

The logo for the American Immunization Registry Association (AIRA) features the acronym "AIRA" in a large, white, stylized font on a purple rectangular background.

AMERICAN
IMMUNIZATION
REGISTRY
ASSOCIATION

Immunization Information Systems for a New Era

Immunization Information System Web Services

Real-time integration using Java
Demonstration of reference implementation

The logo for the American Immunization Registry Association (AIRA) features the acronym "AIRA" in a large, white, stylized font on a purple rectangular background.

AMERICAN
IMMUNIZATION
REGISTRY
ASSOCIATION

Immunization Information Systems for a New Era

© 2012 American Immunization Registry Association | www.immregistries.org

Web Services in Java

- **Java has many techniques and software for creating web services.**
- **Most are very similar to Microsoft .net strategy that was shown.**
- **Cross-compatible with Microsoft .net and other implementations.**
- **Creating a web service from an existing WSDL is a simple process.**

History of Immunization Web Services

- **2011**
 - Transport Layer committee selects Web Services as the recommended standard.
 - Committee develops a common WSDL.
 - Reference server and client created
 - First EHR representative connects .net client
- **2012**
 - First IIS connects to reference web service

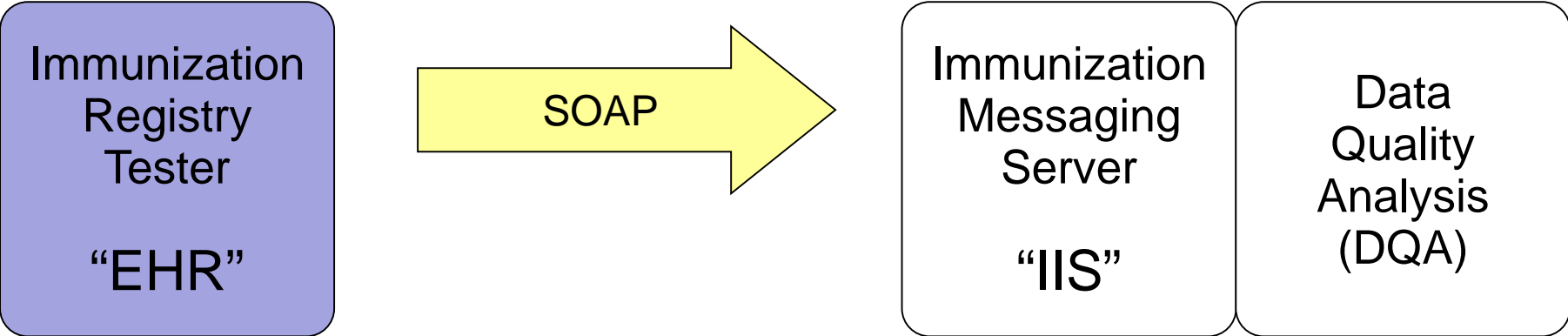
Goal for Reference Implementation

- **Demonstrate feasibility of Web Services.**
- **Test and demonstrate national WSDL.**
- **Provide example source code.**
- **Support testing by EHR vendors.**
- **Provide template for IIS implementations.**
- **Provide additional testing resources.**

Reference Implementation

- **Open Immunization Software (OIS)**
 - Immunization Registry Tester (client)
 - Immunization Messaging Server (server)
 - Data Quality Assurance engine
- **Java 1.6 - NetBeans – GlassFish**
- **Source code hosted at SourceForge**

Reference Implementation Layout



Immunization Registry Tester

- **Create test messages.**
- **Create test cases.**
- **Organize cases into batches.**
- **Connect to any system that supports:**
 - **Standard immunization WSDL**
 - **Legacy standard for HTTPS POST**
- **Run sets of test cases.**
- **Profile test or production HL7 interfaces.**

Reference Implementation Demo

The screenshot shows a Mozilla Firefox browser window titled "Setup - Mozilla Firefox". The address bar contains the URL: `openimmunizationsoftware.us:6394/ImmMessagingClient/LoginServlet?username=Mafai&password=LU0Z23I`. The browser's menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. Below the address bar is a toolbar with various icons and a search bar. The main content area features a navigation menu with links: Home, Setup, Edit, Send Message, Run Tests, Profile Interface, Logout, and Nathan Bunker (Mafai). A red banner displays "Welcome Nathan Bunker".

