

Web Services-III Assignment

Bottom Up Approach of Web services Development

1. Building a service:

- I. Create the following two java classes:

```
-----  
package service.textHandling;  
  
//Service implementation  
public class TextHandlingService {  
    public String toUpperCase(String input){  
        return input.toUpperCase();  
    }  
  
    public String toLowerCase(String input){  
        return input.toLowerCase();  
    }  
    public StringInfo getStringInfo(String input){  
        StringInfo info=new StringInfo();  
        info.setDigitCount(getDigitCount(input));  
        info.setLetterCount(getLetterCount(input));  
        info.setWordCount(getWordCount(input));  
        return info;  
    }  
  
    private int getWordCount(String input) {  
        if(input==null)  
            return 0;  
        int numWords = 0;  
        int index = 0;  
        boolean prevWhitespace = true;  
        while (index < input.length()) {  
            char c = input.charAt(index++);  
            boolean currWhitespace = Character.isWhitespace(c);  
            if (prevWhitespace && !currWhitespace) {  
                numWords++;  
            }  
            prevWhitespace = currWhitespace;  
        }  
        return numWords;  
    }  
  
    private int getLetterCount(String input) {  
        if (input == null)  
            return 0;  
  
        int counter = 0;  
  
        for (int i = 0; i < input.length(); i++) {  
            if (Character.isLetter(input.charAt(i)))  
                counter++;  
        }  
        return counter;  
    }  
}
```

```

    }
    private int getDigitCount(String input) {
        if (input == null)
            return 0;

        int counter = 0;

        for (int i = 0; i < input.length(); i++) {
            if (Character.isDigit(input.charAt(i)))
                counter++;
        }
        return counter;
    }
}

```

```

package service.textHandling;

public class StringInfo {

    int letterCount;
    int wordCount;
    int digitCount;
    public int getLetterCount() {
        return letterCount;
    }
    public void setLetterCount(int letterCount) {
        this.letterCount = letterCount;
    }
    public int getWordCount() {
        return wordCount;
    }
    public void setWordCount(int wordCount) {
        this.wordCount = wordCount;
    }
    public int getDigitCount() {
        return digitCount;
    }
    public void setDigitCount(int digitCount) {
        this.digitCount = digitCount;
    }
}

```

-
- II. Create the services.xml file necessary to deploy such service.
 - III. Create the service archive file.

TO SUBMIT: The .aar file containing the service.

- IV. Deploy the service and test it in your browser

http://localhost:8080/axis2/services/textHandlingService/getStringInfo?input="Hi
, I would like to test you: 201121"

TO SUBMIT: The given response.

2. Building a client:

- I. Write a client to call the "toUpperCase" method in Blocking (Synchronous) manner.

TO SUBMIT: The client source code.

- II. Write a client to call the "toLowerCase" method in a non-Blocking (Asynchronous) manner.

TO SUBMIT: The client source code.

III. Monitoring the service:

- I. Use the TCPMon tool to monitor your service. Run one of your clients and direct your call to the TCPMon tool.

Snapshot of the collected messages inside the TCPMon tool. (See the sample picture)

The screenshot shows the TCPMon application window. At the top, there are controls for 'Admin', 'Sender', and 'Port 6666'. Below that, there are fields for 'Listen Port: 6666', 'Host: 127.0.0.1', and 'Port: 8080'. A table below displays a list of messages with columns for State, Time, Request Host, Target Host, Request..., and Elapsed Time. The selected message is a POST request to /axis2/services/Version HTTP/1.1, which resulted in a 200 OK response. The detailed view of the message shows the SOAP request and response XML.

State	Time	Request Host	Target Host	Request...	Elapsed Time
Error	2008-01-20 14:59:36	127.0.0.1	127.0.0.1	GET /axis2/ HTTP/1.1	
Done	2008-01-20 15:00:08	127.0.0.1	127.0.0.1	GET /axis2/ HTTP/1.1	20953
Done	2008-01-20 15:00:09	127.0.0.1	127.0.0.1	GET /axis2/axis2-web/css/axis-style.css HTTP/1...	110
Done	2008-01-20 15:00:09	127.0.0.1	127.0.0.1	GET /axis2/axis2-web/images/axis_j.jpg HTTP/1...	63
Done	2008-01-20 15:00:15	127.0.0.1	127.0.0.1	POST /axis2/services/Version HTTP/1.1	20047

```
POST /axis2/services/Version HTTP/1.1
Content-Type: text/xml; charset=UTF-8
SOAPAction: "urn:anonOutInOp"
User-Agent: Axis2
Host: 127.0.0.1:6666
Transfer-Encoding: chunked

fc
<?xml version='1.0' encoding='UTF-8'?>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoa...
    <soapenv:Body>
      <ns1:getVersion xmlns:ns1="http://axisversion.sa...
        <ns1:myValue />
      </ns1:getVersion>
    </soapenv:Body>
  </soapenv:Envelope>0

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml; charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 20 Jan 2008 14:00:15 GMT

141
<?xml version='1.0' encoding='UTF-8'?>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap...
    <soapenv:Body>
      <ns:getVersionResponse xmlns:ns="http://axisversio...
        <ns:return>Hello I am Axis2 version service , I...
      </ns:getVersionResponse>
    </soapenv:Body>
  </soapenv:Envelope>0
```

- II. Use the TCPMon tool to send a request to call the "toUpperCase" method.

Snapshot of the Sender window. (See the sample picture)

