DDPAF

Deep Dive into PowerApps and Flow

Building Custom Solutions on the Business Application Platform

Course Manual Version 1.1

About This Training Course

Course Title:	Deep Dive into PowerApps and Flow
Course Code:	DDPAF
Audience:	Technical Specialists
Format:	In-person and Remote
Length	2 Days

Course Description

Deep Dive into Power Apps and Flow is an intensive 2-day online training class for technical specialists, web developers and IT professionals working with SharePoint Online, Power BI, Office 365, and Dynamics 365. This course teaches the essential concepts and visual designer skills required to build advanced business solutions using PowerApps and Microsoft Flow. Students will learn advanced techniques such as writing complex expressions for PowerApps and Flows and accessing REST-based data sources using custom connectors.

The course goes beyond the fundamentals of PowerApps and Flow teaching students how to design and build custom solutions for real-world scenarios in SharePoint Online, Power BI, the Common Data Service for Apps and Dynamics 356. The course examines issues with application lifecycle management (ALM) and explains best practices for building and testing custom solutions built with PowerApps and Flow in an isolated development environment and for packaging custom solutions for deployment to a production environment after quality assurance testing has been completed.

Student Prerequisites

All students will require a Windows PC for lab exercises running Windows 10 or Windows 8.1. Students should already be familiar with Microsoft Excel, Office 365 and SharePoint Online. Due to the accelerated nature of this training class, it is also recommended that students get some hands-on experience with PowerApps and Flow before the start of class by going through some of Microsoft’s introductory tutorials.

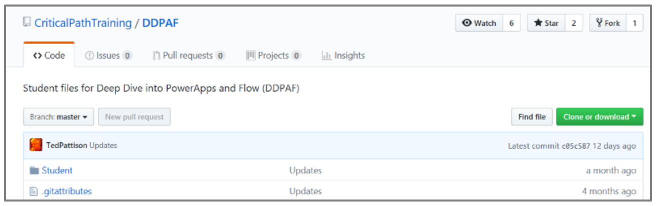
About the Hands-on Labs

Deep Dive with PowerApps and Flow (DDPAF) includes a rich set of hands-on lab exercises designed to reinforce the key concepts and techniques that you learned in the lectures. The lab exercises for this training course were designed to run on a workstation or laptop running the Windows operating system. We recommend that you use Windows 10 but you can also Windows 8.1.

DISCLAIMER: While it is possible to complete many of the lab exercises for this course using a Mac instead of a Windows PC, the writeups for the lab exercises in this training course assume that you are using a Windows PC which means the screenshots will be very different from using a Mac. While you are free to try the lab exercises on a Mac, please keep in mind that we have written and tested these labs using a Windows PC and not a Mac. You have been forewarned. Now, go forward and have some fun with this class.

GitHub Repository for this course

Student files for this course are maintained in a GitHub fil repository at **https://github.com/CriticalPathTraining/DDPAF**.



Student files for Deep Dive into PowerApps and Flow (DDPAF)

TedPattison Updates — Latest commit c05c587 12 days ago

Student	Updates	a month ago
.gitattributes	Updates	4 months ago

Student Workstation Prerequisites

All lab exercises should be completed on a student workstation that has the following software.

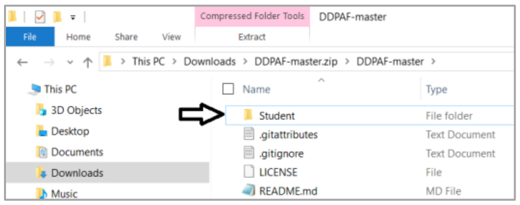
1. Windows 10 or Windows 8.1.
2. One or more Internet browsers (Chrome, Edge, IE, etc.)
3. Microsoft Excel 2016 or Microsoft Excel 2013 - *If you don't already have it, you can optionally install it during lab 1.*

Copying the Student Lab Files to Your Windows PC

The most recent version of the electronic student files for this course are kept in the **Student** folder of the GitHub repository for this course. You can download the zip archive for this repository from the following URL.

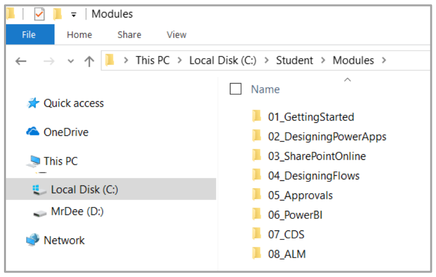
https://github.com/CriticalPathTraining/DDPAF/archive/master.zip

It is recommended that you that you download the master zip archive and make a local copy of the **Student** folder so that you have a local copy of the files you will need on your computer workstation when going through these labs exercise. Once you download the master zip archive, open it and copy the **Student** folder to a new local folder.



Name	Type
Student	File folder
.gitattributes	Text Document
.gitignore	Text Document
LICENSE	File
README.md	MD File

Copy the **Student** folder from the master zip archive to a new local folder at **C:\Student**.



01_GettingStarted
02_DesigningPowerApps
03_SharePointOnline
04_DesigningFlows
05_Approvals
06_PowerBI
07_CDS
08_ALM

Module 01: Getting Started with PowerApps Studio

Module Description

This module introduces students to the PowerApps Service and the fundamentals of creating and testing web and mobile applications using PowerApps Studio. Students will learn to create and configure an isolated development environment for building custom solutions with PowerApps and Microsoft Flow. The module teaches students how to create a modern user interface experience in PowerApps Studio by working with screens and controls and by writing dynamic PowerApps formulas for control properties. The module introduces student to connectors and explains how connectors are used in PowerApps to connect an app to an external data source. The module teaches students how data binding works in PowerApps and demonstrates how to use data binding with gallery controls, form controls and data cards.

Module Agenda

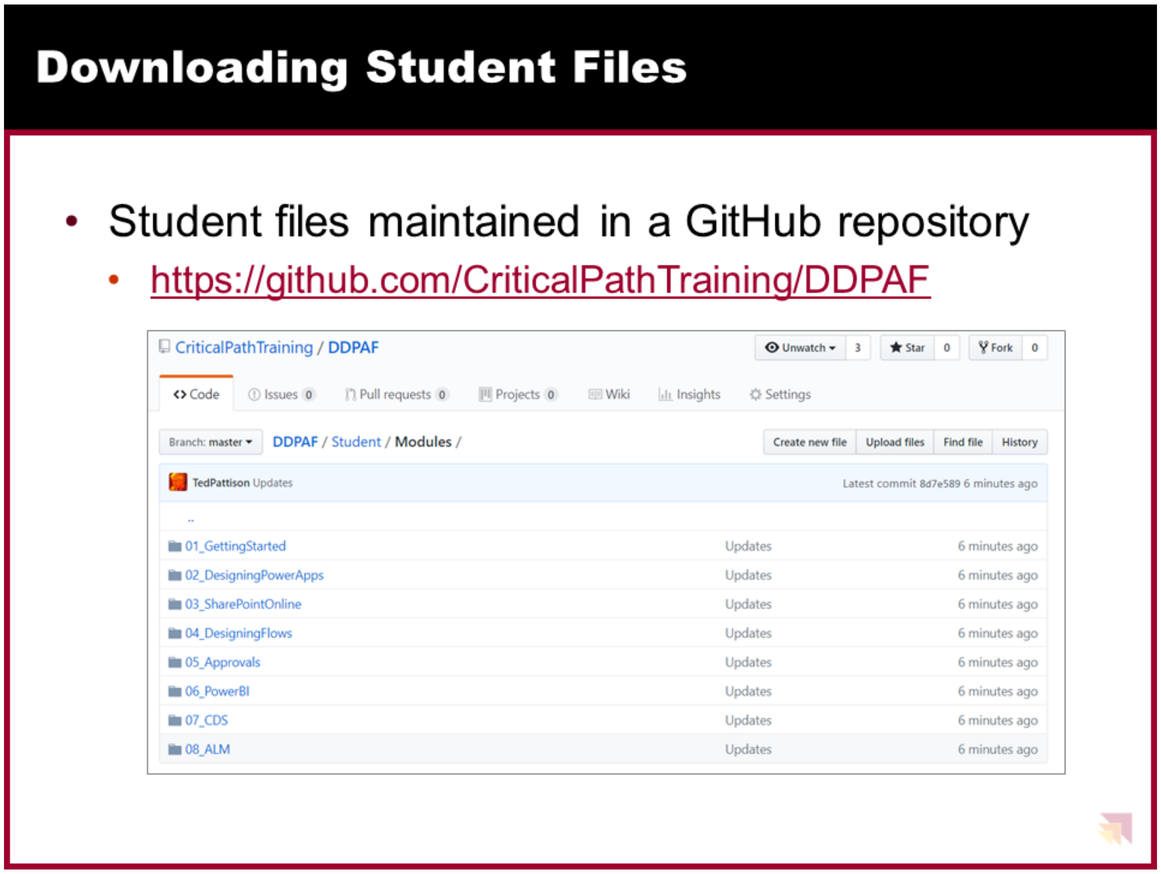
- Getting Started with PowerApps
- Creating and Testing Apps with PowerApps Studio
- Working with Screens and Controls
- Understanding Connectors and Data Binding
- Customizing Forms and Data Cards

Topics Covered

Downloading Student Files
Student Background Questionaire
What is the Business Application Platform?
What is PowerApps?
Creating an Office 365 E5 Trial Tenant
Office 365 Admin Center
Configuring a PowerApps Plan 2 License
PowerApps Admin Center & Environments
Creating New Apps
Getting Started with PowerApps Studio
Running an App from PowerApps Studio
Saving an App to the Cloud
Building Apps using Screens and Controls
Adding Controls to a Screen
Configuring Control Properties
PowerApps Formula Language
Navigating Between Screens
Understanding Connectors & Connections
Standard Connectors vs Custom Connectors
Data Binding with Galleries and Forms
Working with the Data Pane
Understanding Forms and Data Cards
Changing a Field’s Data Card Type
Customizing a Data Card
Populating a Dropdown Combobox

Instructor Demos

- Configuring PowerApps Plan 2 Licenses
- Creating an App with the Budget Tracker App Template
- Creating an App using the Start from Blank Template
- Navigating Between Screens
- Creating an App using the Start from Data Template
- Customizing Forms and Data Cards



Downloading Student Files

- Student files maintained in a GitHub repository
  - https://github.com/CriticalPathTraining/DDPAF

CriticalPathTraining / DDPAF

Branch: master DDPAF / Student / Modules /

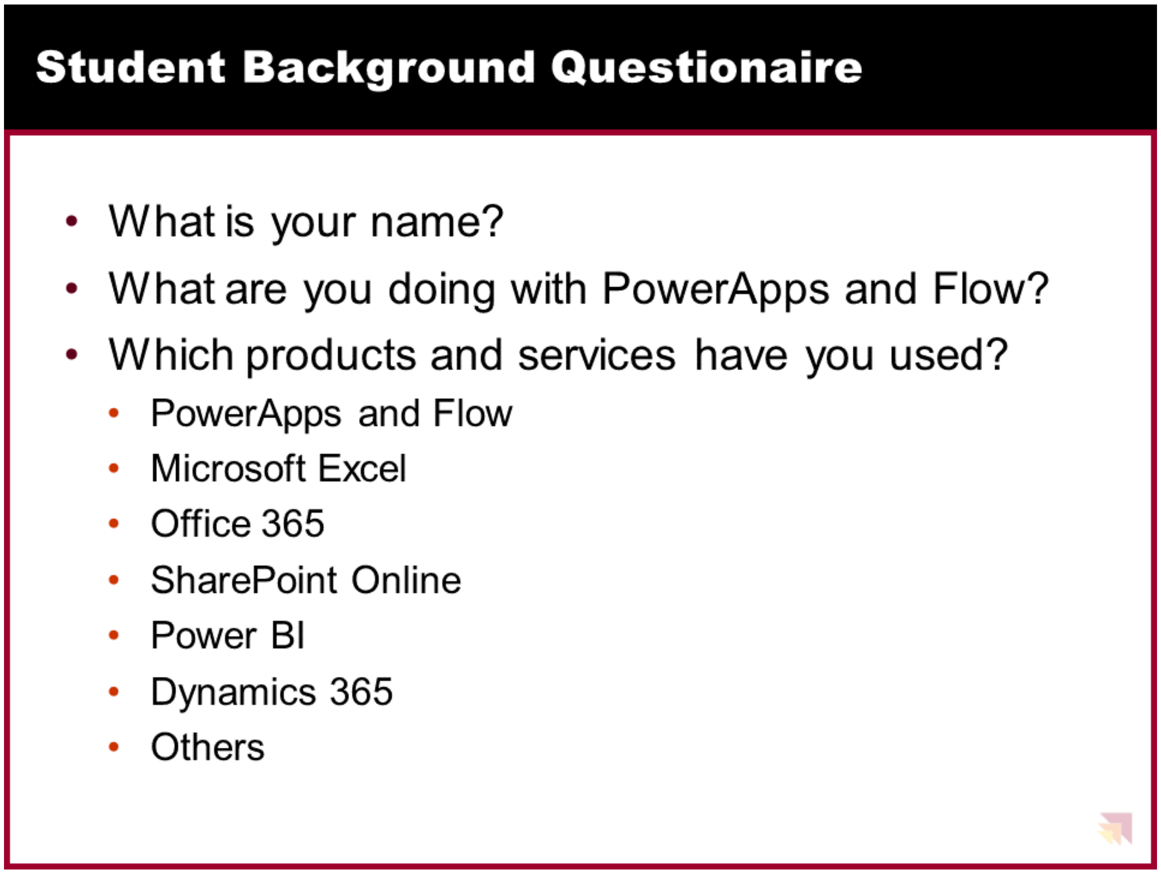
TedPattison Updates Latest commit 8d7e589 6 minutes ago

..		
01_GettingStarted	Updates	6 minutes ago
02_DesigningPowerApps	Updates	6 minutes ago
03_SharePointOnline	Updates	6 minutes ago
04_DesigningFlows	Updates	6 minutes ago
05_Approvals	Updates	6 minutes ago
06_PowerBI	Updates	6 minutes ago
07_CDS	Updates	6 minutes ago
08_ALM	Updates	6 minutes ago

At Critical Path Training, we try our best to stay on top of all of Microsoft's updates as they are applied to their cloud-based services such as PowerApps and Microsoft Flow. As you can imagine, this cloud cadence of monthly updates can pose a challenge to many training companies and content vendors who are faced with the problem of having to update their courseware on a monthly basis.

Critical Path Training has responded to challenge of dealing with monthly updates with a new student experience that we refer to as **Live Labs**. All the electronic student files for running demos and completing the lab exercises are published in a public GitHub repository which can be accessed with a browser at https://github.com/CriticalPathTraining/DDPAF/archive/master.zip.

It is recommended that you that you download the master zip archive and make a local copy of the **Student** folder so that you have a local copy of the files you will need on your computer workstation when going through these labs exercise. Once you download the master zip archive, open it and copy the **Student** folder to a new local folder. Note that each module of the course has its own folder in the **Student\Modules** folder.

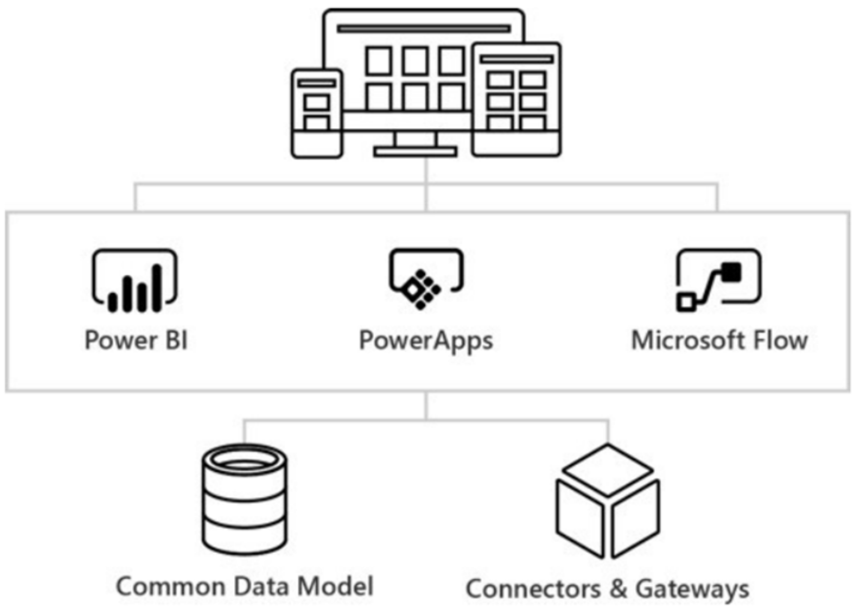


Student Background Questionaire

- What is your name?
- What are you doing with PowerApps and Flow?
- Which products and services have you used?
  - PowerApps and Flow
  - Microsoft Excel
  - Office 365
  - SharePoint Online
  - Power BI
  - Dynamics 365
  - Others

When you are called upon by the instructor, please provide a brief background by answering the following questions.

What is the Business Application Platform?



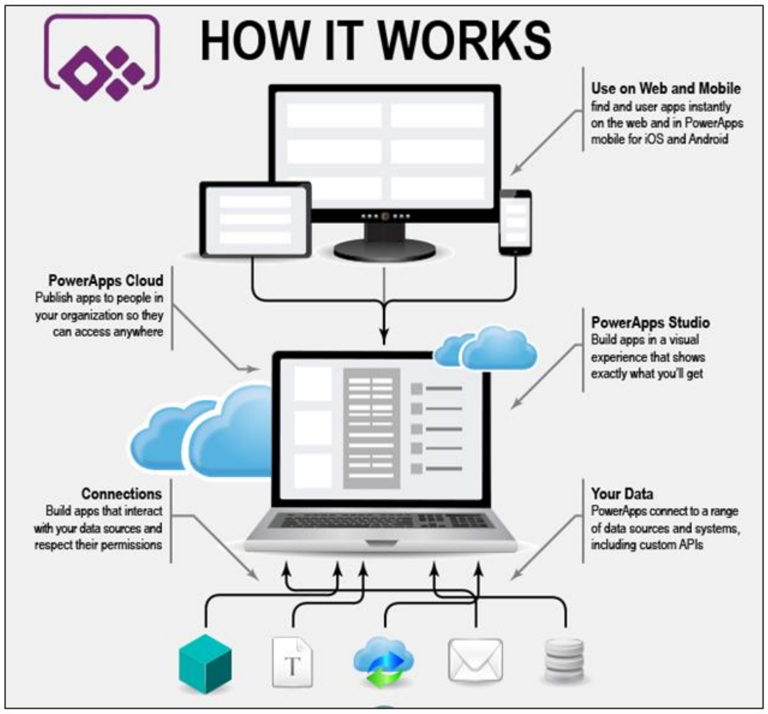
PowerApps and Microsoft Flow are to important pieces of the Microsoft initiative known as the Business Application Platform. Together with Power BI, PowerApps and Flow make up the "Power Trio" of the Business Application Platform

The Business Application Platform has been specifically designed for business users and technical specialists who do not have the background of a software developer. The ultimate goal of the Business Application Platform is to allow business users and technical specialists within an organization to build custom applications and workflow solutions in a fraction of the time compared to the traditional lifecycle of a software development project. The Business Application Platform has also been designed with a mobile-first philosophy which makes it relatively easy and very effective to build applications that target mobile devices such as iPhones and Android phones.

The Business Application Platform consist of these services

- PowerApps is a service for building and consuming web applications and mobile apps that connect to data
- Microsoft Flow is a service for automating workflow across the growing number of apps and SaaS services
- Power BI is a business analytics service which provides self-service BI and great interactive visualizations
- The Common Data Service for Apps provides the native storage of business data in the PowerApps environment
- Connectors and Gateways provide the Business Application Platform with access to external data

What is PowerApps?

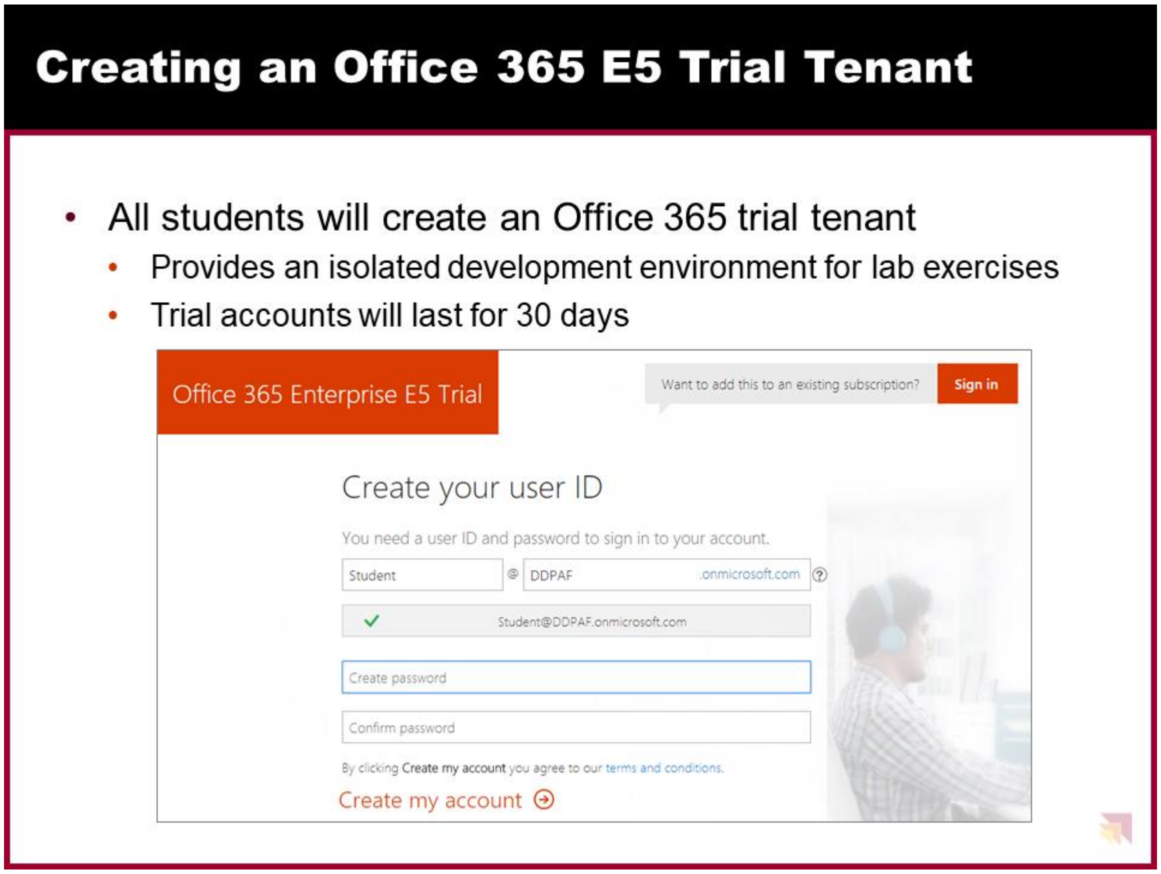

HOW IT WORKS
Use on Web and Mobile
find and user apps instantly on the web and in PowerApps mobile for iOS and Android
PowerApps Cloud
Publish apps to people in your organization so they can access anywhere
PowerApps Studio
Build apps in a visual experience that shows exactly what you'll get
Connections
Build apps that interact with your data sources and respect their permissions
Your Data
PowerApps connect to a range of data sources and systems, including custom APIs

PowerApps is a service for building and consuming business apps. As an app builder, you can use any modern browser you like and you can work on a Windows PC or a Mac. It's you're choice. Once you build an app using PowerApps, it can be consumed by you and by other users in your organization through any modern browser as well. PowerApps can also be consumed on tablets and mobile devices by installing the PowerApps native app published by Microsoft.

