PROXY – Design Pattern using JAX-WS

Definition:

Provides a surrogate or a placeholder for anther object to control access to it.

User the proxy pattern to create a representative object that controls access to anther object, which may be remote, expensive to create or need of securing.

# Variations:

1. Remote Proxy: the proxy act as a local representative for an object that lives on a different JVM. A method call on a proxy results in the call being transferred over the wire, invoked remotely and the result being returned back to the proxy and then to the Client.
2. Protection Proxy: Components must be protected from unauthorized access.
3. Cache Proxy: Multiple local clients can share results from remote components.
4. Synchronization Proxy: Multiple simultaneous accesses to a component must be synchronized.
5. Counting Proxy: Accidental deletion of components must be prevented or usage statistics collected.
6. Virtual Proxy: Processing or Loading a component is costly, while partial information about the component may be sufficient.
7. Firewall Proxy: Local clients should be protected from the outside world.

Context:

To see is JAX-WS implements Remote Proxy.

Problem:

Get the mobile usage data for a given number.

The user will invoke the request from his/her mobile to check the his/her Mobile Usage as of now.

Solution:

Create a JAX-WS Service to get the mobile usage for a given number by querying the CDR Table from a service provider Database.

Service:

# MobileDataUsageIntf - Interface

package com.mobile.data.usage;

import javax.jws.WebMethod;

import javax.jws.WebService;

@WebService

public interface MobileDataUsageIntf {

@WebMethod public String getMobileData(String data);

}

1. **MobileDataUsageImpl – Concreate class**

package com.mobile.data.usage;

import javax.jws.WebService;

@WebService(endpointInterface="com.mobile.data.usage.MobileDataUsageIntf")

public class MobileDataUsageImpl implements MobileDataUsageIntf{

public String getMobileData(String phNumber) {

StringBuilder builder = new StringBuilder();

String totatCalls="23";

String totalSms="5";

String mobileInternateDataUsed="2.5GB";

String billedAmount="257";

String unbilledAmount="500";

String billDate="07/12/2017";

String billPlan="PLAN500";

builder.append("--------------------------Mobile Usage---------------------- \n");

builder.append("Mobile Data Usage for Phone Number: "+phNumber).append("\n");

builder.append("Total Calls: ").append(totatCalls).append("\n");

builder.append("Total SMS: ").append(totalSms).append("\n");

builder.append("Total Mobile Internate Data Used: ").append(mobileInternateDataUsed).append("\n");

builder.append("Billed Amount: ").append(billedAmount).append("\n");

builder.append("Unbilled Amount: ").append(unbilledAmount).append("\n");

builder.append("Bill Date: ").append(billDate).append("\n");

builder.append("Bill Plan: ").append(billPlan).append("\n");

builder.append("--------------------------Mobile Usage----------------------");

return builder.toString();

}

}

WSDL: <http://localhost:8080/WS/MobileDataUsageIntf?wsdl>

This XML file does not appear to have any style information associated with it. The document tree is shown below.

<!--

Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e.

-->

<!--

Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e.

-->

<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsp1\_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://usage.data.mobile.com/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://usage.data.mobile.com/" name="MobileDataUsageImplService">

<types>

<xsd:schema>

<xsd:import namespace="http://usage.data.mobile.com/" schemaLocation="http://localhost:8080/WS/MobileDataUsageIntf?xsd=1"/>

</xsd:schema>

</types>

<message name="getMobileData">

<part name="parameters" element="tns:getMobileData"/>

</message>

<message name="getMobileDataResponse">

<part name="parameters" element="tns:getMobileDataResponse"/>

</message>

<portType name="MobileDataUsageIntf">

<operation name="getMobileData">

<input wsam:Action="http://usage.data.mobile.com/MobileDataUsageIntf/getMobileDataRequest" message="tns:getMobileData"/>

<output wsam:Action="http://usage.data.mobile.com/MobileDataUsageIntf/getMobileDataResponse" message="tns:getMobileDataResponse"/>

</operation>

</portType>

<binding name="MobileDataUsageImplPortBinding" type="tns:MobileDataUsageIntf">

<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>

<operation name="getMobileData">

<soap:operation soapAction=""/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

</operation>

</binding>

<service name="MobileDataUsageImplService">

<port name="MobileDataUsageImplPort" binding="tns:MobileDataUsageImplPortBinding">

<soap:address location="http://localhost:8080/WS/MobileDataUsageIntf"/>

</port>

</service>

</definitions>

Class Diagram:

MobileDataUsageIntf

MobileDataUsageImpl

Note: The same interface (MobileDataUsageIntf) will be called in client stub.