Test Cases

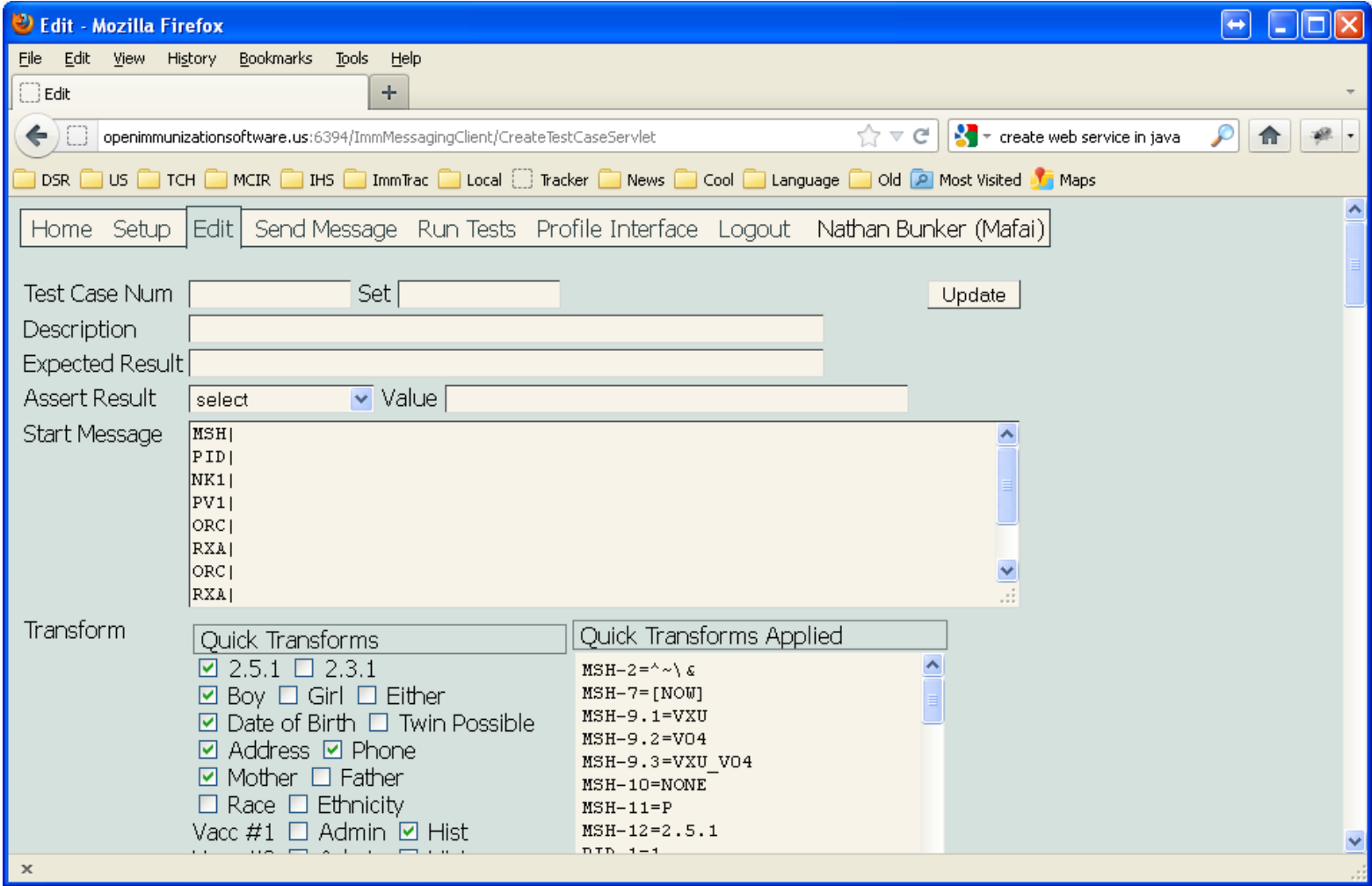
Script

Load Test Cases Download Script Download HL7 Only

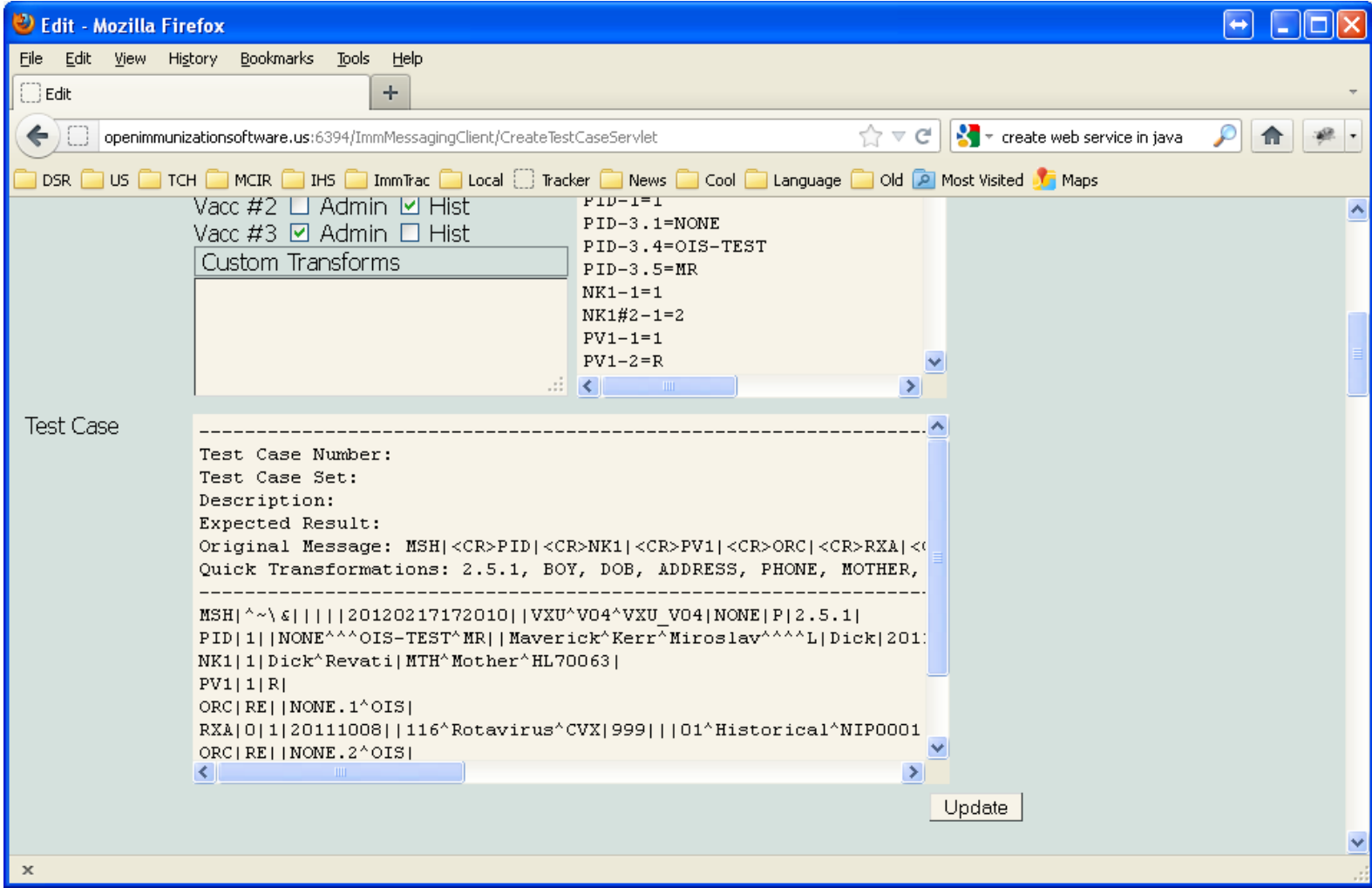
Connections

Label	Type	URL	User Id	Password	Facility Id	
OIS DQA	SOAP	http://openimmunizationsoftw..	CDCTLWG		1200	Delete
	select					Add

Reference Implementation Demo



Reference Implementation Demo



Reference Implementation Demo

The screenshot shows a Mozilla Firefox browser window titled "Send Message - Mozilla Firefox". The address bar displays the URL "openimmunizationsoftware.us:6394/ImmMessagingClient/SubmitServlet". The browser's menu bar includes "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". The browser's toolbar shows various navigation and utility icons. The main content area features a navigation menu with "Home", "Setup", "Edit", "Send Message", "Run Tests", "Profile Interface", "Logout", and "Nathan Bunker (Mafai)".

The "Send Message" section is titled "Connection OIS DQA (SOAP)" and contains a text area for the message. The message content is as follows:

```
MSH|^~\&|||20120217172010||VXU^V04^VXU_V04|NONE|P|2.5.1|
PID|1||NONE^^^OIS-TEST^MR||Maverick^Kerr^Miroslav^^^^L|Dick|20110808|M
NK1|1|Dick^Revati|MTH^Mother^HL70063|
PV1|1|R|
ORC|RE||NONE.1^OIS|
RXA|O|1|20111008||116^Rotavirus^CVX|999|||01^Historical^NIP0001|
ORC|RE||NONE.2^OIS|
RXA|O|1|20111207||48^Hib^CVX|999|||01^Historical^NIP0001|
ORC|RE||NONE.3^OIS|
RXA|O|1|20120217||48^Hib^CVX|0.5|ML||00^Administered^NIP0001|
Z381
```

Below the message text area are "Refresh" and "Submit" buttons. The status section below the message reads: "Status for OIS DQA: Test message 'Sent from client 'ch.astrahosting.com'' received by host 'ch.astrahosting.com' on Fri, 17 Feb 2012 at 17:21:10 UTC".

