Implementing Agents and Copilots using Semantic Kernel and Azure AI Foundry Agent Service (AZAGENTS)

ID AZAGENTS **Prix** sur demande **Durée** 4 jours

A qui s'adresse cette formation

Microsoft 365 & AI Pro-Code Developers

Pré-requis

- Basic Microsoft 365 Platform Development Skills
- Basic Azure Development Skills
- Python, C#, Typescript

Contenu

Module 1: Copilot, Agents & Azure AI Foundry Essentials

Introduction to Azure AI Foundry (Theory / Lab: 2 / 1.25)

- Overview Copilots and Agent Frameworks in the Microsoft Ecosystem
- Azure AI Foundry: Hubs, Projects and Resources
- Hub based projects vs AI Foundry Projects
- Deploy and use Large Language Models (LLM) in Azure AI Foundry
- Visual Studio Code AI Toolkit Extension
- Introduction to Azure AI Foundry SDK
- Deploy AI Apps using Azure Developer CLI

Agent Essentials (Theory / Lab: 1.25 / 1.25)

- Introduction Effective Prompt Engineering
- Introduction to GitHub Models
- Comparing and Prototyping Prompts using GitHub Models
- Retrieval Augmented Generation (RAG) & Agentic Retrieval in Azure AI Search
- Function Calling

Developing & Consuming Model Context Servers (Theory / Lab: 1 / 1.5)

- Model Context Protocol (MCP) Overview
- MCP Core Concepts
- Transports STDIO vs Http Streaming
- Develop MCP Servers
- Testing & Debugging using MCP Inspector
- Publishing MCP's to Azure

Module 2: Develop AI Agents using Azure OpenAI and Semantic Kernel

Semantic Kernel Basics & Concepts (Theory / Lab: 1 / 1)

- Understand the purpose of Semantic Kernel
- Semantic Kernel Components
- Chat History & AI Services Integration
- Chat Completion and Multi-modal capabilities

Optimizing Prompts (Theory / Lab: 0.5 / 0.75)

- Prompt Engineering with Semantic Kernel
- YAML Prompt Templates and Template Formats
- Handlebar Prompt Templates
- Liquid Prompt Templates
- Using Prompty Visual Studio Code Extension

Implement Plugins for Semantic Kernel (Theory / Lab: 1.5 / 1.5)

- Understand the purpose of Semantic Kernel plugins
- Learn how to use pre-made plugins
- Planners, Function Calling and Choice Behaviors
- Implement Native Functions using Prompts
- Integrate existing API's using OpenApi Plugins
- Using MCP Servers in Semantic Kernel
- Invocation-, Prompt Render & Invocation Filters

Kernel Memory & Vector Store Connectors (Theory / Lab: 1 / 1)

- Understand the purpose of Kernel Memory
- Semantic Kernel Memory: In-process & Out-of-the-box-Connectors
- Data Model And Embedding Generation
- Kernel Memory & Retrieval Augmented Generation (RAG)

Semantic Kernel Agent Framework (Theory / Lab: 2 / 1.5)

- Agents Overview
- Completing multi-step tasks with Agents
- Using Personas with Agents
- Implementing Multi Agent Solutions
- Sematic Kernel A2A Integration

- Using .NET Aspire in multi-agent scenarios

Semantic Kernel Process Framework (Theory / Lab: 1.5 / 1)

- Process Framework Overview
- Core Components and Patterns
- Runtimes: Orleans vs Dapr
- Implementing Human in the Loop

Module 3: Develop Agents using Azure AI Foundry Agent Service (Theory / Lab: 2 / 1.5)

