Virginia Tech ■ ECE/CS 4570: Wireless Networks and Mobile Systems ■ Spring 2006

At-home Exercise 4 (E04)

Report Due: February 16, 2006 (at 4:00 PM)

Part I – Objectives and Laboratory Materials

Objectives

The objectives of this assignment are to:

- Introduce the design of a web service server; and
- Introduce using the C# programming language and .NET.

After completing the assignment, you should be able to:

- Design and implement web service clients and servers; and
- Understand the role of middleware (XML and SOAP in this context) and how they enhance communication.

Hardware to be used in this assignment

- Notebook computer
- iPAQ handheld computer
- IEEE 802.11b access point
- IEEE 802.11b network interface card

Software to be used in this assignment

- Microsoft Visual Studio .NET 2003 (VS.NET or, simply, VS)
- .NET Compact Framework (.NETcf)

Overview

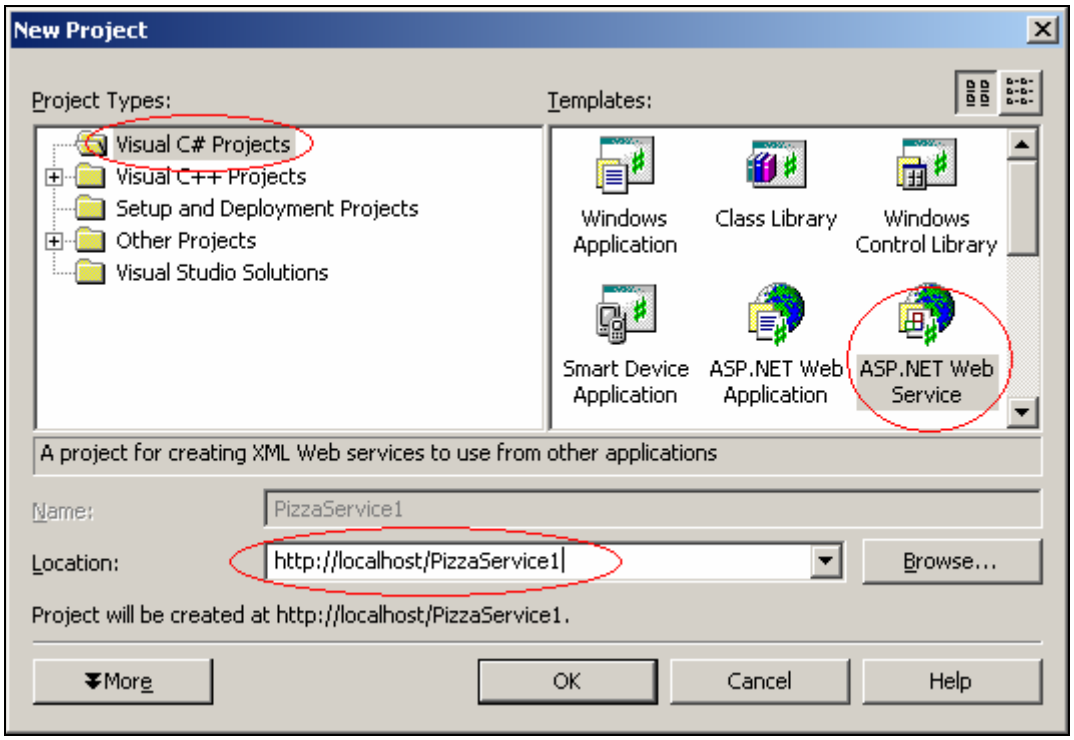
In this assignment you will design a web service server application. In this context, a web service is an application running on the notebook that offers some sort of "service" to the client, which is the iPAQ in this case. Using the pizza client you designed in the in-class laboratory session, you will write the service back-end of the client-server pair. You will also analyze the operation of the web service observed during the in-class laboratory session.

