

Advance Dotnet (2 Month)



Course Content

Introduction

- WCF Using .Net 4.0
- Service Oriented Architecture
- Three Basic Layers
- First Principle
- Communication and Integration
- Integration Styles
- Legacy Applications
- Message Exchange Patterns
- Messaging Patterns
- Business Process Patterns

Service and Data Contracts

- Explaining Contracts
- Define WCF
- Contract Types
- WCF Service-Car Rental Agency
- Code First Service
- Scenario
- Client Implementation/li>
- Auto-Generate WSDL

- Service and Operation
- Attributes
- WSDL Update
- Check Price Per-Day
- Price Estimate Request
- Pass Estimate Request
- Associated XSD File
- DataContractSerializer
- Automatic Endpoints
- Default Bindings Mappings
- Standard Endpoints
- Configuring Multiple Bindings
- DataContract and DataMember Attributes
- Serialize Data Contracts
- Serializing Instance
- The Details
- Add 2-Classes
- KnownType Attribute
- Service Known Type
- Config File for Known Type
- Disclosing Known Types
- Versioning
- Round-Trip
- Contract Versioning
- XML Serialization

Bindings

- ABCs of WCF
- How The Bindings Work
- Addresses
- Service Behavior
- IServiceBehavior
- ServiceMetadataBehavior
- Operation Behaviors
- EndPoint Behaviors
- Contract Behaviors
- Bindings
- Context Enabled
- Configuring Bindings
- System.ServiceModel
- Base Addresses
- Default Configurations
- Automatic Endpoints
- Default Bindings Mappings
- Standard Endpoints
- Configuring Multiple Bindings

Clients

- InstanceContextMode
- Per Call
- Singleton
- Per Session
- Service Life
- Operation Behavior-Service Life
- Session ID
- Conclusions
- InstanceContextMode
- Throttling
- Service Behavior
- Visual Studio-Throttling
- Throttling-Results
- ConcurrencyMode.Multiple
- Final Example
- Best Practices
- Load Balancing

Understanding WCF Security

- Main Principles
- Transport and Message Security
- Configuring Security in WCF
- Security Mode
- Protection Level
- Client Credential Type
- Authentication and Negotiation
- Secure Sessions

WCF Security In Action

- Claims-Based Identity Model
- Claim Types
- Authentication in Action
- Message Security
- Username Authentication Demo
- Transport Security
- Mutual X509 Certificate
- Kerberos Authentication Over Message Security
- Claims Transformation

Hosting WCF Services

- Service Hosting
- Windows Activation Services
- Tracking Endpoints
- Enabling Event Viewer
- Routing Services
- Bridging
- Error Handling
- Cloud-Based Hosting

Create a WCF Service Application

- Creating a WCF
- The Requirements
- The Services
- Best Practices and Standards
- Design Guidelines

Introduction to ASP.NET MVC

- The role of the Model, View, and Controller
- Key benefits of ASP.NET MVC

Getting Started with ASP.NET MVC

- ASP.NET MVC project templates
- Understanding the structure of an ASP.NET MVC project
- Naming conventions
- Creating views
- Defining controllers
- Defining a data model

Creating a Complete ASP.NET MVC 4 Application

- Creating strongly-typed views
- Understanding URLs and action methods
- Using HTML helpers
- Handling form post-backs
- Data validation

Using the Razor View Engine

- Getting started with Razor
- Razor design goals
- Implementing a Razor view
- Razor syntax
- Accessing Model Data in Razor views

Industrial-Strength ASP.NET MVC Applications

- ASP.NET application architecture best practices
- Implementing a Repository and Entity Framework Data Model
- Using Dependency Injection
- Implementing a custom controller factory

View Techniques

- Defining and using custom HTML Helpers
- Defining a layout / MVC Master Page
- Using Styles
- Defining and using partial views
- Razor Helper Method syntax

Implementing Navigation in MVC web apps

- Defining view-model classes
- Implementing Data Filtering in a Controller
- Understanding the Routing mechanism
- Adding custom entries to a route table
- Defining defaults, parameters, and validation
- Generating URLs and Hyperlinks
- Custom Route constraints

MVC State Management

- Using hidden fields
- Session and Application State
- Custom model bindings

Additional Techniques

- View Scaffold Templates
- Controller Scaffold Templates
- Global Filters
- Model binding
- Asynchronous Controllers

Using AJAX and jQuery with ASP.NET MVC

- Overview of AJAX and ASP.NET MVC
- Unobtrusive AJAX
- Using AJAX Action Links
- Overview of jQuery
- jQuery Techniques
- Using jQuery UI

ASP.NET MVC & LINQ - working with Data

- Language features used in LINQ
- Creating simple LINQ queries
- Using LINQ queries in a Web application;
- Defining a data repository;
- Using LINQ to define a data access component

ASP.NET MVC 4 Techniques & Best Practices

- View scaffold templates
- Controller scaffold templates
- Dependency injection

ASP.NET Web API with MVC

- Overview of the ASP.NET Web API
- Building servers and clients
- Content negotiation
- Validation
- Query able Interfaces
- Dependency Injection

Introduction to LINQ

- Need for LINQ
- Various LINQ Providers
- LINQ to Objects, LINQ to SQL, and LINQ to XML Samples

Language Enhancements for LINQ

- New Language Features Required by LINQ
- Implicit Type Declaration
- Object Initialization
- Lambda Expressions
- Extension Methods
- Multiple Columns Using Anonymous Types

LINQ to Objects

- LINQ to Query Collections of In-Memory Objects
- LINQ with Arrays, Generic Lists, Generic Dictionaries, Strings, and Other Non-Generic Collections
- System.Linq.Enumerable Extension Methods, and Compare these Methods to LINQ Keywords

LINQ to SQL

- Map a Data Model to an Object Model
- LINQ to SQL to Query Data in a SQL Server Database
- Modify Data and Save Changes
- Use Stored Procedures to Modify Data

More LINQ to SQL

- Advanced Features of LINQ to SQL
- LINQ to SQL in a Windows Application
- Explore Using LINQ to SQL in a Web Application

LINQ to DataSets

- LINQ to DataSet to Query Data in a DataSet
- Use DataViews with the Results of a LINQ to DataSet Query
- Bind the Results of a LINQ to DataSet Query to Windows Forms Controls

LINQ to XML

- Create XML Content Using Classes in the System.Xml.Linq Namespace
- Search Content Using XPath Expressions with the LINQ-Based API
- Validate XML Content Using the LINQ-Based API
- Query XML Content Using LINQ to XML
- Transform XML Content Using LINQ to XML

LINQ to Entities

- Relationship Between LINQ to Entities and the Entity Framework
- Map a Data Model to an Entity Data Model
- Techniques for Using LINQ to Entities to Query Data in a Database
- Modify Data and Save Changes
- Use Stored Procedures to Modify Data