One of the primary strengths of PowerApps is that is makes it quick and easy to build an app that reads and writes business data which is stored in terms of database records or SharePoint list items. PowerApps provides a "Start from Data" template for building a new app which automatically generates the screens for searching and updating database records.

PowerApps provides app builders the ability to access external data through the use of connectors. You can use a connector to create a connection to a datasource such as a SQL Server database or a SharePoint list. PowerApps provides over 100 other connectors out of the box for connecting to common data sources such as OneDrive, Excel, Azure Active Directory, Azure SQL Server and Dynamics 365.

One really valuable aspect of connectors and connections is that they are able to store and reuse security credentials when connecting to an external data source. When creating a connection, you are often promoted to provide security credential and grant permissions to the app using the connection. After that, PowerApps should be able to establish a secure connection to that datasource using the stored credentials.



Creating an Office 365 E5 Trial Tenant

- All students will create an Office 365 trial tenant
  - Provides an isolated development environment for lab exercises
  - Trial accounts will last for 30 days

Office 365 Enterprise E5 Trial

Want to add this to an existing subscription? Sign in

Create your user ID

You need a user ID and password to sign in to your account.

Student @ DDPAF .onmicrosoft.com

Student@DDPAF.onmicrosoft.com

Create password

Confirm password

By clicking **Create my account** you agree to our terms and conditions.

Create my account

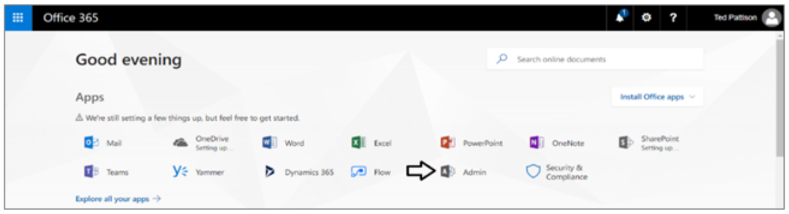
In the lab exercises for this course, you will go through the steps to sign up for an Office 365 Enterprise E5 trial account. By doing this you will create a new Office 365 tenant which makes it possible to create multiple user accounts. A key point here is that you are creating a trial account for an entire Office 365 organization as opposed to creating a trial account for a single user.

When you initially create the new Office 365 tenant, you will be prompted to enter the user name and password for a new user account. This initial user account will be created with full tenant administrator capabilities. This means that this accounts with have full administrative control over user management and group management within the Office 365 tenant. This account will also be able to see and modify the organization-wide administrative settings for important cloud services such as Azure AD, SharePoint Online, Power BI and, of course, PowerApps and Microsoft Flow.

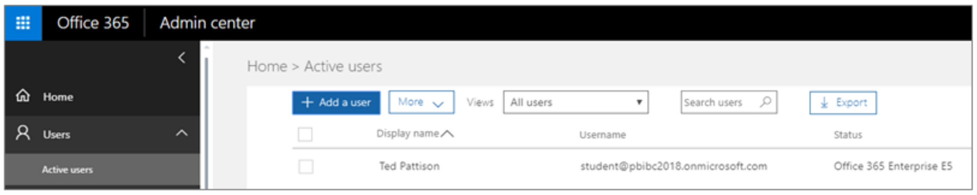
A significant benefit of creating a test environment in this fashion is that you can create additional users which makes it possible to test PowerApps and Flow scenarios such creating custom connectors and configuring an On-Premises Gateway. An Office 365 Enterprise E5 trial account allows you to add up to 25 user accounts for testing purposes. You will also be able to create new PowerApps environments so you can create a setup where you build apps and other components in a development/staging environment and then practice how to package and deploy your work to a production environment.

Office 365 Admin Center

- Navigate to the Office 365 Admin center



- Allows for management of users accounts and licensing



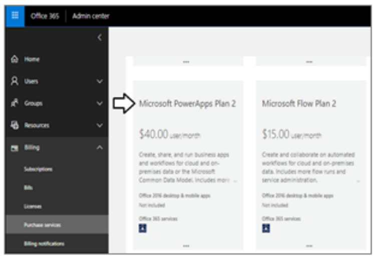
Display name	Username	Status
Ted Pattison	student@pbibc2018.onmicrosoft.com	Office 365 Enterprise E5

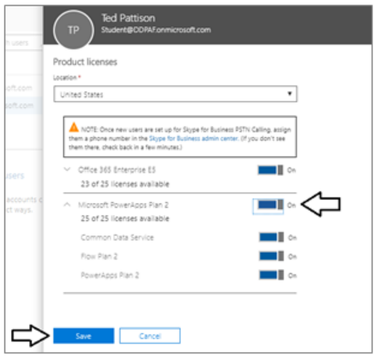
Once you have created your new Office 365 Enterprise E5 trial account, you should become familiar with the process of navigating around inside the Office 365 Admin center. For example, you need to learn how to add new user accounts and groups. In the lab exercises, you will add a secondary user account for testing purposes. For most of the class, you will log in and complete your work using your primary user account which will have tenant-level administrative permissions. The benefit of creating a second user account is that you can log in and test your apps and flows with a typical user account which does not have any administrative permissions.

The Office 365 Admin Center also allows you to purchase new subscriptions and to start trial subscriptions. This is something you will be required to do. More on that as we move to the next slide.

Configuring a PowerApps Plan 2 License

- Certain design tasks require PowerApps Plan 2
  - You can start a 30-day trial for PowerApps Plan 2
  - License must be assigned to individual user accounts



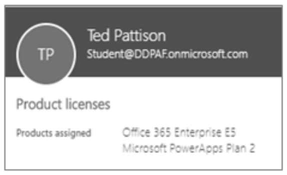

Ted Pattison
Student@DDPAF.onmicrosoft.com

Product licenses

Location *: United States

NOTE: Once new users are set up for Skype for Business PSTN Calling, assign them a phone number in the Skype for Business admin center. (If you don't see them there, check back in a few minutes.)

Office 365 Enterprise E5 — 23 of 25 licenses available	On
Microsoft PowerApps Plan 2 — 25 of 25 licenses available	On
Common Data Service	On
Flow Plan 2	On
PowerApps Plan 2	On


Ted Pattison
Student@DDPAF.onmicrosoft.com

Product licenses

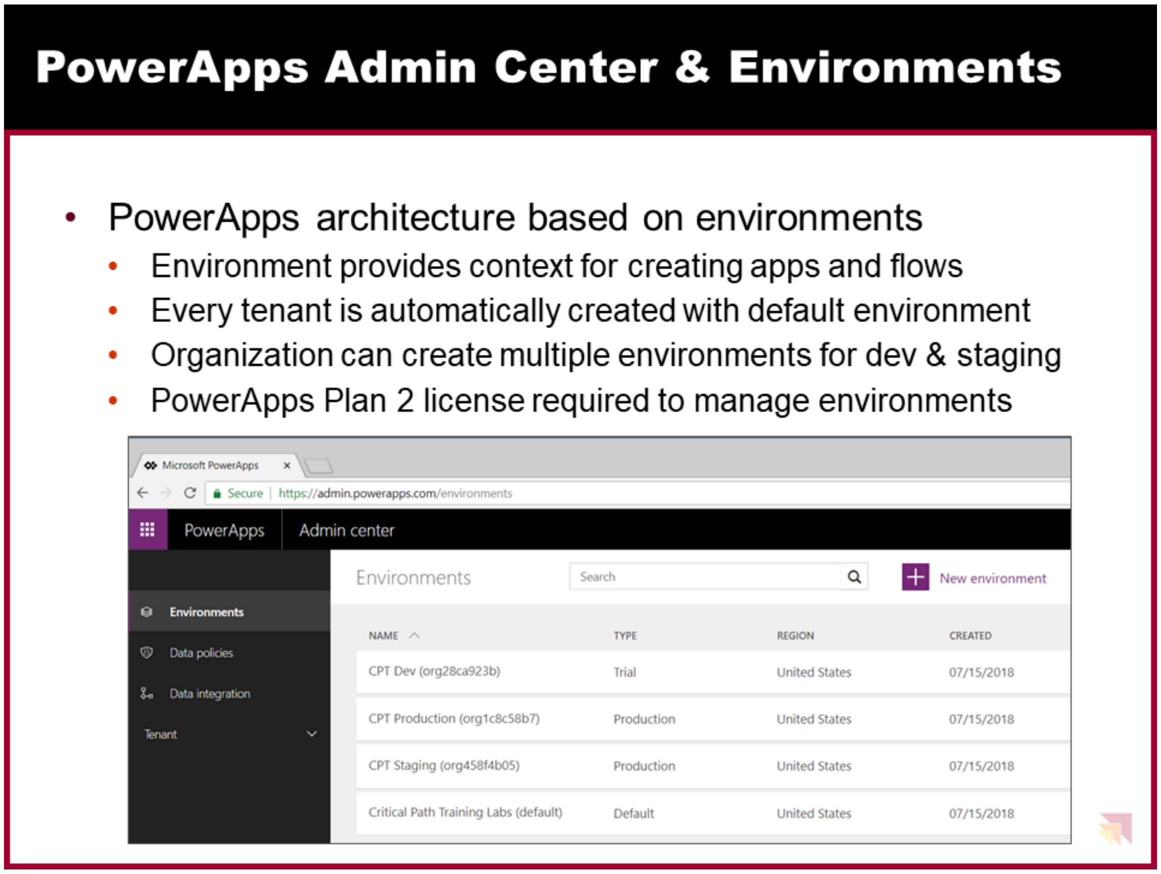
Products assigned: Office 365 Enterprise E5, Microsoft PowerApps Plan 2

In the lab exercises for this module, there are instructions to configure your primary Office 365 user account with a PowerApps plan 2 license. The reason for this is that it provides you with the ability to view and create PowerApps environments and to work with the Common Data Service for Apps to complete design tasks such as creating a custom entity or a model-driven app.

Here are the high-level steps required to configure a PowerApps Plan 2 license.

- Navigate to the Purchase services ion of the Office 365 admin center
- Start a Trial Subscription of Microsoft PowerApps Plan 2
- Configure Your User Account with a Microsoft PowerApps Plan 2 license

Note that the upcoming lab exercises for this module will provide you with the step-by-step instructions needed to complete these steps.



PowerApps Admin Center & Environments

- PowerApps architecture based on environments
  - Environment provides context for creating apps and flows
  - Every tenant is automatically created with default environment
  - Organization can create multiple environments for dev & staging
  - PowerApps Plan 2 license required to manage environments

Microsoft PowerApps
Secure | https://admin.powerapps.com/environments
PowerApps Admin center
Environments | Data policies | Data integration | Tenant
Environments Search New environment

NAME	TYPE	REGION	CREATED
CPT Dev (org28ca923b)	Trial	United States	07/15/2018
CPT Production (org1c8c58b7)	Production	United States	07/15/2018
CPT Staging (org458f4b05)	Production	United States	07/15/2018
Critical Path Training Labs (default)	Default	United States	07/15/2018

The architecture for PowerApps and Flow is based on an important concept of “environments”. Any time you create a new app with PowerApps or a new flow, you are always doing so within the context of a specific environment.

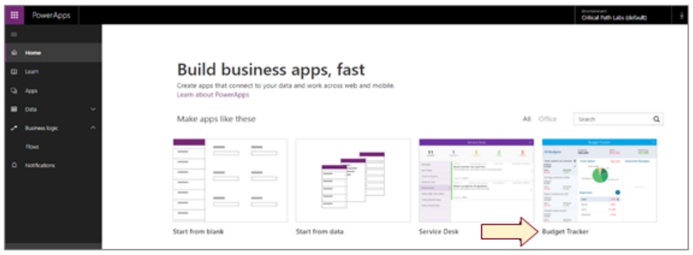
You don’t have to think about environments when you start working with PowerApps. That’s because every new Office 365 tenant is automatically created with default environment. When you first begin to create apps and flows, you will be working within the default environment for the tenant that was created along with your new Office 365 trial account. During the first part of this training course, everything you do will be done in the default environment.

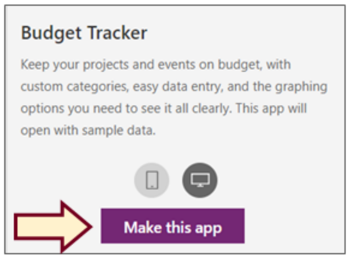
You can view and manage the environments for the current Office 365 tenant by navigating to the PowerApps Admin center at https://admin.powerapps.com. Note that you will not be able to view existing environments or create new environments until your Office 365 user account has been assigned a PowerApps Plan 2 license.

This training course provides a module on application lifecycle management (ALM) with PowerApps and Flow. At that point, the instructor will go into greater detail about creating and managing environments. As you will learn, an organization can benefit from creating multiple environments to assist with developing and staging the apps and flows they build. After building and testing apps and flows in a staging environment, an organization can design a strategy for packaging and deploying them into a production environment.

Creating New Apps

- You create new apps from the PowerApps Home page
  - Located at https://web.powerapps.com
  - Chose an app template
  - Choose between **Mobile App** and **Tablet|Desktop App**
  - Click the **Make this app** button




Budget Tracker

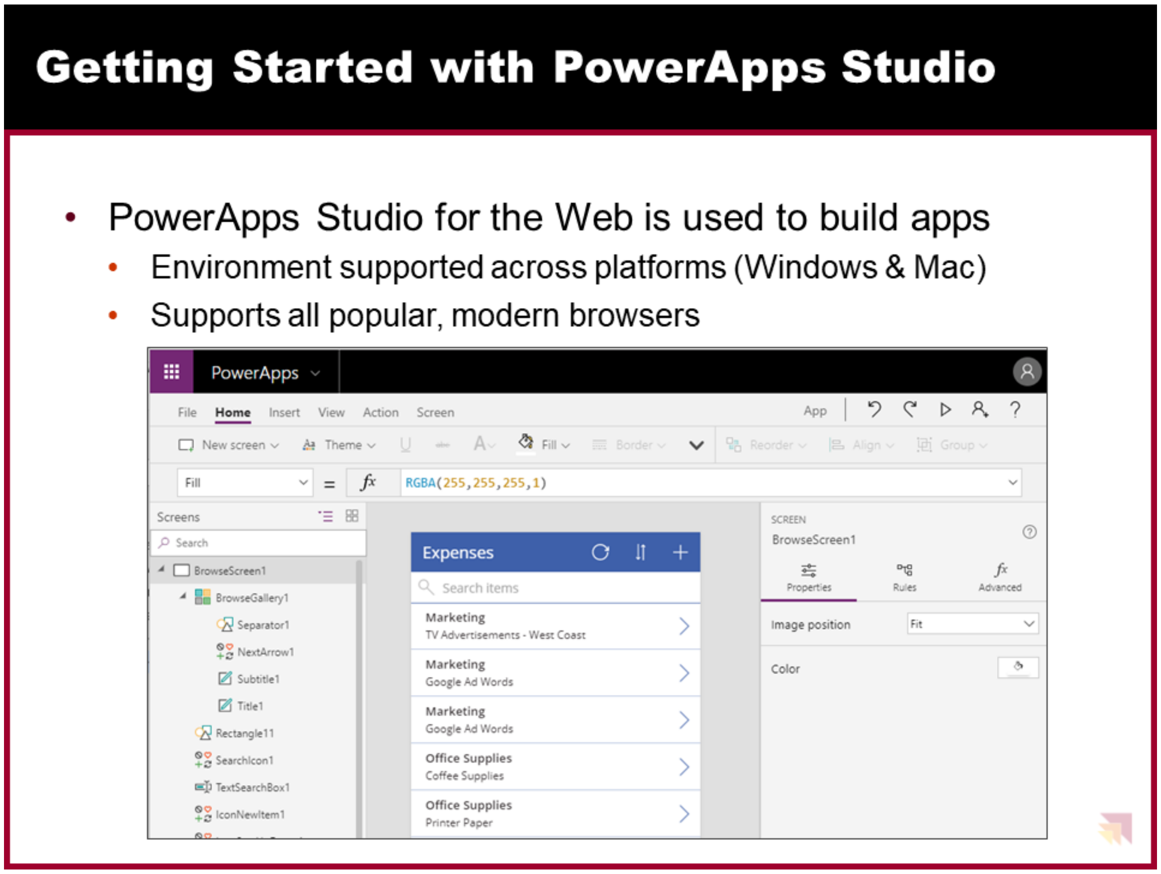
Keep your projects and events on budget, with custom categories, easy data entry, and the graphing options you need to see it all clearly. This app will open with sample data.

- New app creation redirects you to PowerApps Studio
  - Located at https://create.powerapps.com

When you start using PowerApps, there are two important URLs that are easy to confuse with one another. The first is the URL to the PowerApps Home page which is located at https://web.powerapps.com. The second is the URL to PowerApps Studio which is https://create.powerapps.com. It's important you understand the difference between these two locations.

You should navigate to the PowerApps Home page at https://web.powerapps.com whenever you would like to create a new app or when you would like to play, edit or share an app you have already created. There is a navigation menu on the left which allows you to see and manage other PowerApps resources such as connections, custom connectors and custom entities in the Common Data Service for Apps.

Whenever you create a new app or you edit an existing app, you will be redirected to PowerApps Studio at https://create.powerapps.com. In other words, PowerApps Studio is the place where you do all your work designing, testing and debugging the apps you create with PowerApps. When you navigate to PowerApps Studio, you will notice that the browser resolves to a URL that has a locale inside it as in the case of https://us.create.powerapps.com.



Getting Started with PowerApps Studio

- PowerApps Studio for the Web is used to build apps
  - Environment supported across platforms (Windows & Mac)
  - Supports all popular, modern browsers

```
RGBA(255,255,255,1)
```

If this is your first day using PowerApps, it will take a small bit of time to become accustomed to PowerApps Studio. In the first hands-on lab, you will get experience creating and building new apps. Once you create a new app, you must learn how to move from screen to screen and how to add new controls. You must also learn to set property values using the Properties pan and by writing formulas in the Formula bar.

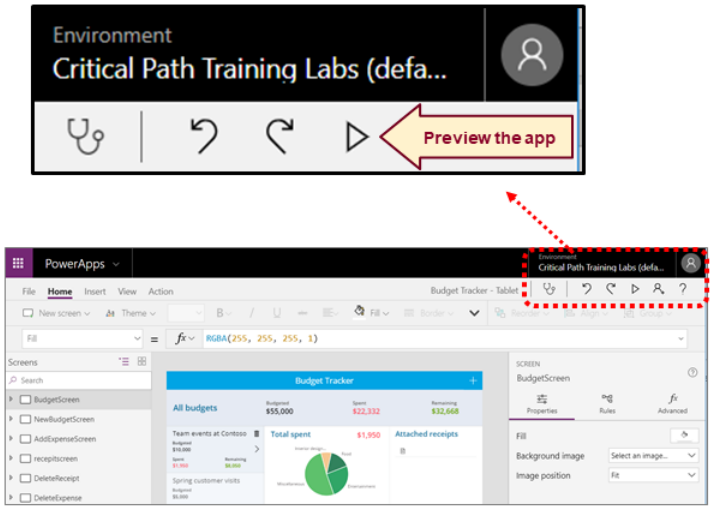
PowerApps Studio provides a user interface using the following components

1. **Left navigation menu** displays app hierarchy of screens and controls.
2. **Screen designer** is the pane in the middle showing the current screen in design mode.
3. **Right-hand pane** displays Properties pane, Rules pane and Advanced pane.
4. **Property dropdown list** allows you to select property for Formula bar.
5. **Formula bar** where you write formulas in PowerApps Formula Language.
6. **Ribbon tabs** provide ribbon buttons to add and customize design elements.

PowerApps Studio is primarily available as a web-based application that is accessible through any modern browser. You should note that there is also a native version of PowerApps Studio that can be installed on the Windows operating system or on a Mac. These native versions of PowerApps Studio were more important in the early preview days of PowerApps because back then they provided functionality that was not available in the web-based version. At this point, Microsoft has improved the web-based version so it now includes all functionality that exists in the native application versions. Furthermore, Microsoft now recommends using the web-based version of PowerApps Studio over the native versions because that's where Microsoft's primary investments will be made moving forward.

Running an App from PowerApps Studio

- You can run the app using the PowerApps Studio toolbar
  - Run the app by clicking the **Preview the App** button
  - Stop a running app to return to design mode



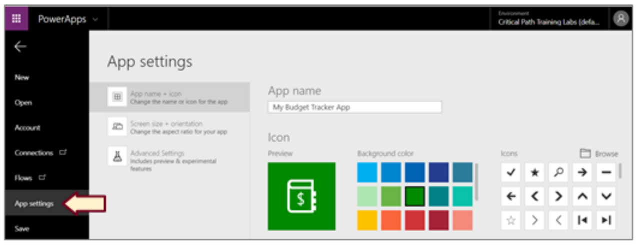
While you are building an app, you will want to periodically run the app in order to test it and make sure the app it does what you want it to do. The way that you run an app is by clicking the ‘Preview the app’ button in the toolbar in the upper right corner of PowerApps Studio. Running the app makes it possible to experience what the consumers of the app will see when they launch the app.

After running and testing an app, you must learn how to quit the app and return back to design view in PowerApps Studio. You can accomplish this by clicking the button with the X icon in the top, right corner of the App Preview window. You should become comfortable running and quitting apps in PowerApps studio because you will be doing that on a regular basis as you build apps.

When you begin to work with variables and collections (covered in the next lecture), you will see that it is often required to run the app to initialize variable and collection values. In other words, you cannot initialize a variable or a collection while the app is in design mode. Instead, you must run the app and execute the actions to initialize the values for variables and collections.

