First note: This document may be updated during the labs. Please download a new version from time to time. Changes will be announced via email.

1 Introduction

From Wikipedia, the free encyclopedia (http://en.wikipedia.org/wiki/Webservice):

A Web service (also Web Service) is defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network".[1] Web services are frequently just Web application programming interfaces (API) that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services. Other approaches with nearly the same functionality as web services are Object Management Group’s (OMG) Common Object Request Broker Architecture (CORBA), Microsoft’s Distributed Component Object Model (DCOM) or SUN’s Java/Remote Method Invocation (RMI).

The W3C Web service definition encompasses many different systems, but in common usage the term refers to clients and servers that communicate over the HTTP protocol used on the Web. Such services tend to fall into one of two camps: Big Web Services and RESTful Web Services.

"Big Web Services" use Extensible Markup Language (XML) messages that follow the Simple Object Access Protocol (SOAP) standard and have been popular with traditional enterprise. In such systems, there is often a machine-readable description of the operations offered by the service written in the Web Services Description Language (WSDL). The latter is not a requirement of a SOAP endpoint, but it is a prerequisite for automated client-side code generation in many Java and .NET SOAP frameworks (frameworks such as Spring, Apache Axis2 and Apache CXF being notable exceptions). Some industry organizations, such as the WS-I, mandate both SOAP and WSDL in their definition of a Web service.

1.1 SOAP

From Wikipedia, the free encyclopedia (http://en.wikipedia.org/wiki/SOAP_(protocol)):

SOAP, originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of
Web Services in computer networks. It relies on Extensible Markup Language (XML) as its message format, and usually relies on other Application Layer protocols (most notably Remote Procedure Call (RPC) and HTTP) for message negotiation and transmission. SOAP can form the foundation layer of a web services protocol stack, providing a basic messaging framework upon which web services can be built.

As a layman’s example of how SOAP procedures can be used, a SOAP message could be sent to a web service enabled web site (for example, a house price database) with the parameters needed for a search. The site would then return an XML-formatted document with the resulting data (prices, location, features, etc). Because the data is returned in a standardized machine-parseable format, it could then be integrated directly into a third-party site.

1.2 WSDL


2 Assignment

In the second assignment you are going to write a webservice and a client for the messaging service.

The global idea is that we make one large messaging service. There is a main webservice available where you can register your webservice. The only reason for this is that others can find you, and you can find others. You are going to write a web service that holds messages sorted by topic. Topics are defined by one keyword. Messages have a number, topic, title and body.

2.1 Netcat

The tool netcat (nc) provides you a way to create and interact with socket servers. See man nc for more information.

2.1.1 HTTP

For example when you start a listening netcat session (act as a server) in your terminal, and access this using your browser, you can see parts of the HTTP protocol.

1. Start a terminal.
2. Type nc -l 8888
3. Start your webbrowser
4. Go to http://localhost:8888/

What do you see in your terminal window? That is a HTTP request! Now we can use that request, so copy it and end your netcat session.

We now use a connecting netcat session (without the -l option) to act as a browser.

1. Type in your terminal nc www.science.uva.nl 80
2. paste your (modified) HTTP request.

What is the response now? Try this with other sites, can you fool them? If we can do this, how safe is HTTP? Keep this in mind while you are surfing over the internet...

Can you get output to your webbrowser using the netcat server in the first example? That means: You got the request from the browser, can you send a valid response back? How hard is it to write a webserver then?

2.1.2 SOAP

Now you have done this using HTTP, SOAP is based on the HTTP protocol, figure out the difference and add this to your report. Also add a the plain request and response to and from your webservice in an appendix of your report.

2.2 Main webservice

The main webservice is provided to you. You can call the following methods to this webservice:

- **hello()** This method says hello back to you and is provided only for debugging and testing purpose.

- **register(name, hostname, port)** This method registers your webservice called name at the machine hostname at port port. It returns a confirmation with your assigned server number. The name should be your own name or team name. If you use a team name, I want to know the who is in which team.

- **deregister(number)** This method deregisters your webservice with the assigned number number.

- **getServices()** This method allows you to get a list of all registered and active web services.

The main webservice is running at host ow121.science.uva.nl:8008. When this machine is restarted for some reason, there is a backup at ow156.science.uva.nl:8008. When you see that one of the services is down, send me an email ASAP.

2.3 Your Webservice

The webservice that you are going to implement should register with the main webservice at startup, and deregister when it shuts down.

