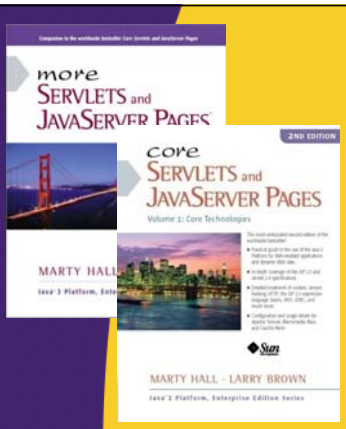
Building Web Services with Apache Axis2

Part II: Clients from Java-First (Bottom-Up) Services

© 2009 Marty Hall



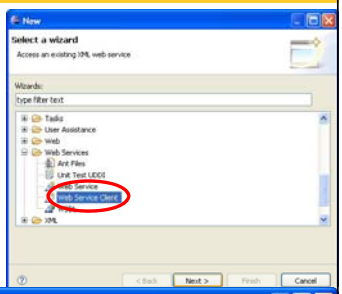
Agenda

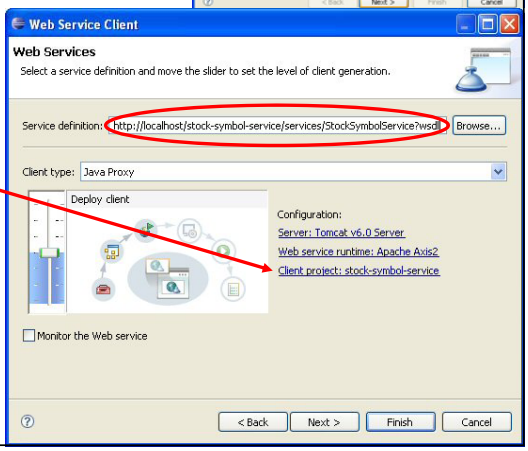
- Making client stubs automatically
- Making clients from client stubs
- Standalone example
- Servlet example

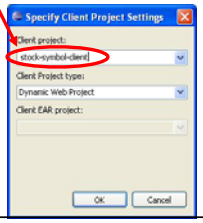
Basic Approach

Make Client Project

- **Specify Web Service Client**
  - File → New → Other → Web Services → Web Service Client → Next
- **Specify WSDL location**
  - For service definition, give URL of WSDL file
- **Specify project name**
  - Click on Client project, enter new name
  - Press OK
  - Press Finish on wizard page

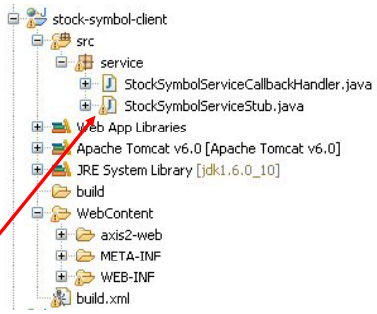






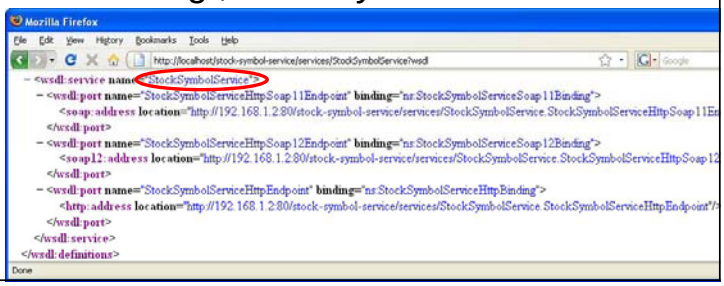
Result: Client Stubs

- **Notes**
  - Your code will use service stub class, but you will not edit the generated files
  - Generated code follows JDK 1.4 style
    - In particular, service stub does *not* use generics



Making a Client: Starting Point

- **Make new class with "main"**
  - I will use package called "client"
- **Specify address of service**
  - http://*host*/*app-name*/services/*service-name*
- **Pass address to constructor of stub**
  - BlahStub stub = new BlahStub(*address-of-service*);
    - Blah is the name of the class from which you built the bottom-up Web Service. E.g., StockSymbolService.
    - More generally, it is the wsdl:service name.



