

# Fusion development in Power Platform

This course will teach you how a fusion development team functions, introduce low-code concepts for professional developers, and explain how a professional developer can enable their citizen developer counterparts to build amazing Power Apps applications using pro-code tools and frameworks they're already familiar and comfortable with, like JavaScript, ASP.NET Core web APIs, and Azure API Management.

**Duration:** 5 Days (40 hours)

**Target Audience:** App Makers, Citizen Developers, Power Platform Developers

**Prerequisites:**

- You should be familiar with basics of power platform.
- You Should be familiar with fundamental of Azure

## Module 1: Transform business software authoring with fusion development teams

Learn how to empower your development team to build better apps, faster

- What is tech intensity?
- The current field inventory management system
- Fusion team development process

## Module 2: Introduction to fusion development

Learn how to bring your pro devs and business together to build apps fast

- What is the fusion development approach?
- Introduction to the sample scenario

## Module 3: Understanding Low Code as a Traditional Developer

Power Apps can be a powerful tool for citizen developers and traditional developers alike. Upon finishing this module, a traditional developer will have learned how Power Apps work, what the formula language is, and how to create an app using Power Apps

- Introduction
- What is low code?
- Understand Power Fx
- Exercise - Create Your First Power Apps App as a Traditional Developer

## Module 4: Building a low-code prototype

The module focus on the design, implementation, and rollout of the mobile apps built with Power Apps.

- Item 1: Field inventory management
- Item 2: Field knowledge base
- Item 3: Field scheduling and notes
- Combining the screens into a single app
- Adding a Home screen to the app

## Module 5: Using Microsoft Dataverse as the data source

- What is Dataverse?
- Defining entities and relationships in Dataverse
- Adding views and business rules
- Defining business activities
- Adding graphical display elements
- Decision to use Dataverse

### **Module 6: Build a Power Apps component**

Build a custom Power Apps component, create a code component solution package, and then test and debug a code component.

- Introduction to creating a code component
- Create a code component solution package
- Test and debug code components

### **Module 7: Creating and publishing a Web API in Azure**

In this module we will learn how app should be sourced from existing systems through a Web API

- Defining the Web API operations: Field inventory management
- Defining the Web API operations: Field Knowledgebase
- Defining the Web API operations: Field scheduling
- Building the Web API: Field inventory management
- Deploying the Web API to Azure: Field inventory management
- Building and deploying the Web API: Field Knowledgebase
- Building and Deploying the Web API: Field Scheduling

### **Module 8: Using the Web API in the app**

We will learn how to combine the app with the Web API

- Understanding the IT operations management requirements for the Web API
- Creating an API Management service
- Publishing the Web API through API Management
- Connecting to API Management from the app
- Updating the app to use the connector: Field inventory management
- Updating the App to use the Connector: Field Scheduling and Notes
- Creating the Azure Cognitive Search service for the Field Knowledgebase
- Creating the custom connector for the Azure Cognitive Search service
- Updating the app to use Azure Cognitive Search: Field Knowledgebase

### **Module 9: Adding functionality to the app**

Adding some extra user interface functionality to make it easier to use

- Adding a photograph to an appointment
- Displaying images of parts
- Tracking appointment history for a customer
- Ordering parts

### **Module 10: Protecting and deploying the app**

Learn to ensure that the solution is safe to deploy, and that they have a mechanism for maintaining it as requirements change in the future.

- Protecting the app and resources
- Personalizing the app
- Deploying the app
- Maintaining the app

### **Module 11: Integrate OpenAPI-enabled Web APIs with Azure API Management through Visual Studio**

Learn how to add an OpenAPI-enabled Web API application to Azure API Management through Visual Studio.

- Introduction
- Use Swashbuckle to create an OpenAPI document
- Exercise - Use Swashbuckle to create an Open API document
- Publish an OpenAPI enabled web API to Azure App Service
- Exercise - Publish an OpenAPI enabled web API to Azure App Service
- Exercise - Integrate openAPI enabled web API with Azure API Management

### **Module 12: Discover and use Web APIs with Power Apps**

In this module, you'll learn how you can build custom connectors to have your Power Apps applications interact with your web APIs.

- Introduction
- What are custom connectors
- Exercise - Create a custom connector from Azure API Management
- What is OpenAPI and why you should use it
- Exercise - Create a custom connector from an OpenAPI document
- Test the custom connector
- Exercise - Test the custom connector
- Exercise - Use the custom connector in Power Apps canvas app