# Getting Started with GitHub Copilot Extensions and Agents (45 min)

## Learning Objectives

By the end of this course, you will be able to:

* Explain what a GitHub Copilot Extension and agent are
* Understand the basics of creating a GitHub Copilot Extension and agent
* Utilize GitHub Copilot’s features to build and enhance GitHub Copilot Extensions and agents
* Implement a Copilot Extension and agent to assist with enable domain specific workflows

### 1. Introduction (2 min)

* Overview of GitHub Copilot Extensions and agents and their role in the software development life cycle
* How GitHub Copilot supports AI pair programming
* Examples of GitHub Copilot Extensions and agents in real-world development workflows

### 2. What is a GitHub Copilot Extension and Agent? (10 min)

* Explain how **Copilot Extensions differ from standard Copilot suggestions**
* **Copilot Extensions** provide **context-aware, task-specific guidance** rather than just inline code suggestions
* Copilot Extensions are integrations that expand the functionality of Copilot Chat, allowing developers to bring external tools, services, and custom behaviors into the Chat experience. You can use Copilot Extensions to extend the capabilities of Copilot Chat in a variety of ways, including:
* **Querying documentation:** A Copilot Extension can allow Copilot Chat to query a third-party documentation service to find information about a specific topic.
* **AI-assisted coding:** A Copilot Extension can use a third-party AI model to provide code suggestions.
* **Data retrieval:** A Copilot Extension can allow Copilot Chat to query a third-party data service to retrieve information about a specific topic.
* **Action execution:** A Copilot Extension can allow Copilot Chat to execute a specific action, such as posting to a message board or updating a tracking item in an external system.
* Importantly, **GitHub Copilot Extensions are powered by GitHub Copilot Agents**, which serve as the underlying extensibility platform. Copilot agents are custom tools embedded in Copilot Extensions. They integrate with Copilot Chat to provide additional functionalities tailored to specific needs.Copilot Agents provide:
  + The ability to integrate external services and APIs into Copilot Chat
  + Support for complex integrations requiring full control over request processing
  + Flexibility to implement custom logic
  + Integration capabilities with other LLMs and the Copilot API
  + Management of conversation context
  + Complete control over user interaction handling
  + While Agents require more engineering effort and ongoing maintenance, they provide the robust foundation needed to support Copilot Extensions and enable sophisticated workflows with maximum flexibility.

### 3. How do GitHub Copilot Extensions and Agents work? (10 min)

* GitHub provides a comprehensive toolkit for extension builders, with code samples, a CLI debugging tool, quickstart SDKs, and a user feedback repository. For more information, see the [copilot-extensions](https://github.com/orgs/copilot-extensions/) organization on GitHub.
* [Configuring your Copilot agent to communicate with the Copilot platform](https://docs.github.com/en/copilot/building-copilot-extensions/building-a-copilot-agent-for-your-copilot-extension/configuring-your-copilot-agent-to-communicate-with-the-copilot-platform)
* Learn how to interact with the Copilot platform by sending and receiving server-sent events with your Copilot agent.
* [Configuring your Copilot agent to communicate with GitHub](https://docs.github.com/en/copilot/building-copilot-extensions/building-a-copilot-agent-for-your-copilot-extension/configuring-your-copilot-agent-to-communicate-with-github)
* Learn how to verify payloads and get resources from GitHub with your Copilot agent.
* [Context passing for your agent](https://docs.github.com/en/copilot/building-copilot-extensions/building-a-copilot-agent-for-your-copilot-extension/context-passing-for-your-agent)
* Learn how to use context passing with your Copilot agent.
* [Using Copilot's LLM for your agent](https://docs.github.com/en/copilot/building-copilot-extensions/building-a-copilot-agent-for-your-copilot-extension/using-copilots-llm-for-your-agent)
* Learn how to use Copilot's LLM for your agent.

### 4. When to use GitHub Copilot Extensions and Agents (10 min)

GitHub Extensions use cases when you want to expand the functionality of GitHub Copilot Chat in the following way.

* Implement custom logic
* Integrate with other LLMs and/or the Copilot API
* Manage conversation context and handle all aspects of the user interaction.

### 5. Exercise: Your first extension for GitHub Copilot (25 min)

**Course reference example:** <https://github.com/burkeholland/prompt-crafting-github-copilot>

* Guide learners through setting up a Copilot extension
* Implement a basic Copilot extension with defined actions and directives
* Showcase how to test the agent’s responses and refine its functionality

In this hands-on GitHub Skills exercise, participants will build a GitHub Copilot extension for a practical developer workflow as well as content from Subject Matter Experts (SME) in a specific domain. Learners will:

1. Start a pre-configured development environment with required tools.
2. Configure a web service in python or javascript to support your Copilot extension.
3. Create a GitHub App and configure it to use with Copilot.
4. Add custom abilities to make your extension unique.
5. Try your extensions on real github.com and in VS Code.
6. Optionally deploy your service to Azure and publish it to the Copilot extensions marketplace.

#### **Exercise Topic**

**Domain Knowledge** – Build an extension that incorporates your industry’s nomenclature, lingo, and domain knowledge to create product specific features.

In the exercise we will build a GitHub Copilot that specializes in the education domain.  
Each role has both specialized knowledge and overlapping areas with other positions

communication and collaboration are essential across all roles

Most positions require both technical expertise and interpersonal skills

documentation and compliance appear as common threads. Technology integration is relevant to nearly every role. The GitHub Copilot extension goal is to offer subject matter expert content in each of the domains from different role perspectives.

**Education Domain Knowledge: Role-Based Perspective**

* School Administration
  + Principal
  + Assistant Principal
* Instructional Staff
  + Classroom Teacher
  + Special Education Teacher
  + Student Support
  + Educational Specialist
* **Support Staff**
  + School Counselor
  + School Psychologist
  + Instructional Coach
* **Specialized Roles**
  + Technology Coordinator
  + Media Specialist/Librarian
  + Parent Coordinator
* **Administrative Support**
  + Registrar
  + Administrative Assistant

**6. Knowledge Check**

* Multiple-choice and scenario-based questions to reinforce learning objectives

1. What are the use cases for GitHub Copilot Extensions?
   1. AI prompt generation for coding
   2. Querying documentation and data retrieval
   3. Inline suggestions
   4. Knowledgebases
2. What are the components involved with a GitHub Copilot Extension?
   1. GitHub Extensions, Actions, and Apps
   2. GitHub Apps, CodeQL, Issues
   3. GitHub Apps, Agents, Extensions
   4. GitHub Issue Templates, Actions, Agents
3. There are only two levels for GitHub Copilot Extension, public and private
   1. True
   2. False
4. What client/IDE support GitHub Copilot Extensions
   1. Copilot CLI
   2. Xcode
   3. Visual Studio Code
   4. GitHub Codespaces

**7. Summary**

* Recap of key points
* Sign up for 50-day challenge
* Further learning/resources
* Potential Link to Guinness attestation/badging