

Web-services Lab Book

1. Hello World - Exercise

```
package basicpack;
import javax.ws.HandlerChain;
import javax.ws.WebMethod;
import javax.ws.WebParam;
import javax.ws.WebService;
import javax.ws.soap.SOAPBinding;
import javax.ws.soap.SOAPBinding.Style;

@WebService(endpointInterface="basicpack.Hello", name=
"myservice")
public class HelloWorld implements Hello{
    public Employee sayHello(String name)
    {
        System.out.println("Say hello
called.....");

        return "Welcome to webservices...:"+name;
    }
}

@WebService
interface Hello
{
    @WebMethod
    public String sayHello(String name);
}

package basicpack;
import java.net.URL;
import javax.xml.namespace.QName;
```

```

import javax.xml.ws.Service;

public class Client{

    public static void main(String[] args) throws
Exception {

        URL url =
new URL("http://localhost:9999/ws/hello?wsdl");
        QName qname = new QName("http://basicpack/",
"HelloWorldService");
        Service service = Service.create(url,
qname);
        Hello hello = service.getPort(Hello.class);
        System.out.println(hello.sayHello("Falah
Technologies"));

    }
}

package basicpack;
import javax.xml.ws.Endpoint;

//Endpoint publisher
public class HelloPublisher{

    public static void main(String[] args) {
        System.out.println("service ready....");
        Endpoint.publish("http://localhost:8888/ws/hello"
, new HelloWorld());

    }
}

```

2. With Handler Example

```
package basicpack;
import javax.xml.ws.HandlerChain;
import javax.xml.ws.WebMethod;
import javax.xml.ws.WebParam;
import javax.xml.ws.WebService;
import javax.xml.ws.soap.SOAPBinding;
import javax.xml.ws.soap.SOAPBinding.Style;

@WebService(endpointInterface="basicpack.Hello", name=
"myservice")
@HandlerChain(file="handler-chain.xml")
public class HelloWorld implements Hello {
    public Employee sayHello(Employee emp)
    {
        System.out.println("Say hello
called.....");
        System.out.println("Welcome to
Webservices....:"+emp.getName()+":"+emp.getAge());
        return emp;
    }
}

@WebService
interface Hello
{
    @WebMethod
    public Employee sayHello(Employee emp);
}

package basicpack;
```

```

import java.io.FileOutputStream;
import java.io.PrintStream;
import java.util.Set;
import javax.xml.namespace.QName;
import javax.xml.soap.SOAPMessage;
import javax.xml.ws.handler.MessageContext;
import javax.xml.ws.handler.soap.SOAPHandler;
import javax.xml.ws.handler.soap.SOAPMessageContext;

public class MyHandler implements SOAPHandler<SOAPMessageContext>{
    @Override
    public void close(MessageContext context) {
        // TODO Auto-generated method stub
    }

    @Override
    public boolean handleMessage(SOAPMessageContext context) {
        Boolean result = (Boolean)
context.get(MessageContext.MESSAGE_OUTBOUND_PROPERTY)
;
        if(result)
        {
            System.out.println("out bound message -
response");
            try{
                SOAPMessage msg=context.getMessage();
                PrintStream
out=new PrintStream(System.out);
                msg.writeTo(out);
                out.println();
            }
            catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

        FileOutputStream os =
new FileOutputStream("response.xml");
        msg.writeTo(os);

        }catch(Exception e){}
    }
    elseif(!result)
    {
        System.out.println("in bound message -
request");
        try{
            SOAPMessage msg=context.getMessage();
            PrintStream
out=new PrintStream(System.out);
            msg.writeTo(out);
            FileOutputStream os =
new FileOutputStream("request.xml");
            msg.writeTo(os);
        }catch(Exception e){}
    }
    System.out.println("Result...:"+result);
    return true;
}

@Override
public boolean handleFault(SOAPMessageContext context)
{
    // TODO Auto-generated method stub
    return false;
}

@Override
public Set<QName>getHeaders() {
    // TODO Auto-generated method stub

```

```
        return null;
    }
}
```

```
package basicpack;
```

```
public class Employee {
    private String name;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    private int age;
}
```

```
package basicpack;
```

```
import javax.xml.ws.Endpoint;
```

```
//Endpoint publisher
```

```
public class HelloPublisher{

    public static void main(String[] args) {
        System.out.println("service ready....");
        Endpoint.publish("http://localhost:8888/ws/hello"
, new HelloWorld());
    }
}
```

```
}  
}
```

