

Διαδίκτυο και Εφαρμογές

3<sup>η</sup> Εργαστηριακή Άσκηση

Υπηρεσίες web με χρήση SOAP

---



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ  
ΠΟΛΥΤΕΧΝΕΙΟ**

8<sup>ο</sup> Εξάμηνο  
2017-2018

Λεούσης Σάββας  
Α.Μ. 03114945

## Υπηρεσίες web με χρήση SOAP

Στην παρούσα άσκηση ζητήθηκε η υλοποίηση μιας υπηρεσίας SOAP-RPC, , με βάση με τους κώδικες των σημειώσεων, η οποία διαχειρίζεται εγγραφές σε μορφή JavaBean με την επέκταση, ότι το JavaBean, πέρα από βασικές μεταβλητές ως ιδιότητες, θα έχει και ένα δεύτερο JavaBean ενσωματωμένο.

Συγκεκριμένα, η δομή του JavaBean VehicleBean θα είναι η εξής:

- String VModel
- ManBean VMan
- String VYear

όπου η δομή του VMan είναι:

- String ManName
- Integer ManYear
- String ManCountry

Το Hashtable στην υπηρεσία του server έχει ένα πεδίο για το VModel, ένα πεδίο για τον κατασκευαστή (Manufacturer) του κάθε Vehicle, και ένα πεδίο για το VYear. Παρακάτω παρατίθενται οι τροποποιημένοι κώδικες του παραδείγματος των σημειώσεων (VehicleBean.java, ManBean.java, BVCatalog.java, BVAdderLister.java) , καθώς και ο κατάλληλος Deployment Descriptor της υπηρεσίας σε .xml, ώστε να ικανοποιούνται οι παραπάνω προδιαγραφές.

```

1 package BVShop;
2 //if present 'package' MUST be the first statement (possibly) followed by 'import'
3 // The following class follows the structure of a 'Java Bean'
4
5 public class VehicleBean {
6     /**The properties of the VehicleBean*/
7     String VModel;
8     ManBean VMan;
9     String VYear;
10
11     /** The following non-argument constructor MUST be always present*/
12     /** for this class to be a Java Bean*/
13     public VehicleBean(){}
14
15     /** Another constructor CAN also be always present*/
16     /** the following initializes all properties whenever a new object is
instantiated (with 'new') - */
17     public VehicleBean(String model, ManBean manu, String year){
18         VModel= model; VMan = manu; VYear= year;}
19
20     /** get & set methods for all properties MUST be present*/
21     public String getVModel(){return VModel;}
22     public void setVModel(String model){VModel= model;}
23     public ManBean getVManufacturer(){return VMan;}
24     public void setVManufacturer(ManBean manu){VMan= manu;}
25     public String getVYear(){return VYear;}
26     public void setVYear(String year){VYear= year;}
27
28     //toString is an in-built method for every Java object. Outputs an
informative string
29     public String toString(){

```

```

30         return "" + VModel + " manufactured by " + VMan.toString() + " during year
        " + VYear; }
31 }

```

Listing 1: Ο κώδικας VehicleBean.java

```

1 package BVShop;
2 //if present 'package' MUST be the first statement (possibly) followed by 'import'
3 // The following class follows the structure of a 'Java Bean'
4
5 public class ManBean {
6     /**The properties of the VehicleBean*/
7     String ManName;
8     Integer ManYear;
9     String ManCountry;
10
11     /** The following non-argument constructor MUST be always present*/
12     /** for this class to be a Java Bean*/
13     public ManBean(){ }
14
15     /** Another constructor CAN also be always present*/
16     /** the following initializes all properties whenever a new object is
17     instantiated (with 'new') - */
18     public ManBean(String name,Integer year,String country){
19         this.ManName= name; this.ManYear = year; this.ManCountry= country;}
20
21     /** get & set methods for all properties MUST be present*/
22     public String getManName(){return ManName;}
23     public void setManName(String name){this.ManName = name;}
24     public String getManYear(){return ManYear;}
25     public void setManYear(Integer year){this.ManYear = year;}
26     public String getManCountry(){return ManCountry;}
27     public void setManCountry(String country){this.ManCountry = country;}
28
29     /**toString is an in-built method for every Java object. Outputs an
30     informative string
31     public String toString(){
32         return "" + ManName + " in " + ManCountry + " (est. " + ManYear.toString
33         () + ")"; }
34 }

```

Listing 2: Ο κώδικας ManBean.java

```

1 package BVShop;
2 /*if present 'package' MUST be the first statement,*/
3 /* (possibly) followed by 'import'*/
4 import java.util.Hashtable;
5
6 public class BVCatalog {
7     /**The vehicles as Hashtable*/
8     /** the car Model is the key for finding entries */
9     private Hashtable catalog;
10
11     public BVCatalog(){
12         /** the old catalog remains*/
13         /** but now the catalog (Hashtable) will contain also beans !!!*/
14         catalog=new Hashtable();
15
16         /**Some content - we NOW polulate with some objects!!!*/

