

Building MCP Servers for AI Applications with Azure AI Foundry & VS Code

Duration: 3 Days (24 Hours)

Course Overview

Model Context Protocol (MCP) is emerging as a standard approach for connecting AI applications with external tools, APIs, and enterprise systems. This hands-on course provides developers with practical knowledge of designing, developing, and integrating MCP Servers with AI applications.

Participants will learn how MCP differs from traditional REST APIs and OpenAPI-based integrations, how to build MCP Servers using ASP.NET Core, how to connect MCP Servers with Visual Studio Code AI extensions, and how to integrate MCP tools with Azure AI Foundry Agents.

The course focuses on real-world enterprise scenarios, including custom MCP tool development, security, and deployment patterns.

Target Audience

- .NET Developers
- Azure Developers
- AI Application Developers
- Cloud Engineers
- Solution Architects
- Technical Leads

Prerequisites

Participants should have:

- Basic knowledge of C# and ASP.NET Core
- Understanding of REST APIs
- Basic Azure concepts
- Familiarity with Generative AI concepts is recommended

Day 1: Introduction to MCP Architecture & Building Your First MCP Server

Module 1: Introduction to Model Context Protocol (MCP)

Topics Covered

- Evolution of AI application integration
- Introduction to Model Context Protocol
- Why MCP was introduced
- Challenges with traditional API integrations
- MCP ecosystem overview
- Benefits of MCP for AI applications
- MCP use cases in enterprise applications

Module 2: Understanding MCP Architecture

Topics Covered

- MCP architecture overview

- MCP Host
- MCP Client
- MCP Server
- MCP Tools
- MCP Resources
- MCP Prompts
- MCP communication flow

Architecture Discussion

Traditional Application Integration:

Application
|
REST API
|
Backend Service

MCP-based AI Integration:

AI Application
|
MCP Client
|
MCP Protocol
|
MCP Server
|
Tools / Data / APIs

Module 3: MCP Server vs REST API vs OpenAPI Tools

Topics Covered

- Understanding REST API-based integrations

- OpenAPI specifications and AI tool calling
- Limitations of traditional endpoints
- MCP advantages for AI applications
- Tool discovery and metadata
- Context-aware tool execution
- Selecting the right integration approach

Hands-on Exercise

Compare:

- REST API endpoint
- OpenAPI tool
- MCP Tool

Design the same business capability using all three approaches.

Module 4: Developing MCP Server using ASP.NET Core

Topics Covered

- MCP Server development architecture
- MCP SDK overview
- Creating MCP Server projects
- Registering MCP tools
- Tool definitions and metadata
- Input parameters
- Tool responses
- Error handling

Hands-on Lab

Build:

Customer Order MCP Server

Tools:

- Get Customer Details
- Get Order Information
- Create Order
- Track Order Status

Technology Stack:

- ASP.NET Core
- C#
- SQL Server

Day 2: MCP Development with VS Code & Azure AI Foundry Integration

Module 5: Using MCP Servers with Visual Studio Code

Topics Covered

- Understanding VS Code as MCP Host
- MCP client architecture
- MCP configuration
- Connecting local MCP Servers
- MCP tool discovery
- Using AI coding assistants with MCP

Hands-on Lab

Create and connect:

Developer Productivity MCP Server

Tools:

- Read Files
- Search Files
- Generate Information
- Execute Developer Utilities

Integrate with VS Code.

Module 6: Advanced MCP Server Development

Topics Covered

- Designing effective MCP tools
- Multiple tool implementation
- Tool input validation
- Response formatting
- Exception handling
- Logging and diagnostics
- MCP Server best practices

Hands-on Lab

Enhance Customer Order MCP Server:

Additional tools:

- Check Product Availability
- Calculate Discount
- Generate Order Summary

Module 7: Integrating MCP Server with Azure AI Foundry Agents

Topics Covered

- Azure AI Foundry Agent overview
- Agent tools architecture
- MCP tool integration
- Tool discovery process
- Agent reasoning and tool execution flow
- Connecting external capabilities with AI Agents

Architecture

User

|

Azure AI Foundry Agent

|

MCP Tool

|

ASP.NET Core MCP Server

|

Enterprise Application

Hands-on Lab

Integrate:

Azure AI Foundry Agent

↓

Custom MCP Server

↓

Business Operations

Scenario:

Customer Support AI Assistant

Module 8: Enterprise API Integration using MCP

Topics Covered

- Exposing existing APIs through MCP
- API wrapping patterns
- MCP Server as an abstraction layer
- Connecting databases and enterprise services
- Designing reusable MCP tools

Hands-on Lab

Convert existing APIs:

Existing:

- Customer API
- Product API
- Order API

into MCP tools:

- Search Customer
- Get Products
- Create Order
- Track Shipment

Day 3: Enterprise MCP Solutions, Security & Deployment

Module 9: Low-Code MCP Server Implementation

Topics Covered

- Introduction to low-code MCP approaches
- Azure Logic Apps integration
- Azure API Center overview
- API discovery and governance
- Dataverse integration scenarios
- Connecting business systems with AI Agents

Architecture

Azure AI Foundry Agent

|

MCP

|

Azure Logic Apps

|

Dataverse / Enterprise Data

Hands-on Lab

Build a low-code AI assistant:

Scenario:

Employee Assistant

Capabilities:

- Retrieve employee information
- Check leave balance
- Create requests using Dataverse

Module 10: Securing MCP Servers for Enterprise Applications

Topics Covered

- Security considerations for MCP
- Authentication and authorization
- Microsoft Entra ID integration
- OAuth 2.0 concepts
- Managed Identity
- Secure secrets management
- Protecting enterprise data access

Hands-on Lab

Secure MCP Server:

- Configure authentication
- Protect MCP tools
- Manage application secrets

Module 11: Deploying and Operating MCP Servers

Topics Covered

- MCP Server hosting options
- Deploying ASP.NET Core MCP Servers
- Azure App Service deployment
- Azure Container Apps deployment
- Configuration management
- Logging and monitoring

Hands-on Lab

Deploy:

ASP.NET Core MCP Server

|

Azure Cloud Hosting

|

Azure AI Foundry Agent

Module 12: Final Capstone Project

Build an Enterprise AI Assistant using MCP

Participants will create:

User

|

Azure AI Foundry Agent

|

MCP Server

|

ASP.NET Core MCP Tools

|

Enterprise Application

Implementation:

- Create MCP Server
- Develop custom tools
- Integrate with VS Code
- Connect with Azure AI Foundry Agent
- Secure MCP communication
- Deploy MCP solution

Learning Outcomes

After completing this course, participants will be able to:

- ☒ Understand MCP architecture and ecosystem
- ☒ Explain MCP vs REST APIs and OpenAPI tools
- ☒ Develop MCP Servers using ASP.NET Core
- ☒ Create reusable MCP tools for AI applications
- ☒ Connect MCP Servers with VS Code
- ☒ Integrate MCP tools with Azure AI Foundry Agents
- ☒ Build enterprise MCP-based solutions
- ☒ Implement security and deployment practices