Handler-chain.xml - Place this file in the classpath

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<javaee:handler-chains  
  xmlns:javaee="http://java.sun.com/xml/ns/javaee"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
  <javaee:handler-chain>  
    <javaee:handler>  
      <javaee:handler-  
class>basicpack.MyHandler</javaee:handler-class>  
    </javaee:handler>  
  </javaee:handler-chain>  
</javaee:handler-chains>
```

3. Mtom Exercise

```
package mtompack;
```

```
import java.awt.Image;  
import java.awt.image.BufferedImage;  
import java.io.File;  
import javax.imageio.ImageIO;  
import javax.jws.HandlerChain;  
import javax.jws.WebService;  
import javax.jws.soap.SOAPBinding;  
import javax.jws.soap.SOAPBinding.Style;  
import javax.xml.ws.soap.MTOM;
```

```
@MTOM
```

```
@WebService(endpointInterface = "mtompack.MtomInter")
```

```

@HandlerChain(file="handler-chain.xml")
public class MtomService implements MtomInter{
    @Override
    public Image getImage(String name) throws
Exception {
        // TODO Auto-generated method stub
        File f=new File(name);
        return ImageIO.read(f);
    }
    @Override
    public void setImage(Image img,String name) {
        try{
            File fos=new File(name);

            BufferedImage bimage=(BufferedImage)img;
            ImageIO.write(bimage, "jpg", fos);
        }catch(Exception e){e.printStackTrace();}
    }
}
@WebService
//@SOAPBinding(style = Style.RPC)
interface MtomInter
{
    public Image getImage(String name) throws
Exception;

    public void setImage(Image img,String name);

}
package mtompack;
import javax.xml.ws.Endpoint;

//Endpoint publisher
public class MtomPublisher{

```



```

    public static void main(String[] args) {
        System.out.println("service ready....");
        Endpoint.publish("http://localhost:8888/ws/hello"
, new MtomService());

    }
}

```

```

package mtompack;
import java.awt.Image;
import java.io.File;
import java.net.URL;
import javax.imageio.ImageIO;
import javax.xml.namespace.QName;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.Service;
import javax.xml.ws.soap.SOAPBinding;

```

```

public class Client{

    public static void main(String[] args) throws
Exception {

        URL url = new
URL("http://localhost:8888/ws/hello?wsdl");
        QName qname = new QName("http://mtompack/",
"MtomServiceService");

        Service service = Service.create(url, qname);

        MtomInter mtom=service.getPort(MtomInter.class);
        BindingProvider bp = (BindingProvider) mtom;
    }
}

```

```

SOAPBinding binding = (SOAPBinding) bp.getBinding();
binding.setMTOMEnabled(true);
    Image img=ImageIO.read(new
File("first.jpg"));
mtom.setImage(img,"earrings.jpg");

    }
}

```

```

package mtompack;
import java.awt.Image;
import java.net.URL;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;

public class ReadClient{
    public static void main(String[] args) throws
Exception {
        URL url = new
URL("http://localhost:8888/ws/hello?wsdl");
        QName qname = new QName("http://mtompack/",
"MtomServiceService");
        Service service = Service.create(url, qname);
        MtomInter mtom=service.getPort(MtomInter.class);
        Image img= mtom.getImage("earrings.jpg");
        JFrame frame = new JFrame();
        frame.setSize(300, 300);
        JLabel label = new JLabel(new ImageIcon(img));
        frame.add(label);
        frame.setVisible(true);
    }
}

```

}



Handler Example

```
package handler;
import java.io.IOException;
import java.util.Iterator;
import java.util.Set;
import javax.xml.namespace.QName;
import javax.xml.soap.Node;
import javax.xml.soap.SOAPBody;
import javax.xml.soap.SOAPConstants;
import javax.xml.soap.SOAPEnvelope;
import javax.xml.soap.SOAPException;
import javax.xml.soap.SOAPFault;
import javax.xml.soap.SOAPHeader;
import javax.xml.soap.SOAPMessage;
import javax.xml.ws.handler.MessageContext;
import javax.xml.ws.handler.soap.SOAPHandler;
import javax.xml.ws.handler.soap.SOAPMessageContext;
import javax.xml.ws.soap.SOAPFaultException;

public class MacAddressValidatorHandler implements
SOAPHandler<SOAPMessageContext>{
```

```

@Override
    public boolean handleMessage(SOAPMessageContext
context) {

        System.out.println("Server :
handleMessage().....");

        Boolean isRequest = (Boolean)
context.get(MessageContext.MESSAGE_OUTBOUND_PROPERTY)
;