```

```

17         addV(new VehicleBean("Buick",new ManBean("General Motors",1908,"USA"),"
18         1948"));
19         addV(new VehicleBean("Mustang",new ManBean("Ford",1903,"USA"),"1960"));
20         addV(new VehicleBean("4CV",new ManBean("Citroen",1919,"France"),"1950"));
21         addV(new VehicleBean("Jeep",new ManBean("General Motors",1908,"USA"), "
22         1942"));
23         addV(new VehicleBean("Beatle",new ManBean("Volkswagen",1937,"Germany"),"
24         1938"));
25     }
26     /** FIRST METHOD TO BE EXPOSED - addV */
27     /** input argument is now a single ('complex') object !***/
28     /** nothing is returned*/
29     public void addV(VehicleBean vObj) {
30         if (vObj == null){
31             throw new IllegalArgumentException("The object provided cannot be
32             null.");
33         }
34         /** entry format for the Hashtable: key,data */
35         /** vObj.getVModel gets the 'key' out of the bean object*/
36         catalog.put(vObj.getVModel(),vObj);
37         System.out.println("Addition at server side: " + vObj.getVModel());
38     }
39
40     /** SECOND METHOD TO BE EXPOSED - getVehicleBean */
41     /** input argument is a String */
42     /** a ('complex') object is returned */
43     public VehicleBean getVehicleBean(String model) {
44         if (model==null){
45             throw new IllegalArgumentException("carModel cannot be null.");
46         }
47
48         //Return the requested vehicle - NOW AS AN OBJECT !!!!
49         return (VehicleBean)catalog.get(model);
50     }
51
52     /** THIRD METHOD TO BE EXPOSED - listV */
53     /** NO input argument - a whole table is returned */
54     /** ... , and that with ('complex') objects as entries !*** */
55     public Hashtable listV(){return catalog;}
56 }

```

Listing 3: Ο κώδικας BVCatalog.java

```

1 package BVShop;
2 import java.net.URL;
3 import java.util.Enumeration;
4 import java.util.Hashtable;
5 import java.util.Vector;
6 import org.apache.soap.Constants;
7 import org.apache.soap.Fault;
8 import org.apache.soap.SOAPException;
9
10 import org.apache.soap.encoding.SOAPMappingRegistry;
11 import org.apache.soap.encoding.soapenc.BeanSerializer;
12
13 import org.apache.soap.rpc.Call;
14 import org.apache.soap.rpc.Parameter;
15 import org.apache.soap.rpc.Response;
16
17 import org.apache.soap.util.xml.QName;

```

```

18
19 public class BVAdderLister {
20
21     public void addlist(URL url, String vmodel,String mname, Integer myear,
22         String mcountry, String vyear )
23         throws SOAPException{
24
25         // Build the object with the data given by user
26         VehicleBean vObj = new VehicleBean(vmodel, new ManBean(mname,myear,
27             mcountry), vyear);
28
29         //VehicleBean must now be mapped ... so SOAP can use it
30         SOAPMappingRegistry reg = new SOAPMappingRegistry();
31         BeanSerializer serializer = new BeanSerializer();
32         reg.mapTypes(Constants.NS_URI_SOAP_ENC,
33             new QName("urn:VBean_xmlns","vObj"),
34             VehicleBean.class, serializer, serializer);
35         reg.mapTypes(Constants.NS_URI_SOAP_ENC,
36             new QName("urn:MBean_xmlns","mObj"),
37             ManBean.class, serializer, serializer);
38         //Build the Call object
39         Call call = new Call();
40         //How to map, where to send, method to call, encoding "style"
41         call.setSOAPMappingRegistry(reg);
42         call.setTargetObjectURI("urn:BVehicleCatalog");
43         call.setMethodName("addV");
44         call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);
45
46         //----- A D D I N G -----
47         System.out.println("Adding vehicle model '" + vmodel + "' by " + mname);
48
49         // Set up the parameters of the call
50         Vector params = new Vector();
51
52         //in the instructions given to the 'Serializer' - see Depl. Descr.
53         params.addElement(new Parameter("vObj", VehicleBean.class, vObj, null));
54         call.setParams(params);
55
56         // Invoke the call
57         Response response;
58         response = call.invoke(url, "");
59         //We do not expect something back, unless there is a fault!!
60         if (!response.generatedFault())
61             {
62                 System.out.println("Server reported NO FAULT while adding
63                 vehicle");}
64         else {
65             Fault fault = response.getFault();
66             System.out.println("Server reported FAULT while adding:");
67             System.out.println(fault.getFaultString());}
68
69         //----- L I S T I N G -----
70         //We use the same Call Object and change this as appropriate
71         /* Another method is now called*/
72         call.setMethodName("listV");
73         /* NO parameters here !!*/
74         /*(we cannot have a call with arguments as before)*/
75         call.setParams(null);
76         // Invoke the call; here we expect something back !!
77         response = call.invoke(url, "");