Making a Client: Starting Point

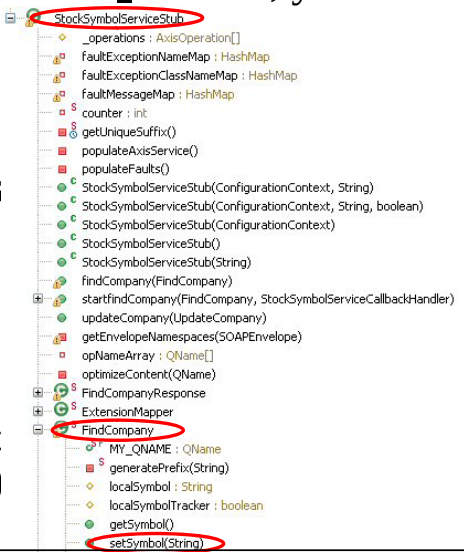
```
package client;

import service.*;

public class StockSymbolClient {
  ...
  public static StockSymbolServiceStub getStub()
      throws Exception {
    String address =
      "http://localhost/stock-symbol-service/" +
      "services/StockSymbolService";
    StockSymbolServiceStub stub =
      new StockSymbolServiceStub(address);
    return(stub);
  }
}
```

Java class name (and wsdl:service name) was StockSymbolService, so you use StockSymbolServiceStub.

Web Services ADB Client: Making Request Object

- **Instantiate a request object**
  - Inner class in the service stub, named after op name
  - For example, if op (method) name is findData, you do:
    - BlahStub.FindData request = new BlahStub.FindData();
- **Set parameters**
  - Find method in bottom-up Java code. If method param name is foo, you do
    - request.setFoo(...)
- **Example**

```
StockSymbolServiceStub.FindCompany request
   new StockSymbolServiceStub.FindCompany()
request.setSymbol("ibm");
```



Web Services ADB Client: Getting Response Data

- **Get response object**
  - E.g., assume service name (original bottom-up Java classname) is Blah and operation (method name in original bottom-up Java class) is findData.
    - BlahStub stub = new BlahStub(*address-of-service*);
    - BlahStub.FindData request = new BlahStub.FindData();
    - BlahStub.FindDataResponse response = stub.findData(request);
      - If there is no return value, just call stub.doSideEffect
      - No need to use response object
- **Extract data from response**
  - SomeType data = response.get_return();
    - The name get_return is specified in WSDL file, but in the case of clients built from a WSDL file that came from a bottom-up client, it will always be get_return.

Notes for Advanced Usage

- **Class and method names**
  - Really come from WSDL file
  - But when building client based on Java-first (bottom-up) service, it is easiest to just look at method and parameter names in Java class (POJO) from which service was built.
- **Asynchronous requests**
  - Simplest approach is to use synchronous (RPC-style) calls to service, as shown here.
  - But you can also use asynchronous (callback-style) calls to service, as will be shown in later section.

Standalone Example

Getting Stub

```
public static StockSymbolServiceStub getStub()
    throws Exception {
  String address =
    "http://localhost/stock-symbol-service/" +
    "services/StockSymbolService";
  StockSymbolServiceStub stub =
    new StockSymbolServiceStub(address);
  return(stub);
}
```


Invoking Operations: Example 1 (Getting Return Value)

```
public static String findCompany(StockSymbolServiceStub stub,
                                 String stockSymbol)
    throws Exception {
  StockSymbolServiceStub.FindCompany request =
    new StockSymbolServiceStub.FindCompany();
  request.setSymbol(stockSymbol);
  StockSymbolServiceStub.FindCompanyResponse response =
    stub.findCompany(request);
  return(response.get_return());
}
```

Invoking Operations: Example 1 (Continued)

```
public static void main(String[] args) {
  try {
    StockSymbolServiceStub stub = getStub();
    String  stockSymbol = "ORCL";
    System.out.printf("Company for '%s' is '%s'%n",
                      stockSymbol,
                      findCompany(stub, stockSymbol));
  } catch(Exception e) {
    System.err.println("StockSymbolService error");
    e.printStackTrace();
  }
}
```

Output: Company for 'ORCL' is 'Oracle Corp.'