At the bottom of the page, there is a section titled "How To Use This Page".

Reference Implementation Demo

Send Message - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Send Message +

openimmunizationsoftware.us:6394/ImmMessagingClient/SubmitServlet

DSR US TCH MCIR IHS ImmTrac Local Tracker News Cool Language Old Most Visited Maps

Home Setup Edit **Send Message** Run Tests Profile Interface Logout Nathan Bunker (Mafai)

Connection OIS DQA (SOAP)

Message

```
MSH|^~\&|||20120217172010||VXU^V04^VXU_V04|NONE|P|2.5.1|
PID|1||NONE^^^OIS-TEST^MR||Maverick^Kerr^Miroslav^^^L|Dick|20110808|M
NK1|1|Dick^Revati|MTH^Mother^HL70063|
PV1|1|R|
ORC|RE||NONE.1^OIS|
RXA|0|1|20111008||116^Rotavirus^CVX|999|||01^Historical^NIP0001|
ORC|RE||NONE.2^OIS|
RXA|0|1|20111207||48^Hib^CVX|999|||01^Historical^NIP0001|
ORC|RE||NONE.3^OIS|
RXA|0|1|20120217||48^Hib^CVX|0.5|ML||00^Administered^NIP0001|
Z381
```

Refresh Submit

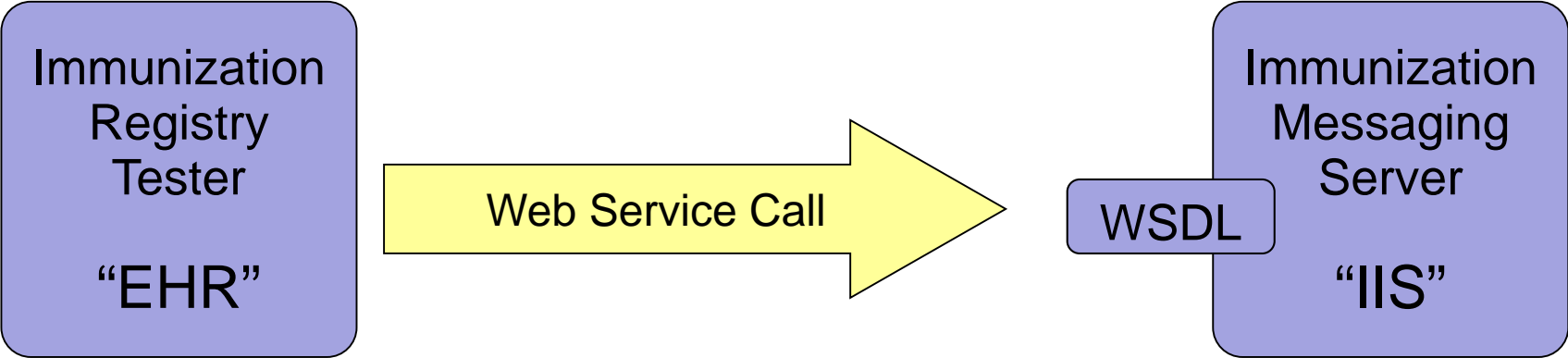
Response from OIS DQA:

```
MSH|^~\&|MCIR|MCIR|||20120217172145+0000||ACK^V04|20120217172145+0000.1|P|2.5.1|
SFT|OIS|0.13d|DQA|
MSA|AA|NONE|Message accepted with 3 vaccinations|
```

Status for OIS DQA:

x

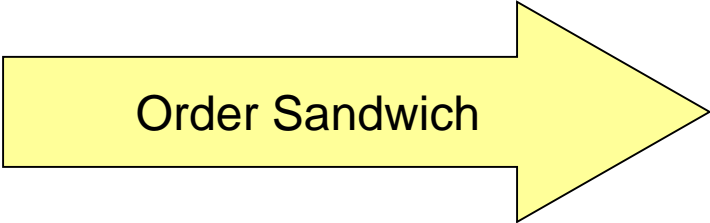
Creating a Web Service



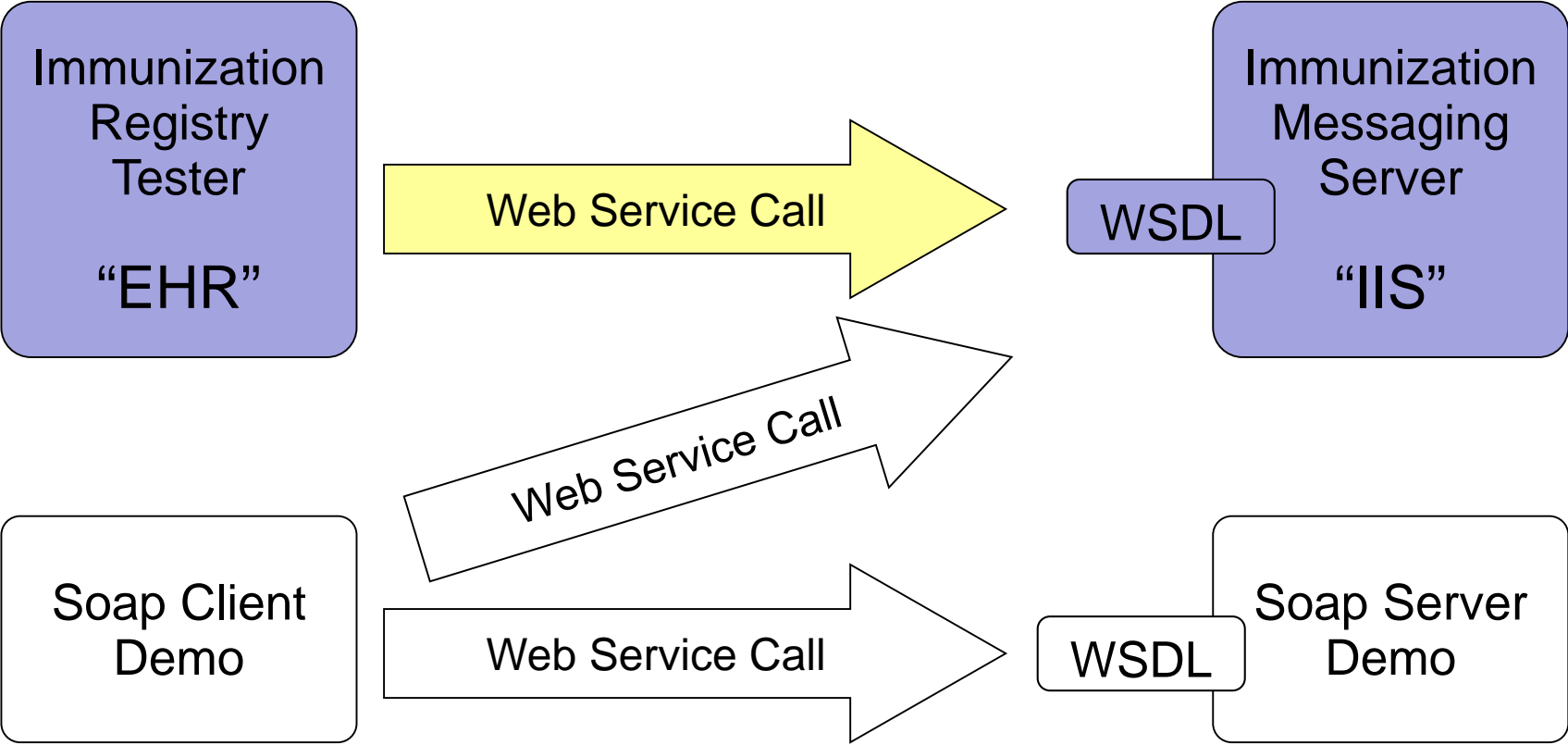
Do you want a WSDL for Lunch?