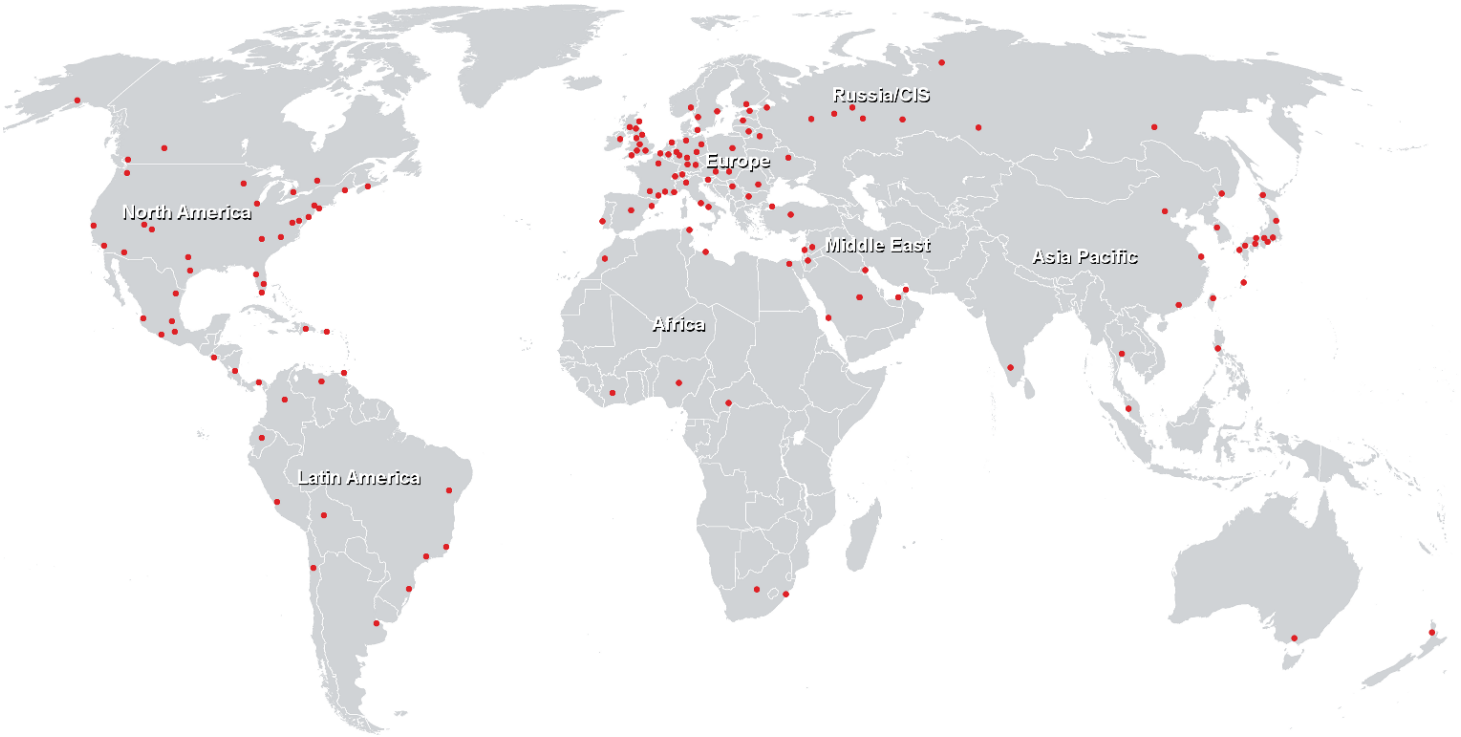
- Introduction to Azure AI Foundry Agent Service
- Using Action Tools: Code Interpreter, Function Calling, Azure Functions and OpenAPI Tools
- Using Knowledge Tools: File Search, Azure AI Search and Bing Grounding
- Connect MCP tools to Azure AI Agent Service
- Automating UI Tasks using Computer Use Agent
- Designing and implementing connected Agents
- Orchestrate Multi-Agent-Solutions using Semantic Kernel

Module 4: Securing, Monitoring and Evaluating Agents (Theory / Lab: 1 / 1)

- Agent Guardrails and Data Controls
- Ensuring App Behaviour using Evaluations
- Monitoring Risk and Alerts
- Azure AI Foundry Agent Governance and Observability
- Ensuring App Behaviour using Evaluations

Implementing Agents and Copilots using Semantic Kernel and Azure AI Foundry Agent Service (AZAGENTS)

Centres de formation dans le monde entier



Fast Lane Institute for Knowledge Transfer (Switzerland) AG

Husacherstrasse 3
CH-8304 Wallisellen
Tel. +41 44 832 50 80

info@flane.ch, https://www.flane.ch