```

```

75
76      /*Extract the value returned in the form of a 'Parameter' Object*/
77      Parameter returnValue = response.getReturnValue();
78      /*Cast the 'Parameter' Object onto a Hashtable Object*/
79      Hashtable catalog = (Hashtable)returnValue.getValue();
80      Enumeration e = catalog.keys();
81      while (e.hasMoreElements()) {
82          String VModel = (String)e.nextElement();
83          VehicleBean vo = (VehicleBean)catalog.get(VModel);
84          System.out.println("  " + vo.getVModel() + " by " + vo.
getVManufacturer().getManName() + "(" + vo.getVManufacturer().getManCountry() +
" " + vo.getVManufacturer().getManYear() + "), year " + vo.getVYear());
85      }
86  }
87
88
89
90      public static void main(String[] args) {
91          if (args.length != 6)
92              {System.out.println("Put url, model, manufacturer name,year,county &
model year as arguments !!");
93                  return;}
94
95          try {
96              // URL for SOAP server to connect to
97              URL urlink = new URL(args[0]);
98              // Get values for the new vehicle
99              String model = args[1];
100              String manufacturer_name = args[2];
101              Integer manufacturer_year = Integer.parseInt(args[3]);
102              String manufacturer_country = args[4];
103              String year = args[5];
104              //Add the new vehicle,
105              BVAdderLister adderlist = new BVAdderLister();
106              adderlist.addlist(urlink, model, manufacturer_name,
manufacturer_year, manufacturer_country, year);
107          } catch (Exception e) {e.printStackTrace();}
108      }
109  }

```

Listing 4: Ο κώδικας BVAdderLister.java

```

1  <isd:service xmlns:isd ="http://xml.apache.org/xml-soap/deployment" id = "
    urn:BVehicleCatalog">
2
3      <isd:provider type="java" scope ="Application" methods="addV getVehicleBean
listV">
4          <isd:java class="BVShop.BVCatalog" static="false" />
5      </isd:provider>
6
7      <isd:faultListener>org.apache.soap.server.DOMFaultListener</isd:faultListener
>
8
9      <isd:mappings>
10         <isd:map encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
11             xmlns:x="urn:VBean_xmlns" qname="x:vObj"
12             javaType="BVShop.VehicleBean"
13             java2XMLClassName="org.apache.soap.encoding.soapenc.
BeanSerializer"
14             xml2JavaClassName="org.apache.soap.encoding.soapenc.

```

```

15     BeanSerializer"/>
16         <isd:map encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
17             xmlns:x="urn:MBean_xmlns" qname="x:mObj"
18             javaType="BVShop.ManBean"
19             java2XMLClassName="org.apache.soap.encoding.soapenc.
20             BeanSerializer"
21             xml2JavaClassName="org.apache.soap.encoding.soapenc.
22             BeanSerializer"/>
23     </isd:mappings>
24 </isd:service>

```

**Listing 5:** O Deployment Descriptor BVCatalogDD.xml

Χρησιμοποιώντας το TCP Monitor για τη θύρα 8080, στην οποία τρέχει ο Tomcat Server, ανιχνεύεται το παρακάτω:

```

1  =====
2
3  Listen Port: 8080
4  Target Host: 127.0.0.1
5  Target Port: 8080
6  ==== Request ====
7  POST /soap/servlet/rpcrouter HTTP/1.0
8  Host: 127.0.0.1:8080
9  Content-Type: text/xml; charset=utf-8
10 Content-Length: 789
11 SOAPAction: ""
12
13 <?xml version='1.0' encoding='UTF-8'?>
14 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
15     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.
16     org/2001/XMLSchema">
17 <SOAP-ENV:Body>
18 <ns1:addV xmlns:ns1="urn:BVehicleCatalog" SOAP-ENV:encodingStyle="http://schemas.
19     xmlsoap.org/soap/encoding/">
20 <vObj xmlns:ns2="urn:VBean_xmlns" xsi:type="ns2:vObj">
21 <VManufacturer xmlns:ns3="urn:MBean_xmlns" xsi:type="ns3:mObj">
22 <manCountry xsi:type="xsd:string">Japan</manCountry>
23 <manName xsi:type="xsd:string">Toyota</manName>
24 <manYear xsi:type="xsd:integer">1937</manYear>
25 </VManufacturer>
26 <VModel xsi:type="xsd:string">Avensis</VModel>
27 <VYear xsi:type="xsd:string">2008</VYear>
28 </vObj>
29 </ns1:addV>
30 </SOAP-ENV:Body>
31 </SOAP-ENV:Envelope>

```

**Listing 6:** Το SOAP request προς τον Tomcat Server