Eating at a Restaurant	Integrating with Web Services
Food	Data
Waiter	Server
Customer	Client
Menu	WSDL

Creating a Web Service



Creating a Web Service



Web Service Ingredients

- **NetBeans IDE 6.9.1**
- **GlassFish 3.0**
- **Java 1.6**

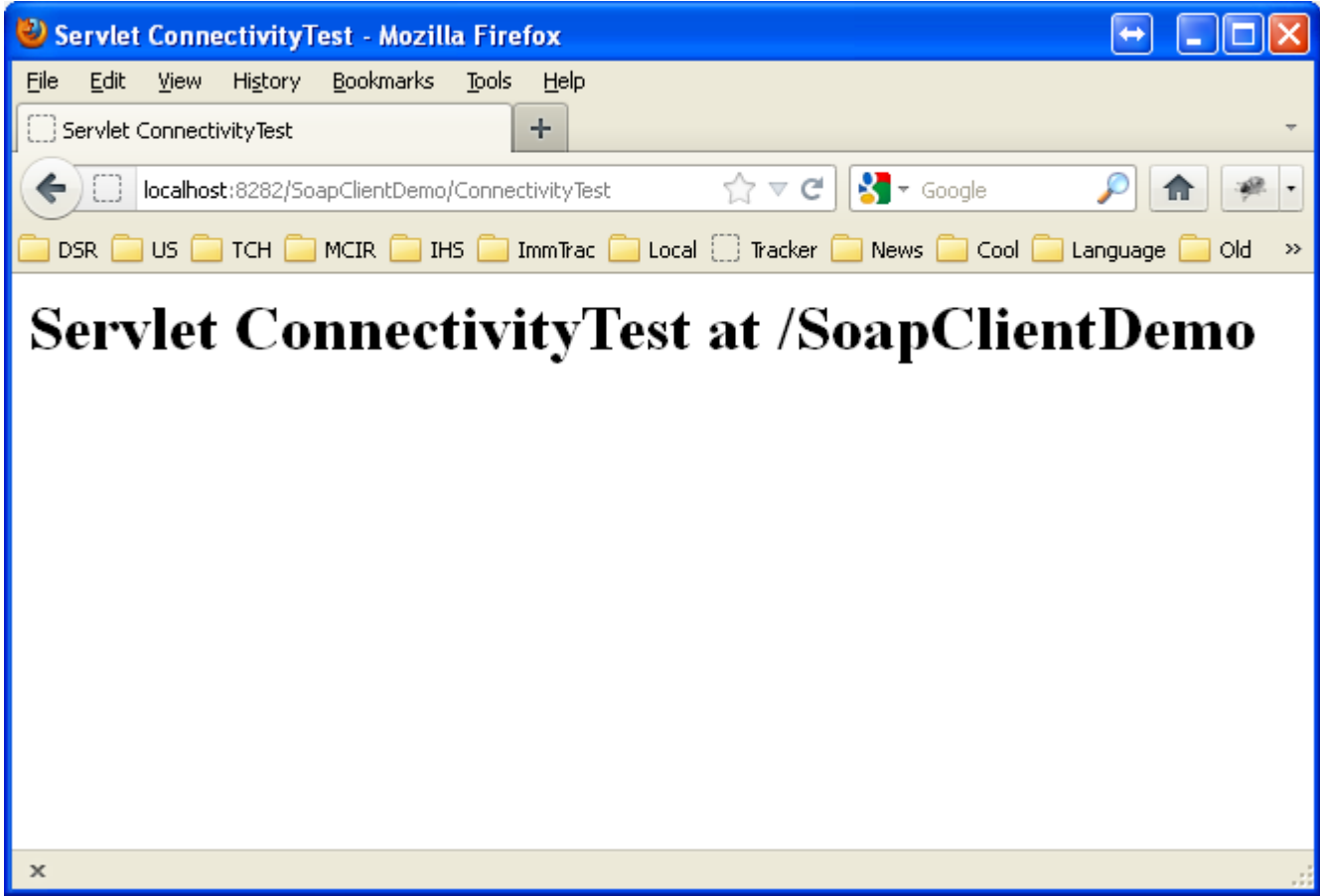


index.jsp x ConnectivityTest.java x



```
26 * @param response servlet response
27 * @throws ServletException if a servlet-specific error occurs
28 * @throws IOException if an I/O error occurs
29 */
30 protected void processRequest (HttpServletRequest request, HttpServletResponse response)
31 throws ServletException, IOException {
32     response.setContentType("text/html;charset=UTF-8");
33     PrintWriter out = response.getWriter();
34     try {
35         out.println("<html>");
36         out.println("<head>");
37         out.println("<title>Servlet ConnectivityTest</title>");
38         out.println("</head>");
39         out.println("<body>");
40         out.println("<h1>Servlet ConnectivityTest at " + request.getContextPath () + "</h1>");
41         out.println("</body>");
42         out.println("</html>");
43     } finally {
44         out.close();
45     }
46 }
47
48 + HttpServlet methods. Click on the + sign on the left to edit the code.
83
84 }
85
```

New servlet running



Create new client

New Web Service Client

Steps

1. Choose File Type
2. **WSDL and Client Location**

WSDL and Client Location

Specify the WSDL file of the Web Service.

Project:

Local File:

WSDL URL:

Specify a package name where the client java artifacts will be generated:

Project: SoapClientDemo

Package:

Client Style: JAX-WS Style

Generate Dispatch code

Enter the URL of the service you wish to use.