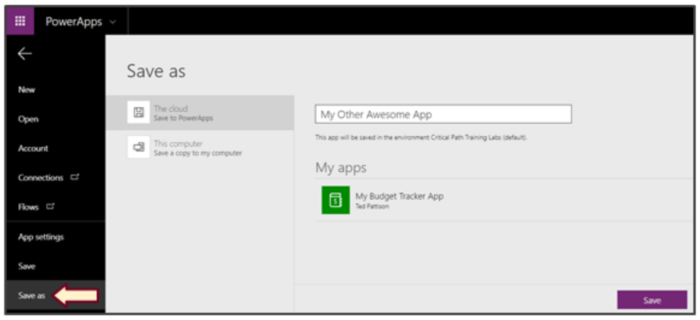
Saving an App to the Cloud

- Before saving, first you should configure App settings


PowerApps
Environment
Critical Path Training Labs (defa...
New
Open
Account
Connections
Flows
App settings
Save
App settings
App name + icon
Change the name or icon for the app
Screen size + orientation
Change the aspect ratio for your app
Advanced Settings
Includes preview & experimental features
Icon
Preview
Background color
Icons
Browse

App name: My Budget Tracker App

- Save app to cloud using Save or Save As command



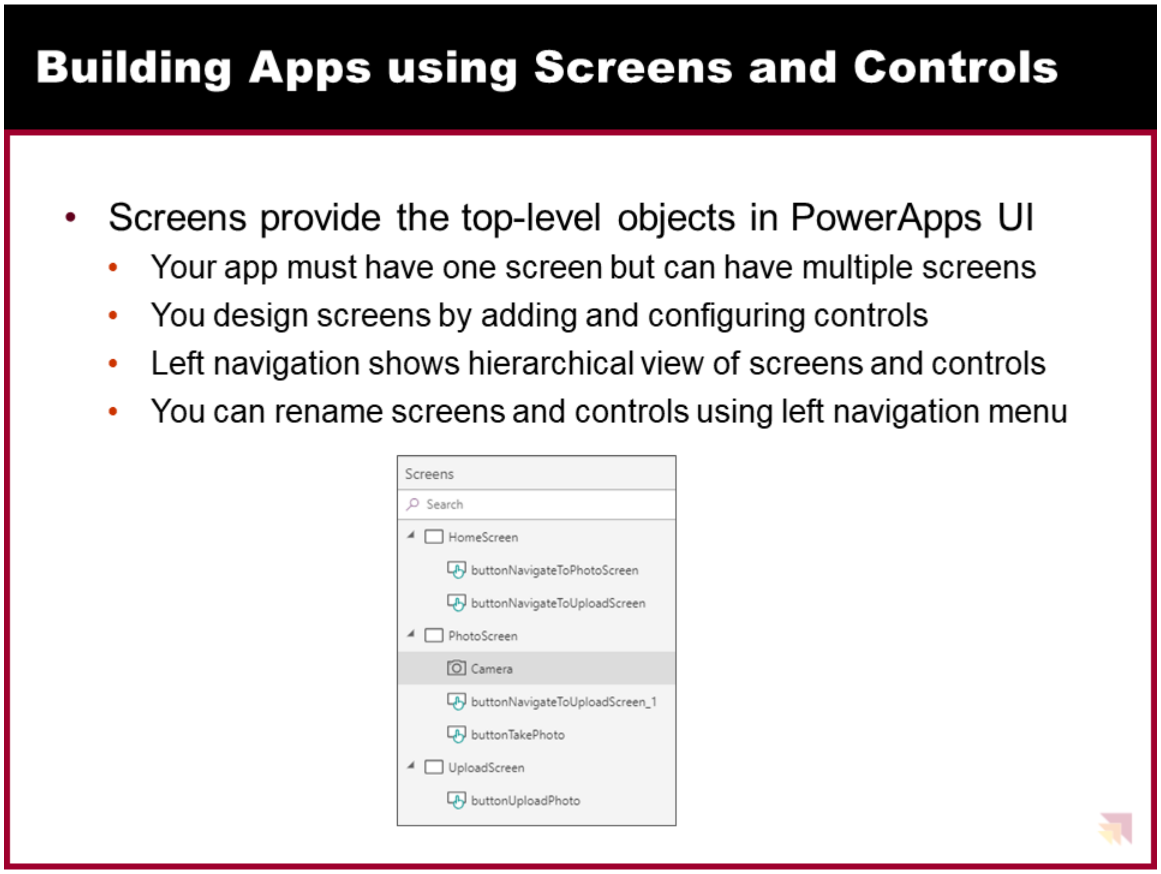
My Other Awesome App

This app will be saved in the environment Critical Path Training Labs (default).

When you create a new app from the PowerApps Home page, you are redirected from the URL of https://web.powerapps.com over to PowerApps Studio at https://create.powerapps.com. While you are building and testing an app, everything you do is simply stored in memory by the browser. If you close the browser without saving your work, then the app you are creating will be discarded. Therefore, you must save your app at periodic intervals to ensure you do not lose any work.

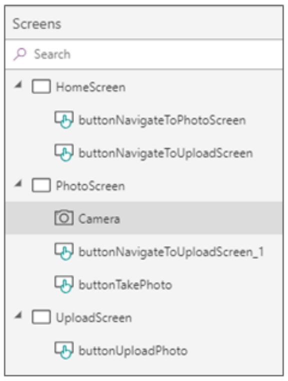
An app can be saved into storage in the cloud by using the Save command or the Save As command. However, before saving an app for the first time, you should navigate to the App settings page so you can enter an app title as well as configure the app color and the app icon. The App settings page also provides access to the settings for the app's screen size and orientation as well as advanced settings which makes it possible to enable new features that are still in preview.

After you save an app, you can later return to it to run the app or to open it up in edit mode to continue developing it. The user who creates an app is automatically assigned as the app owner. By default, the app is not available to other users. However, you can share an app you have created with other users so they can play it and use it as an app consumer. You can also share an app with edit permissions so another user can open the app in edit mode and continue the design work by adding new screens and controls.



Building Apps using Screens and Controls

- Screens provide the top-level objects in PowerApps UI
  - Your app must have one screen but can have multiple screens
  - You design screens by adding and configuring controls
  - Left navigation shows hierarchical view of screens and controls
  - You can rename screens and controls using left navigation menu



```
Screens
Search
◢ HomeScreen
    buttonNavigateToPhotoScreen
    buttonNavigateToUploadScreen
◢ PhotoScreen
    Camera
    buttonNavigateToUploadScreen_1
    buttonTakePhoto
◢ UploadScreen
    buttonUploadPhoto
```

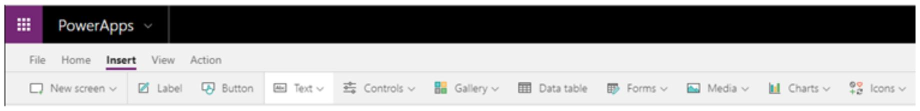
Screens represent the top-level objects in any user interface you create with PowerApps. Every app must contain at least one screen. However, an app can have multiple screens and it's common to add secondary screens when designing real-world apps. Note that only one screen can serve as the startup screen for an app. The screen which is located at the top of the left navigation menu is considered to be the startup screen. That means the top screen in the left navigation will be automatically displayed whenever the app is run. When you add secondary screens to an app you have created with PowerApps Studio, you must extend the app with extra behavior to display and navigate to these secondary screens.

You design screens by adding and configuring controls. Controls represent standard user interface elements such as buttons, labels, textboxes and combo boxes. When you want to add a new control, you must add it to a specific form. However, once you have added a control to a screen and configured its properties, you can copy that control to the clipboard and paste a copy of it into another screen within the same app.

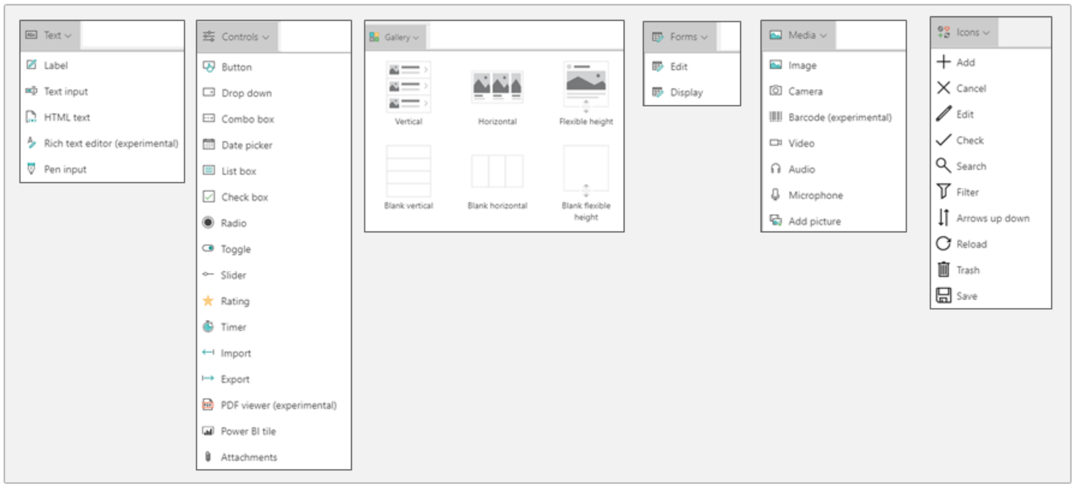
The left navigation menu is important because it displays a hierarchical view of screens and controls that make up your app. Using the left navigation menu is often the best way to navigate to a particular screen or control. This is especially true once you have a screen with a large number of controls when it becomes hard to select the control you want in the PowerApps screen designer. Note that the left navigation menu also makes it possible to rename screens and controls to give them names that are more meaningful. While renaming screens and controls can be tedious, it is a good practice to get into because it makes apps easier to understand and to maintain.

Adding Controls to a Screen

- You add controls to a screen using the **Insert** ribbon tab



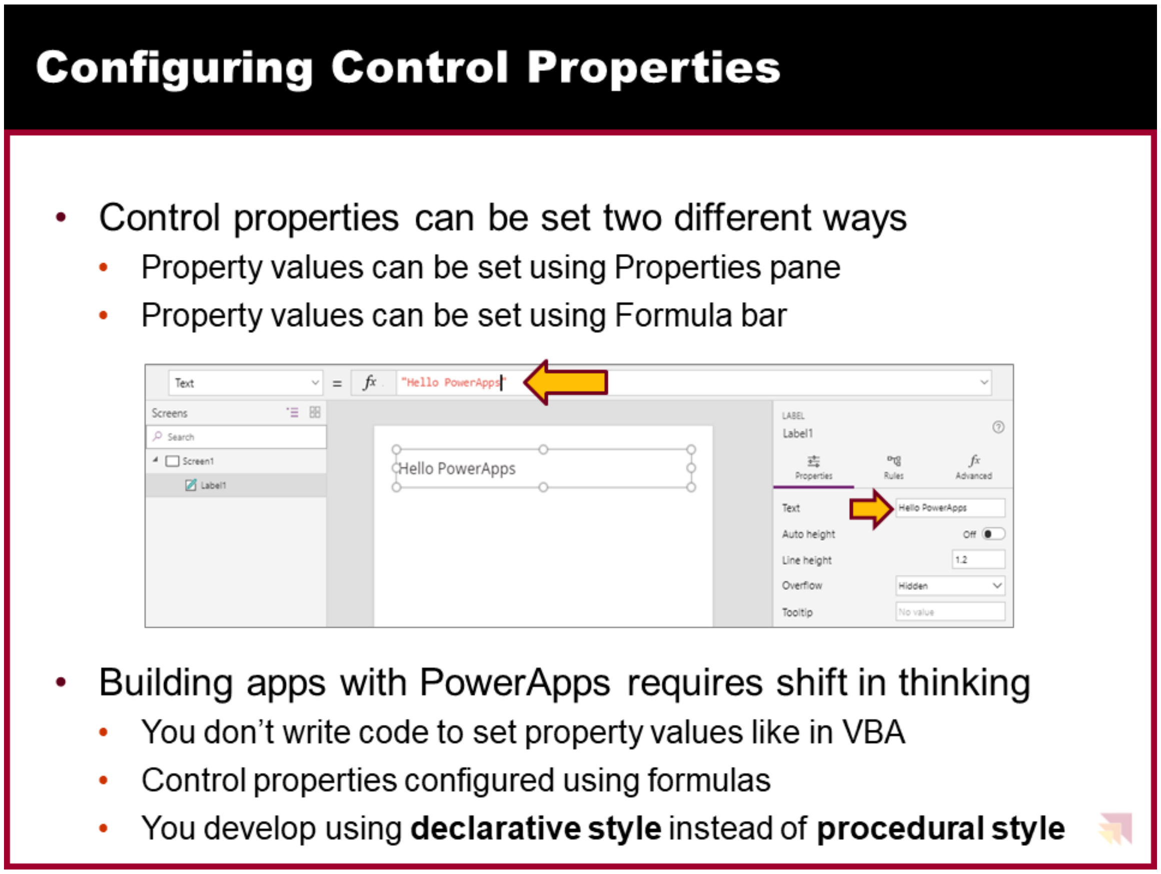
- PowerApps provides extensive set of controls for web and mobile apps



PowerApps provide an extensive set of controls for building whatever user experience your app requires. To create a new control, you should first select the correct screen in the left navigation and then navigate to the Insert tab in the ribbon. Next, you can use one of the dropdown menu buttons on the Insert tab to add the type of control you want.

The Text menu provides controls that deal with text values such as Labels and Textboxes. The Controls menu provides controls such as Combo Box, Date picker and Slider that provide a more convenient user experience for entering data. You can also add controls that support data binding such as a gallery, a display form or an edit form. These controls will be explained in more details over the next few slides.

Take a look at the controls available in the Media menu. As you can see there are controls such as Camera, Microphone and Barcode that can be used to build apps that take advantage of the capabilities of a modern mobile device such as an iPhone or an Android phone. There is also an Icons menu that makes it quick and easy to add style to an app by adding familiar icons for tasks such as saving, filtering and searching through data.



Configuring Control Properties

- Control properties can be set two different ways
  - Property values can be set using Properties pane
  - Property values can be set using Formula bar

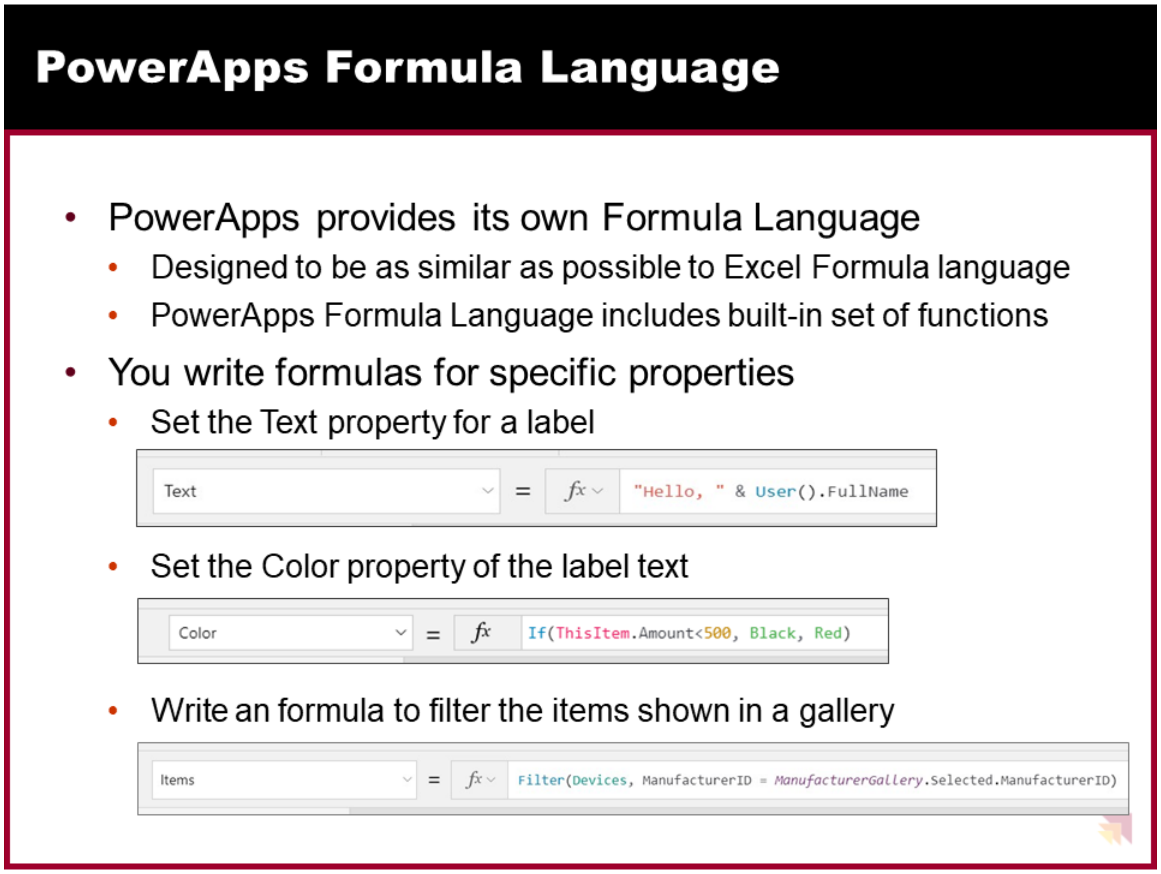
- Building apps with PowerApps requires shift in thinking
  - You don't write code to set property values like in VBA
  - Control properties configured using formulas
  - You develop using **declarative style** instead of **procedural style**

When it comes to configuring the property values for a control, you have two choices. First, you can configure property values using the Properties pane on the right. In the example shown on the slide above, you can see that the Properties pane displays an editable textbox for the Text property of a Label control. This textbox in the Properties pane makes it possible to view and update the Text property of the label.

As an alternative to using the Properties pane, you can edit the Text property of a control using the Formula bar which is located above the PowerApps screen designer. To edit a control property using the Formula bar, you must first select the control. Next, you must select the property you want to configure in the property selector dropdown menu to the left of the Formula bar. After that, you can configure the control property by writing a formula in the Formula bar. In the example shown on the slide above, the formula in the Formula bar sets the Text property of the Label control. While the formula used in this example is just a static string value, you can write formulas which are dynamic. The ability to write dynamic formulas for screen and control properties adds a very important aspect to the PowerApps development model.

Let's take a step back and think about how PowerApps allows you to set control properties using dynamic formulas. This leads to a development style that is quite different than times in the past when you may have written code in a programming language such as VBA, VB or C#. In these other languages, it is common to write procedural code that sets control properties. For example, you might write code behind the Reset button on a form to set the Text property of every textbox back to an empty string. This style of procedural coding is not the same style of development that you should be using when you begin working with PowerApps.

In PowerApps, you will not be writing any procedural code to set control properties. Instead, you configure control properties by writing formulas. Therefore, control properties take care of setting their own values. This leads to a development style in PowerApps that is 'declarative' instead of 'procedural'. However, in order to master this new declarative style of development, you must first learn about the syntax of the PowerApps Formula Language.



PowerApps Formula Language

- PowerApps provides its own Formula Language
  - Designed to be as similar as possible to Excel Formula language
  - PowerApps Formula Language includes built-in set of functions
- You write formulas for specific properties
  - Set the Text property for a label

```
Text = fx "Hello, " & User().FullName
```

  - Set the Color property of the label text

```
Color = fx If(ThisItem.Amount<500, Black, Red)
```

  - Write an formula to filter the items shown in a gallery

```
Items = fx Filter(Devices, ManufacturerID = ManufacturerGallery.Selected.ManufacturerID)
```

PowerApps defines its own Formula language for writing the formulas used to configure control properties. When Microsoft created the PowerApps Formula language, they tried to make it as similar as possible to the Excel Formula language. The reason for this is that are already 100 trillion (OK, maybe that’s an exaggeration) business users that know how to write formulas in an Excel workbook. Many Excel users will feel at home as they begin to write formulas in PowerApps because it uses a syntax and functions that they are already familiar with.

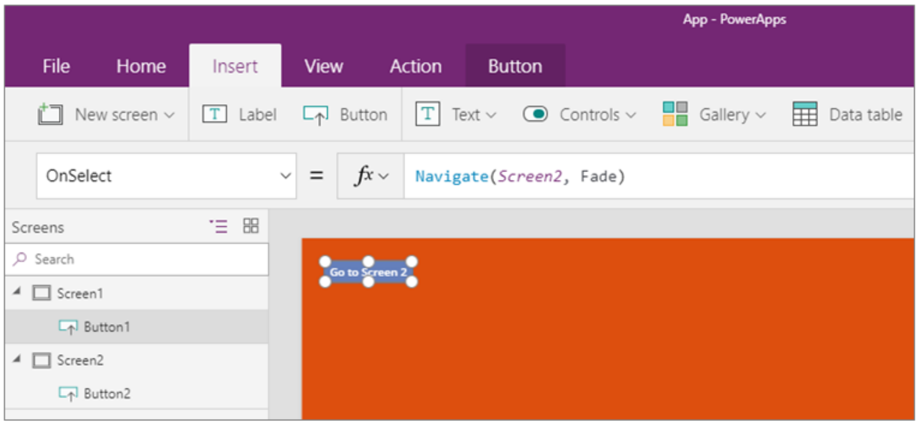
The PowerApps Formula language includes a built-in set of functions. Here are a few simple examples. You can use the Text function to apply formatting to a currency value. You can use the User function to return an object with information about the current user. You can use the Filter function to filter the set of records displayed in a Gallery control.

In the next module, you will see that PowerApps Formula language provides additional syntax for compound data structures such as tables and records. You will also learn that some functions provided by the PowerApps Formula language return a value while other perform an action. If you want to take a look at Microsoft documentation for this language, you can navigate to the following URL:

https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/formula-reference

Navigating Between Screens

- Navigate to a screen using Navigate function
  - Call **Navigate** function from **OnSelect** property of **Button** control
  - **Navigate** function performs action instead of returning a value

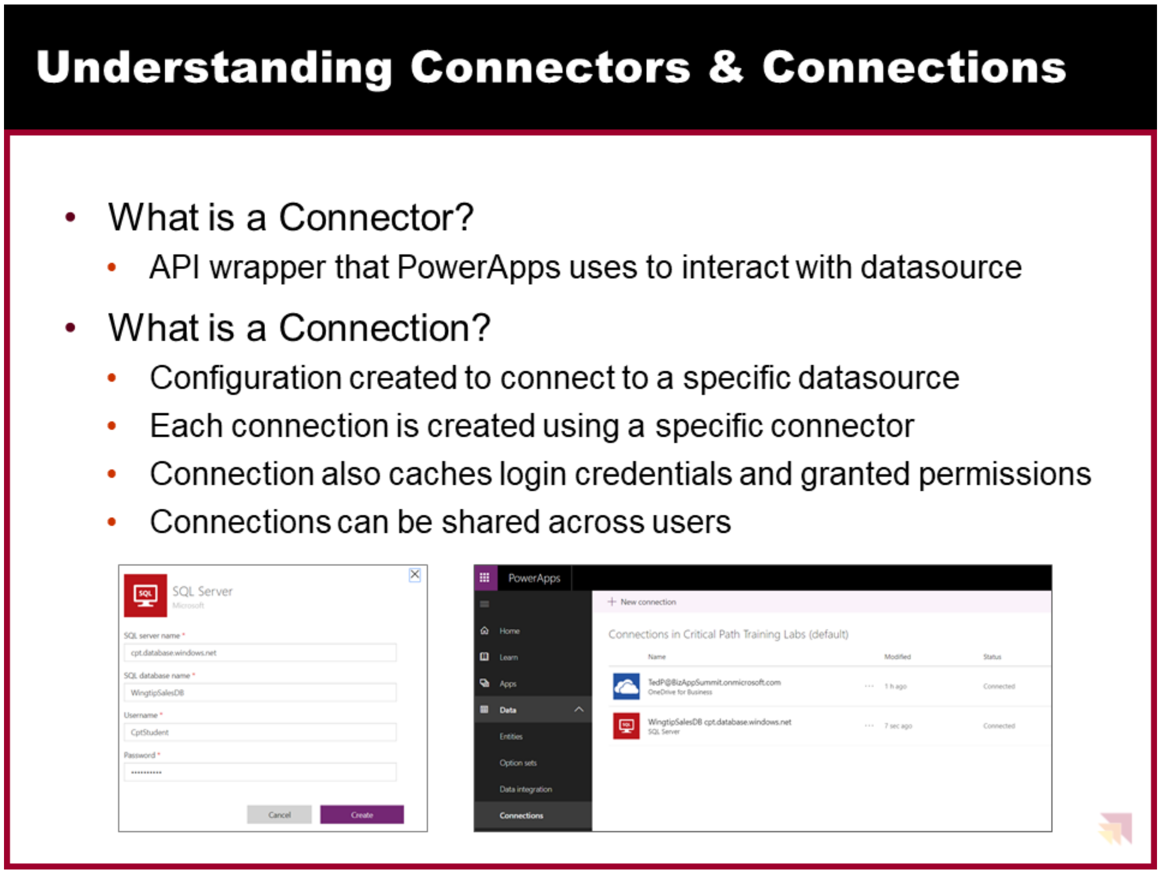


OnSelect = `Navigate(Screen2, Fade)`

When an app is run, PowerApps automatically displays the startup screen which is defined as the top screen in the Left Navigation menu. If you add secondary screens to an app, you must extend your app with the behavior to navigate to secondary screens. This can be accomplished by adding a Button control and configuring its OnSelect property with a formula that calls the Navigate function.