Part II – At-home Laboratory Assignment

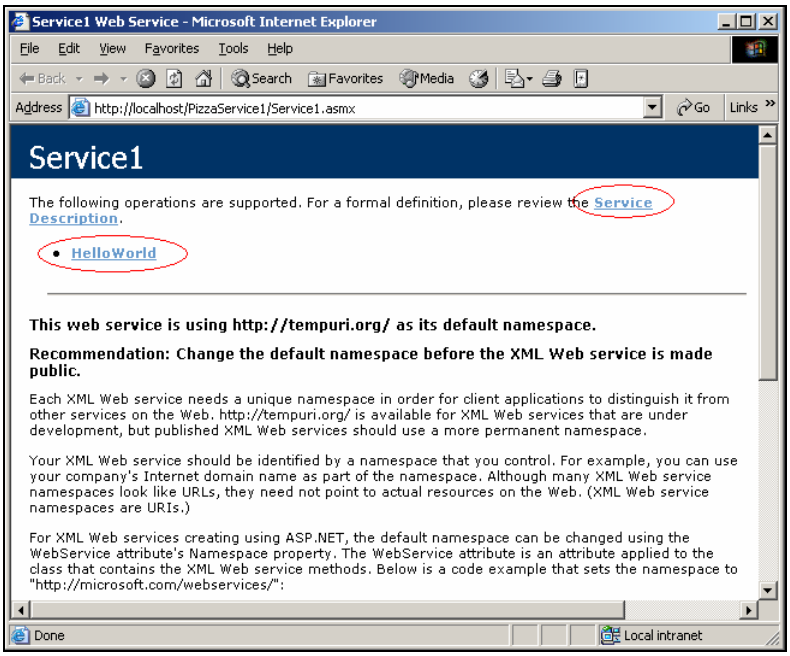
Create a web service

The first step is to create a new web service project using VS.NET.

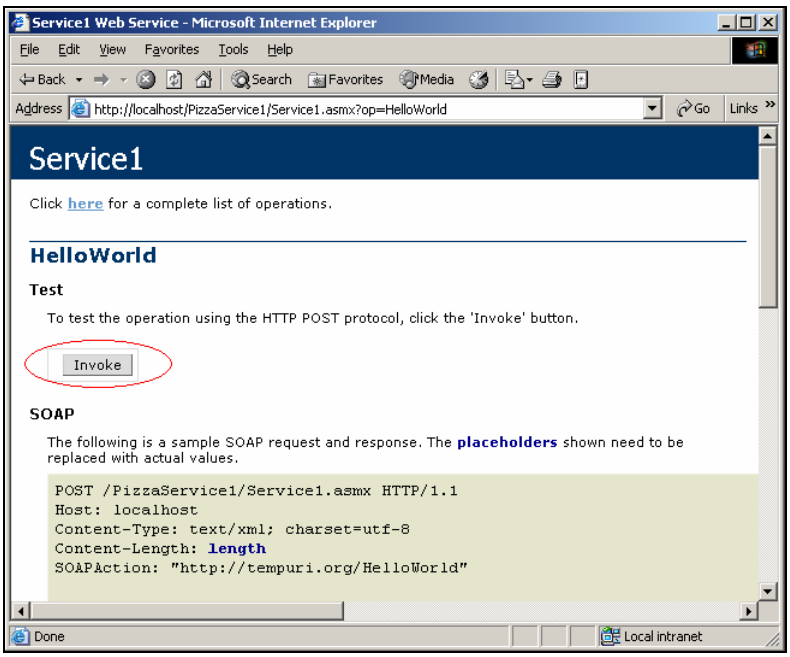
1. Start a new project in VS using the Visual C# project template of type ASP.NET Web Service, as shown below. Give the project a meaningful name.



2. Next, find the web service. Find the directory that contains this project. It is typically in the "Visual Studio" directory in the "My Documents" folder. Inside the project directory there should be two files. This should be raise some questions, since after all, VS projects typically have a lot more than that files! The reason for this is that the rest of the project is in the web server directory tree. Open the directory "C:\Inetpub\wwwroot." In this directory you will find a directory with the same name as your project. Opening that directory will reveal something that looks more like a typical VS project. This step has nothing to do with the creation of the project, it just provides some possibly useful information.
3. Returning to VS, click the "click here to switch to code view" link in the middle of the page. This is the main file of the project. Scroll down to the bottom and you will see a commented section beginning with "[WebMethod]." The [WebMethod] line specifies that this method will be a web service method. Only methods preceded by [WebMethod] will be available to the world. The example web method provided is, of course, a "hello world" method. Uncomment the five lines of the method and build the project. Run the project.
4. Conduct the following simple test to verify that the project was built and runs correctly.
    a. When you run the project, a web browser will open and point you to a web page. (Note the URL of this page. You might need to look back at it.) This page is generated by the local web server, Internet Information Server (IIS). This page gives a link to the XML service description. Feel free to take a look at this. The next section shows the available web methods. Every web method you defined in this project should be here. For now, we only have "HelloWorld." Click the link for HelloWorld (see below).