Mozilla Firefox

File Edit View History Bookmarks Tools Help

Servlet ConnectivityTest x http://openimmunizat...client_Service?wsdl x +

openimmunizationsoftware.us:6394/ImmMessagingServer/client_Service?wsdl

DSR US TCH MCIR IHS ImmTrac Local Tracker News Cool Language Old Most Visited Maps

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 at http://jax-ws.dev.java.net. RI's version is Metro/2.1 (branches/2.1-6728; 2011-02-03T1  
  -->  
- <!--  
  Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Metro/2.1 (branches/2.1-6728; 2011-02-03T1  
  -->  
- <definitions targetNamespace="urn:cdciisb:2011" name="IISServiceNew">  
  <!-- schema for types -->  
  <types>  
    <xsd:schema elementFormDefault="qualified" targetNamespace="urn:cdciisb:2011">  
      <xsd:complexType name="connectivityTestRequestType">  
        <xsd:sequence>  
          <xsd:element name="echoBack" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>  
        </xsd:sequence>  
      </xsd:complexType>  
      <xsd:complexType name="connectivityTestResponseType">  
        <xsd:sequence>  
          <xsd:element name="return" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>  
        </xsd:sequence>  
      </xsd:complexType>  
    </types>  
</definitions>
```

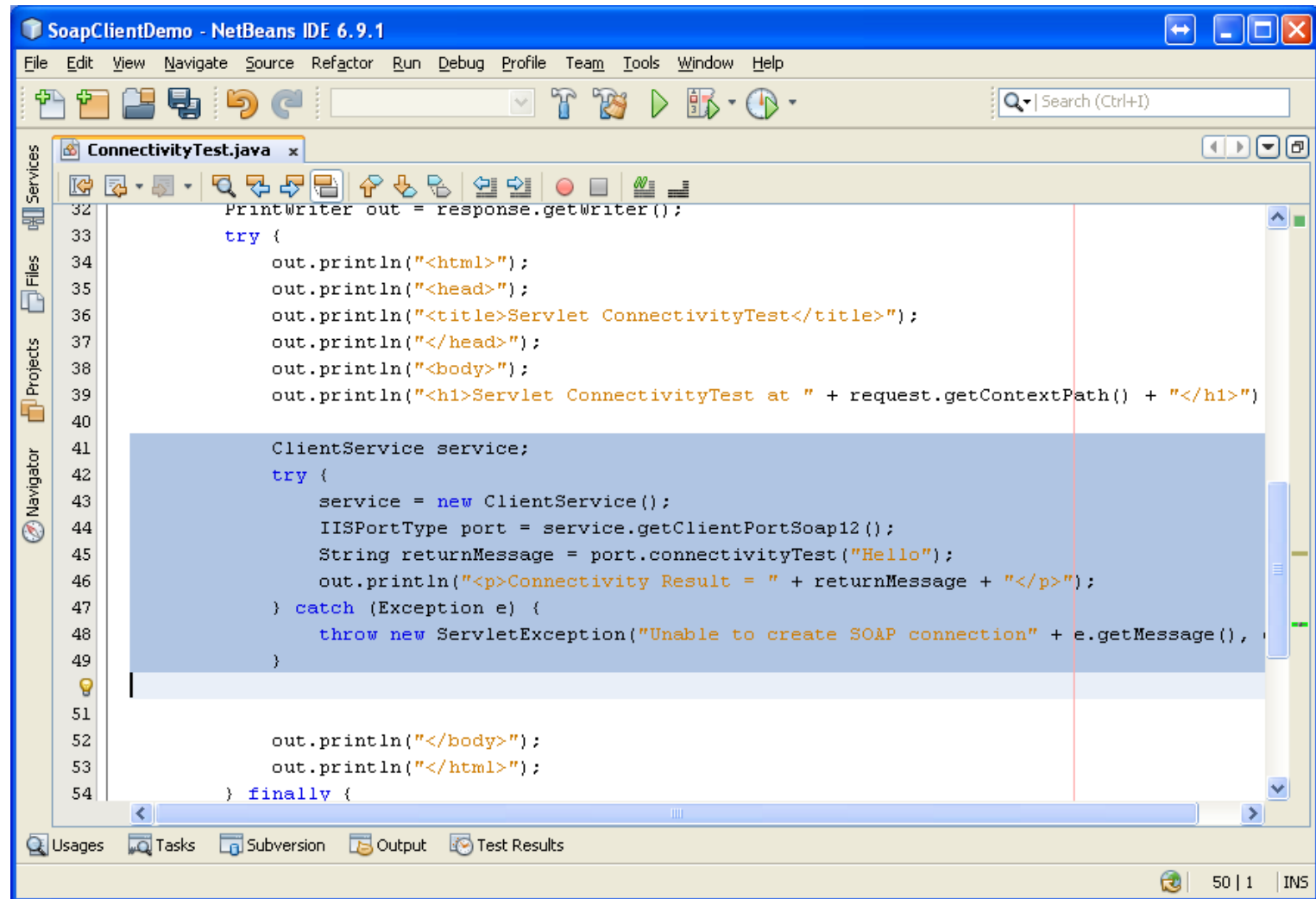
Point to WSDL link

The screenshot shows the 'New Web Service Client' wizard window. The title bar reads 'New Web Service Client'. On the left, a 'Steps' pane shows two steps: '1. Choose File Type' and '2. WSDL and Client Location', with the second step being the active one. The main area is titled 'WSDL and Client Location' and contains the following fields and controls:

- Instruction: 'Specify the WSDL file of the Web Service.'
- Radio buttons for selection:
 - Project: [text box] [Browse...]
 - Local File: [text box] [Browse...]
 - WSDL URL: [text box containing 'js:6394/ImmMessagingServer/client_Service?wsdl'] [Set Proxy...]
- Instruction: 'Specify a package name where the client java artifacts will be generated:'
- Project: SoapClientDemo
- Package: [text box containing 'com.soapclientdemo'] [dropdown arrow]
- Client Style: [text box containing 'JAX-WS Style'] [dropdown arrow]
- Generate Dispatch code

At the bottom of the window are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Add new code



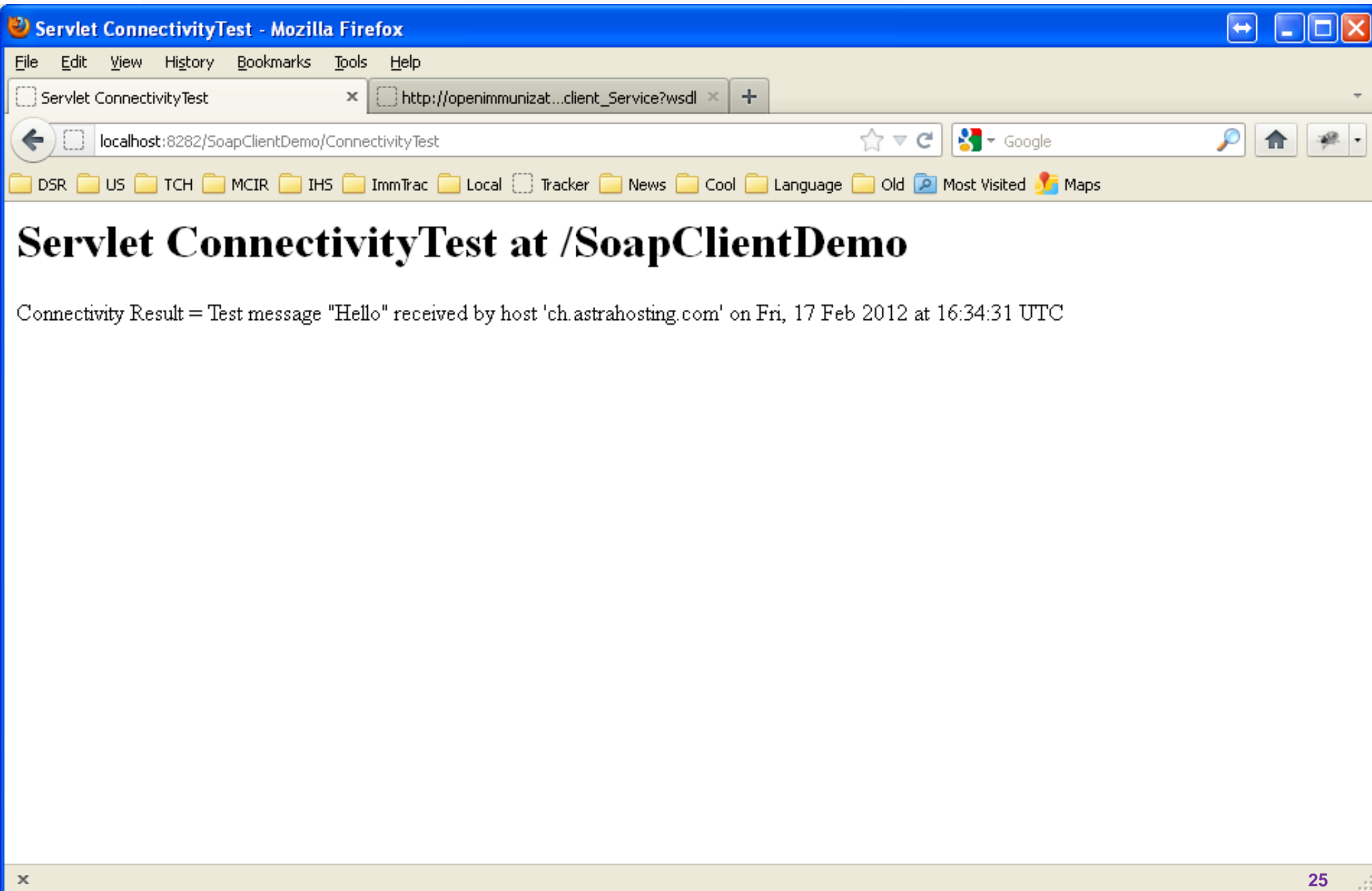
```
SoapClientDemo - NetBeans IDE 6.9.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
ConnectivityTest.java x
Services
Files
Projects
Navigator
32 PrintWriter out = response.getWriter();
33
34 try {
35     out.println("<html>");
36     out.println("<head>");
37     out.println("<title>Servlet ConnectivityTest</title>");
38     out.println("</head>");
39     out.println("<body>");
40     out.println("<h1>Servlet ConnectivityTest at " + request.getContextPath() + "</h1>")
41
42     ClientService service;
43     try {
44         service = new ClientService();
45         IISPortType port = service.getClientPortSoap12();
46         String returnMessage = port.connectivityTest("Hello");
47         out.println("<p>Connectivity Result = " + returnMessage + "</p>");
48     } catch (Exception e) {
49         throw new ServletException("Unable to create SOAP connection" + e.getMessage(),
50
51
52         out.println("</body>");
53         out.println("</html>");
54     } finally {
```

Usages Tasks Subversion Output Test Results

50 | 1 | INS

Add new code