The Navigate function is an example of a function that performs an action instead of returning a value. When you call the Navigate function, the first parameter indicates the screen you want to display. The second parameters controls whether there will be a visual animation such as a fade when moving from one screen to another.

The Navigate function is complemented with a similar function named Back. The Back function provides the behavior of returning to the previous screen. For example, imagine you have an app that has a Customers screen that displays a list of customers and also a Customer screen that displays the data for a single customer at a time. On the Customers screen you can call the Navigate function to navigate to the Customer screen to display the data for a specific customer. The Customer screen could provide a Button control with an OnSelect property which calls the Back function to navigate back the Customers screen.



Understanding Connectors & Connections

- What is a Connector?
  - API wrapper that PowerApps uses to interact with datasource
- What is a Connection?
  - Configuration created to connect to a specific datasource
  - Each connection is created using a specific connector
  - Connection also caches login credentials and granted permissions
  - Connections can be shared across users

SQL Server
Microsoft

SQL server name *
cpt.database.windows.net
SQL database name *
WingtipSalesDB
Username *
CptStudent
Password *
••••••••••

Cancel Create

Connections in Critical Path Training Labs (default)

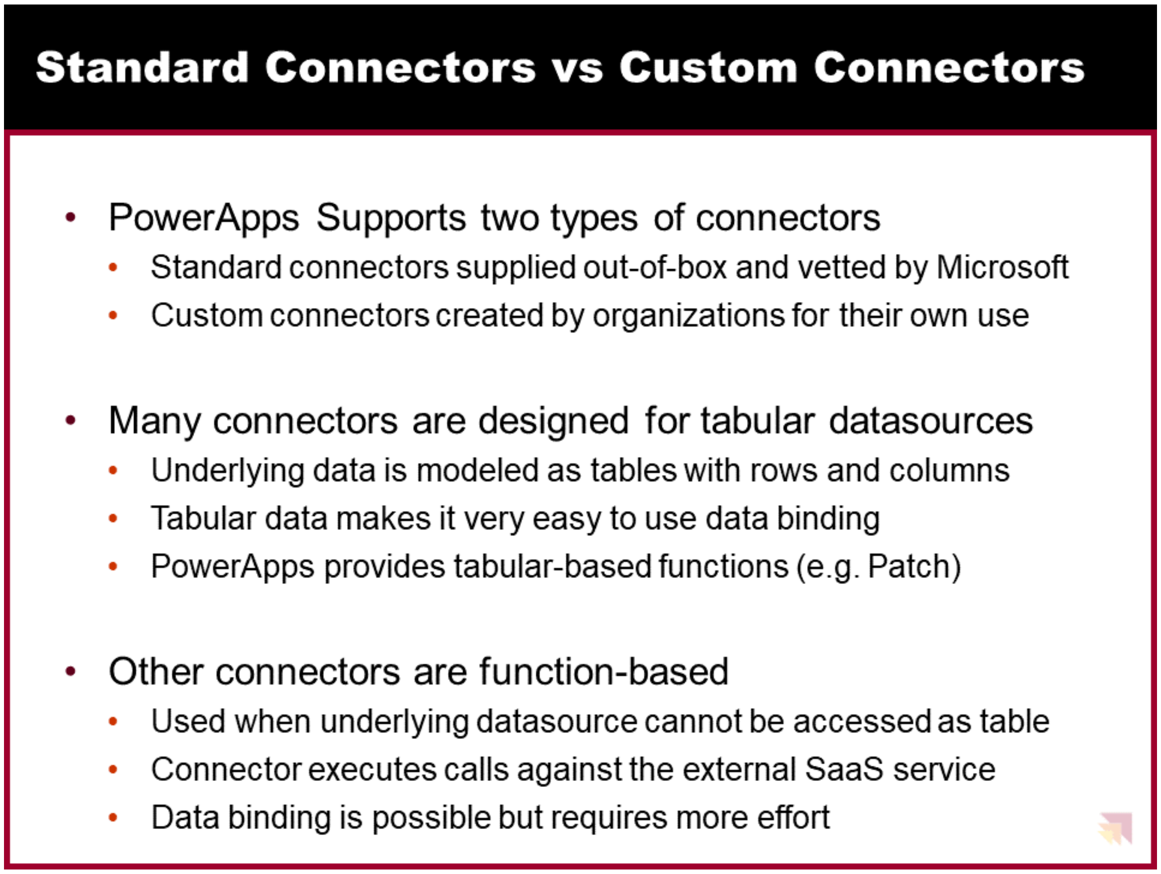
Name	Modified	Status
TedP@BizAppSummit.onmicrosoft.com OneDrive for Business	1 h ago	Connected
WingtipSalesDB cpt.database.windows.net SQL Server	7 sec ago	Connected

One of the strengths of PowerApps is that it makes it relatively easy to build an app that connects to data. Keep in mind that the data you need to access may be stored in many different types of datasources. You bring that data into your app by creating connections. Each connection is created by a specific user and may be shared across users. If you navigate to the PowerApps home page, you can see your current set of connections by navigating to Data > Connections node in the left navigation menu as shown in the screenshot in the slide above.

To become proficient at building apps with PowerApps, you must learn the key concepts involved with connectors. A connector is a wrapper around an API that PowerApps uses to interact with a specific type of datasource. Each connection you create is based on a specific connector which knows how to communicate with the datasource to which you are connecting. PowerApps provides out-of-the-box connectors for many popular services and on-premises datasources including SQL Server, SharePoint, OneDrive for Business, Azure AD, Dynamics 365, Salesforce and Twitter.

There are two important aspects of connectors and connections that relate to security. When using a connection for the first time, the user is often prompted to provide security credentials such as a user name and password to establish a secure connection to the datasource. After the user supplies security credentials the first time, the connection is able to cache those credentials and reuse them to reestablish connections to that datasource in the future without any need for user interaction.

The second security-related aspect of connections has to do with users delegating permissions to an app. A connection often allows an app to read and write to a datasource on behalf of the current user. PowerApps prompts users with an interactive dialog which allows a user to grant permissions to a connection which is required in order to read and write to a datasource on a user's behalf.



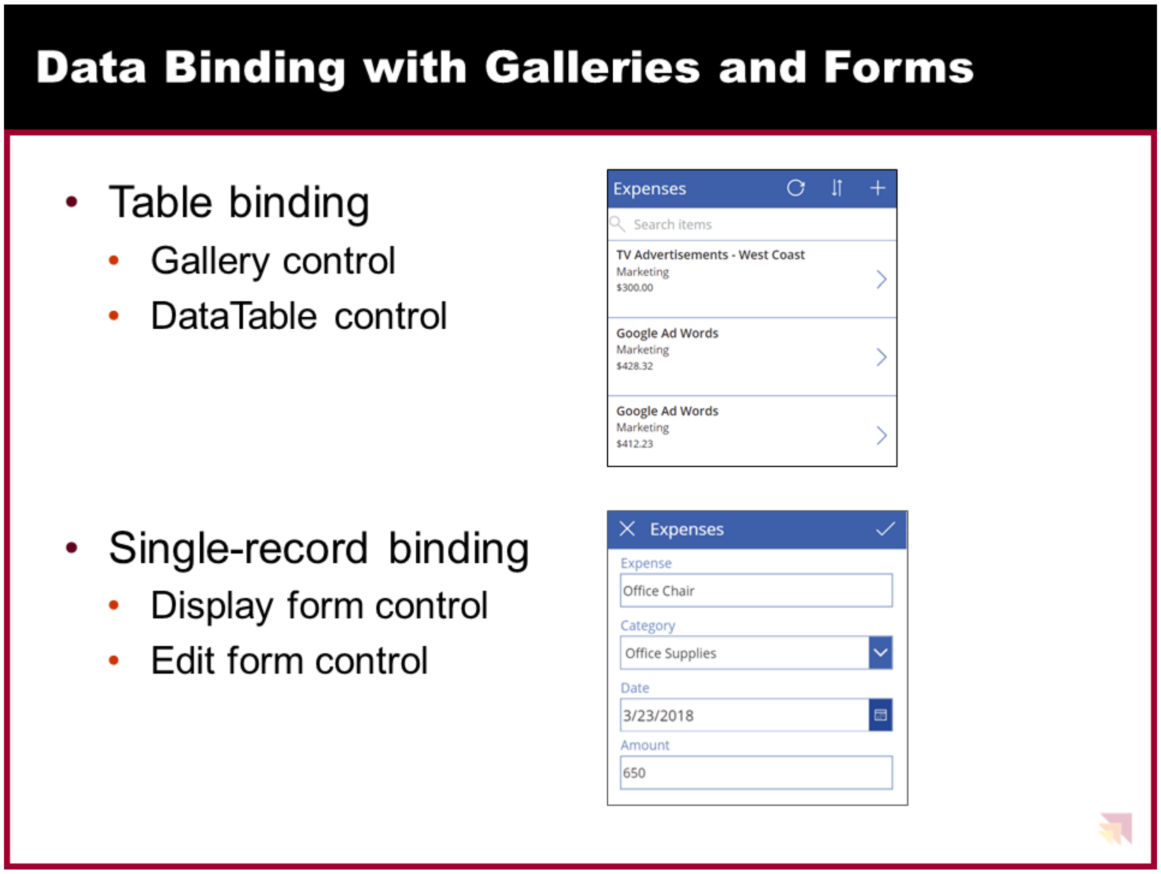
Standard Connectors vs Custom Connectors

- PowerApps Supports two types of connectors
  - Standard connectors supplied out-of-box and vetted by Microsoft
  - Custom connectors created by organizations for their own use
- Many connectors are designed for tabular datasources
  - Underlying data is modeled as tables with rows and columns
  - Tabular data makes it very easy to use data binding
  - PowerApps provides tabular-based functions (e.g. Patch)
- Other connectors are function-based
  - Used when underlying datasource cannot be accessed as table
  - Connector executes calls against the external SaaS service
  - Data binding is possible but requires more effort

PowerApps supports both standard connectors and custom connectors. The difference is that Microsoft provides standard connectors out-of-the-box as part of the PowerApps platform. If you're connecting to a data source that PowerApps supports with a standard connector, you should use that connector. If you need to connect to a datasource for which Microsoft does not provided a standard connector, you can create a custom connector to integrate the data from that datasource.

Many connectors are designed to work with tabular datasources such as tables in Excel workbooks, tables in SQL Server and lists in SharePoint. With a tabular-based connector, data can be modeled as a table with rows and columns. PowerApps provides its functions for tabular datasources as Patch, Collect and Update which you can call to interact with data in an underlying table. You should also keep in mind that tabular datasources make it easy to use data binding with Gallery controls and Form controls.

PowerApps also supports function-based connectors to interact with datasources that do not fit within the tabular connector model. For example, there are connectors that connect to SaaS services such as Office 365, Twitter and Facebook. When working with function-based connectors, PowerApps is able to move data back and forth between an app and a datasource by executing web service calls against the external Saas service. You can still use these types of connectors to bind data to galleries and forms, but it usually requires a little more work. That’s because you must typically cache the data returned from a function-based connector into a variable or collection in order to bind the data to a gallery or a form.



Data Binding with Galleries and Forms

- Table binding
  - Gallery control
  - DataTable control
- Single-record binding
  - Display form control
  - Edit form control

Expenses

Search items

TV Advertisements - West Coast
Marketing
$300.00

Google Ad Words
Marketing
$428.32

Google Ad Words
Marketing
$412.23

Expenses

Expense	Office Chair
Category	Office Supplies
Date	3/23/2018
Amount	650

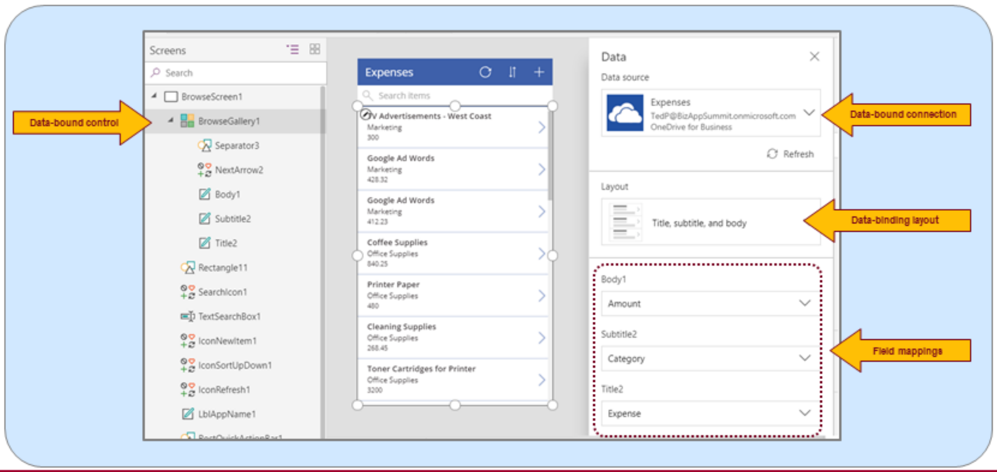
Data binding is one of the key features in PowerApps that dramatically simplifies the process of building an app to access data. When you create a new app using the ‘Start from data’ template, PowerApps Studio will use data binding when it generates the connection, the screens and the controls for the app. This means you can use data binding without really understanding how it works. However, learning how data binding really works in PowerApps will help you become a better app builder.

Data binding in PowerApps is based tabular data which is laid out in terms of tables and records. Some controls support data binding to a table of records while other controls support binding to a single record at a time. The Gallery control and the DataTable control are examples of controls that can be bound to a table of records. For example, you can create a Gallery control to display a set of expenses.

Forms are a special type of control that support data binding with a single record of data at a time. You can use a Display Form control when you want to display a single record of data in a read-only fashion. You can use an Edit Form control when you want to build a screen that allows a user to add a new record or to edit an existing record.

Working with the Data Pane

- You use the Data pane to configure data binding
  - Select a data-bound control and then display Data pane
  - Data pane allows you to change layout for data binding
  - Once you select layout, you can then map fields below

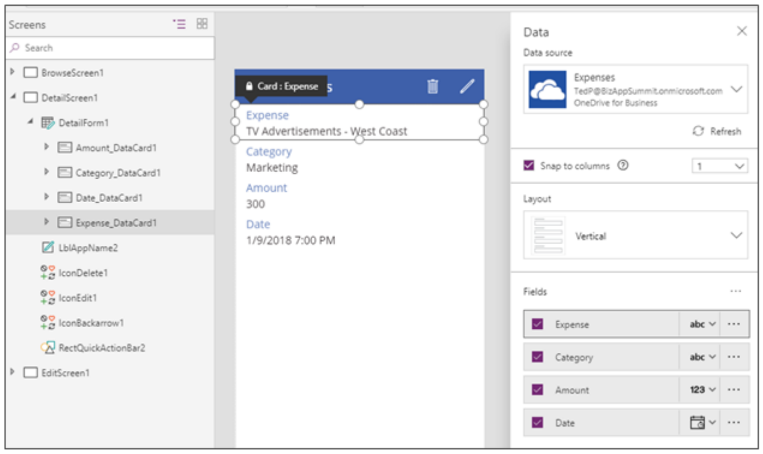


In order to customize the way data binding works, you must learn to use the Data Pane in PowerApps Studio. You use the Data Pane when you need to configure or customize the data binding for a data bound control such as the Gallery control. In order to use the Data pane, you must first select the data-bound control in the left navigation and then you can open the Data pane by navigating to the View tab in the ribbon and clicking the Data sources button. When a Gallery control is selected, the Data pane shows the details of the data binding including the connection, the data binding layout and the field mappings. It's easy to switch back and forth between different layouts and to update the field mappings to creating a better view of the underling data.

There is a common design pattern used in PowerApps where an app is created with three screens including (1) a browser screen, (2) a detail screen and (3) an edit screen. All three screens are used to data bind to the same underlying table. The browser screen contains a Gallery control with data binding to display many records at once. The detail screen uses a Display Form control to display the data for a single record in a read-only fashion. The edit screen contains an Edit Form control which allows users of the app to add new records and edit existing records.

Understanding Forms and Data Cards

- Form acts as a container for data cards
  - Each form binds to a single record
  - Within a form, each data card binds to an underlying field
  - Each data card contains an encapsulated set of child controls



In PowerApps, you use display forms and edit forms for scenarios where you need to bind to a single record at a time. Display forms and edit forms act as containers that hold another type of control known as a 'data card' which is also called a 'card control'. Each form contains a set of data cards that bind to individual fields from the underlying record. Each data card has a DataField property that determines which field it binds to within the underlying record.

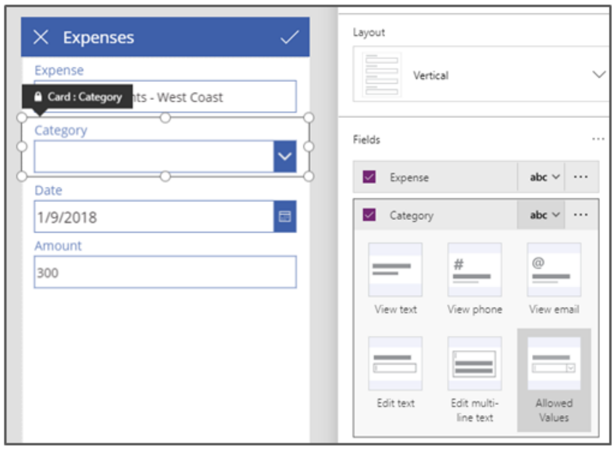
A data card contains its own set of child controls. The child controls inside a data card make up the experience for displaying and editing a single field. For example, a number data card may consist of a Label control to provide the display name of the field and a Text input control to provide an editor for the value of the field. Most of the data cards supplied by PowerApps also provide a label control to display any validation errors that occur due to bad user input.

When you are writing a formula for a Form control property and a data card propertythe current record is available using an object named ThisItem. The ThisItem object for a Form control contains properties for each field in the underlying record.

When you are writing a formula for the properties of a child control inside of a data card, you can use the Parent object. For example, a child control inside a data card should use Parent.Default to read the initial state of the field from the data source. By using Parent instead of directly accessing the information that you want, the data card is better encapsulated. That means you can use the same data card with multiple fields in the same record without breaking internal formulas.

Changing a Field’s Data Card Type

- Fields in Form control get default data card
  - Use Data pane to change data card used by any field
  - Different data cards offer different editing experiences



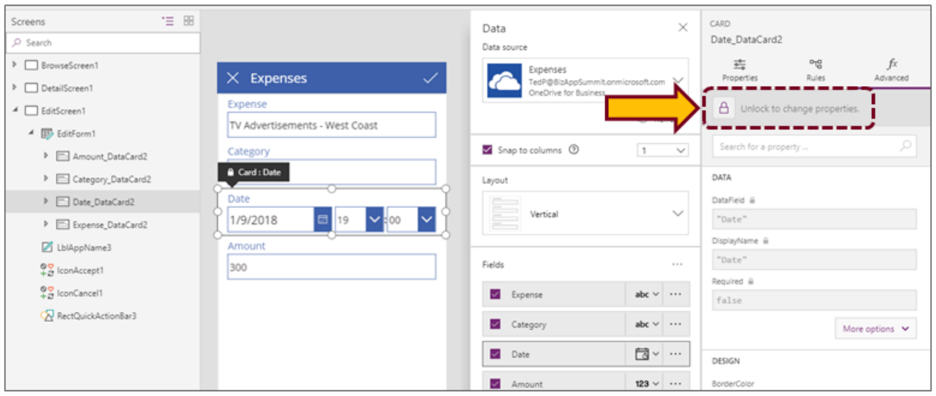
PowerApps provides several different types of data cards which can be used with various data types to create customized user interface experiences. When you use the ‘Start from data’ template to create a new app, PowerApps automatically generates a new view form and a new edit form for you with a default set of data cards.

Sometimes PowerApps will create a default set of data cards that isn’t exactly what you want. However, you can swap out the default data card used for a field with a different type of data card that improves the user experience. For example, imagine you have an Expenses table with a Category field which only allows 4 predefined category values. You can change the default data card which uses a standard textbox to a different data card that supplies the user with a dropdown list of choices.

Working with data cards can be a little tricky at first because you must select the form in the left navigation and then display the Data pane. Once you’ve done that, you can then view and edit the type of data card used for each field. The slide above shows an example of changing from a data card with a textbox over to a better data card named ‘Allowed Values’ that provides a dropdown menu.

Customizing a Data Card

- By default, data cards are locked and cannot be edited
  - In many scenarios, you should leave data cards locked
  - Some scenarios call for unlocking data cards to customize them



Expense: TV Advertisements - West Coast
Category:
Date: 1/9/2018 19 : 00
Amount: 300
Expenses TedP@BizAppSummit.onmicrosoft.com
Snap to columns: 1
Layout: Vertical
Fields: Expense abc, Category abc, Date, Amount 123
DataField: "Date"
DisplayName: "Date"
Required: false

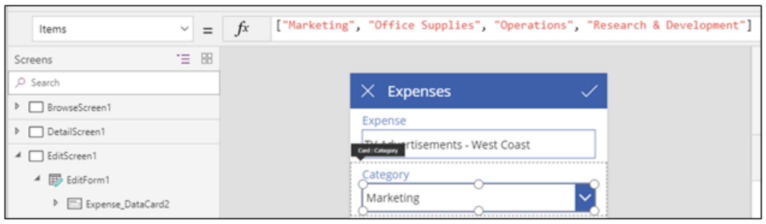
The standard data cards provided by Microsoft are locked by default. When a data card is locked, you can modify a small number of properties of the data card and child controls inside. If a property is locked and can't be modified, it appears in the Property pane with a lock icon next to its name.

In many cases, leaving data cards in their locked state is helpful because it prevents you from making unintended changes that could disrupt the way that data binding is working. In other cases where the data card requires some customization, you'll be required to unlock a data card so you can edit all aspects of its properties and its child controls.

You can unlock a data card by navigating to the Advanced pane on the right with the data card selected and clicking the button with the lock icon with the caption 'Unlock to change properties'. Once the data card is unlocked, you are free to customize the child controls of a data card by resizing them, moving them or hiding them. You might also want to add new child controls to a data card to achieve a particular user interface experience. It's also possible to start with an entirely blank data card known as a 'custom card' which allows you to add all the child controls from scratch.