        //for response message only, true for
outbound messages, false for inbound
        if(!isRequest){

            try{
                SOAPMessage soapMsg =
context.getMessage();
                SOAPEnvelope soapEnv =
soapMsg.getSOAPPart().getEnvelope();
                SOAPHeader soapHeader =
soapEnv.getHeader();

                //if no header, add one
                if (soapHeader == null){
                    soapHeader = soapEnv.addHeader();
                    //throw exception
                    generateSOAPErrorMessage(soapMsg, "No
SOAP header.");
                }

                //Get client mac address from SOAP
header

```

```

        Iterator it =
soapHeader.extractHeaderElements(SOAPConstants.URI_SO
AP_ACTOR_NEXT);

        //if no header block for next actor
found? throw exception
        if (it == null || !it.hasNext()){
            generateSOAPErrorMessage(soapMsg, "No
header block for next actor.");
        }

        //if no mac address found? throw
exception
        Node macNode = (Node) it.next();
        String macValue = (macNode == null) ?
null : macNode.getValue();

        if (macValue == null){
            generateSOAPErrorMessage(soapMsg, "No
mac address in header block.");
        }

        //if mac address is not match, throw
exception
        if(!macValue.equals("90-4C-E5-44-B9-
8F11111")){
            generateSOAPErrorMessage(soapMsg,
"Invalid mac address, access is denied.");
        }

        //tracking
        soapMsg.writeTo(System.out);

```

```

        }catch(SOAPException e){
            System.err.println(e);
        }catch(IOException e){
            System.err.println(e);
        }

    }

    //continue other handler chain
    return true;
}

@Override
public boolean handleFault(SOAPMessageContext
context) {

    System.out.println("Server :
handleFault().....");

    return true;
}

@Override
public void close(MessageContext context) {
    System.out.println("Server : close().....");
}

@Override
public Set<QName> getHeaders() {
    System.out.println("Server :
getHeaders().....");
    return null;
}

```

```

    private void generateSOAPErrorMessage(SOAPMessage
msg, String reason) {
    try {
        SOAPBody soapBody =
msg.getSOAPPart().getEnvelope().getBody();
        SOAPFault soapFault = soapBody.addFault();
        soapFault.setFaultString(reason);
        throw new SOAPFaultException(soapFault);
    }
    catch(SOAPException e) { }
}
}

```

Handler Example which generate SOAP Fault

```

package com.mypack.handler;

import java.io.IOException;
import java.util.Iterator;
import java.util.Set;
import javax.xml.namespace.QName;
import javax.xml.soap.Node;
import javax.xml.soap.SOAPBody;
import javax.xml.soap.SOAPConstants;
import javax.xml.soap.SOAPEnvelope;
import javax.xml.soap.SOAPException;
import javax.xml.soap.SOAPFault;
import javax.xml.soap.SOAPHeader;
import javax.xml.soap.SOAPMessage;
import javax.xml.ws.handler.MessageContext;
import javax.xml.ws.handler.soap.SOAPHandler;
import javax.xml.ws.handler.soap.SOAPMessageContext;
import javax.xml.ws.soap.SOAPFaultException;

```

```

public class MacAddressValidatorHandler implements
SOAPHandler<SOAPMessageContext>{

    @Override
    public boolean handleMessage(SOAPMessageContext
context) {

        System.out.println("Server :
handleMessage().....");

        Boolean isRequest = (Boolean)
context.get(MessageContext.MESSAGE_OUTBOUND_PROPERTY)
;

        //for response message only, true for
outbound messages, false for inbound
        if(!isRequest){

            try{
                SOAPMessage soapMsg =
context.getMessage();
                SOAPEnvelope soapEnv =
soapMsg.getSOAPPart().getEnvelope();
                SOAPHeader soapHeader =
soapEnv.getHeader();

                //if no header, add one
                if (soapHeader == null){
                    soapHeader = soapEnv.addHeader();
                    //throw exception
                    generateSOAPErrorMessage(soapMsg, "No
SOAP header.");
                }
            }
        }
    }
}

```



```

//Get client mac address from SOAP
header
    Iterator it =
soapHeader.extractHeaderElements(SOAPConstants.URI_SO
AP_ACTOR_NEXT);

//if no header block for next actor
found? throw exception
    if (it == null || !it.hasNext()){
        generateSOAPErrorMessage(soapMsg, "No
header block for next actor.");
    }

//if no mac address found? throw
exception
    Node macNode = (Node) it.next();
    String macValue = (macNode == null) ?
null : macNode.getValue();

    if (macValue == null){
        generateSOAPErrorMessage(soapMsg, "No
mac address in header block.");
    }