```
ClientService service;
try {
    service = new ClientService();
    IISPortType port = service.getClientPortSoap12();
    String returnMessage = port.connectivityTest("Hello");
    out.println("<p>Connectivity Result = " + returnMessage + "</p>");
} catch (Exception e) {
    throw new ServletException("Unable to create SOAP connection" + e.getMessage(), e);
}
```

Adding submit

```
ClientService service;
try {
    service = new ClientService();
    IISPortType port = service.getClientPortSoap12();
    String returnMessage = port.connectivityTest("Hello");
    out.println("<p>Connectivity Result = " + returnMessage + "</p>");
    String username = "CDCTLWG";
    String password = "H3490";
    String facilityId = "1200";
    String message = "MSH|^~\&|||20120217094337||VXU^V04^VXU_V04|NONE|P|2.5.1|\rPID|1";
    String ack = port.submitSingleMessage(username, password, facilityId, message);
    out.println("<h2>Sent</h2><pre>" + message + "</pre>");
    out.println("<h2>Received</h2><pre>" + ack + "</pre>");
} catch (Exception e) {
    throw new ServletException("Unable to create SOAP connection" + e.getMessage(), e);
}
```

Servlet ConnectivityTest at /SoapClientDemo

Connectivity Result = Test message "Hello" received by host 'ch.astrahosting.com' on Fri, 17 Feb 2012 at 16:48:02 UTC

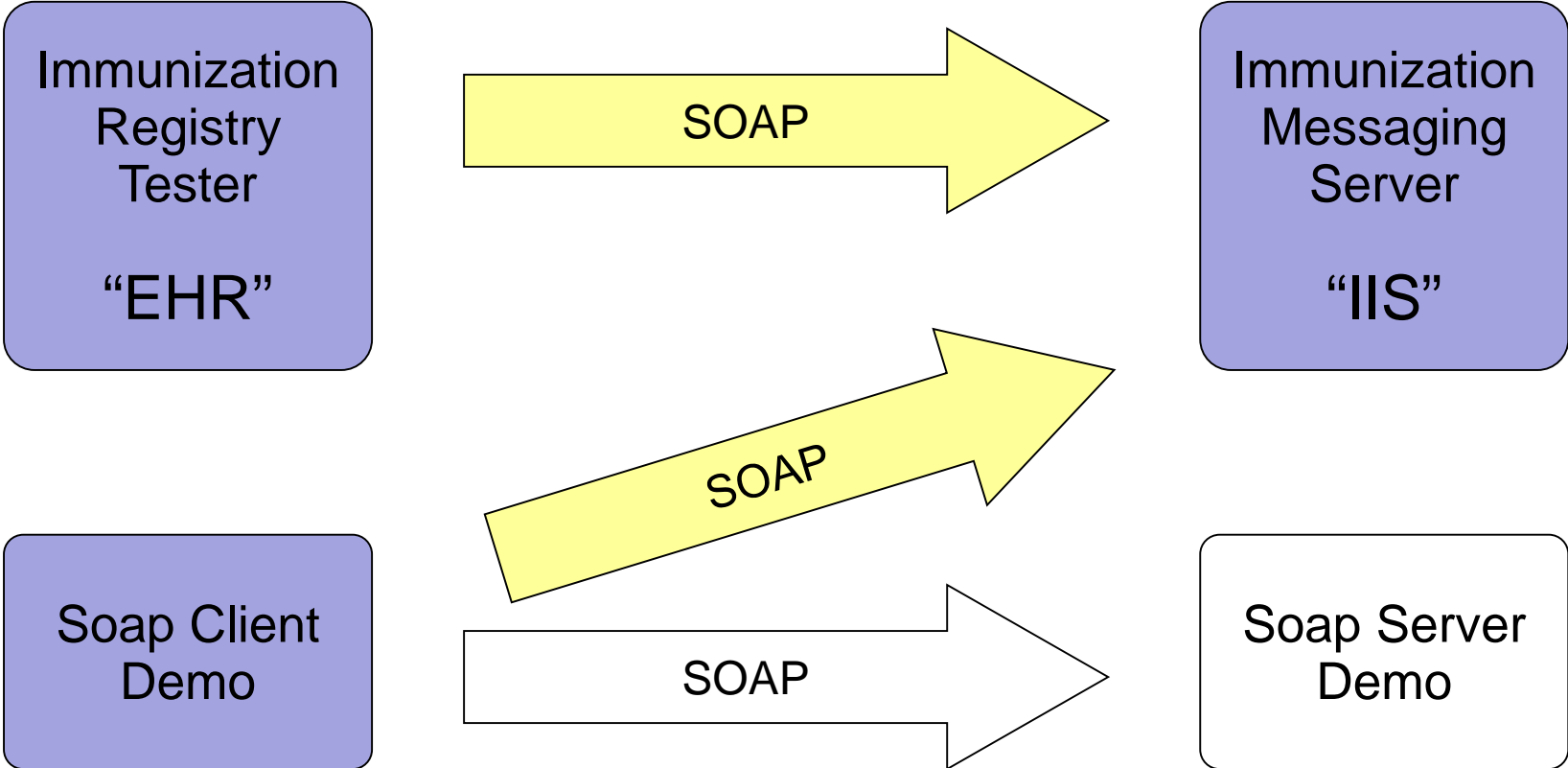
Sent

