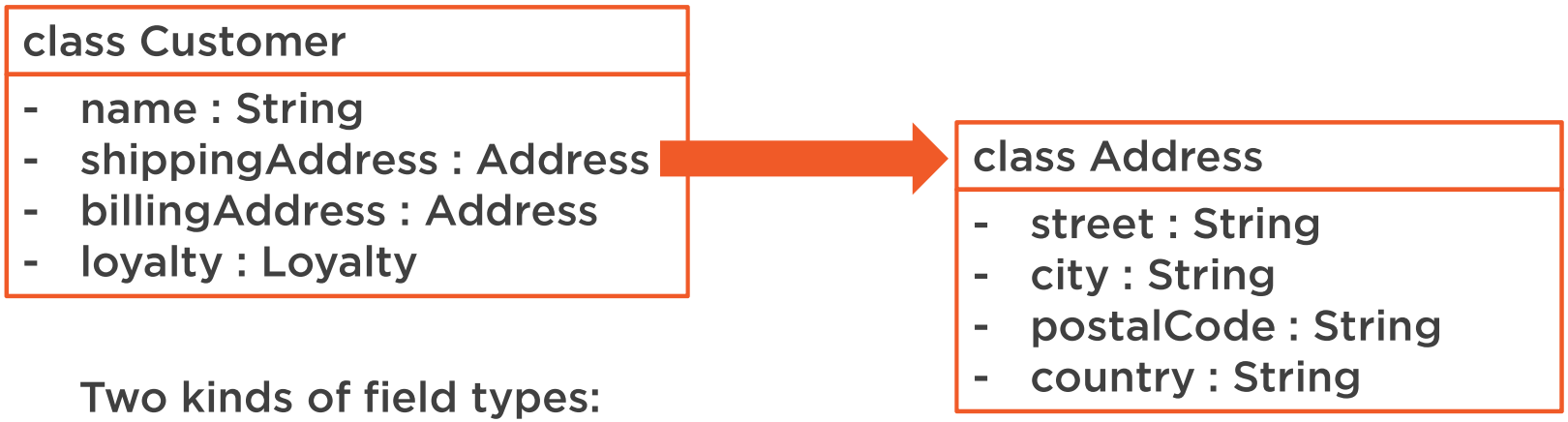
- JAXB Annotations
- How JAXB Handles Domain Model Classes
- Elements and Attributes
- Element Ordering
- Mapping Classes to Simple Types
- Enums
- Collections
- Adapters