The methods that you need to implement are:

- **addMessage(topic, title, body)** This method adds a message with topic topic, title title and body body to the message list. The method should return a confirmation.

- **search(topic)** This method searches all available webservises for messages with topic topic. It is intended for use by the clients. The method returns for each message that is found the server-number and title.
• searchLocal(topic) This method searches the local webservice for messages with topic topic. It is intended for use by the other servers. The method returns for each message that is found the server-number and title.

• getMessage(server, number) Retrieves the message with number number from server server and returns it.

• ping() This method returns a confirmation.

You should also provide a WSDL file for your webservice.

2.4 Your webservice client

Your client should be able to search the global pool by connecting to your webservice.

Providing a command line client is enough for this assignment. A dummy execution of your client could be:

Welcome to this webservice client.
Now connected to: ow152.science.uva.nl:8080
Choose:
1. Write message.
2. Search messages.
3. Read message.
4. Quit.

1
Give a topic:
telematica
Give a title:
practical assignment 2
Give the message (end with <CRLF>.<CRLF>):
In the second assignment you are going to write a webservice and a client for the webservice.

The global idea is that we make one large messaging service. There is a main webservice available where you can register your webservice.

Your message "practical assignment 2" is added to "telematica".

Choose:
1. Write message.
2. Search messages.
3. Read message.
4. Quit.

2
Give a topic:
multimedia
Searching for messages about "multimedia".
- No messages found about "multimedia".

Choose:
1. Write message.
2. Search messages.
3. Read message.
4. Quit.

2
Give a topic:
telematica

Searching for messages about "telematica".
- found 3 messages about "telematica"
  server-number title
  23-2     assignment 1
  15-4     college 4 - sheets online
  23-30    practical assignment 2

Choose:
1. Write message.
2. Search messages.
3. Read message.
4. Quit.

3
Give the message server-number:
23-30

Topic: telematica
Title: practical assignment 2
Message:
In the second assignment you are going to write a webservice and a client for the webservice.

The global idea is that we make one large messaging service.
There is a main webservice available where you can register your webservice.

Choose:
1. Write message.
2. Search messages.
3. Read message.
4. Quit.

4
Bye!
3 Implementation

Part of this assignment is to figure out how you use and create web services. In most cases you need special tools or programs for this. Java webservices work only in combination with a webservice. There is also a good SOAP library with build-in webservice for Python. Any language is allowed, but you should first discuss this with me. If nothing suits for you, you may implement your webservice just using a socket, but note that you should follow the specifications of SOAP and WSDL.

4 Report

Your report should contain an introduction, describing web services, SOAP and WSDL. It should also contain your implementation decisions and problems. Of course, you should end with a conclusion, remarks and a sample execution.

Provide a small tutorial for me to run your code.

4.1 SOAP

When you describe web services and SOAP, add the execution in the section netcat 2.1 to motivate your answers.

5 Planning

For this assignment you have again three weeks, but there is one week of “vacation” in between. That means that you actually can work four weeks on the assignment!

There will be NO lab session on Wednesday April 29.

5.1 Week 1

In this week you should read about web services, SOAP and WSDL. You should also choose your language and make a start with your implementation.

5.2 Week 2

Finish your webservice and WSDL file. Also, try to start with your client.

5.3 Week 3

Finish your client and write the report. Show your working webservice during this lab session. When you can not make it to be there, please contact me for an additional date.

Deadline: Tuesday May 19, 23:59:59.

Note that the deadline is very strict. If, for some good reason, you can not hand in your assignment before the deadline, please contact me as soon as possible. We can find a way to solve the problem. This means that an email five minutes before the deadline will not result in extension of the deadline, but
a few days before might. The assignment should be submitted via blackboard. Also read the instructions there!

6 Assessment

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Handing in before the deadline</td>
<td>1.0</td>
</tr>
<tr>
<td>Your service</td>
<td>2.5</td>
</tr>
<tr>
<td>Your WSDL file</td>
<td>0.5</td>
</tr>
<tr>
<td>Your client</td>
<td>2.0</td>
</tr>
<tr>
<td>coding style and comments</td>
<td>1.0</td>
</tr>
<tr>
<td>Report</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>9.0</strong></td>
</tr>
</tbody>
</table>

7 Bonus

Of course, you can get the extra credit if you do some more work. Think of a way to extend your service and discuss with me if and how many extra credits you can earn.

8 Contact

If you have any questions, feel free to ask them during the lab session. I am there to help you. If you have additional questions after the labs, you may send me an email.

Good luck with the assignment.