```
MSH|^~\&|||20120217094337||VXU^V04^VXU_V04|NONE|P|2.5.1|
PID|1||NONE^^^OIS-TEST^MR||Major^Damion^Z^^^L|Howard|20110814|M||307 Bexar Ln^^Coppersville^MI^49404^USA|| (616) 871-6
NK1|1|Howard^Idonia|MTH^Mother^HL70063|
PV1|1|R|
ORC|RE||NONE.1^OIS|
RXA|O|1|20111014||20^DTaP^CVX|999|||01^Historical^NIP0001|A|
ORC|RE||NONE.2^OIS|
```

Received

```
MSH|^~\&|MCIR|MCIR||20120217164802+0000||ACK^V04|20120217164802+0000.1|P|2.5.1|
SFT|OIS|O.13d|DQA||
MSA|AE|NONE|Message rejected: HL7 RXA segment is missing, in repeat #2|
ERR|||O|E|||HL7 RXA segment is missing, in repeat #2|
```

Creating a Web Service



Creating Server

New Web Service from WSDL

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Web Service Name: IIS

Project: SoapServerDemo

Location: Source Packages

Package: com.soapdemo

Select Local WSDL File or Enter WSDL URL:

http://openimmunizationsoftware.us:6394/ImmMessagingServer/client

If WSDL file defines more services and ports press Browse button to select port from which web service will be generated.

Web Service Port: client_Service#client_Port_Soap12

Use Provider

Implement Web Service as Stateless Session Bean

< Back Next > Finish Cancel Help

New class created

```
] /**
 *
 * @author nathan
 */
@WebService(serviceName = "client_Service", portName = "client_Port_Soap12", endpointInterface = "cdc.iisb._2011
@BindingType(value = "http://java.sun.com/xml/ns/jaxws/2003/05/soap/bindings/HTTP/")
public class IIS {

]   public java.lang.String connectivityTest(java.lang.String echoBack) throws cdc.iisb._2011.UnknownFaultMessag
    //TODO implement this method
    throw new UnsupportedOperationException("Not implemented yet.");
-   }

]   public java.lang.String submitSingleMessage(java.lang.String username, java.lang.String password, java.lang.
    //TODO implement this method
    throw new UnsupportedOperationException("Not implemented yet.");
-   }

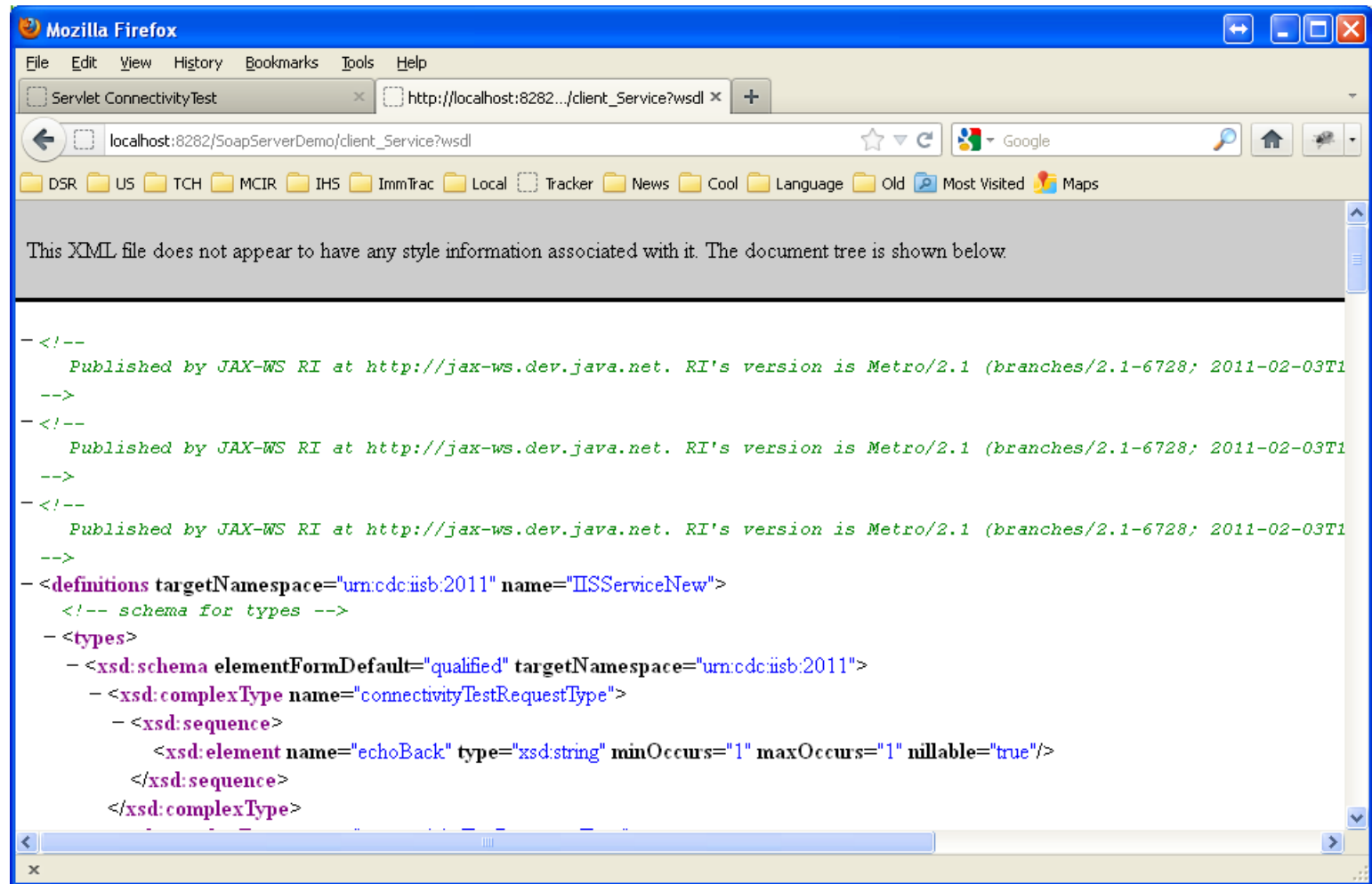
}

}
```

Implemented the methods

```
@WebService(serviceName = "client_Service", portName = "client_Port_Soap12", e:  
@BindingType(value = "http://java.sun.com/xml/ns/jaxws/2003/05/soap/bindings/H  
public class IIS {  
  