Client:

MobileDataUsageClient.java:

package com.mobile.data.usage.client;

import com.mobile.data.usage.MobileDataUsageImplService;

import com.mobile.data.usage.MobileDataUsageIntf;

public class MobileDataUsageClient {

public static void main(String[] args) {

MobileDataUsageImplService mobileDataUsageImplService = new MobileDataUsageImplService();

MobileDataUsageIntf mobileData = mobileDataUsageImplService.getMobileDataUsageImplPort();

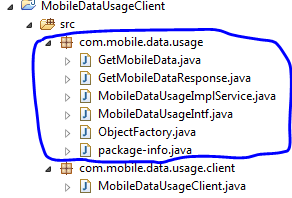
System.out.println(mobileData.getMobileData("97979877787"));

}

}

Client Stubs which act as a remote proxy are generated from the following command:

wsimport -s . http://localhost:8080/WS/MobileDataUsageIntf?wsdl



Class Diagram:

MobileDataUsageIntf

MobileDataUsageImplService

Note: The same interface (MobileDataUsageIntf) will be called in client stub.

Proof:

This means that both the Proxy (MobileDataUsageImplService) and the Real Subject (MobileDataUsageImpl) implement the same Subject Interface (MobileDataUsageIntf).

This allows any client to treat the proxy (MobileDataUsageImplService) just like the Real Subject (MobileDataUsageImpl).

Since the Server code resides on any application server which means on a different JVM and the client code being called from the mobile device that is from a different JVM, hence it is an example of Remote Proxy.

MobileDataUsageImplService - generated code

Line no. 57 to 60 – proxy implementing Subject Ineterface.

1: package com.mobile.data.usage;

2: import java.net.MalformedURLException;

3: import java.net.URL;

4: import javax.xml.namespace.QName;

5: import javax.xml.ws.Service;

6: import javax.xml.ws.WebEndpoint;

7: import javax.xml.ws.WebServiceClient;

8: import javax.xml.ws.WebServiceException;

9: import javax.xml.ws.WebServiceFeature;

10: /\*\*

11: \* This class was generated by the JAX-WS RI.

12: \* JAX-WS RI 2.2.9-b130926.1035

13: \* Generated source version: 2.2

14: \*

15: \*/

16: @WebServiceClient(name = "MobileDataUsageImplService", targetNamespace = "http://usage.data.mobile.com/", wsdlLocation = "http://localhost:8080/WS/MobileDataUsageIntf?wsdl")

17: public class MobileDataUsageImplService

18: extends Service

19: {

20: private final static URL MOBILEDATAUSAGEIMPLSERVICE\_WSDL\_LOCATION;

21: private final static WebServiceException MOBILEDATAUSAGEIMPLSERVICE\_EXCEPTION;

22: private final static QName MOBILEDATAUSAGEIMPLSERVICE\_QNAME = new QName("http://usage.data.mobile.com/", "MobileDataUsageImplService");

23: static {

24: URL url = null;

25: WebServiceException e = null;

26: try {

27: url = new URL("http://localhost:8080/WS/MobileDataUsageIntf?wsdl");

28: } catch (MalformedURLException ex) {

29: e = new WebServiceException(ex);

30: }

31: MOBILEDATAUSAGEIMPLSERVICE\_WSDL\_LOCATION = url;

32: MOBILEDATAUSAGEIMPLSERVICE\_EXCEPTION = e;

33: }

34: public MobileDataUsageImplService() {

35: super(\_\_getWsdlLocation(), MOBILEDATAUSAGEIMPLSERVICE\_QNAME);