How JAXB Handles Domain Model Classes

Domain Model Representations



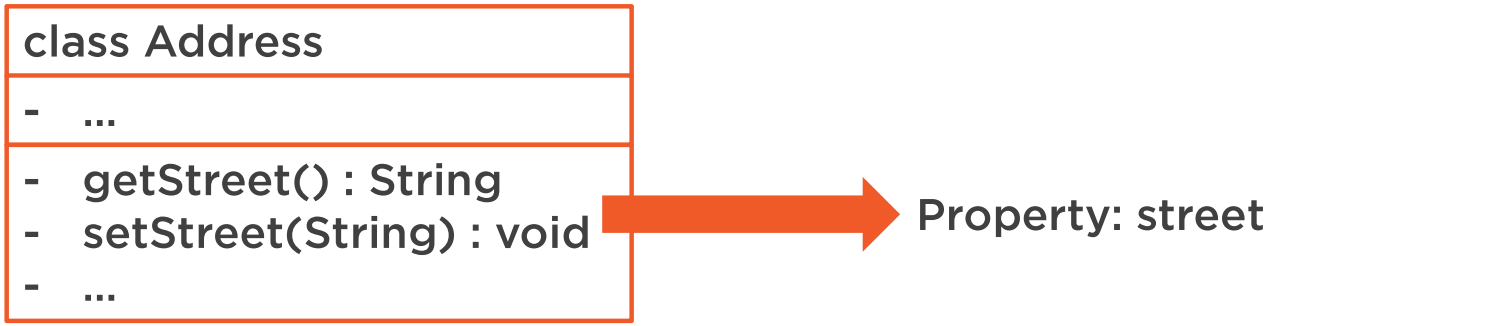
Java Domain Model



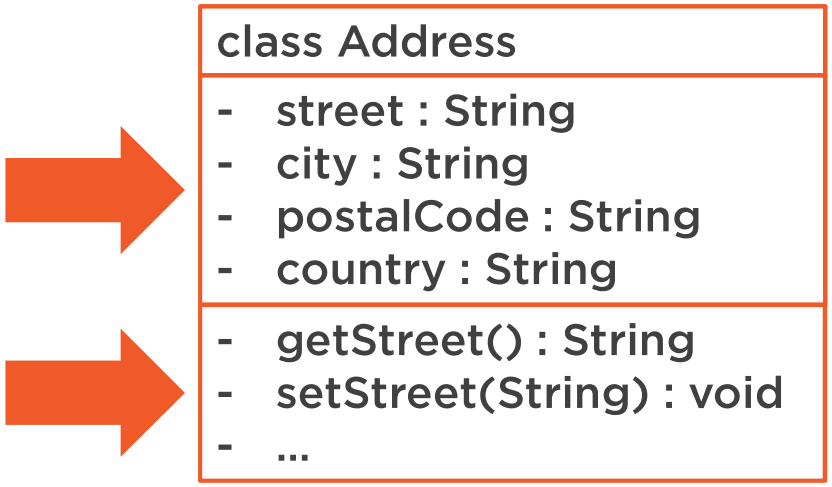
Two kinds of field types:

- Value types
- Class types

Java Domain Model - JavaBeans



Java Domain Model - JAXB



Java Domain Model

Annotating a field

```
@XmlElement
private String street;
```

Annotating a property

```
@XmlElement
public void getStreet() {
    // ...
}
```

XML Schema Domain Model

Simple and Complex Types

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<purchaseOrder>
    <customer>
        <billingAddress>
            <city>Exampleville</city>
            <country>USA</country>
            <postalCode>12345</postalCode>
            <street>123 Main Street</street>
        </billingAddress>
        <loyalty>SILVER</loyalty>
        <name>John Doe</name>
        <shippingAddress>
            <city>Exampleville</city>
            <country>USA</country>
            <postalCode>12345</postalCode>
            <street>123 Main Street</street>
        </shippingAddress>
    </customer>
```

Summary

- Two ways to look at domain model classes
  - Access fields directly
  - Access as JavaBeans properties
- Two kinds of types
  - Value types – Simple types
  - Class types – Complex types

Annotations for Elements and Attributes

Annotations for Elements and Attributes

```
@XmlRootElement
@XmlAccessorType
@XmlElement
@XmlTransient
@XmlAttribute
```

ps-jaxb-04-anno-basic

XmlAccessorType

- Direct field access or JavaBeans properties
- Default: Public fields and JavaBeans properties
- Use @XmlAccessorType to specify behavior
- Can be used on a class or on a package

Annotations for Element Ordering

Order of XML Elements

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<purchaseOrder>
    <customer>
        <billingAddress>
            <city>Exampleville</city>
            <country>USA</country>
            <postalCode>12345</postalCode>
            <street>123 Main Street</street>
        </billingAddress>
        <loyalty>SILVER</loyalty>
        <name>John Doe</name>
        <shippingAddress>
            <city>Exampleville</city>
            <country>USA</country>
            <postalCode>12345</postalCode>
            <street>123 Main Street</street>
        </shippingAddress>
    </customer>
    <items>
        <comment>Blue ink</comment>
        <price>8.95</price>
        <productName>Ballpoint Pen</productName>
        <quantity>20</quantity>
    </items>
    <items>
        <price>2.95</price>
        <productName>Pencil</productName>
        <quantity>10</quantity>
    </items>
</purchaseOrder>
```