b. This leads to a new page. This page has an invoke button and some definitions concerning the use of the method. If there were any parameters to be passed to the function, there would be space to input those parameters. Click the “Invoke” button.



c. The returned XML document is the response that a client would receive if they called the web method, “HelloWorld,” with the parameters we specified, in this case, none.

d. This is a way of testing your web methods to see if there is an issue with the web method or the calling client method. Unfortunately, you cannot specify object attributes.

e. You may now close the browser.

How does it work?

For web services to work, the web server must know what to do with the connection it is receiving. For our setup using IIS, IIS knows how to interpret the .asmx file extension type. Active Server Pages .NET (ASP.NET) is used when dealing with .asmx files. When a file with the .asmx extension is requested, IIS knows what to do with the data being sent to it. More about this will be seen in the latter part of this lab.

Your task

For this section of the at-home assignment, you will design the back-end that was provided during the in-class laboratory session. It should match the method definition presented in the in-class laboratory session. It should return the same values as well. It **does not** need to store the values of the orders placed; it can simply return the string with a summary of the information provided by the client. The return string has the following structure. Note that each identifier is followed by a single space.

<clientID> <string for size> <string for crust> <topping1> <topping2>…

For testing, set up the iPAQ and notebook to communicate using IEEE 802.11b. You can use either ad hoc mode or infrastructure mode with the access point. Either way works. Assign each device an IP address from the private address space (192.168.*x*.*y*) and an appropriate network mask. Test the connectivity by pinging the iPAQ from the notebook.

Test your server with the client you developed in the in-class laboratory session. You will have to modify the web reference to point to your notebook's server instead of the service provided on the GTA's notebook that was used in the in-class laboratory session. This will involve changing the URL to point to the new service and rebuilding the client project. (See step 4.a above for an easy way of finding the service URL.)

You will also need to define the *pizzaOrder* object in your **service** code. It does not need to have any bracketed [ ] definitions added; just simply copy and paste it in.

Part III – The Report

Your report should cover both the in-class and the at-home aspects of this week's work. The report will be graded for both form and content. The report must be submitted in electronic form to the Digital Dropbox on the Blackboard site for this course. Each group must submit *one* report for the entire group.

Provide a report that answers each of the following questions *in the order specified here*. (If needed, feel free to do some "Googling" to answer some questions.)

Report Part I: Report for the In-class Laboratory Session

Use the trace you captured with Ethereal to do the following.

1. Neatly format the trace file. For the XML contents, add tabs to structure the text so that it is easy to discern the beginning and end of the different sections.
2. Divide the overall session into application-level transfers and identify each transfer as being data from the client to the server or data from the server to the client.
3. For the SOAP contents, identify the values being passed **into** and **returned from** the web method call.
4. Search the session for instances of "http://tempuri.org." Why is this link present and where does it point (semi-trick question)?

Report Part II: Report for the At-home Assignment

1. Include your code for the *placeOrder* web method that you designed.
2. Assuming that the pizza orders were written to a file or "executed," what are three other functions that could be added to the pizzeria's web services to help the employees and the management?
3. Web services serve the same functionality as remote procedure calls (RPCs). List the advantages and disadvantages of using web services instead of RPCs. List at least two advantages and at least two disadvantages.
4. What are two other application areas for web services besides ordering pizza (outside of a pizzeria environment)?

Report Part III: Conclusions

What did you learn from this lab? What sections are unclear or you would like to know more about? What are some suggestions, comments, and criticisms of this lab and the tools used?