

Building Web Services The Right Way

Yasser Shohoud
Program Manager
XML Messaging
Microsoft



Agenda

- How we build Web services today
- What are Web services
- The right way to build Web services

Interesting Quote

“The most widespread tools take an existing Java class or COM object and then generate a WSDL definition. This is backwards and half-assed. It's backwards because -- as we should have learned with earlier infrastructures -- the right thing to do is write the interface first. It's half-assed because while everybody's generating interfaces, nobody is capable of automatically consuming them.”

Rich Salz, May 15, 2002

<http://www.xml.com/pub/a/2002/05/15/ends.html>

What Are Web Services

- A platform for building *distributed interoperable* applications
- The platform has a standards
 - XML 1.0 serialization format
 - W3C XML Schemas type system
 - WSDL interface description

How Do You Build Them Today?

- Create a class
- Add methods
- Expose the class as a Web service
- Your focus
 - On the class and its methods/parameters
 - In the language you use e.g. C#, VB, Java etc.

The Illusion

- Web service tools perform illusions
 - To make it look like you are receiving an object or returning one
 - It's really all about XML and XML Schemas

What Your Focus Should Be

- On the XML message
 - Both message content and message exchange pattern
- Model the message not the interface
 - Using XSD

Building a Web Service

- Design your messages
- Design your operations
- Implement the service

Versioning a Web Service

- Design new messages
- Define new operations
- Implement new operations

Summary

- Web services are all about XML messages
 - Interfaces are messages and their exchange patterns
- Begin by designing your interface
 - Focus on what matters
- Always think of the wire-level message
 - Throughout the process

Resources

- <http://msdn.microsoft.com/webservices>
- <http://www.LearnXmlws.com>
- Real World XML Web Services, Addison Wesley
 - <http://www.LearnXmlws.com/book>



We're Hiring

- Test Lead – Messaging Framework
- Test Lead – Security Framework
- Test Lead – Scenarios
- Test Lead – Performance & Benchmark
- Software Design Engineers in Test
 - Different levels across all feature teams

Skills Desired

- ***Exceptional*** Testing Skills
- Working knowledge
 - Distributed systems
 - Multi-threaded & Asynchronous software
 - Networking protocols
 - Internet protocols
 - Server Products Development/Test Experience
- Strong Algorithms, Data structures, C, C++, C#
- SOAP & XML