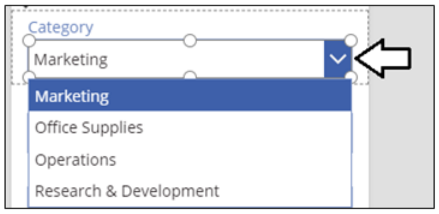
Populating a Dropdown Combobox

- Once a data card is unlocked you can customize it
  - Add formula for **Items** property to populate combo box



```
["Marketing", "Office Supplies", "Operations", "Research & Development"]
```

  - Dropdown list provides better user experience than textbox



Here is a common example of when it is necessary to unlock a data card. Let's say you can change the data card used for a text field to use the Allowed Values data card which provides a user experience with a dropdown menu of predefined choices. When you first update the field to use the Allowed Values data card, all the child controls are locked so you cannot configure the dropdown menu with a set of values.

Once you unlock an Allowed Values data card, you can then write a custom formula for the Items property of the child control which provides the dropdown menu. The screenshot on top in the slide above shows how to write a formula for the Items property to populate the set of choices provided by this dropdown menu. The screenshot on the bottom of the slide shows what the dropdown menu looks like to a user who is a consumer of the app.

Module 01 Lab: Getting Started with PowerApps Studio

Lab Time: 60 minutes

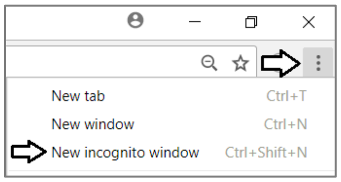
Lab Folder: C:\Student\Modules\01_GettingStarted\Lab

Lab Overview: This lab covers how to get up and running with PowerApps by creating a new Office 365 tenant with trial subscriptions to Office 365, PowerApps and Flow. The act of creating and configuring this new Office 365 tenant will yield an isolated testing and development environment for building and testing PowerApps and Flows. One valuable aspect of creating a new and isolated Office 365 tenant is that you will have tenant-level administrative permissions allowing you to configure the tenant with multiple user accounts for testing your PowerApps and Flows in isolation from any existing Office 365 tenancy.

Exercise 1: Create a new Office 365 Trial Tenant

In this exercise, you will create a new Office 365 tenant which allows you to create up to 25 user accounts with Enterprise E5 trial licenses. Note that the Enterprise E5 trial license provides the benefits of the PowerApps, Flow, Power BI Pro and SharePoint licenses. Being able to create multiple Office 365 user accounts in your Power BI testing environment will be important so that you can test the effects of sharing PowerApps and Flows between users within an organization.

1. Navigate to the Office 365 trial sign up page using an Incognito browser window.
   a) Launch the Chrome browser.
   b) Using the dropdown menu in the upper right, select the command to open a **New incognito window**.

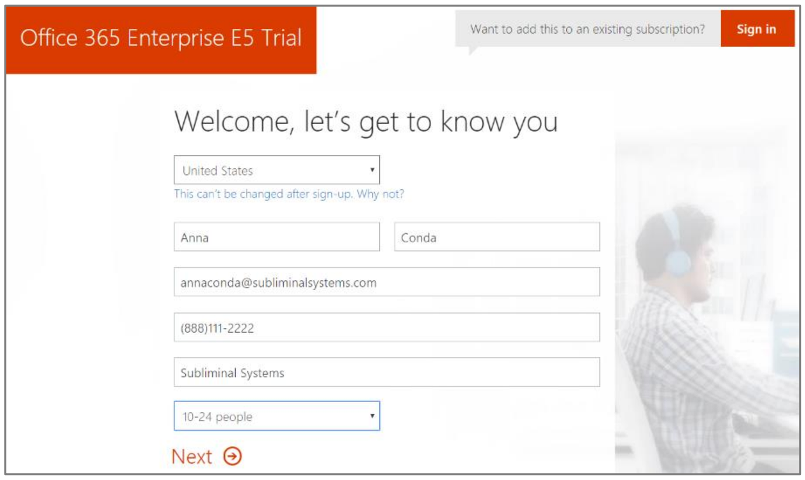


   c) Copy and paste the following URL into the address bar of the incognito window to navigate to the sign up page.

```
https://go.microsoft.com/fwlink/p/?LinkID=698279&culture=en-US&country=US
```

It's not always necessary to sign up for an Office 365 trial account using an incognito window. However, most errors that occur when attempting to sign up are caused by cached browser settings such as residue from an earlier Office 365 trial account. The solution to overcoming most errors when signing up for a trial account is using an incognito window.

2. Fill out the form with your personal information and click **Next**.



Office 365 Enterprise E5 Trial

Want to add this to an existing subscription? Sign in

Welcome, let's get to know you

United States

This can't be changed after sign-up. Why not?

Anna | Conda

annaconda@subliminalsystems.com

(888)111-2222

Subliminal Systems

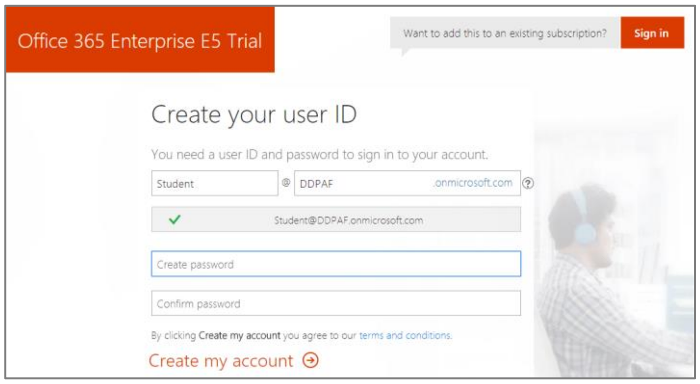
10-24 people

Next

The information you provide on the next page of the signup process will be used to name your new Office 365 tenant.

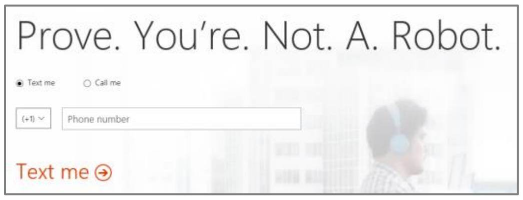
3. On the Create your user ID page…

a) Enter a user name
b) Enter a unique company name (you might have to try a few before you get one that's unique)
c) Enter a password that you will remember.

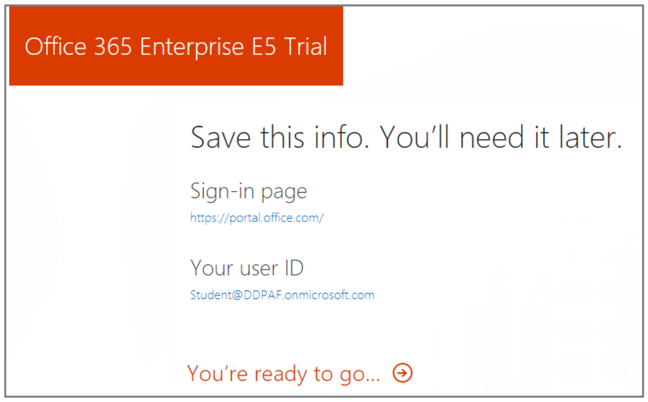

Office 365 Enterprise E5 Trial
Want to add this to an existing subscription?
Sign in
Create your user ID
You need a user ID and password to sign in to your account.
Student
DDPAF
.onmicrosoft.com
Student@DDPAF.onmicrosoft.com
Create password
Confirm password
By clicking Create my account you agree to our terms and conditions.
Create my account

Note that the company name you enter on this page will be used to create the domain name for your new Office 365 trial tenant. For example, if you were to enter a company name of **ddpaf**, it would result in the creation of a new Office 365 tenant within a domain of **ddpaf.onMicrosoft.com**. The user name you enter will be used to create the first user account which will be given administrative rights within the Office 365 trial tenant. If you enter a user name of **Student**, then the email address as well as user principal name for this account will be **Student@ddpaf.onMicrosoft.com**.

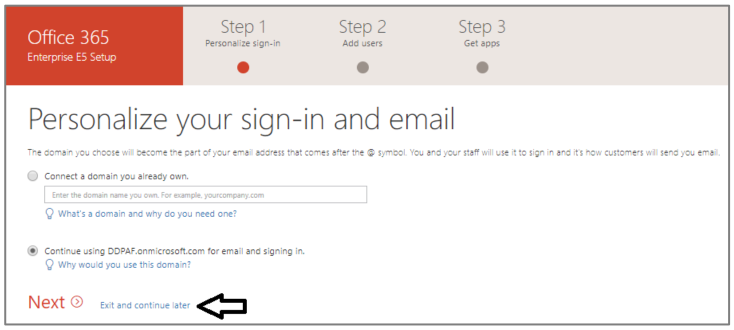
4. Click **Next** to continue to step 3.
5. Complete the validation form in step 3 by proving you are not a robot.
   a) Select the **Text me** option and provide the number of your mobile phone.
   b) When you go through this process, a Microsoft service will send you a text message that contains an access code.
   c) You retrieve the access code form your mobile device and use it to complete the validation process.



6. Once you have completed the validation process, click the **You're ready to go…** link to navigate to the portal welcome page for your new Office 365 trial tenant. Note that you should already be logged on using the user account that was created during the signup process.


Office 365 Enterprise E5 Trial
Save this info. You'll need it later.
Sign-in page
https://portal.office.com/
Your user ID
Student@DDPAF.onmicrosoft.com
You're ready to go...

7. If you are prompted with the **Personalize your sign-in and email**, click the **Exit and continue later** link at the bottom of the page.


Office 365
Enterprise E5 Setup
Step 1
Personalize sign-in
Step 2
Add users
Step 3
Get apps

Personalize your sign-in and email

The domain you choose will become the part of your email address that comes after the @ symbol. You and your staff will use it to sign in and it's how customers will send you email.

Connect a domain you already own.

Enter the domain name you own. For example, yourcompany.com

What's a domain and why do you need one?

Continue using DDPAF.onmicrosoft.com for email and signing in.

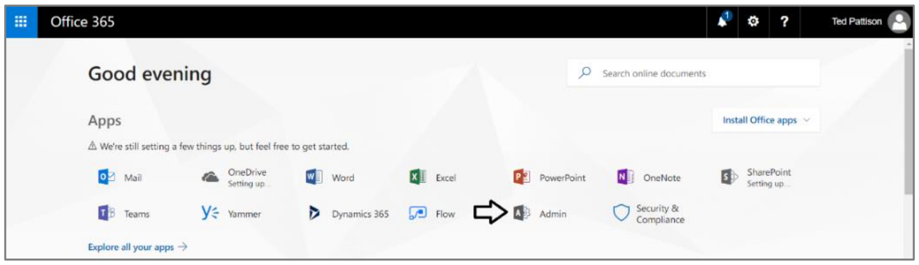
Why would you use this domain?

Next Exit and continue later

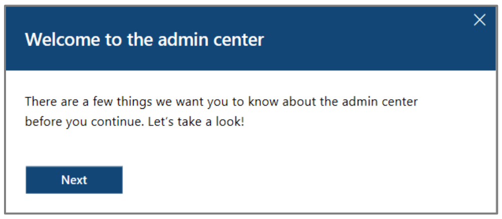
At this point, you have already created your new Office 365 tenant which can support creating up to 25 user accounts with Office 365 Enterprise E5 trial licenses. Note that some Office 365 services within your new Office 365 tenant such as the Office 365 admin center, PowerApps, Flow and Power BI can be accessed immediately. Other services in your Office 365 tenant such as SharePoint Online, OneDrive for Business and Outlook will not be ready immediately and can take some time to provision.

There is no more need to run the browser in incognito mode anymore because it's only required to get through the signup process. You can now return to using a standard browser window. However, it's always a good thing to check to see who you are logged in as because sometimes the browser may log you on using a different Office 365 account you have instead of your new trial account.

8. At this point, you should be located on an Office 365 welcome page. Click the **Admin** tile to go to the Office **365 admin center**.


Office 365
Ted Pattison
Good evening
Search online documents
Apps
Install Office apps
We're still setting a few things up, but feel free to get started.
Mail
OneDrive Setting up...
Word
Excel
PowerPoint
OneNote
SharePoint Setting up...
Teams
Yammer
Dynamics 365
Flow
Admin
Security & Compliance
Explore all your apps

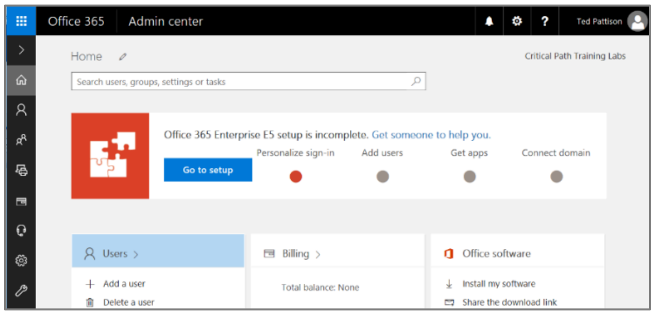
9. If you are presented with the Office 365 admin center welcome dialog, close it by clicking the **X** menu in the upper right corner.


Welcome to the admin center

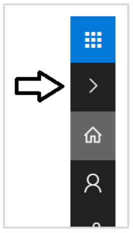
There are a few things we want you to know about the admin center before you continue. Let's take a look!

Next

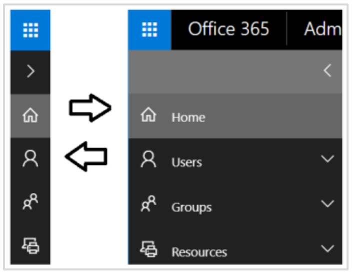
10. Verify that you are able to access the home page of the **Office 365 admin center**.
    a) The following screenshot shows the Office 365 Admin home page.


Office 365 Admin center
Ted Pattison
Home
Critical Path Training Labs
Search users, groups, settings or tasks
Office 365 Enterprise E5 setup is incomplete. Get someone to help you.
Go to setup
Personalize sign-in
Add users
Get apps
Connect domain
Users
Add a user
Delete a user
Billing
Total balance: None
Office software
Install my software
Share the download link

b) Locate the top **Menu** button for the left navigation menu. It's the second button from the top with the arrow icon which sits just beneath the Office 365 App Launcher menu button.

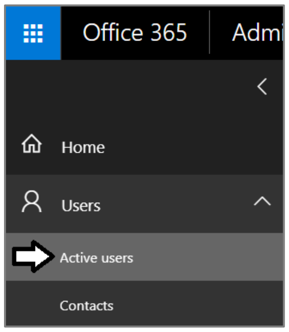


c) Click the top **Menu** button several times and see how it toggles the left navigation between a collapsed and expanded mode.

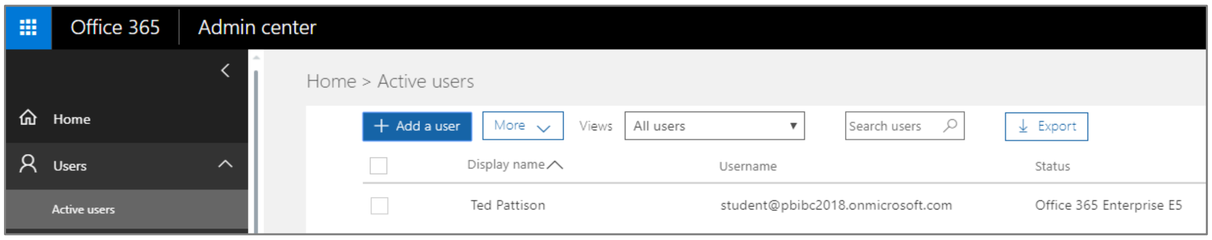


Over the next few steps, you will configure your new Office 365 tenant by creating a secondary user account that you will need later when you begin experimenting with sharing PowerApps and Flows with other users.

11. Make sure you are in the browser at the home page of the Office 365 admin center.
12. Inspect the set of Active Users in the current tenancy.
    a) In the left navigation menu, expand the **Users** node and click **Active Users** to navigate to the **Active Users** page.

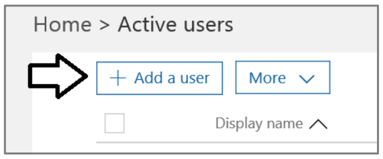


    b) Once the **Active Users** page is displayed, you should be able to verify that the user account you are currently logged on as is the only user account that exists in the current tenancy. Remember that this account has been set up as a Global Administrator to the tenant because it is the account that was used when creating your new Office 365 tenant.

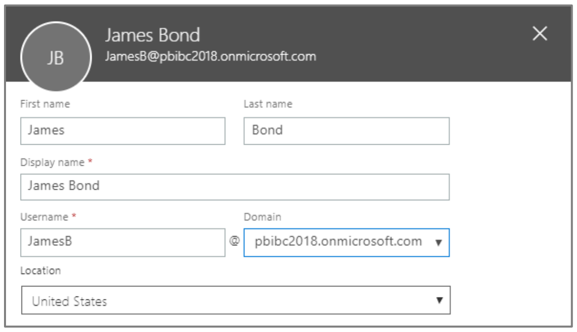


Display name	Username	Status
Ted Pattison	student@pbibc2018.onmicrosoft.com	Office 365 Enterprise E5

13. Create a new user account.
    a) On the **Active Users** page, click the button **Add a user** button to create a new user account



b) Fill in the **Create new user account** form with information for a new user account. When creating this account, you can use any name you would like. These lab instructions will demonstrate this by creating a user account for a person named **James Bond** with a user name and email of **JamesB@pbibc2018.onmicrosoft.com**.


JB
James Bond
JamesB@pbibc2018.onmicrosoft.com

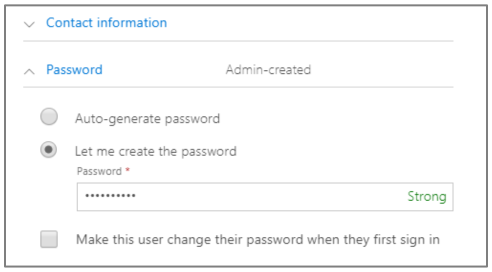
First name: James
Last name: Bond
Display name *: James Bond
Username *: JamesB @ Domain: pbibc2018.onmicrosoft.com
Location: United States

c) Expand **Password** section under **Contact Information** section.

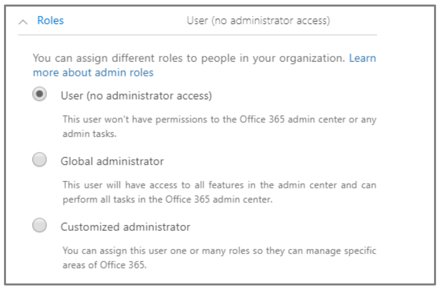
i. Select the option for Let me create the password.

ii. Enter a password of **pass@word1** into the textboxes labeled **Password** and **Retype Password**.

iii. Uncheck the checkbox for the option labeled Make this user change their password when they first sign in.


Contact information
Password Admin-created
○ Auto-generate password
● Let me create the password
Password *: •••••••••• Strong
☐ Make this user change their password when they first sign in

d) Expand the roles section. You do not need to change anything in this section, although you should note that this new user account will be created as a standard user account without any administrator access or privileges.


Roles User (no administrator access)

You can assign different roles to people in your organization. Learn more about admin roles

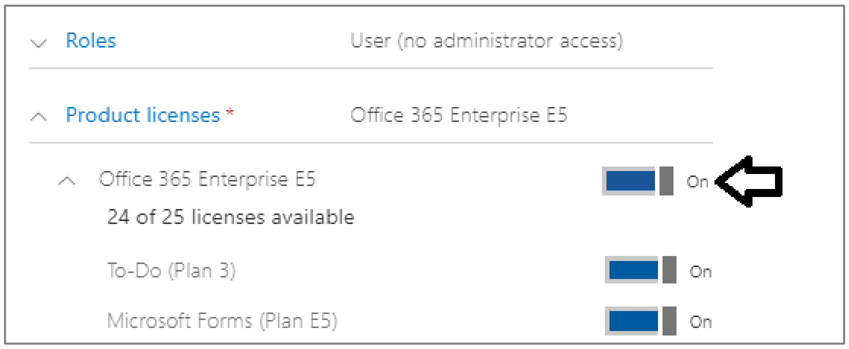
● User (no administrator access)
This user won't have permissions to the Office 365 admin center or any admin tasks.

○ Global administrator
This user will have access to all features in the admin center and can perform all tasks in the Office 365 admin center.

○ Customized administrator
You can assign this user one or many roles so they can manage specific areas of Office 365.

Note that the new account is usually assigned a trial license for **Office 365 Enterprise E5** plan. However, it's a good practice to check and make sure the new user has been assigned a license for **Office 365 Enterprise E5** which includes the **Power BI Pro** license.

e) In the **Product licenses** section, make sure the **Office 365 Enterprise E5** license is set to **On**..

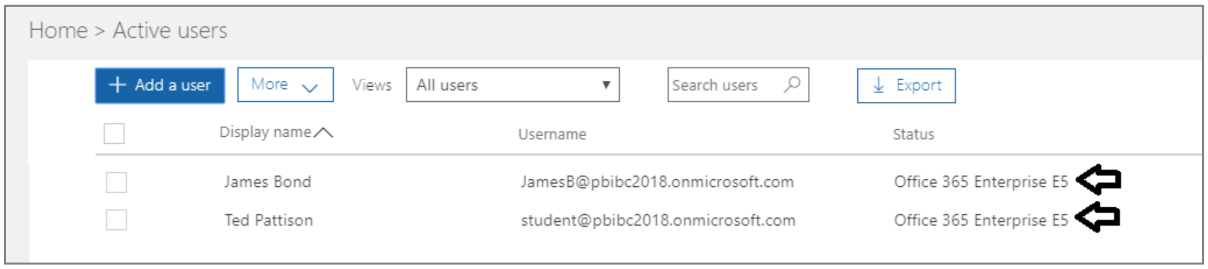


Roles — User (no administrator access)
Product licenses * — Office 365 Enterprise E5
Office 365 Enterprise E5 — On
24 of 25 licenses available
To-Do (Plan 3) — On
Microsoft Forms (Plan E5) — On

f) Click the **Save** button at the bottom of the new user form to create the new user account.

g) When you see the **User was added** message, click **Send email and close** to dismiss the **Add new user** task pane.

h) Verify that the new user account has been created and is displayed along with your primary user account.



Home > Active users

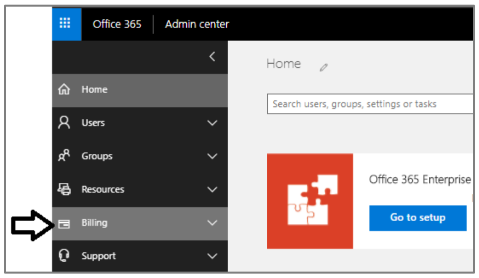
Display name	Username	Status
James Bond	JamesB@pbibc2018.onmicrosoft.com	Office 365 Enterprise E5
Ted Pattison	student@pbibc2018.onmicrosoft.com	Office 365 Enterprise E5

Exercise 2: Create a Trial Subscription for PowerApps Plan 2

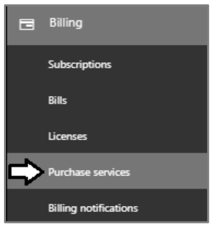
In this exercise, you will configure your new Office 365 tenant by creating a new subscription based on PowerApps Plan 2.