    public java.lang.String connectivityTest(java.lang.String echoBack)  
        throws cdc.iisb._2011.UnknownFaultMessage,  
            cdc.iisb._2011.UnsupportedOperationFaultMessage {  
        return "You said '" + echoBack + "'.";  
    }  
  
    public java.lang.String submitSingleMessage(java.lang.String username,  
        java.lang.String password,  
        java.lang.String facilityID,  
        java.lang.String hl7Message)  
        throws cdc.iisb._2011.UnknownFaultMessage,  
            cdc.iisb._2011.MessageTooLargeFaultMessage,  
            cdc.iisb._2011.SecurityFaultMessage {  
        return "Thank your for the message.";  
    }  
}
```

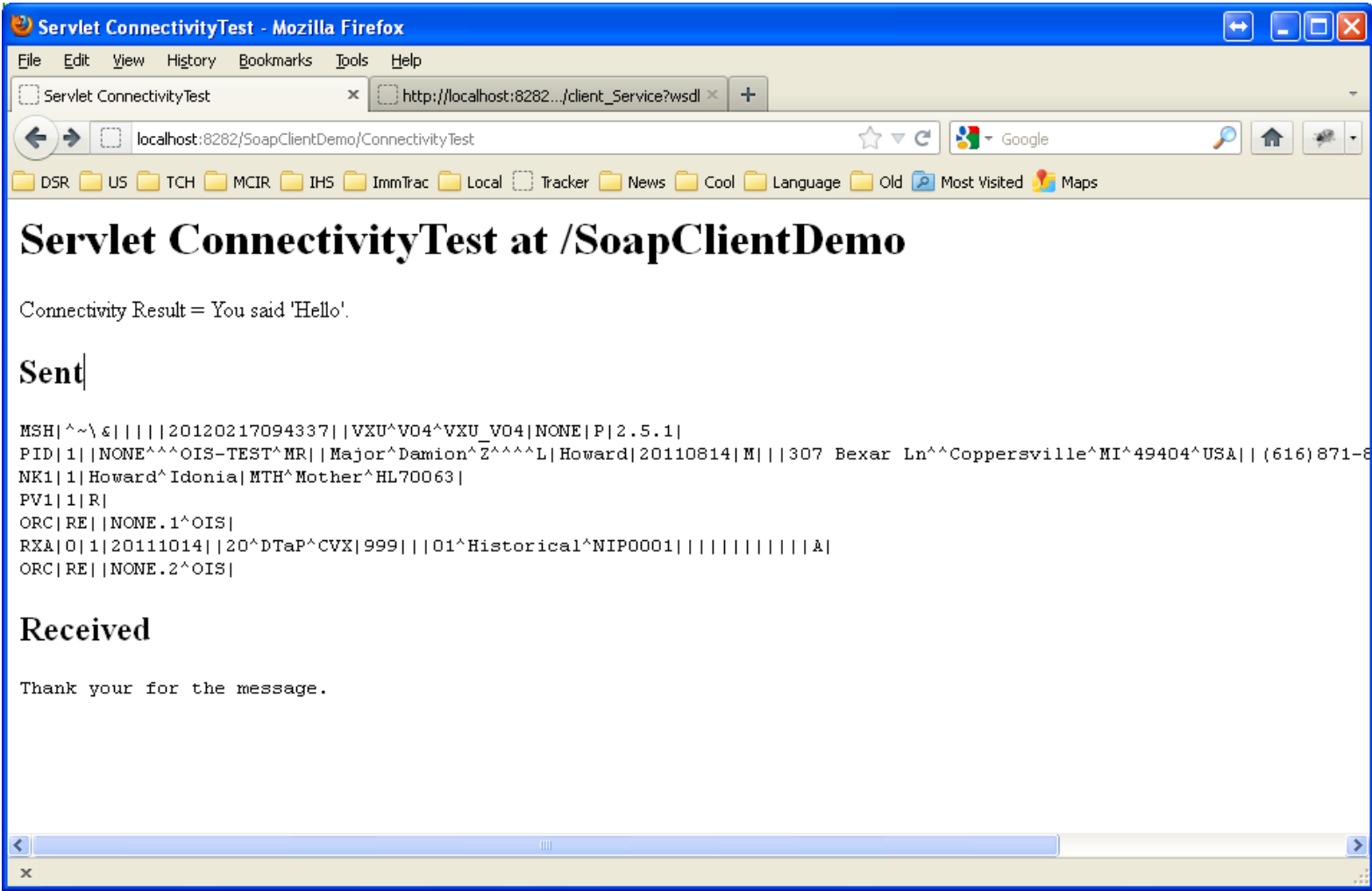
New WSDL up and running



Changed URL in service client

```
ClientService service;
try {
    service = new ClientService(new URL("http://localhost:8282/SoapServerDemo/client_Service?wsdl"));
    IISPortType port = service.getClientPortSoap12();
    String returnMessage = port.connectivityTest("Hello");
    out.println("<p>Connectivity Result = " + returnMessage + "</p>");
    String username = "CDCTLWG";
    String password = "H3490";
    String facilityId = "1200";
    String message = "MSH|^~\&||||20120217094337||VXU^V04^VXU_V04|NONE|P|2.5.1|\rPID|1||NONE^^^OIS-TEST^MR";
    String ack = port.submitSingleMessage(username, password, facilityId, message);
    out.println("<h2>Sent</h2><pre>" + message + "</pre>");
    out.println("<h2>Received</h2><pre>" + ack + "</pre>");
} catch (Exception e) {
    throw new ServletException("Unable to create SOAP connection" + e.getMessage(), e);
}
```

Demo client and server connected



Additional Resources

- **Source code hosted at SourceForge:**
 - **IIS Messaging Client**
<https://sourceforge.net/projects/iismclient/>
 - **IIS Messaging Server**
<https://sourceforge.net/projects/iismserver/>
- **Fully functional implementation:**
 - <http://openimmunizationsoftware.us/>
 - **For client account access or service credentials please contact Nathan Bunker**

Contact Information

Nathan Bunker

435-635-1532

Nathan.Bunker@openimmunizationsoftware.net

<http://openimmunizationsoftware.org/>