36: }

37: public MobileDataUsageImplService(WebServiceFeature... features) {

38: super(\_\_getWsdlLocation(), MOBILEDATAUSAGEIMPLSERVICE\_QNAME, features);

39: }

40: public MobileDataUsageImplService(URL wsdlLocation) {

41: super(wsdlLocation, MOBILEDATAUSAGEIMPLSERVICE\_QNAME);

42: }

43: public MobileDataUsageImplService(URL wsdlLocation, WebServiceFeature... features) {

44: super(wsdlLocation, MOBILEDATAUSAGEIMPLSERVICE\_QNAME, features);

45: }

46: public MobileDataUsageImplService(URL wsdlLocation, QName serviceName) {

47: super(wsdlLocation, serviceName);

48: }

49: public MobileDataUsageImplService(URL wsdlLocation, QName serviceName, WebServiceFeature... features) {

50: super(wsdlLocation, serviceName, features);

51: }

52: /\*\*

53: \*

54: \* @return

55: \* returns MobileDataUsageIntf

56: \*/

57: @WebEndpoint(name = "MobileDataUsageImplPort")

58: public MobileDataUsageIntf getMobileDataUsageImplPort() {

59: return super.getPort(new QName("http://usage.data.mobile.com/", "MobileDataUsageImplPort"), MobileDataUsageIntf.class);

60: }

61: /\*\*

62: \*

63: \* @param features

64: \* A list of {@link javax.xml.ws.WebServiceFeature} to configure on the proxy. Supported features not in the <code>features</code> parameter will have their default values.

65: \* @return

66: \* returns MobileDataUsageIntf

67: \*/

68: @WebEndpoint(name = "MobileDataUsageImplPort")

69: public MobileDataUsageIntf getMobileDataUsageImplPort(WebServiceFeature... features) {

70: return super.getPort(new QName("http://usage.data.mobile.com/", "MobileDataUsageImplPort"), MobileDataUsageIntf.class, features);

71: }

72: private static URL \_\_getWsdlLocation() {

73: if (MOBILEDATAUSAGEIMPLSERVICE\_EXCEPTION!= null) {

74: throw MOBILEDATAUSAGEIMPLSERVICE\_EXCEPTION;

75: }

76: return MOBILEDATAUSAGEIMPLSERVICE\_WSDL\_LOCATION;

77: }

78: }

Interface Used by Proxy –same as used by Real Subject

MobileDataUsageIntf - generated code

package com.mobile.data.usage;

import javax.jws.WebMethod;

import javax.jws.WebParam;

import javax.jws.WebResult;

import javax.jws.WebService;

import javax.xml.bind.annotation.XmlSeeAlso;

import javax.xml.ws.Action;

import javax.xml.ws.RequestWrapper;

import javax.xml.ws.ResponseWrapper;

/\*\*

\* This class was generated by the JAX-WS RI.

\* JAX-WS RI 2.2.9-b130926.1035

\* Generated source version: 2.2

\*

\*/

@WebService(name = "MobileDataUsageIntf", targetNamespace = "http://usage.data.mobile.com/")

@XmlSeeAlso({

ObjectFactory.class

})

public interface MobileDataUsageIntf {

/\*\*

\*

\* @param arg0

\* @return

\* returns java.lang.String

\*/

@WebMethod

@WebResult(targetNamespace = "")

@RequestWrapper(localName = "getMobileData", targetNamespace = "http://usage.data.mobile.com/", className = "com.mobile.data.usage.GetMobileData")

@ResponseWrapper(localName = "getMobileDataResponse", targetNamespace = "http://usage.data.mobile.com/", className = "com.mobile.data.usage.GetMobileDataResponse")

@Action(input = "http://usage.data.mobile.com/MobileDataUsageIntf/getMobileDataRequest", output = "http://usage.data.mobile.com/MobileDataUsageIntf/getMobileDataResponse")

public String getMobileData(

@WebParam(name = "arg0", targetNamespace = "")

String arg0);

}

# Sequence Diagram:

Server

Proxy

Client

Mobile

getMobileData

getMobileData

User

returnMobileData

returnMobileData

Conclusion:

#### JAX-WS implements the remote proxy design pattern.