Invoking Operations: Example 2 (No Return Value)

```
public static void updateCompany(StockSymbolServiceStub stub,
                                 String stockSymbol,
                                 String company)
    throws Exception {
  StockSymbolServiceStub.UpdateCompany request =
    new StockSymbolServiceStub.UpdateCompany();
  request.setSymbol(stockSymbol);
  request.setCompany(company);
  stub.updateCompany(request);
}
```

Invoking Operations: Example 2 (Continued)

```
public static void main(String[] args) {
  try {
    StockSymbolServiceStub stub = getStub();
    String  stockSymbol = "ORCL";
    System.out.printf("Company for '%s' is '%s'%n",
                      stockSymbol,
                      findCompany(stub, stockSymbol));
    updateCompany(stub, stockSymbol, "MySQL Corp.");
    System.out.printf("Company for '%s' is '%s'%n",
                      stockSymbol,
                      findCompany(stub, stockSymbol));
  } ...
```

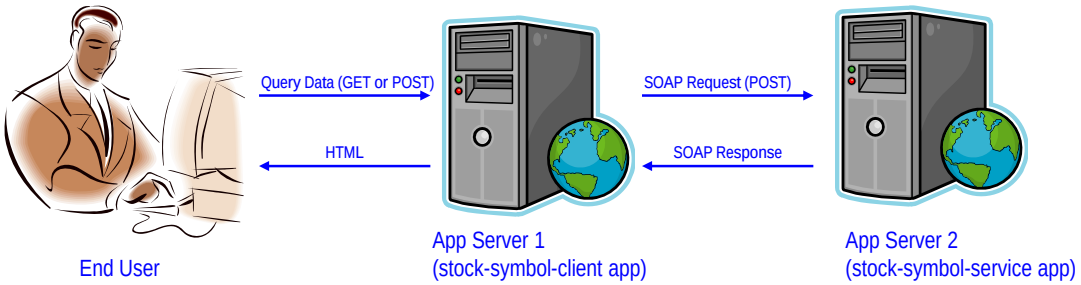
Output: Company for 'ORCL' is 'Oracle Corp.'
Company for 'ORCL' is 'MySQL Corp.'

Servlet Example

Idea

- **Start with code from previous section**
  - StockSymbolClient.getStub
  - StockSymbolClient.findCompany
  - StockSymbolClient.updateCompany
- **Make servlet that looks up company name**
  - Read request param for stock symbol
  - Call findCompany
  - Use MVC
- **Make servlet that updates company name**
  - Read request params for stock symbol and company
  - Call updateCompany
  - Use MVC

Architecture



Lookup: Servlet Code

```
public class StockSymbolServlet extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
    throws ServletException, IOException {
    String stockSymbol =
      request.getParameter("stockSymbol");
    String address;
```


Lookup: Servlet Code (Continued)

```
    if (isEmpty(stockSymbol)) {
      address = "missing-symbol.jsp";
    } else {
      try {
        StockSymbolServiceStub stub =
          StockSymbolClient.getStub();
        String companyName =
          StockSymbolClient.findCompany(stub, stockSymbol);
        CompanyBean company =
          new CompanyBean(stockSymbol, companyName);
        request.setAttribute("company", company);
        address = "show-company.jsp";
      } catch(Exception e) {
        address = "service-error.jsp";
      }
    }
```

Lookup: Servlet Code (Continued)

```
    address = "/WEB-INF/results/" + address;
    RequestDispatcher dispatcher =
      request.getRequestDispatcher(address);
    dispatcher.forward(request, response);
  }

  private boolean isEmpty(String value) {
    return((value == null) ||
           (value.trim().equals("")));
  }
}
```


Lookup: web.xml

```
...
  <servlet>
    <servlet-name>Stock Symbol Lookup Servlet</servlet-name>
    <servlet-class>client.StockSymbolServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Stock Symbol Lookup Servlet</servlet-name>
    <url-pattern>/stock-symbol-lookup</url-pattern>
  </servlet-mapping>
...
```

Lookup: Bean

```
public class CompanyBean implements Serializable {
  private String stockSymbol;
  private String companyName;

  public CompanyBean(String stockSymbol,
                     String companyName) {
    setStockSymbol(stockSymbol);
    setCompanyName(companyName);
  }
  public String getStockSymbol() {
    return(stockSymbol);
  }
  public void setStockSymbol(String stockSymbol) {
    this.stockSymbol = stockSymbol;
  }
  public String getCompanyName() {
    return(companyName);
  }
  public void setCompanyName(String companyName) {
    this.companyName = companyName;
  }
}
```