1. Navigate to the home page of the Office 365 Admin center.
2. Create a new subscription for PowerApps Plan 2.

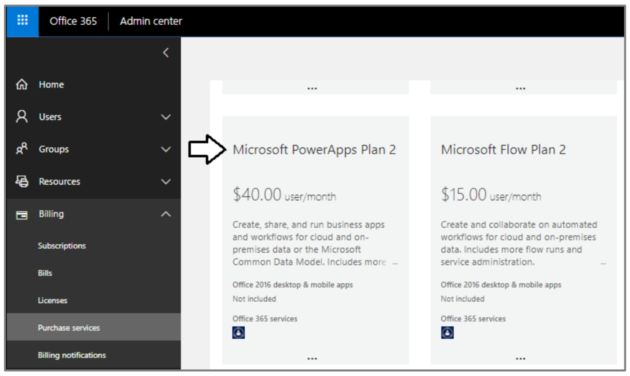
a) Click on **Billing** in the left navigation to expand the menu items underneath.



b) Click on the **Purchase services** navigation link.

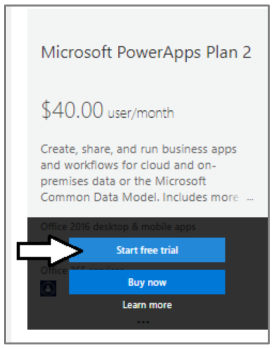


c) Scroll down the page and find the subscription with the name **Microsoft PowerApps Plan 2**.

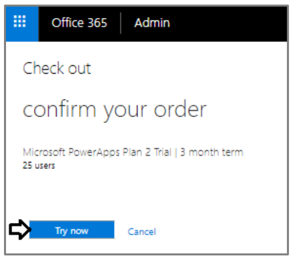

Office 365
Admin center
Home
Users
Groups
Resources
Billing
Subscriptions
Bills
Licenses
Purchase services
Billing notifications
Microsoft PowerApps Plan 2
$40.00 user/month
Create, share, and run business apps and workflows for cloud and on-premises data or the Microsoft Common Data Model. Includes more ...
Office 2016 desktop & mobile apps
Not included
Office 365 services
Microsoft Flow Plan 2
$15.00 user/month
Create and collaborate on automated workflows for cloud and on-premises data. Includes more flow runs and service administration.
Office 2016 desktop & mobile apps
Not included
Office 365 services

d) Hover over the ellipse (…) menu at the center on the bottom to see the flyout menu.

e) Click the **Start free trial** button


Microsoft PowerApps Plan 2
$40.00 user/month
Create, share, and run business apps and workflows for cloud and on-premises data or the Microsoft Common Data Model. Includes more ...
Start free trial
Buy now
Learn more

f) When prompted to confirm your order, click **Try now**.

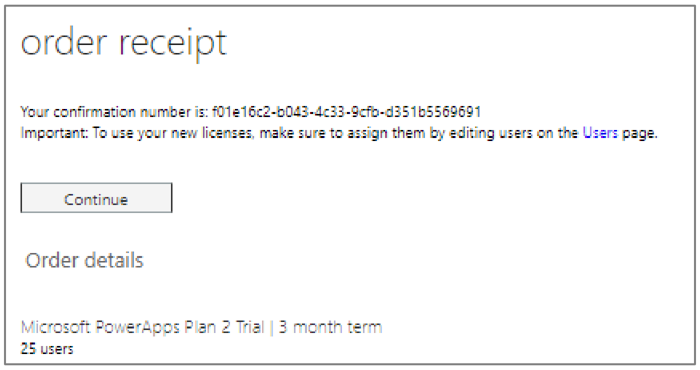


Check out

confirm your order

Microsoft PowerApps Plan 2 Trial | 3 month term
25 users

g) You should see an order receipt to confirm you have created the new trial subscription.



order receipt

Your confirmation number is: f01e16c2-b043-4c33-9cfb-d351b5569691
Important: To use your new licenses, make sure to assign them by editing users on the Users page.

Continue

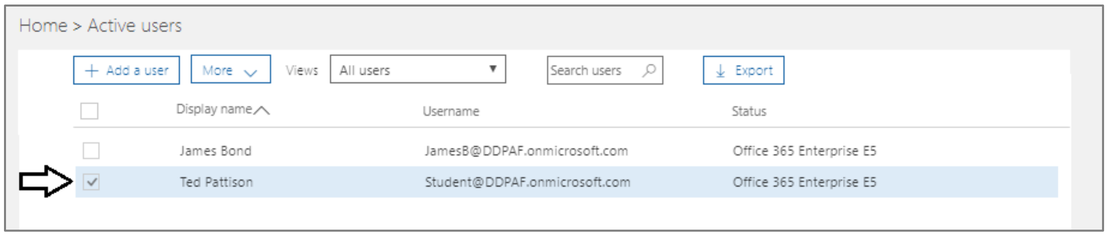
Order details

Microsoft PowerApps Plan 2 Trial | 3 month term
25 users

3. Configure your user account by assigning a PowerApps Plan 2 license.

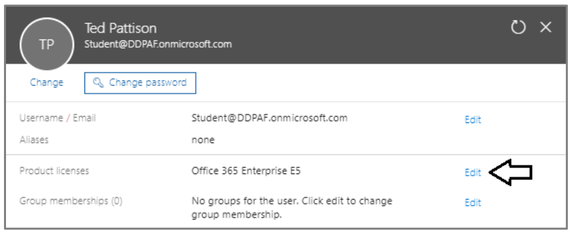
a) Navigate back to the Active Users page in the Office 365 Admin center.

b) Click on your user account to edit it.


Home > Active users

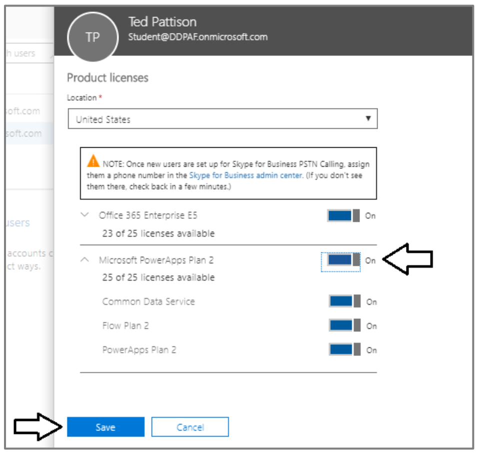
Display name	Username	Status
James Bond	JamesB@DDPAF.onmicrosoft.com	Office 365 Enterprise E5
Ted Pattison	Student@DDPAF.onmicrosoft.com	Office 365 Enterprise E5

c) Click the **Edit** link for **Product licenses**.


Ted Pattison
Student@DDPAF.onmicrosoft.com

Username / Email	Student@DDPAF.onmicrosoft.com	Edit
Aliases	none	
Product licenses	Office 365 Enterprise E5	Edit
Group memberships (0)	No groups for the user. Click edit to change group membership.	Edit

d) Enable the **Microsoft PowerApps Plan 2** subscription and then click Save below.


Ted Pattison
Student@DDPAF.onmicrosoft.com

Product licenses

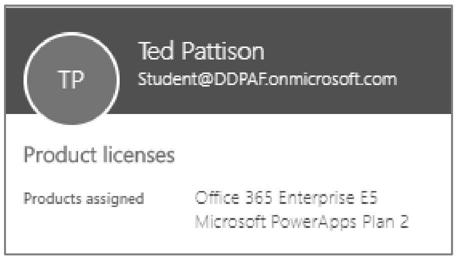
Location *
United States

NOTE: Once new users are set up for Skype for Business PSTN Calling, assign them a phone number in the Skype for Business admin center. (If you don't see them there, check back in a few minutes.)

Office 365 Enterprise E5 — 23 of 25 licenses available	On
Microsoft PowerApps Plan 2 — 25 of 25 licenses available	On
Common Data Service	On
Flow Plan 2	On
PowerApps Plan 2	On

After creating a new subscription for PowerApps Plan 2, it might take 3-5 minutes before it shows up in the Product licenses dialog.

e) You should be able to confirm your user account has been configured with a **Microsoft PowerApps Plan 2** subscription.


Ted Pattison
Student@DDPAF.onmicrosoft.com

Product licenses

Products assigned: Office 365 Enterprise E5, Microsoft PowerApps Plan 2

You will not actually need the Microsoft PowerApps Plan 2 subscription until day 2 in this course. However, plan 2 is needed to connect to data using premium connectors and to design custom solutions which use the Common Data Service for Apps.

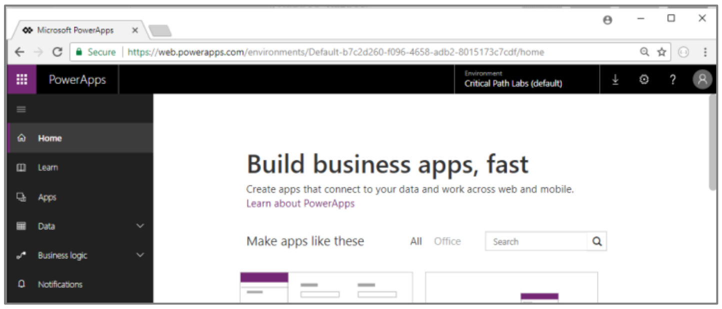
Exercise 3: Create a New App from a PowerApps Template

In this exercise, you will begin working with PowerApps Studio for the web.

1. Launch PowerApps Studio for the Web.
   a) Using a browser such as Chrome, navigate to the following URL and log in using your new Office 365 trial account.

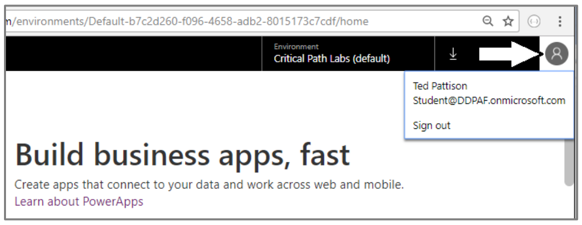
```
https://web.powerapps.com
```

   b) You should now be at the home page of PowerApps Studio as shown in the following screenshot.



Build business apps, fast
Create apps that connect to your data and work across web and mobile.
Learn about PowerApps

   c) Drop down the welcome menu at the top, right of the page and verify you are logged on with your new Office 365 trial account.

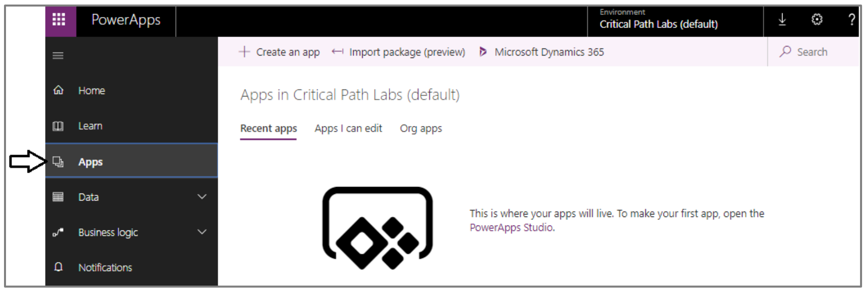

n/environments/Default-b7c2d260-f096-4658-adb2-8015173c7cdf/home
Environment
Critical Path Labs (default)
Sign out

Ted Pattison
Student@DDPAF.onmicrosoft.com

Build business apps, fast
Create apps that connect to your data and work across web and mobile.
Learn about PowerApps

If you are logged in with another account, click the **Sign out** link in the welcome menu and log back in using your new user account.

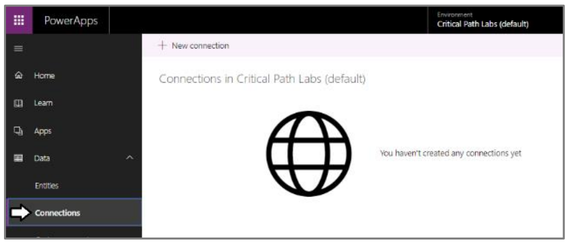
   d) Click the **Apps** link in the left navigation. You should be able to confirm that you currently have no apps.



Apps in Critical Path Labs (default)

This is where your apps will live. To make your first app, open the PowerApps Studio.

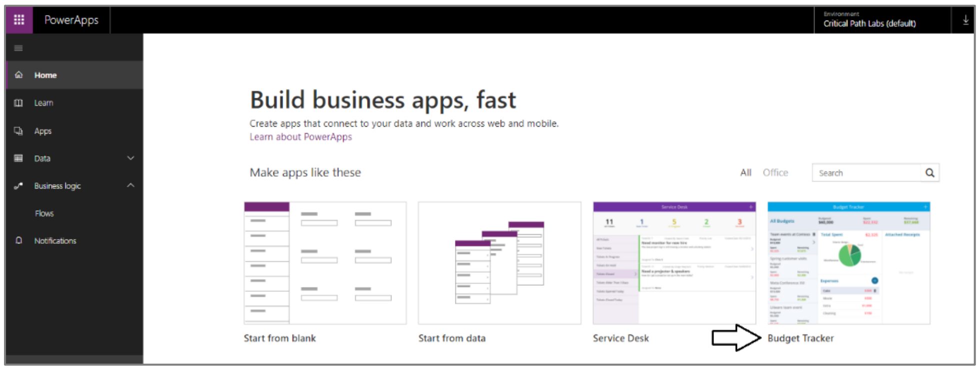
   e) Click on the **Connections** link in the left navigation. You should be able to confirm that you currently have no connections.



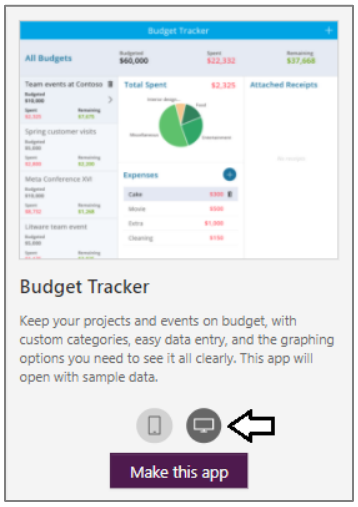
Connections in Critical Path Labs (default)

You haven't created any connections yet

2. Create a new app using the **Budget Tracker** app template.
   a) Return to the home page for PowerApps Studio for the Web.
   b) Locate and click on the tile for the **Budget Tracker** app template.

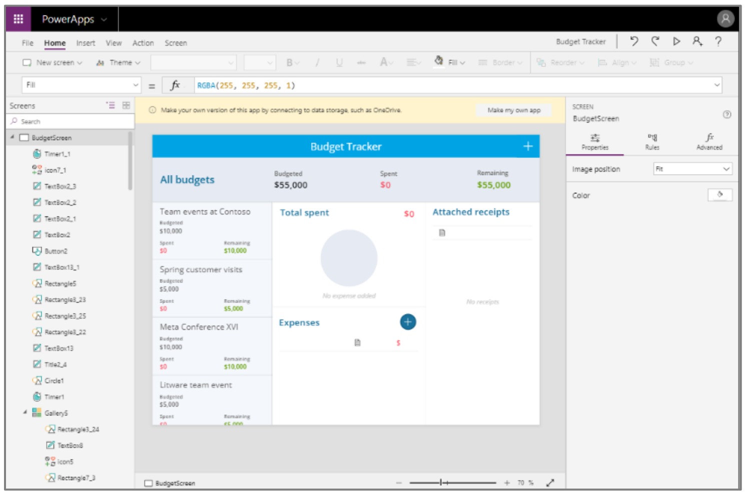


   c) Make sure the icon for a tablet app on the right is selected.


Budget Tracker

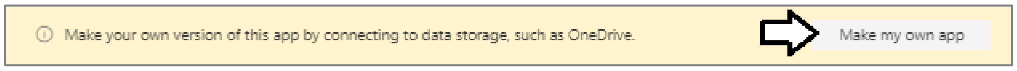
Keep your projects and events on budget, with custom categories, easy data entry, and the graphing options you need to see it all clearly. This app will open with sample data.

   d) Click the **Make this app** button and wait a minute or two until the new app has been created.
   e) The new tablet version of the **Budget Tracker** app should appear as shown in the screenshot below.

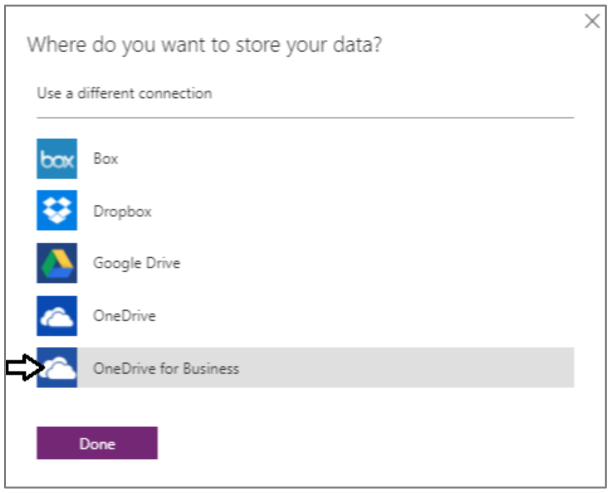


RGBA(255, 255, 255, 1)

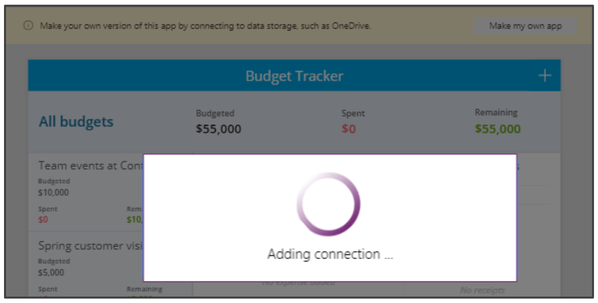
3. Configure the Budget Tracker app to store its data in an excel workbook in OneDrive for Business.
   a) In the yellow bar at the top of the Budget Tracker app, click the **Make my own app** button.


Make your own version of this app by connecting to data storage, such as OneDrive.

   b) When prompted with the Where do you want to store your data? dialog, select OneDrive for Business and click Done.

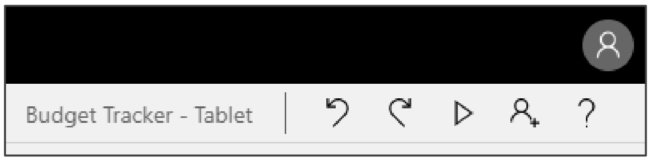

Where do you want to store your data?

   c) Wait while PowerApps Studio creates the new Excel workbook and configure a new connection.

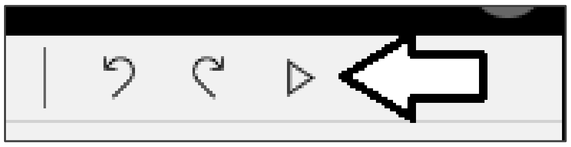


The Budget Tracker app has now been created. It's time to run it and see how it behaves.

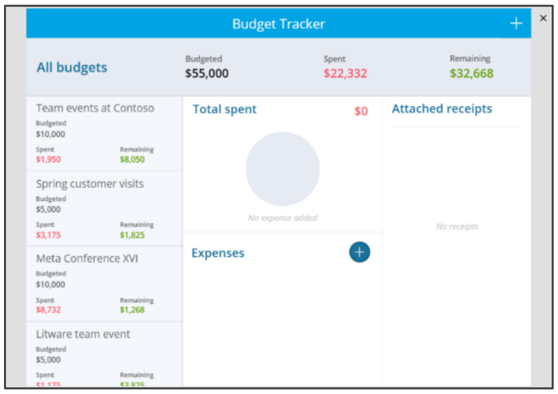
4. Test the Budget Tracker app by starting it up and adding a new budget.
   a) Locate the toolbar in the upper, right-hand side of the PowerApps Studio window.



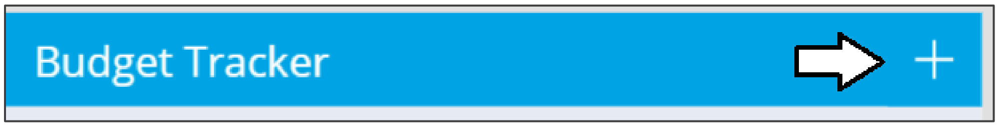
   b) Run the Budget Tracker app by clicking the **Run** button. The **Run** button is the one with the arrow icon.



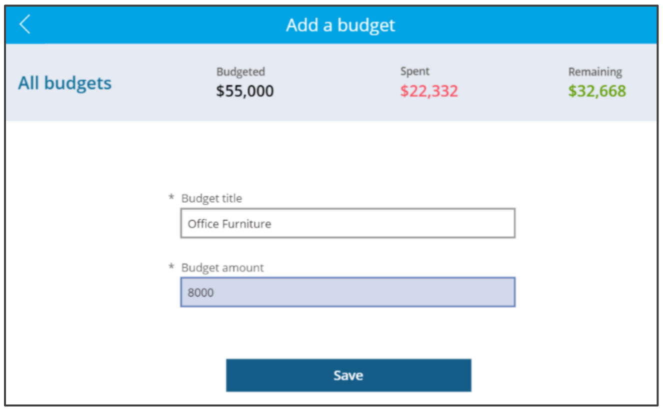
c) The Budget Tracker app should start in the browser and appear like the app shown in the following screenshot.



d) Click the button with the **+** icon in the upper right to add a new budget.



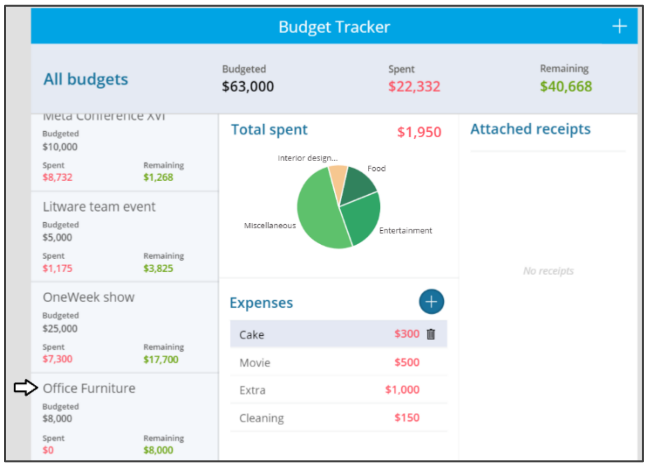
e) In Add a budget dialog, enter a Budget title of Office Furniture and a Budget amount of $8000 and then click Save.



* Budget title: Office Furniture

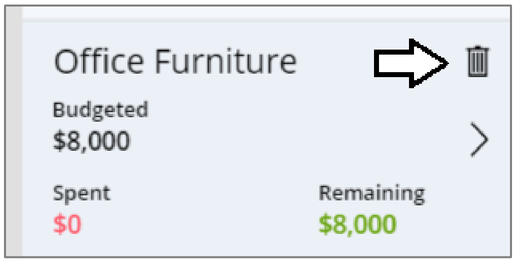
* Budget amount: 8000

f) On the bottom, left, you should be able to verify you can see the new Office Furniture budget you just added.

