Web Services & .NET Technologies

Ahmed Abu Jbara
Web Services & .NET Technologies

- Why Web Services?
- Before Web services.
- Advantages of Web Services.
- Web Services basics and architecture.
- Development environments for Web Services.
- .NET and Web Services.
- Creating web services using .Net.
- .Net vs. other technologies (J 2EE)
- Conclusion
Why Web Services?

- Human being that makes sense of a web page.
- Computers can’t understand what is going in a browser.
- Web services provide computer-to-computer interactions.
Before Web Services

- Service Oriented Architecture (SOA).
- Applications were designed as services that run on clusters.
- Tightly coupled, distributed computing protocols, such as DCOM, CORBA, or RMI, were used to enable clients to access these services.
- They limit the flexibility of the whole system and the reusability of individual services.
- Web Services represent the convergence between SOA and the web.
Advantages of Web Services

- Integration: Web services offer a big step towards application-to-application communication.
- Service Reuse: ability to reuse a service many times rather than creating a new service for the same function.
- Business Flexibility: Web services allow to build new solutions quickly, at a reasonable cost, and with the reusability feature more flexibility is achieved.
Web Services basics and architecture

- Communicate over open protocols such as HTTP
- Using structured forms of XML
- SOAP (Simple Object Access Protocol)
- WSDL (Web Services Description Language)
- UDDI (Universal Description, Discovery and Integration)
Development environments for web services

• Standards enable the opportunity for web services deployment.
• Success of web services depend on the ability of programming environments to provide developers with tools.
• Microsoft, IBM, and other companies developed the SOAP XML-based format and WSDL.
• IBM WebSphere SDK, SunOpen Net Environment (Sun ONE), Mono project, Microsoft .NET
Microsoft .NET and Web Services

- "Net is a series of Microsoft technologies interconnecting the world of information, people and devices", Microsoft’s definition.
- .NET is a collection of resources that includes development tools and languages, server software, and protocols.
- The foundation of .NET platform is the incorporation of the eXtensible Markup Language (XML) and the Simple Object Access Protocol (SOAP).
- The infrastructure that ties .NET technologies together is the .NET Framework that allows applications to access system services.
Microsoft .NET and Web Services

- .NET architecture is mainly composed of six layers

![Diagram of .NET architecture layers]

- Web services
- Frameworks and libraries: ASP.NET, ADO.NET, Window Forms
  - Interchange standards: SOAP, WSDL
  - Common development tools: Visual Studio.NET
- Component model
- Object model and common language specification
- Common language runtime
Microsoft .NET and Web Services

- .NET removes the distinction between traditional software application development and web development.
- .NET offers mechanisms that make a Web page useful as both a human interface and an application program interface.
- .NET provides development support for source code in three major programming languages, Visual C++, Visual C#, and Visual Basic .NET.
- A main example of the usage of Web Services in .NET technology is the Microsoft "Passport" technology which is built-in windows XP.
Microsoft .NET and Web Services

• .NET provides Web Services developers with a variety of tools to create Web Services.
• The developer begins by selecting the project type ASP Web Service, executable, etc and the programming language to be used.
The developer also names the service and specifies the location of the web server that will run the Web Service.

The developer then creates the application using the many graphical resources and wizards provided with the development environment.

The build process not only performs compilation but also creates the auxiliary files needed to successfully implement a Web Service, SOAP file, WSDL file, etc.

.NET even provides a Web Services registration facility that allows developers to search for or publish Web Services through the globally available yellow pages of Web Services UDDI.
Creating Web Services in .NET

- We begin with creating a new project.
• Visual Studio .NET will automatically create all the necessary files along with a default web service .asmx file which contains at the top of it the reference to the file containing the XML web service and the language for the service.

```csharp
Imports System.Web.Services
<Service(Namespace:="http://tempuri.org/MSEC/Service1"> Public Class MSEC
```
Creating Web Services in .NET

- The default Web Service page includes the following commented code:

```csharp
' WEB SERVICE EXAMPLE
' The HelloWorld() example service returns the string Hello World.
' To build, uncomment the following lines then save and build the project.
' To test this web service, ensure that the .asmx file is the start page
' and press F5.
'
'<WebMethod()>
'Public Function HelloWorld() As String
' Return "Hello World"
'End Function
```

- Using the example given in the default code we can create a function that to be exposed as a Web Service.
Example

```csharp
<WebMethod()> Public Function MSEC_Courses(ByVal srtType As String) As String
    Dim courses As String
    If srtType = "Core" Then
        courses = "EC 511 E-commerce Basic IT"
    End If
    If srtType = "IT" Then
        courses = "CS650 Database Engineering or INFS 614 Database"
    End If
    If srtType = "Bus" Then
        courses = "MBA 623 Marketing Management, Four Elective Courses"
    End If
    If srtType = "Pub" Then
        courses = "Five Elective Courses"
    End If
    If srtType = "Health" Then
        courses = "HSCI 707 Health Care Management"
    End If
    Return courses
End Function
End Class
```
Creating Web Services in .NET

- When the project is built the Web service discovery file, WSDL document is created.
- http://palmms.com/ MSEC
Creating Web Services in .NET

- A client application can be build easily to consume the Web Service.
- The first step is to add a reference to the Web Service that we created and published before.
Creating Web Services in .NET

• This reference to the Web Service allows treating it like any other object within the project.
• In the file .aspx.vb which contains the code behind page the following should be declared:

  Dim MSECServiec As MSEC.MSEC = New MSEC.MSEC

• This line creates an object MSECServices of MSEC.MSEC type.
• When the client application is built the compiler creates a proxy class using the WSDL document.
Creating Web Services in .NET

• The client web application can be accessed using http://palmms.com/MSEC_client.

• When using SOAP for the request and response messages, the proxy creates a SOAP envelope that is passed to the Web Service.

• After the invoked method at the Web Service executes the return value is passed back to the proxy in a SOAP envelope over HTTP.

• The client application converts the SOAP XML response to the format required by the client web application.
**.Net vs. other technologies (J2EE)**

- Development and deployment of Web Services doesn’t require any specific underlying technology platform.
- The Web Services platform debate is often viewed as a Sun J 2EE vs. Microsoft .NET.
- Web services do not rely on either J 2EE or .NET, they depend mainly on SOAP, XML, and other components.
- .NET is runs on a single platform (windows) but it supports multiple languages.
- J 2EE is a platform independent environment and uses a single language (Java).
.Net vs. other technologies (J2EE)

- .NET will probably become the standard for client-side application development because Microsoft is so dominant there.
- On the other hand J2EE is already the main tool in building and deploying enterprise and large-scale Web applications.
- The deployment platform is also likely to be an important factor.
Conclusion

• Web Services provide computer-to-computer interaction.
• Web Services provide standards but need development environments.
• Microsoft .NET environment is built and structured to be mostly integrated and related to Web Services concepts.
• .NET is mostly the standard for client side.