



KD-05106 (Workshop)

Low-Code for Developer in a Day

Power Platform



About this course.

This course covers how to be a successful developer for Microsoft Power Platform.

Length.

1 Day.

Prerequisites.

Before attending this course, students must have Software Development background, or equivalent work experience with Microsoft Power Platform. In order to complete the lab, dedicated tenant and user permissions for developers will be required.

Attendees learn how to:

- Power Platform Background.
- Extending the User Experience.
- Custom Connector.
- Power Platform API's in Azure.
- Application Lifecycle Management.
- Low Code or Code first?

Exam.

None.

Course outline.

Module 1: Power Platform Overview.

This module sets the foundation for the rest of the course and provides an overview of the Power Platform.

- Power Platform is built on Azure and can be further extended using Azure.
- Microsoft is not trying to convert traditional code developers to low-code only developers but to use traditional skills to close platform gaps and extend with innovative ideas.
- Power Platform development is a team sport – low code, IT Pros and pro devs should all work together. This can be called a fusion team.
- Provide an overview of the extensibility options.
- Introduce Dataverse and its capabilities above just a simple data repository.
- Introduce solutions and their role in ALM.
- Introduce the Power Platform tooling.

Lab: Getting started.

Module 2: Extending the user experience.

This module focuses on building Power Apps code components to extend the user experience.

- Types of components, e.g. canvas components vs code components.
- Scaffold code component using Power Platform CLI.
- Key parts of a code component.





Lab: Build a code component (PCF).

Module 3: Custom Connectors.

This module focuses on building custom connectors for existing APIs. You're introducing them to the process of making an API available to makers as a connector.

- A connector is a wrapper around an API that allows the underlying service to talk to Power Automate and Power Apps.
- Can build for existing APIs or new ones.
- Developers can add Custom code to transform request and response payloads beyond the scope of existing policy templates.

Lab: CC for existing API.

Module 4: Power Platform APIs in Azure.

This module continues the discussion on custom connectors but explores building them in Azure.

- Use cases for custom APIs – can help makers be more productive and simplify their app building.
- Options for implementing – Azure has some advantages, but any service that can host a RESTful API can be used.
- Adding OpenAPI attributes automatically generates OpenAPI (Swagger) definitions and makes API ready for import.
- Azure API Management has benefits for use including licensing with Dataverse for Teams use of custom connectors.
- How to secure with Azure AD and provide on behalf of Dataverse access in the context of the app and flow connection user.

Lab: Azure Functions.

Module 5: ALM.

This module focuses on how the Power Platform helps facilitate good Application Lifecycle Management (ALM) using solutions and build tools. Solution basics and an overview of the build tools are covered.

- Solutions are used to distribute components across environments through export and import.

- Types of solutions and when they are used.
- Solutions can depend on other solutions – they saw this with the custom connector lab.
- Source control should be your source of truth for storing and collaborating on your components – using Power Platform Build tools the lab will demonstrate this.
- Continuous integration and continuous delivery (CI/CD) such as Azure DevOps or GitHub allow you to automate your build, test, and deployment pipeline.

Lab: ALM.

Module 6: Low Code or Code First.

This module highlights how low code and code first can work together. Take advantage of the examples in your organization to spark a discussion.

- Examples of how fusion teams are working together.
- How to identify opportunities to get started.
- More resources for next steps.