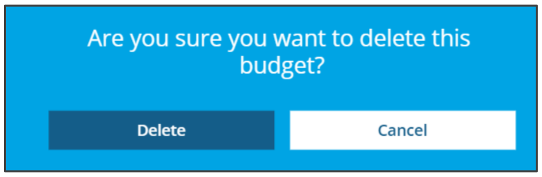


g) Using the mouse, hover over where the app displays **Office Furniture** until you see the icon with the garbage can.

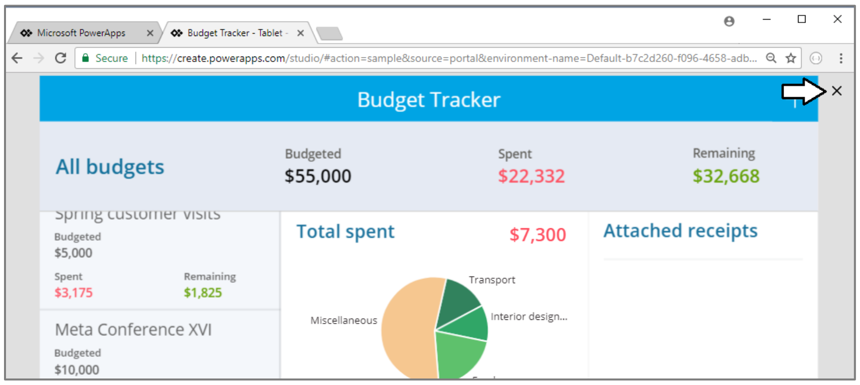
h) Click the button with the garbage can icon to delete the **Office Furniture** budget.



i) When prompted whether you are sure you want to delete the budget, click **Delete** to confirm.



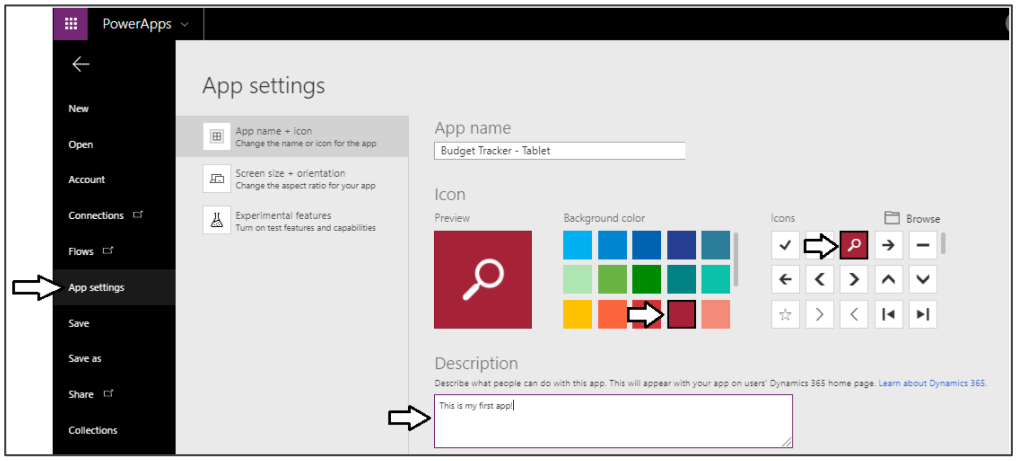
j) Now that you have tested the app, click the **X** icon in the upper right to stop the app.



5. Save the Budget Tracker app.

   a) Click on the **App Settings** link on the right.

   b) Select a **Background color** and an **Icon** for your app and enter a short **Description**.



Budget Tracker - Tablet

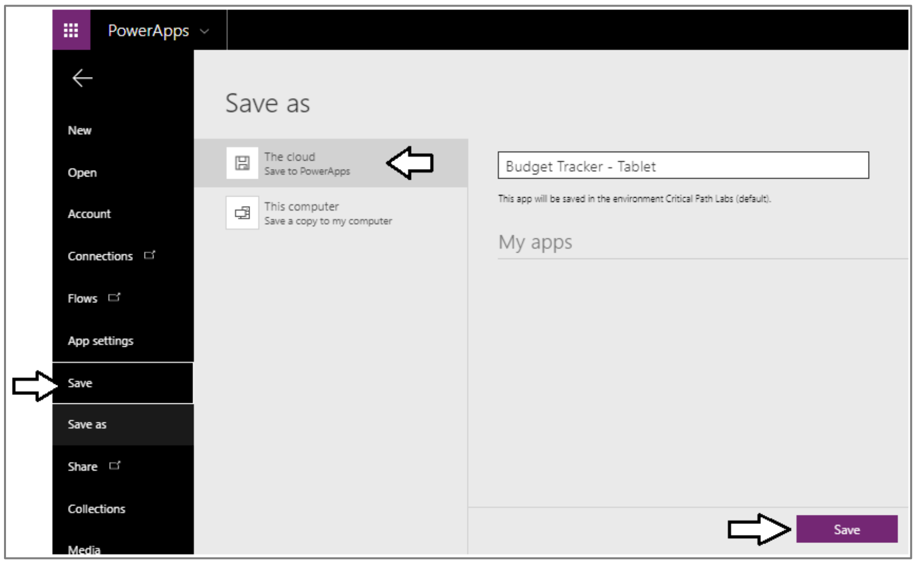
Describe what people can do with this app. This will appear with your app on users' Dynamics 365 home page. Learn about Dynamics 365.

This is my first app!

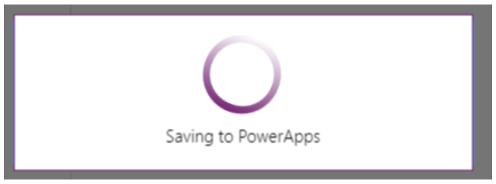
c) Click the **Save** navigation link in the left navigation.

d) Make sure The Cloud is selected in the middle of the screen.

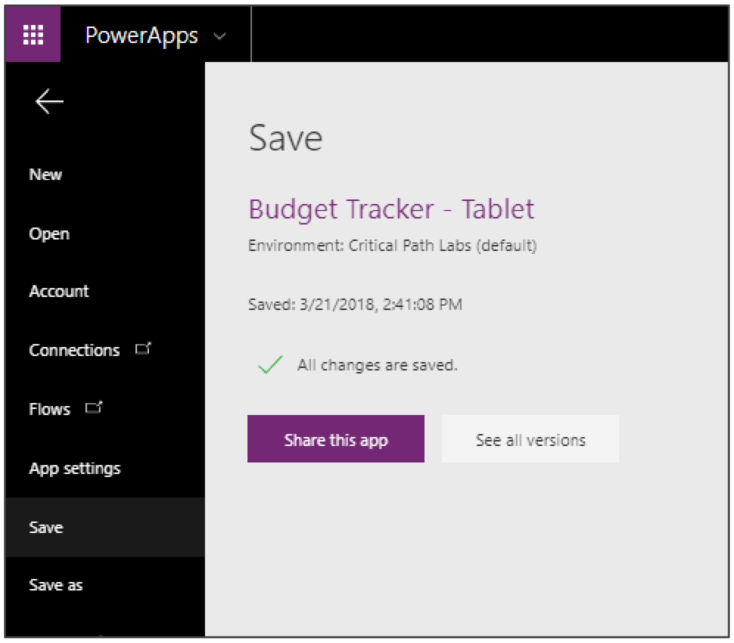
e) Click the **Save** button at the bottom right of the screen.



f) Wait while PowerApps Studio saves the app.

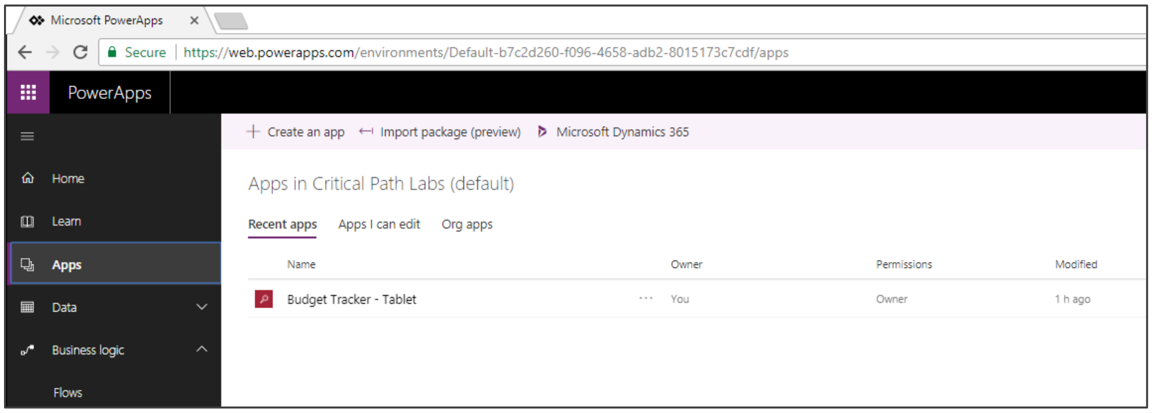


g) After the app has been saved to the cloud, PowerApps Studio should display the screen shown in the following screenshot.



h) Click the button with the back arrow in the left navigation to return to the home page of PowerApps Studio.

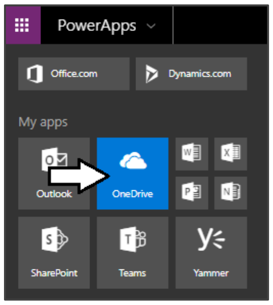
i) Click the **Apps** link in the left navigation and confirm you can see the new app you just created.



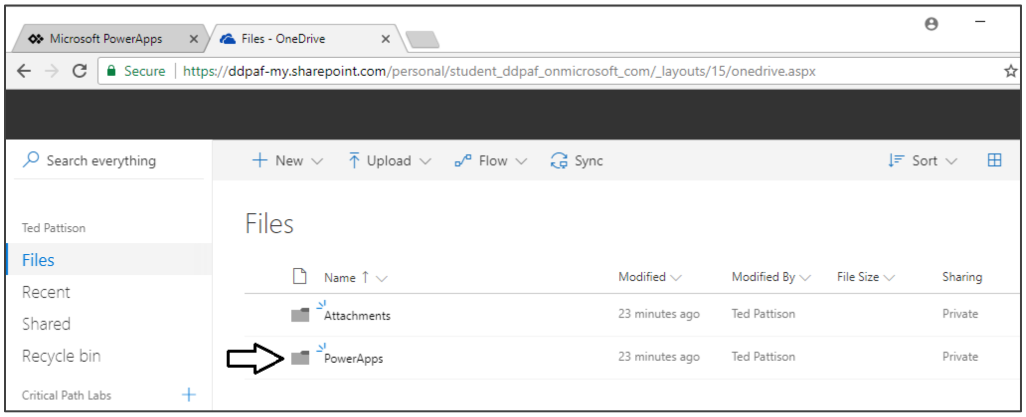
Name	Owner	Permissions	Modified
Budget Tracker - Tablet	You	Owner	1 h ago

6. Inspect the Excel workbook which holds the data for your new Budget Tracker app.

a) Drop down the Office 365 app launcher menu and select **OneDrive** to navigate to your **Files** collection.



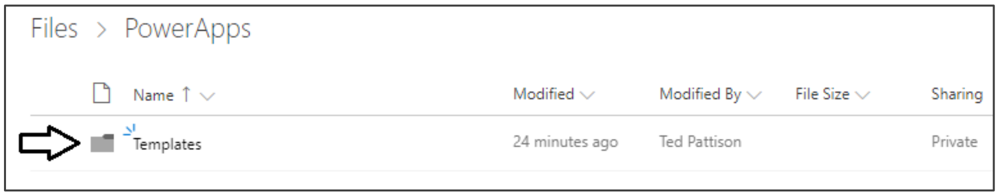
b) When you see the page with the top-level **Files** folder, click the **PowerApps** folder.



Files

Name	Modified	Modified By	File Size	Sharing
Attachments	23 minutes ago	Ted Pattison		Private
PowerApps	23 minutes ago	Ted Pattison		Private

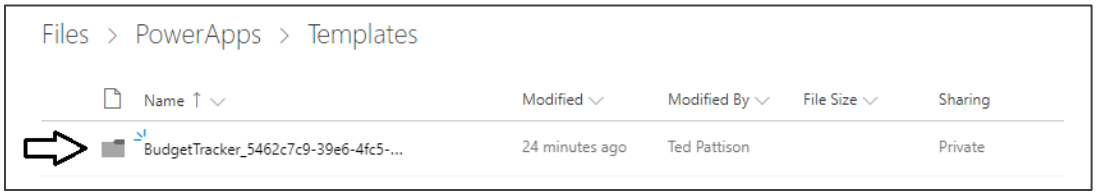
c) Next click on the **Templates** folder to navigate to **Files > PowerApps > Templates**.



Files > PowerApps

Name	Modified	Modified By	File Size	Sharing
Templates	24 minutes ago	Ted Pattison		Private

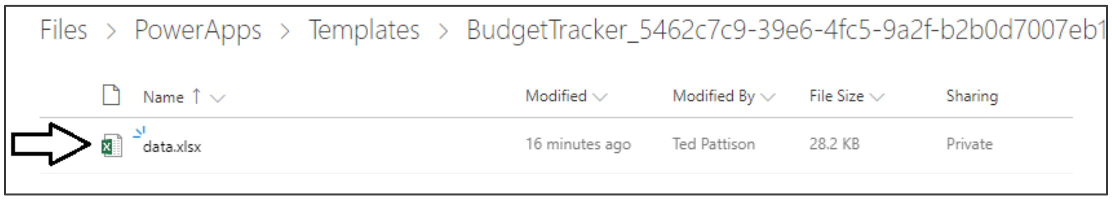
d) Click on the folder whose name begins with **BudgetTracker**.



Files > PowerApps > Templates

Name ↑	Modified	Modified By	File Size	Sharing
BudgetTracker_5462c7c9-39e6-4fc5-...	24 minutes ago	Ted Pattison		Private

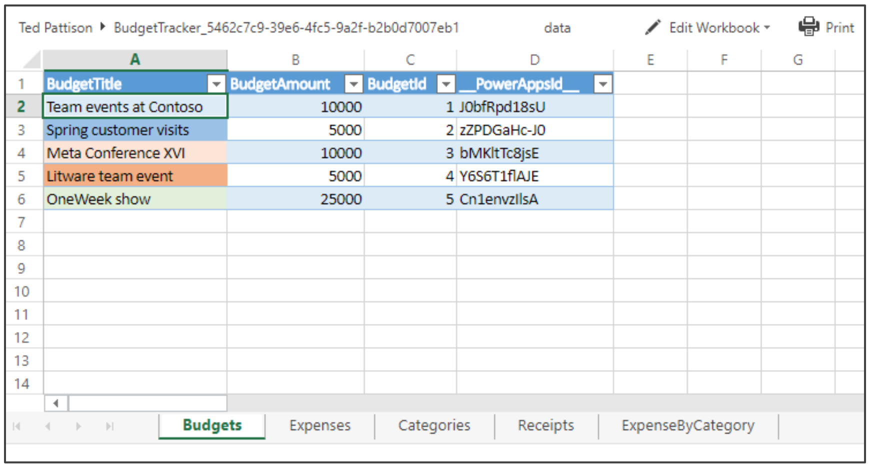
e) You should now see an Excel workbook file named **data.xlsx**. Click on **data.xlsx** to open it in Excel Online.



Files > PowerApps > Templates > BudgetTracker_5462c7c9-39e6-4fc5-9a2f-b2b0d7007eb1

Name ↑	Modified	Modified By	File Size	Sharing
data.xlsx	16 minutes ago	Ted Pattison	28.2 KB	Private

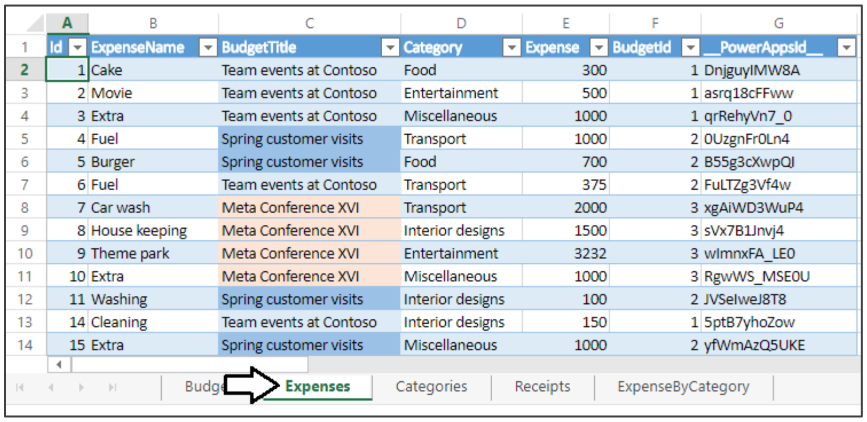
f) Once the Excel workbook opens, you should see the **Budgets** table in the first worksheet.



Ted Pattison ▸ BudgetTracker_5462c7c9-39e6-4fc5-9a2f-b2b0d7007eb1 data

	A	B	C	D
1	BudgetTitle	BudgetAmount	BudgetId	__PowerAppsId__
2	Team events at Contoso	10000	1	J0bfRpd18sU
3	Spring customer visits	5000	2	zZPDGaHc-J0
4	Meta Conference XVI	10000	3	bMKltTc8jsE
5	Litware team event	5000	4	Y6S6T1flAJE
6	OneWeek show	25000	5	Cn1envzIlsA

g) Inspect the four other tables in the other worksheets inside **data.xlsx**.



	A	B	C	D	E	F	G
1	Id	ExpenseName	BudgetTitle	Category	Expense	BudgetId	__PowerAppsId__
2	1	Cake	Team events at Contoso	Food	300	1	DnjguyIMW8A
3	2	Movie	Team events at Contoso	Entertainment	500	1	asrq18cFFww
4	3	Extra	Team events at Contoso	Miscellaneous	1000	1	qrRehyVn7_0
5	4	Fuel	Spring customer visits	Transport	1000	2	0UzgnFr0Ln4
6	5	Burger	Spring customer visits	Food	700	2	B55g3cXwpQI
7	6	Fuel	Team events at Contoso	Transport	375	2	FuLTZg3Vf4w
8	7	Car wash	Meta Conference XVI	Transport	2000	3	xgAiWD3WuP4
9	8	House keeping	Meta Conference XVI	Interior designs	1500	3	sVx7B1Jnvj4
10	9	Theme park	Meta Conference XVI	Entertainment	3232	3	wImnxFA_LE0
11	10	Extra	Meta Conference XVI	Miscellaneous	1000	3	RgwWS_MSE0U
12	11	Washing	Spring customer visits	Interior designs	100	2	JVSelweJ8T8
13	14	Cleaning	Team events at Contoso	Interior designs	150	1	5ptB7yhoZow
14	15	Extra	Spring customer visits	Miscellaneous	1000	2	yfWmAzQ5UKE

The key point is that PowerApps automatically created the Excel workbook named **data.xlsx** to store all the data for the Budget Tracker app. Creating this Excel worksheet and setting up the connection was all done transparently behind the scenes.

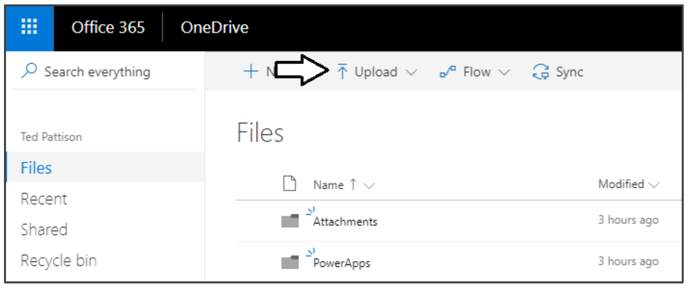
Exercise 4: Create a New App using Data from an Excel Workbook

In this exercise you will create a new app using New App from Data template.

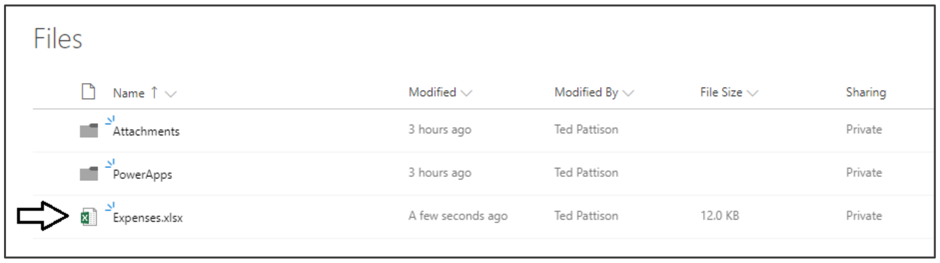
1. Upload the Excel workbook named **Expenses.xlsx** to OneDrive for Business.
   a) Using Windows Explorer, verify that there is an Excel workbook file named **Expenses.xlsx** located at the following path.

```
C:\Student\Modules\01_GettingStarted\Lab\Expenses.xlsx
```

   b) Drop down the Office 365 app launcher menu and select **OneDrive** to navigate to your **Files** collection.
   c) Click the **Upload** button and then select **Expenses.xlsx** to upload this file to OneDrive for Business.



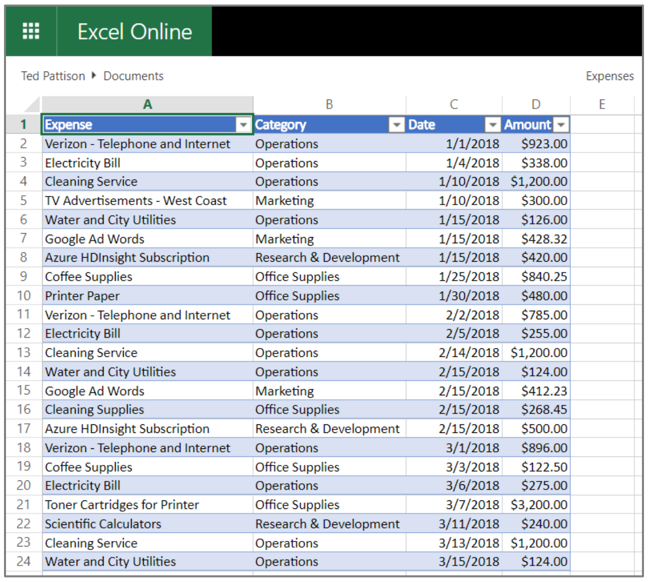
   d) Verify that **Expenses.xlsx** has been uploaded to your **Files** folder.