Annotations for Ordering Elements

```
@XmlAccessorOrder
@XmlType
```

ps-jaxb-05-anno-order

Mapping Classes to Simple Types

Annotations for Simple Types

```
@XmlSchemaType
@XmlValue
```

ps-jaxb-06-anno-simpletypes

Mapping Enums

Annotations for Simple Types

```
@XmlEnum
@XmlEnumValue
```

ps-jaxb-07-anno-enum

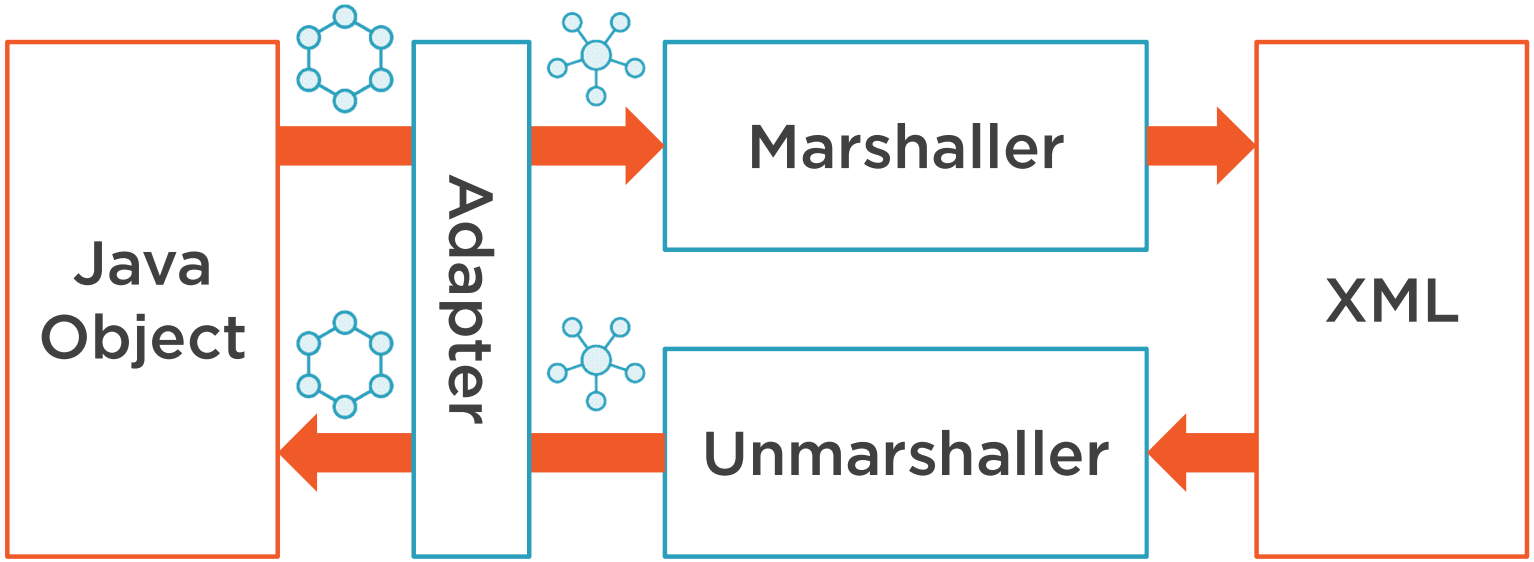
Mapping Collections

Annotations for Collections

ps-jaxb-08-anno-lists-1
ps-jaxb-09-anno-lists-2
ps-jaxb-10-anno-maps

Custom Mapping with Adapters

Custom Mapping with Adapters



ps-jaxb-11-anno-adapters

Summary

Direct field access vs JavaBeans

Java model and XML Schema model

- Value types – Simple types
- Class types – Complex types

Elements and attributes

Element Ordering

Mapping to Simple Types

Mapping Enums

Mapping Collections

Adapters

Coming Up

Generating an XML Schema from Java Classes