

How to transfer *Generic Collection* in Web Service

Problem:

When `Java.util.List<T>` is mapped to a WSDL document, the schema looks like

```
<xs:element maxOccurs="unbounded" minOccurs="0" name="data" nillable="true" type="xs:anyType"/>
```

When you use `wsimport` to generate stub and other client classes, the schema is mapped to `List<Object>`. but in the runtime, the objects are in the format of XML elements. So developers have to map the XML Elements to meaningful objects, which could be a very hard task.

Adding `@XmlSeeAlso` will only map additional classes to the WSDL schema. The web service runtime will not automatically map the XML elements to the classes.

Solution 1:

So we could create a derived classed of `Java.util.ArrayList<T>`

for example,

```
public class MyList extends Java.util.List<MyVO>
```

and `MyList` will be mapped to

```
<xs:element maxOccurs="unbounded" minOccurs="0" name="data" nillable="true" type="MyVO"/>
```

The generated class can map list elements to `MyVO` objects.

Solution 2

Use `@XmlTransient` and `@XmlElement`

Here are some .

```
public abstract class AbstractPagedList<T> {  
    protected Integer rowCount;  
    protected Integer rowIndex;  
    protected Integer totalCount;
```

```

@XmlTransient
abstract public List<T> getData();

public Integer getRowCount() {
    return rowCount;
}

public Integer getRowIndex() {
    return rowIndex;
}

public Integer getTotalCount() {
    return totalCount;
}

abstract public void setData(List<T> data);

public void setRowCount(Integer rowCount) {
    this.rowCount = rowCount;
}

public void setRowIndex(Integer rowIndex) {
    this.rowIndex = rowIndex;
}

public void setTotalCount(Integer totalCount) {
    this.totalCount = totalCount;
}
}

public class GenericPagedList<T> extends AbstractPagedList<T>{
    private static final long serialVersionUID = -3561936664480969952L;
    protected List<T> data;

    public List<T> getData() {
        return data;
    }

    public void setData(List<T> data) {
        this.data = data;
    }
}

public class LdapAuthInfoPagedList extends
    SerializablePagedList<LdapAuthInfoVO> {
    private static final long serialVersionUID = -2950098887571376141L;

    @Override
    @XmlElement
    public List<LdapAuthInfoVO> getData() {
        return super.getData();
    }

    @Override
    public void setData(List<LdapAuthInfoVO> data) {
        super.setData(data);
    }
}

```

```

}

public class LdapAuthInfoPagedList extends
    SerializablePagedList<LdapAuthInfoV0> {
    private static final long serialVersionUID = -2950098887571376141L;

    @Override
    public List<LdapAuthInfoV0> getData() {
        return super.getData();
    }

    @Override
    public void setData(List<LdapAuthInfoV0> data) {
        super.setData(data);
    }
}

```

Without @XmlTransient and @XmlElement, the generated WSDL looks like:

```

<xs:complexType name='ldapAuthInfoPagedList'>
<xs:complexContent>
  <xs:extension base='tns:serializablePagedList'>
    <xs:sequence/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name='serializablePagedList'>
<xs:complexContent>
  <xs:extension base='tns:genericPagedList'>
    <xs:sequence/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name='genericPagedList'>
<xs:complexContent>
  <xs:extension base='tns:abstractPagedList'>
    <xs:sequence/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType abstract='true' name='abstractPagedList'>
  <xs:sequence>
    <xs:element maxOccurs='unbounded' minOccurs='0' name='data' nillable='true'
type='xs:anyType' />
    <xs:element minOccurs='0' name='rowCount' type='xs:int' />
    <xs:element minOccurs='0' name='rowIndex' type='xs:int' />
    <xs:element minOccurs='0' name='totalCount' type='xs:int' />
  </xs:sequence>
</xs:complexType>

```

The runtime can't map XML Elements to LdapAuthInfoV0 for LdapAuthInfoPagedList.

If @XmlTransient is added to List<T> getData() in abstractPagedList, List<T> getData() will not be mapped to WSDL for abstractPagedList and its derived classes if @XmlElement is not found on the override function. The real generated WSDL will look like

```
<xs:complexType name='ldapAuthInfoPagedList'>
  <xs:complexContent>
    <xs:extension base='tns:serializablePagedList'>
      <xs:sequence>
        <xs:element maxOccurs='unbounded' minOccurs='0' name='data'
type='tns:ldapAuthInfoVO' />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name='serializablePagedList'>
  <xs:complexContent>
    <xs:extension base='tns:genericPagedList'>
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name='genericPagedList'>
  <xs:complexContent>
    <xs:extension base='tns:abstractPagedList'>
      <xs:sequence/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType abstract='true' name='abstractPagedList'>
  <xs:sequence>
    <xs:element minOccurs='0' name='rowCount' type='xs:int' />
    <xs:element minOccurs='0' name='rowIndex' type='xs:int' />
    <xs:element minOccurs='0' name='totalCount' type='xs:int' />
  </xs:sequence>
</xs:complexType>
```