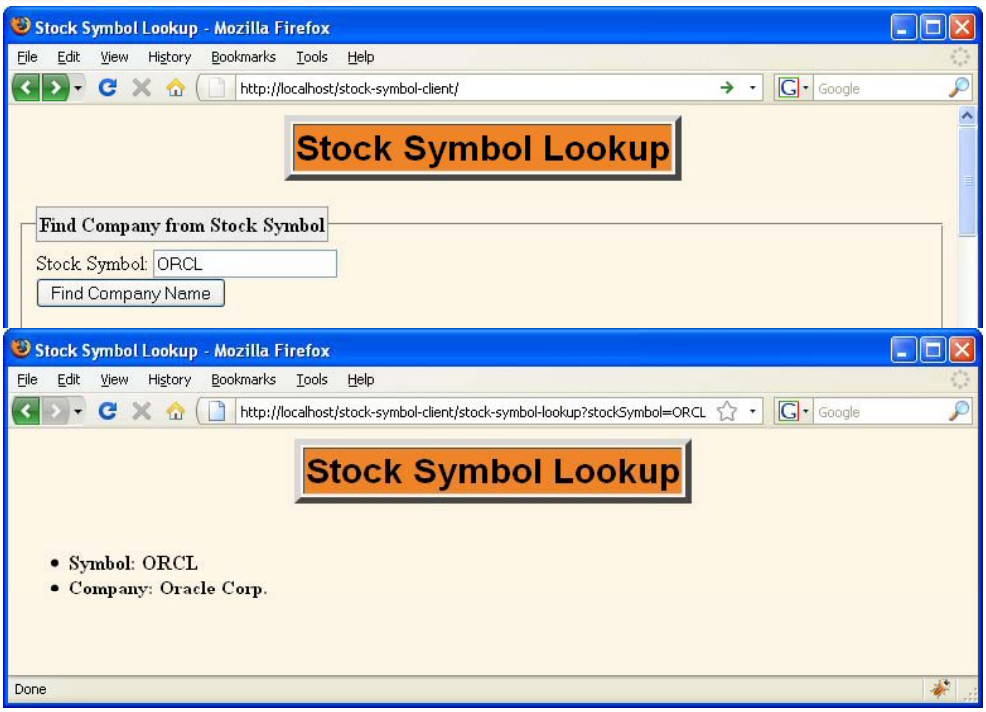

Lookup: Main Result Page

```
<!DOCTYPE ...>
<HTML>
<HEAD>
<TITLE>Stock Symbol Lookup</TITLE>
<LINK REL=STYLESHEET
      HREF="styles.css"
      TYPE="text/css">
</HEAD>
<BODY>
<TABLE ALIGN="CENTER" BORDER="5">
  <TR><TH CLASS="TITLE">Stock Symbol Lookup</TH></TR>
</TABLE><BR>
<UL>
  <LI><B>Symbol: ${company.stockSymbol}</B></LI>
  <LI><B>Company: ${company.companyName}</B></LI>
</UL>
</BODY></HTML>
```

Lookup: Input Form

```
...
<FIELDSET>
  <LEGEND>Find Company from Stock Symbol</LEGEND>
  <FORM ACTION="./stock-symbol-lookup">
    Stock Symbol:
    <INPUT TYPE="TEXT" NAME="stockSymbol"><BR>
    <INPUT TYPE="SUBMIT" VALUE="Find Company Name">
  </FORM>
</FIELDSET>
...
```


Lookup: Results



Update: Servlet Code

```
public class UpdateServlet extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    String stockSymbol =
      request.getParameter("stockSymbol");
    String companyName =
      request.getParameter("companyName");
    String address;
```


Update: Servlet Code (Continued)

```
    if (isEmpty(stockSymbol) || isEmpty(stockSymbol)) {
      address = "missing-input.jsp";
    } else {
      try {
        StockSymbolServiceStub stub =
          StockSymbolClient.getStub();
        StockSymbolClient.updateCompany(stub,
                                        stockSymbol,
                                        companyName);
        CompanyBean company =
          new CompanyBean(stockSymbol, companyName);
        request.setAttribute("company", company);
        address = "show-update.jsp";
      } catch(Exception e) {
        System.err.println("StockSymbolService error");
        address = "service-error.jsp";
      }
    }
```

Update: Servlet Code (Continued)

```
    address = "/WEB-INF/results/" + address;
    RequestDispatcher dispatcher =
      request.getRequestDispatcher(address);
    dispatcher.forward(request, response);
  }

  private boolean isEmpty(String value) {
    return((value == null) ||
           (value.trim().equals("")));
  }
}
```


