

Introduction to Service Oriented Architecture

| | |
|-----------|----------|
| Duration: | 1 day |
| Type: | beginner |

Description

This is a single day course which introduces delegates to the principles, standards and technologies behind Service-Oriented Architectures.

The course is delivered in two parts, the first providing a high-level introduction to SOA and the second a hands-on look at the technologies involved. During the latter delegates can choose to focus on either JEE or the .NET platform. Similarly the delivery can be tailored to stress SOAP'y or REST'ful Web Services as required.

Prerequisites

Ideally delegates will have good knowledge of Java or C# programming, a familiarity with XML / XML Schema and practical experience of building Web Applications using technologies such as ASP Web Controls, JSF and Rails.

List of Modules

The Evolution of SOA

- The challenge of application integration
- The evolution of middleware and messaging
- How DCOM and CORBA emerged in the 1990's
- Reasons why distributed OO technologies failed
- The emergence of XML as a universal data format
- How XML merged into middleware as Web Services
- The idea of a Service Oriented Architecture (SOA)
- Key differences between Web Services and SOA

Key Characteristics of SOA

- The importance of vendor / language neutrality
- The definition of a Service in a business context
- Creating interfaces to define business services
- Hosting services on a network as endpoints
- Publishing, finding and binding to endpoints
- The idea of a universal service bus

Advantages and Drawbacks of SOA

- Raising the level of abstraction in the enterprise
- Loose coupling and the flexibility to respond to change
- The idea of Orchestration and 'programming in the large'
- Business scenarios where SOA will be successful
- Scenarios where SOA should not be attempted
- The gap between the SOA vision and reality

Key Technologies Involved in SOA

- A brief introduction to XML and XML Schema
- Using Web Services Description Language (WSDL)
- Why WSDL is used for all types of Service (not just WS)
- An overview of the structure of a WSDL document
- Differences between WSDL versions 1.1 and 2.0
- Message exchange patterns supported by WSDL
- How WS-Policy adds extra functionality on top of WSDL
- Understanding the format of the WS-Policy language
- How UDDI enables service descriptions to be registered
- The role of UDDI in building SOA's through composition
- Using workflows and BPEL4WS to compose services

The Role of Web Services in SOA

- Why Web Services are the default choice in SOA
- The importance of WS-I Basic Profile in Web Services
- An introduction to the SOAP messaging protocol
- The format of SOAP requests, responses and faults
- Alternatives to transporting messages over HTTP
- Considerations when building SOAP Web Services
- Issues with the SOAP'y model of Web Services
- Using Representational State Transfer (REST)
- Advantages and issues with REST'ful Web Services

Advanced Technologies for Web Services in SOA

- Guaranteed delivery using WS-ReliableMessaging
- Adding transactions to services using WS-Transaction
- Security with XKMS, SAML, XML Signature and Encryption
- Building a PKI with WS-Trust, Privacy and Authorization

Developing Web Services in Java

- A short history of Java Web Service toolkits
- Installing the Metro WS toolkit with Tomcat
- Creating services using the JAX-WS annotations
- Creating WS proxies using the Metro Ant tasks
- Serializing objects using the JAXB annotations
- Making a JAX-WS based Web Service stateful
- Using the Dispatch API and SAAJ to alter messages
- Accessing REST'ful services using JAX-RS and Jersey

Developing Web Services in .NET

- Understanding Windows Communication Foundation (WCF)
- Writing WCF based Web Services in Visual Studio 2010
- Creating proxies to Web Services in Visual Studio
- Support available in WCF for the WS-* technologies
- Using WCF to access REST'ful Web Services