

Angular GitHub Co-pilot Development

Prerequisites: Knowledge of TypeScript and JavaScript Programming

Day 1: Introduction to Angular and GitHub Copilot

Introduction to Angular

- Overview of Angular: Architecture, components, modules, and services
- Setting up the Angular development environment
- Creating a new Angular project using Angular CLI

GitHub Copilot Basics

- Introduction to GitHub Copilot: Features and setup
- Integrating GitHub Copilot with your IDE (e.g., Visual Studio Code)
- Basic usage: Generating code snippets, functions, and components

Angular Component Development with Copilot

- Developing Angular components using GitHub Copilot
- Implementing data binding and event handling with AI assistance
- Using Copilot to create reusable components and templates

Lab 1 - Building a Simple Angular Application with Copilot

- Hands-on lab: Create a basic Angular application
- Practice generating code snippets and components using GitHub Copilot
- Review and refine the generated code for best practices

Day 2: Advanced Angular Features and Copilot Integration

Angular Services and Dependency Injection

- Creating and using Angular services
- Implementing dependency injection in Angular
- Using GitHub Copilot to generate service classes and methods

Angular Routing and Navigation

- Setting up routing and navigation in Angular applications
- Implementing lazy loading and nested routes

- Leveraging Copilot to automate routing module creation

Reactive Forms and Validation

- Introduction to reactive forms in Angular
- Implementing form validation and error handling
- Using GitHub Copilot to generate form controls and validation logic

Lab 2 - Developing Advanced Angular Features with Copilot

- Hands-on lab: Implement Angular services, routing, and forms
- Use GitHub Copilot to streamline development
- Test and debug the application with AI-generated suggestions

Day 3: API Integration and State Management

Consuming RESTful APIs with Angular

- Making HTTP requests and handling responses in Angular
- Integrating third-party APIs into Angular applications
- Utilizing GitHub Copilot to generate API service methods

State Management in Angular

- Introduction to state management concepts in Angular
- Using services and RxJS for simple state management
- Implementing NgRx for complex state management with Copilot assistance

Error Handling and Best Practices

- Implementing error handling and logging in Angular applications
- Ensuring best practices in code structure and modularity
- Using GitHub Copilot to enforce coding standards and error handling

Lab 3 - API Integration and State Management

- Hands-on lab: Integrate a RESTful API into an Angular application
- Implement state management using services or NgRx
- Use GitHub Copilot to assist in coding and debugging

Day 4: Testing, Deployment, and Final Project

Testing Angular Applications

- Introduction to testing in Angular with Jasmine and Karma
- Writing unit tests for components, services, and pipes
- Using GitHub Copilot to generate test cases and assertions

Deployment of Angular Applications

- Preparing Angular applications for production
- Deploying Angular apps to hosting platforms
- Automating deployment processes with GitHub Actions and Copilot