

# Smooks Example - xml-to-java

The primary goal of this example is to illustrate how [Smooks](#) can be used to transform XML into a Java Object Graph.

- [The Input Message](#)
- [The Java Beans \(POJOs\)](#)
- [The Smooks Configuration](#)
- [Executing Smooks](#)

## SVN - Download - Other Tutorials

To Build: "mvn clean install"

To Run: "mvn exec:java"

**Additional information relevant to this example:**

1. Check out the ["xml-to-java"](#) example.
2. Check out the [javadoc for the Javabean Cartridge](#) (see the BeanPopulator class).

## The Input Message

The input message is an order as follows:

```
<order>
  <header>
    <date>Wed Nov 15 13:45:28 EST 2006</date>
    <customer number="123123">Joe</customer>
  </header>
  <order-items>
    <order-item>
      <product>111</product>
      <quantity>2</quantity>
      <price>8.90</price>
    </order-item>
    <order-item>
      <product>222</product>
      <quantity>7</quantity>
      <price>5.20</price>
    </order-item>
  </order-items>
</order>
```

## The Java Beans (POJOs)

We have 3 basic types - Order, Header and OrderItem.

The Order (getters and setters not shown) contains a Header and a list of OrderItems:

```
public class Order {
    private Header header;
    private List<OrderItem> orderItems;
}
```

The Header (getters and setters not shown):

```
public class Header {
    private Date date;
    private Long customerNumber;
    private String customerName;
}
```

The OrderItem (getters and setters not shown):

```
public class OrderItem {
    private long productId;
    private Integer quantity;
    private double price;
}
```

## The Smooks Configuration

The Smooks config required to bind the data from the Input Message into the Order object graph is as follows:

```
<!-- smooks-config.xml -->
<?xml version="1.0"?>
<smooks-resource-list xmlns="http://www.milyn.org/xsd/smooks-1.1.xsd"
xmlns:jb="http://www.milyn.org/xsd/smooks/javabean-1.1.xsd">

    <!--
    Create an "example.beans.Order" bean instance when we visit the start
of the <order> element.
    Assign the new bean instance to the beanId of "order".
    Wire in the "header" and "orderItems" beans.
    -->
    <jb:bindings beanId="order" class="example.model.Order"
createOnElement="order">
        <jb:wiring property="header" beanIdRef="header" />
        <jb:wiring property="orderItems" beanIdRef="orderItems" />
    </jb:bindings>

    <!--
    Create an ArrayList bean instance when we visit the start of the
<order> element.
    This bean is wired into the "order" bean.
    -->
    <jb:bindings beanId="orderItems" class="java.util.ArrayList"
createOnElement="order">
        <jb:wiring beanIdRef="orderItem" />
    </jb:bindings>

    <!--
    Create an "example.beans.Header" bean instance when we visit the start
of the <header> element.
```

```
This bean is wired into the "order" bean.
-->
<jb:bindings beanId="header" class="example.model.Header"
createOnElement="header">
  <jb:value property="date" decoder="Date" data="header/date">
    <jb:decodeParam name="format">EEE MMM dd HH:mm:ss z
yyyy</jb:decodeParam>
    <jb:decodeParam name="locale-language">en</jb:decodeParam>
    <jb:decodeParam name="locale-country">IE</jb:decodeParam>
  </jb:value>
  <jb:value property="customerNumber" decoder="Long"
data="header/customer/@number" />
  <jb:value property="customerName" data="header/customer" />
</jb:bindings>

<!--
  Create an "example.beans.OrderItem" bean instance when we visit the
  start of the <order-item> element.
  This bean is wired into the "orderItems" ArrayList bean.
-->
<jb:bindings beanId="orderItem" class="example.model.OrderItem"
createOnElement="order-item">
  <jb:value property="productId" decoder="Long"
data="order-item/product" />
  <jb:value property="quantity" decoder="Integer"
data="order-item/quantity" />
  <jb:value property="price" decoder="Double" data="order-item/price"
/>
</jb:bindings>
```

```
</smooks-resource-list>
```

For more info on Javabeen decoders, see the [DataDecoder Javadocs](#).

## Executing Smooks

As with the [java-basic](#) tutorial, we use the [Smooks](#) class as follows:

```
// Instantiate Smooks with the config...
Smooks smooks = new Smooks("smooks-config.xml");
JavaResult javaResult = new JavaResult();

// Filter the input message to the JavaResult...
smooks.filter(new StreamSource(messageIn), javaResult);

// Extract the Order bean from the JavaResult using the beanId...
Order order = (Order) javaResult.getBean("order");
```

Of course, you'd typically cache the [Smooks](#) instance.

See the [example/Main.java](#) in the example source.