//if mac address is not match, throw
exception
    if(!macValue.equals("90-4C-E5-44-B9-
8F1")){
        generateSOAPErrorMessage(soapMsg,
"Invalid mac address, access is denied.");
    }

//tracking
soapMsg.writeTo(System.out);

```

```

        }catch(SOAPException e){
            System.err.println(e);
        }catch(IOException e){
            System.err.println(e);
        }
    }

    //continue other handler chain
    return true;
}

@Override
public boolean handleFault(SOAPMessageContext
context) {

    System.out.println("Server :
handleFault().....");

    return true;
}

@Override
public void close(MessageContext context) {
    System.out.println("Server : close().....");
}

@Override
public Set<QName> getHeaders() {
    System.out.println("Server :
getHeaders().....");
    return null;
}

```

```

    }

    private void generateSOAPErrorMessage(SOAPMessage
msg, String reason) {
        try {
            SOAPBody soapBody =
msg.getSOAPPart().getEnvelope().getBody();
            SOAPFault soapFault = soapBody.addFault();
            soapFault.setFaultString(reason);
            throw new SOAPFaultException(soapFault);
        }
        catch(SOAPException e) { }
    }
}

```

HTTP Headers - UserName and Password Authentication Example.

```

package com.mkyong.ws;

import java.util.List;
import java.util.Map;

import javax.annotation.Resource;
import javax.jws.WebService;
import javax.xml.ws.WebServiceContext;
import javax.xml.ws.handler.MessageContext;

//Service Implementation Bean
@WebService(endpointInterface =
"com.mybservice.HelloWorld")
public class HelloWorldImpl implements HelloWorld{

    @Resource

```

```

WebServiceContext wsctx;

@Override
public String getHelloWorldAsString() {

    MessageContext mctx =
wsctx.getMessageContext();

    //get detail from request headers
    Map http_headers = (Map)
mctx.get(MessageContext.HTTP_REQUEST_HEADERS);
    List userList = (List)
http_headers.get("Username");
    List passList = (List)
http_headers.get("Password");

    String username = "";
    String password = "";

    if(userList!=null){
        //get username
        username = userList.get(0).toString();
    }

    if(passList!=null){
        //get password
        password = passList.get(0).toString();
    }

    //Should validate username and password with
database
    if (username.equals("ramu") &&
password.equals("secret")){
        return "Hello World JAX-WS - Valid User!";
    }
}

```

```

    }else{
        return "Unknown User!";
    }

}

}

```

Client for the USERNAME and AUTHENTICATION

```

package com.myservice.client;

import java.net.URL;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.xml.namespace.QName;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.Service;
import javax.xml.ws.handler.MessageContext;

import com.myservice.HelloWorld;

public class HelloWorldClient{

    private static final String WS_URL =
"http://localhost:8888/ws/hello?wsdl";

    public static void main(String[] args) throws
Exception {

```

```

        URL url = new URL(WS_URL);
        QName qname = new
QName("http://myservice.com/",
"HelloWorldImplService");

        Service service = Service.create(url, qname);
        HelloWorld hello =
service.getPort(HelloWorld.class);

        /******UserName & Password*****
******/
        Map<String, Object> req_ctx =
((BindingProvider)hello).getRequestContext();

req_ctx.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY
, WS_URL);

        Map<String, List<String>> headers = new
HashMap<String, List<String>>();
        headers.put("Username",
Collections.singletonList("ramu"));
        headers.put("Password",
Collections.singletonList("secret"));

req_ctx.put(MessageContext.HTTP_REQUEST_HEADERS,
headers);

/******
******/

System.out.println(hello.getHelloWorldAsString());

    }

```

}

- JAX-WS : wsimport tool example

The wsimport tool is used to parse an existing Web Services Description Language (WSDL) file and generate required files (JAX-WS portable artifacts) for web service client to access the published web services.

Client – Access the published service.

For CompB, to develop a web service client to access the CompA published web service, they can use `wsimport` tool to parse CompA's WSDL file and generate files (JAX-WS portable artifacts) to access CompA's published service.

Command :wsimport command to parse CompA WSDL file

```
C:\>wsimport -keep-verbose http://compA.com/ws/server?wsdl

parsing WSDL...

generating code...

com\ws\ServerInfo.java

com\ws\ServerInfoImplService.java
```

- JAX-WS :wsген tool example
The wsген tool is used to parse an existing web service implementation class and generates required files (JAX-WS portable artifacts) for web service deployment.

- D:/> wsген -verbose -keep - cp .com.ws.ServerInfo