Update: web.xml

```
...
  <servlet>
    <servlet-name>Stock Symbol Update Servlet</servlet-name>
    <servlet-class>client.UpdateServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Stock Symbol Update Servlet</servlet-name>
    <url-pattern>/stock-symbol-update</url-pattern>
  </servlet-mapping>
...
```

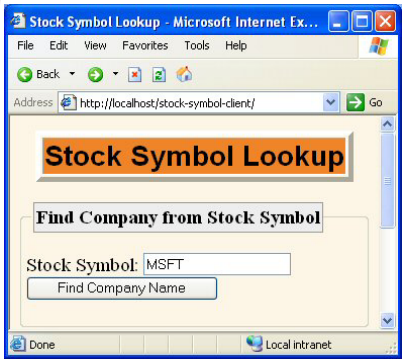
Update: Main Result Page

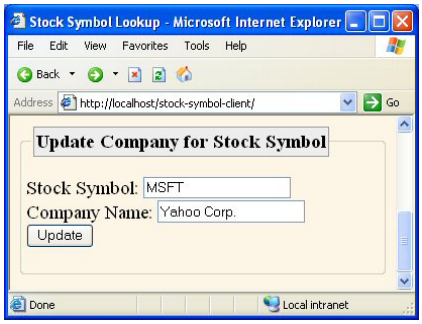
```
<!DOCTYPE ...>
<HTML>
<HEAD>
<TITLE>Successful Update</TITLE>
<LINK REL=STYLESHEET
      HREF="styles.css"
      TYPE="text/css">
</HEAD>
<BODY>
<TABLE ALIGN="CENTER" BORDER="5">
  <TR><TH CLASS="TITLE">Successful Update</TH></TR>
</TABLE><BR>
<UL>
  <LI><B>Symbol: ${company.stockSymbol}</B></LI>
  <LI><B>Company: ${company.companyName}</B></LI>
</UL>
</BODY></HTML>
```

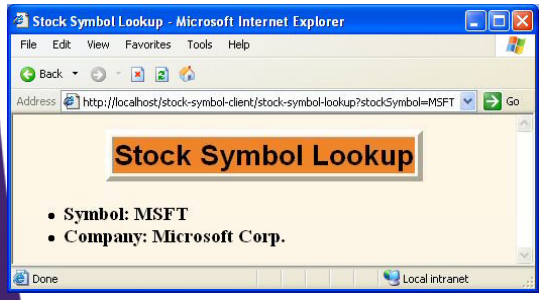

Update: Input Form

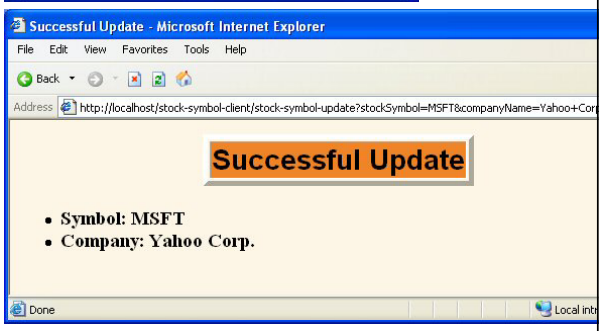
```
...
<FIELDSET>
  <LEGEND>Update Company for Stock Symbol</LEGEND>
  <FORM ACTION="./stock-symbol-update">
    Stock Symbol:
    <INPUT TYPE="TEXT" NAME="stockSymbol"><BR>
    Company Name:
    <INPUT TYPE="TEXT" NAME="companyName"><BR>
    <INPUT TYPE="SUBMIT" VALUE="Update">
  </FORM>
</FIELDSET>...
```

Update: Results









Summary

- **Make client app with stubs**
  - File → New → Other → Web Services → Web Service Client → Next
  - For service definition, give URL of WSDL file
- **Make client code (standalone or servlet)**
  - Make stub
    - BlahStub stub = new BlahStub(*address-of-service*);
  - Get request object
    - BlahStub.FindData request = new BlahStub.FindData();
  - Set parameters
    - request.setFoo(...);
  - Get response object and extract data
    - BlahStub.FindDataResponse response = stub.findData(request);
    - SomeType data = response.get_return();

Questions?