2524 – Developing XML Services Using MS Visual Studio.NET 3 Day

The goal of this course is to provide students with the knowledge and skills that are required to develop Extensible Markup Language (XML) Web services-based solutions to solve common problems in the distributed application domain. The course focuses on using Microsoft Visual Studio .NET and Microsoft ASP.NET to enable students to build, deploy, locate, and consume Web services.

WHO SHOULD ATTEND

This course is intended for experienced software developers who have previously built component-based applications.

PREREQUISITES

Before attending this course, students must have:

- Familiarity with C# or Microsoft Visual Basic .NET.
- Programming in C++, Java, or Microsoft Visual Basic.
- An understanding of how to read and write XML documents.
- Experimented with simple C# applications.
- Developed distributed applications by using Visual Basic, Java, or C++.

AT COURSE COMPLETION

After completing this course, students will be able to:

- Explain how Web services solve problems encountered with traditional approaches to designing distributed applications.
- Describe the architecture of a Web services-based solution.
- Describe the underlying technologies of Web services and explain how to use the Microsoft .NET Framework to implement them.
- Implement a Web service consumer by using Visual Studio .NET.
- Implement a simple Web service by using Visual Studio .NET.
- Publish and deploy a Web service.
- Secure a Web service.
- Implement caching in a Web service.
- Evaluate the trade-offs and issues that are involved in designing a real-world Web service.

- Implement nonstandard Web services such as Hypertext Markup Language (HTML) screen scraping and aggregating Web services.

LESSON TOPICS

Module 1: The Need for XML Web Services

Take a closer look: Download Sample Module 1 (Portable Document Format, 707 KB).

This module provides students with an understanding of the problem space that Web services address. The module compares various approaches to implementing distributed applications. Because the Web services in this course are implemented by using Microsoft ASP.NET and the Microsoft .NET Framework, alternate options for implementing distributed applications by using the .NET Framework are discussed to better define what kinds of solutions Web services are appropriate for.

After completing this module, you will be able to explain how Web services emerged as a solution to the problems with traditional approaches to designing distributed applications. This includes:

- Describing the evolution of distributed applications.
- Identifying the problems with traditional distributed application architectures and technologies.
- Describing Web services and briefly explaining how they address the design problems in traditional distributed applications.
- Listing the alternate options for distributed application development.
- Identifying the kinds of scenarios where Web services are an appropriate solution.

Module 2: XML Web Service Architectures

Take a closer look: Download Sample Module 2 (Portable Document Format, 676 KB).

This module broadly describes the service-oriented architecture, which is a conceptual architecture. Then, the module explains the roles and how Web service architectures are a type of service-oriented architecture.

After completing this module, you will be able to describe the architecture of a Web services-based solution. This includes:

- Identifying how Web service architectures are a type of service-oriented architecture.
- Describing the elements of a Web service architecture and explaining their roles.

- Describing the Web service programming model.

Module 3: The Underlying Technologies of XML Web Services

After completing this module, you will be able to describe the underlying technologies of Web services and explain how to use the .NET Framework to implement Web services by using these technologies.

This includes:

- Describing the structures of an HTTP request and response.
- Issuing HTTP POST and GET requests and processing the responses by using the .NET Framework.
- Describing data types by using the XML Schema Definition language (XSD).
- Explaining how to control the way a .NET Framework object is serialized to XML.
- Describing the structures of a Simple Object Access Protocol (SOAP) request and response.
- Issuing a SOAP request and processing the response by using the .NET Framework.

Module 4: Consuming XML Web Services

After completing this module, you will be able to implement a Web service consumer by using Visual Studio .NET.

This includes:

- Explaining the structure of a Web Service Description Language (WSDL) document.
- Explaining the Web services discovery process.
- Locating service contracts by using Disco.exe.
- Generating Web service proxies by using Wsdl.exe.
- Implementing a Web service consumer by using Visual Studio .NET.
- Invoking a Web service synchronously and asynchronously by using a Web service proxy.

Module 5: Implementing a Simple XML Web Service

This module provides students with the skills that are required to implement and debug a Web service by using Visual Studio .NET.

This includes:

- Creating a Web service project.

- Implementing Web service methods, exposing them, and controlling their behavior.
- Managing state in an ASP.NET-based Web service.
- Debugging Web services.

Module 6: Publishing and Deploying XML Web Services

This module teaches students how to deploy and publish Web services and locate Web services by using the Microsoft Universal Description, Discovery, and Integration (UDDI) software development kit (SDK). A local development UDDI registry is used in the demonstrations for this module, but the mechanics of publishing and finding Web services is no different on the public UDDI registry nodes.

After completing this module, you will be able to publish and deploy a Web service. This includes:

- Explaining the role of UDDI in Web services.
- Publishing a Web service in a UDDI registry by using the UDDI SDK.
- Searching a UDDI registry to locate Web services by using the UDDI SDK.
- Explaining the various options for publishing a Web service on an intranet.
- Explaining some of the options for modifying the default configuration of a Web service.

Module 7: Securing XML Web Services

This module teaches students how to use the security services of the Microsoft Windows operating system, Microsoft Internet Information Services (IIS), and the.NET Framework and common language runtime to secure Web services.

After completing this module, you will be able to secure a Web service. This includes:

- Identifying the differences between authentication and authorization.
- Explaining how to use the security mechanisms that Microsoft Internet Information Services (IIS) and Windows provide for authentication.
- Using SOAP headers for authentication in a Web service.
- Using role-based security and code access security for authorization in a Web service.
- Encrypting the communication between a Web service consumer and a Web service.

Module 8: Designing XML Web Services

This module teaches students which design issues to consider when designing real-world Web services. The issues discussed are related to data type constraints, performance, reliability, versioning,

deployment in Internet Service Provider (ISP) and Application Service Provider (ASP) scenarios, and aggregating Web services. The module also discusses HTML screen scraping as a pseudo Web service.

After completing this module, you will be able to evaluate the trade-offs and issues that are involved in designing a real-world Web service. This includes:

- Identifying the restrictions that are imposed on data types by the various Web services protocols.
- Explaining how the use of Application and Session state can affect the performance and scaling of Web services.
- Explaining how to use output and data caching to improve Web service performance.
- Implementing caching in a Web service.
- Explaining how asynchronous Web service methods can improve performance.
- Explaining the need for instrumenting Web services.
- Identifying the components of a Web service that can be versioned.
- Explaining how to implement a virtual Web service by using screen scraping.
- Implementing a Web service that uses multiple Web services.
- Identifying the trade-offs in the techniques that are used for exposing aggregated Web services.

Module 9: Global XML Web Services Architecture

This module teaches students how to use the security services of the Microsoft Windows operating system, IIS, and the .NET Framework and common language runtime to secure Web services.

After completing this module, you will be able to:

- Describe limitations inherent to the specifications with which today's Web services are built.
- Describe the design principles and specifications of Global XML Web services Architecture (GXA).
- Describe Web service application scenarios made possible by Web Services Routing Protocol (WS-Routing) and Web Services Referral Protocol (WS-Referral).
- Explain how to use Web Services Security Language (WS-Security) and Web Services License Language (WS-License) to perform authentication and authorization for Web services.
- Design Web services that anticipate and can leverage the features that GXA will offer when released.