Files

Name	Modified	Modified By	File Size	Sharing
Attachments	3 hours ago	Ted Pattison		Private
PowerApps	3 hours ago	Ted Pattison		Private
Expenses.xlsx	A few seconds ago	Ted Pattison	12.0 KB	Private

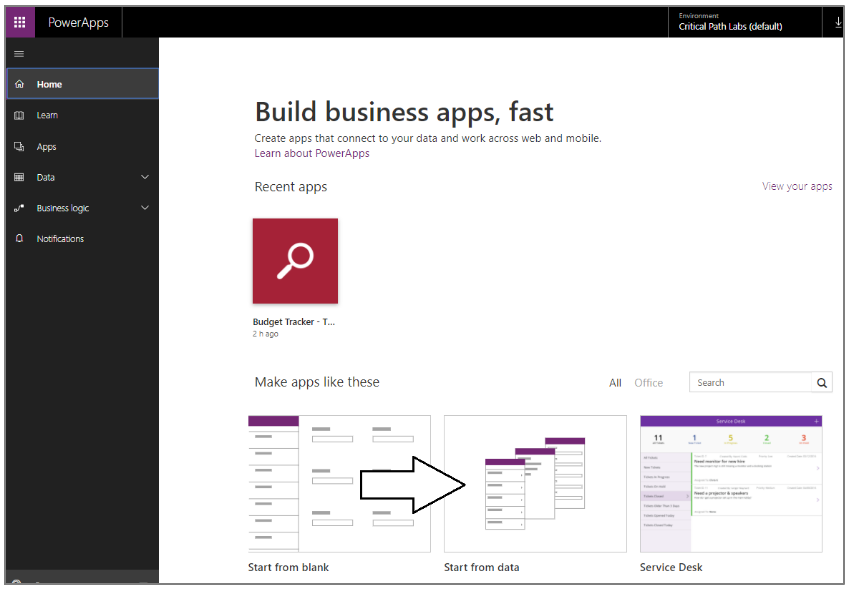
   e) Click on **Expenses.xlsx** to open this worksheet in Excel Online and inspect its contents.



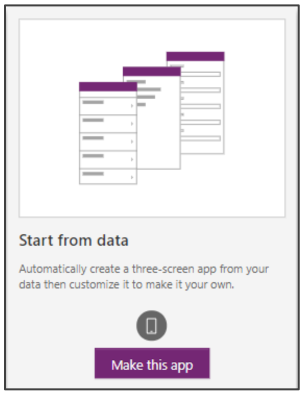
	Expense	Category	Date	Amount
2	Verizon - Telephone and Internet	Operations	1/1/2018	$923.00
3	Electricity Bill	Operations	1/4/2018	$338.00
4	Cleaning Service	Operations	1/10/2018	$1,200.00
5	TV Advertisements - West Coast	Marketing	1/10/2018	$300.00
6	Water and City Utilities	Operations	1/15/2018	$126.00
7	Google Ad Words	Marketing	1/15/2018	$428.32
8	Azure HDInsight Subscription	Research & Development	1/15/2018	$420.00
9	Coffee Supplies	Office Supplies	1/25/2018	$840.25
10	Printer Paper	Office Supplies	1/30/2018	$480.00
11	Verizon - Telephone and Internet	Operations	2/2/2018	$785.00
12	Electricity Bill	Operations	2/5/2018	$255.00
13	Cleaning Service	Operations	2/14/2018	$1,200.00
14	Water and City Utilities	Operations	2/15/2018	$124.00
15	Google Ad Words	Marketing	2/15/2018	$412.23
16	Cleaning Supplies	Office Supplies	2/15/2018	$268.45
17	Azure HDInsight Subscription	Research & Development	2/15/2018	$500.00
18	Verizon - Telephone and Internet	Operations	3/1/2018	$896.00
19	Coffee Supplies	Office Supplies	3/3/2018	$122.50
20	Electricity Bill	Operations	3/6/2018	$275.00
21	Toner Cartridges for Printer	Office Supplies	3/7/2018	$3,200.00
22	Scientific Calculators	Research & Development	3/11/2018	$240.00
23	Cleaning Service	Operations	3/13/2018	$1,200.00
24	Water and City Utilities	Operations	3/15/2018	$124.00

Next, you will create a new app in PowerApps Studio that will read and write to the **Expenses** table in this Excel workbook.

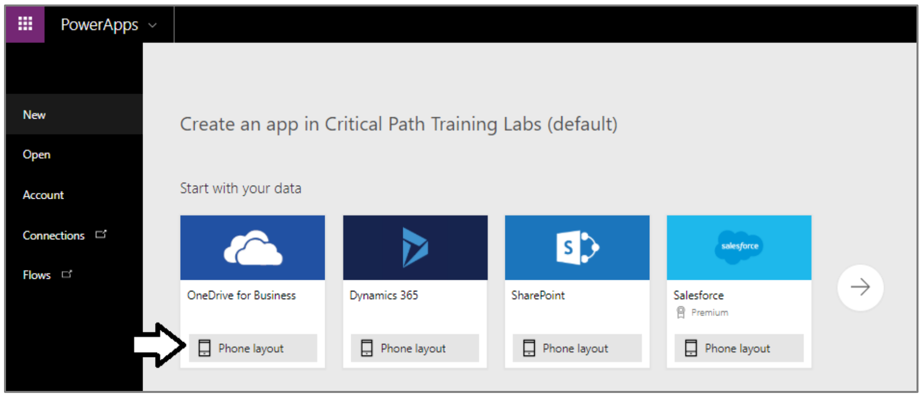
2. Create the new app using the data in the **Expenses.xlsx** workbook.
    a) Navigate back to the home page of PowerApps Studio for the Web at https://web.powerapps.com.
    b) Click on the **Start from data** tile to begin the process of creating the new app.



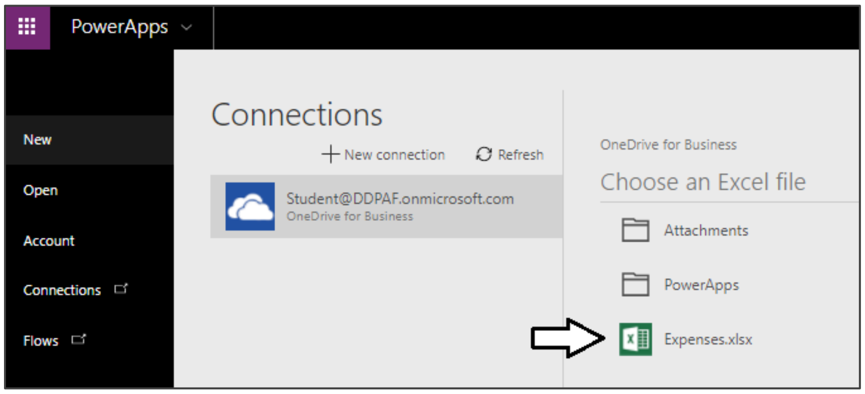
    c) Next, click the **Make this app** button.


Start from data
Automatically create a three-screen app from your data then customize it to make it your own.
Make this app

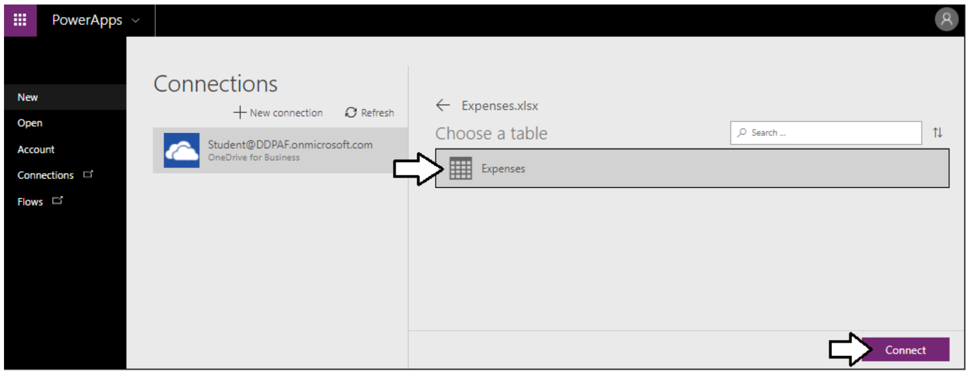
    d) Next, click the **Phone layout** button inside the **OneDrive for Business** tile.



e) When prompted to **Choose an Excel file** on the **Connections** page, click the Excel workbook file named **Expenses.xlsx**.

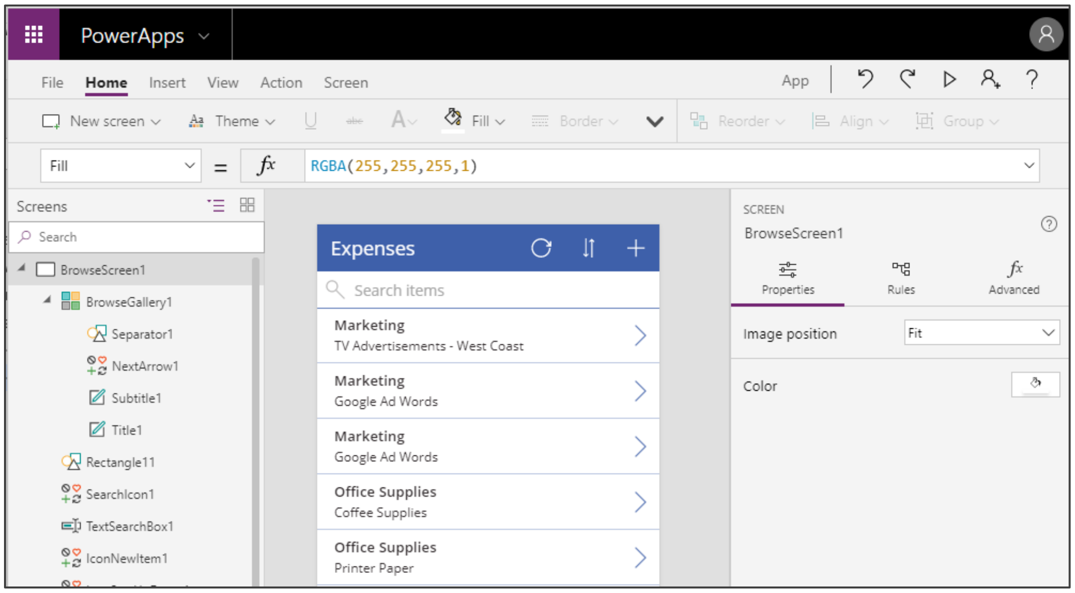


f) When prompted to **Choose a table** on the **Connections** page, select the **Expenses** table and then click **Connect**.



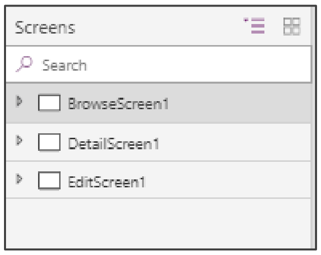
g) Wait while PowerApps Studio generates the starting point for your app.

h) Once PowerApps Studio has created the new app, it should appear as the one in the following screenshot.

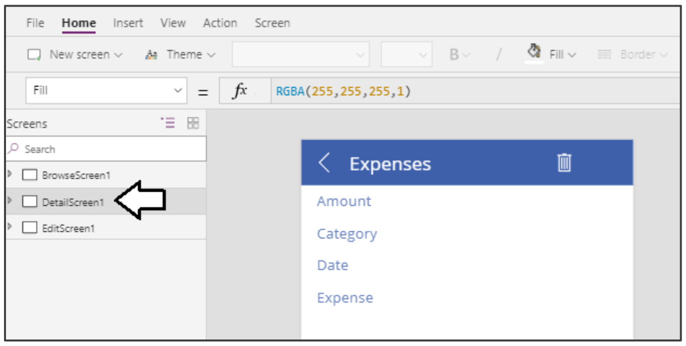


The new app has been created with three different screens. The browse screen shows many expenses at one time. The detail screen and the edit screen are both designed to display only one expense at a time.

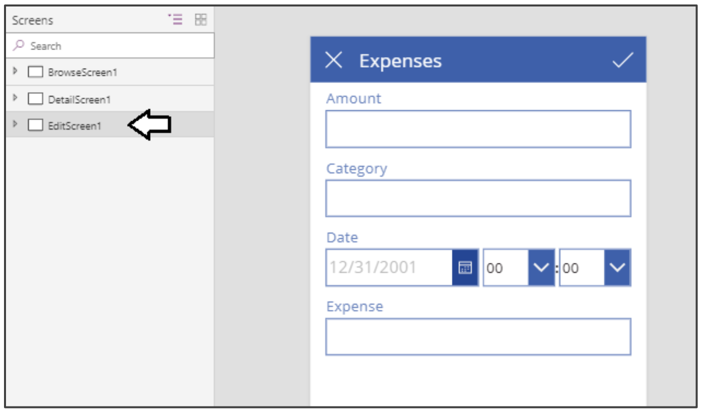
i) Collapse the screen nodes in the left navigation for **BrowserScreen1**, **DetailScreen1** and **EditScreen1**.



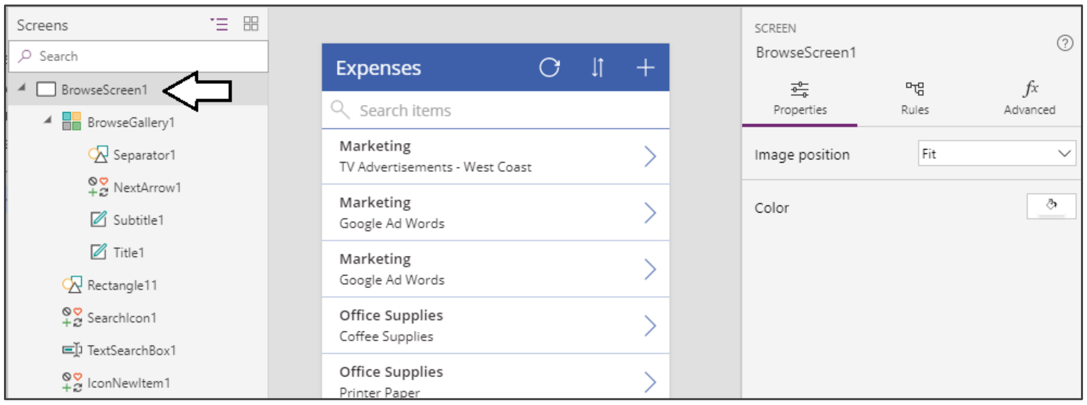
j) Click on **DetailScreen1** in the left navigation to inspect the detail form.



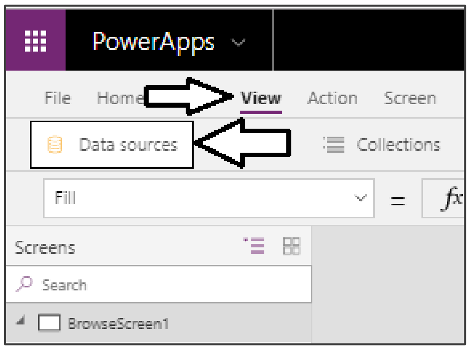
k) Click on **EditScreen1** in the left navigation to inspect the edit form.



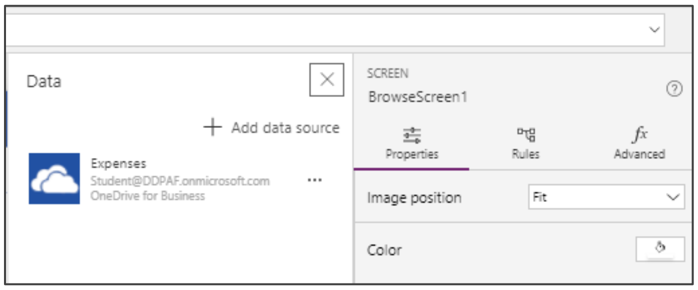
l) Click on **BrowseScreen1** and expand its node in the left navigation.



m) Click the **View** menu and then click the **Data sources** button in the ribbon.

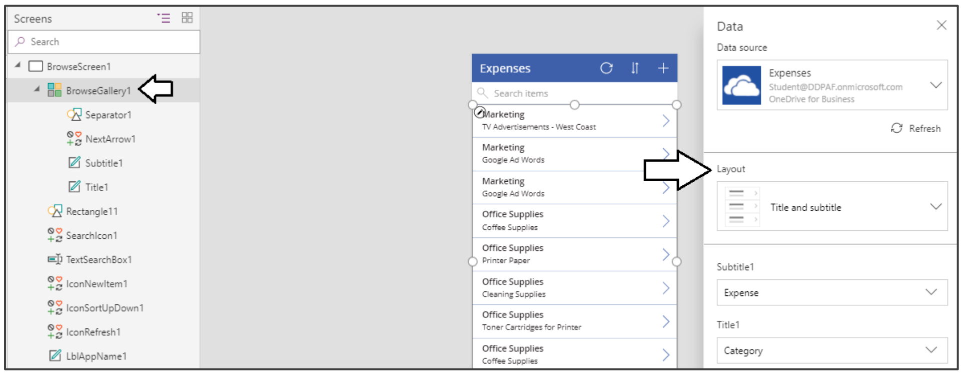


n) The **Data** pane should appear to the left or the **Properties** pane.

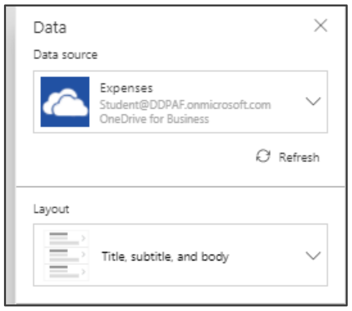


o) Select the **BrowserGallery1** control in the left navigation.

p) When the **BrowseGallery1** control is selected, the **Data** pane should show that the **BrowseGallery1** control has a **Layout** setting of **Title and subtitle**.



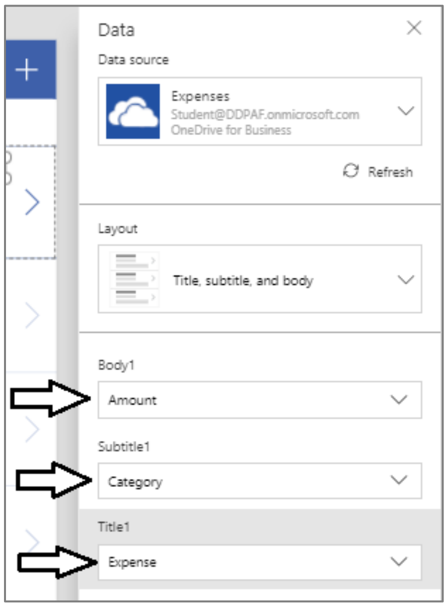
q) Update the Layout setting for BrowseGallery1 to a value of Title, subtitle and body.



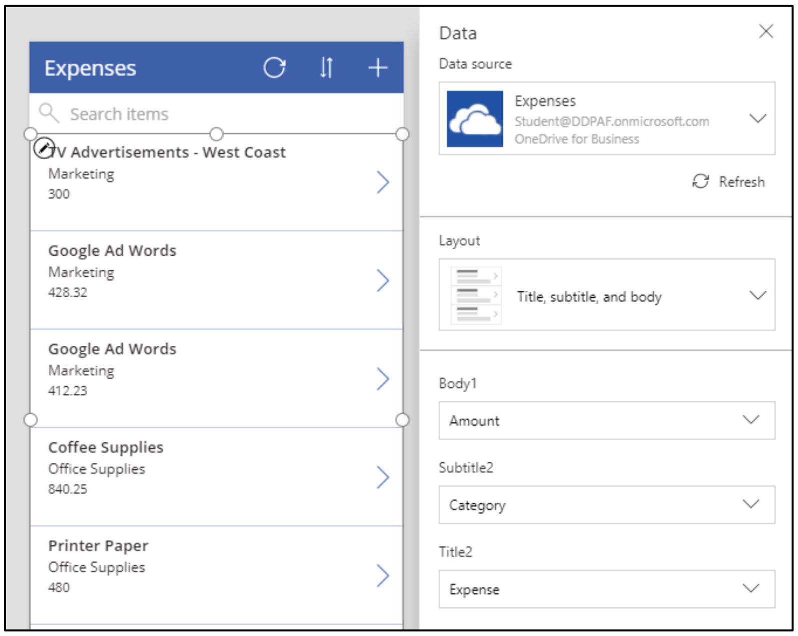
r) Set **Body1** to the **Amount** field.

s) Set **Subtitle1** to the **Category** field.

t) Set **Title1** to the **Expense** field.

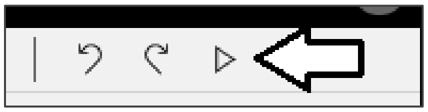


u) The browse screen should now display its fields ordered by **Expense**, **Category** and **Amount**.

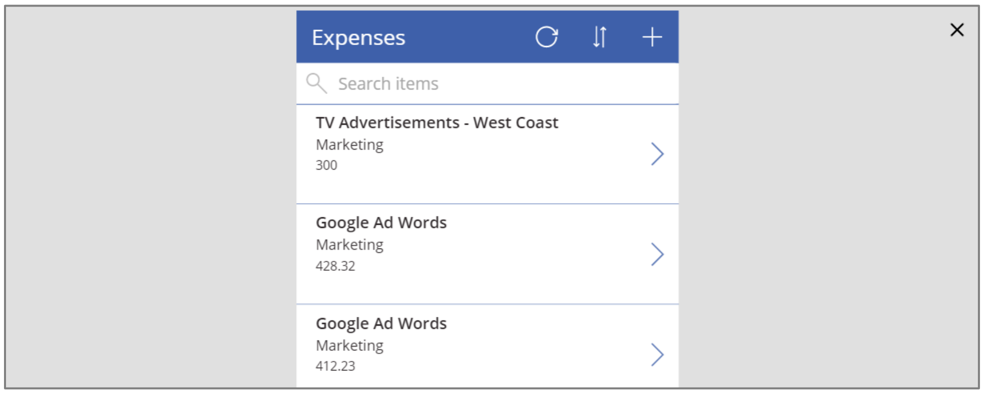


3. Test the app by starting it up and testing the search functionality.

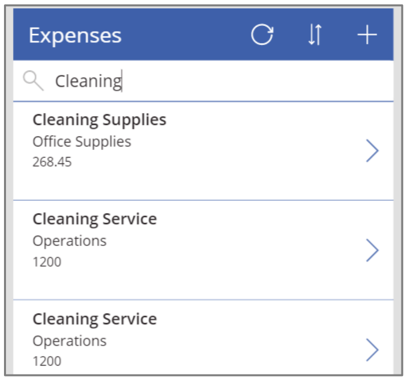
   a) Click the Start button with the arrow icon to launch the app for testing.



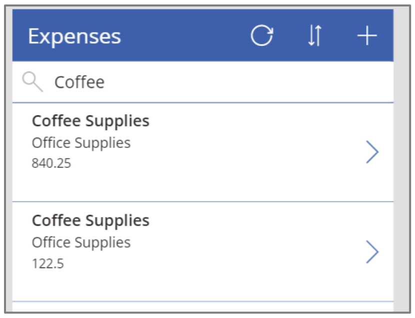
b) The app should start and appear as shown in the following screenshot.



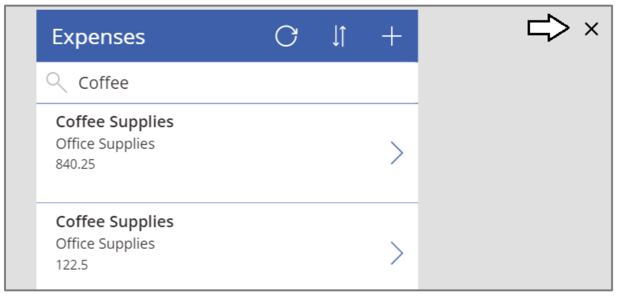
c) Test search functionality by typing the word “Cleaning” in the search box.



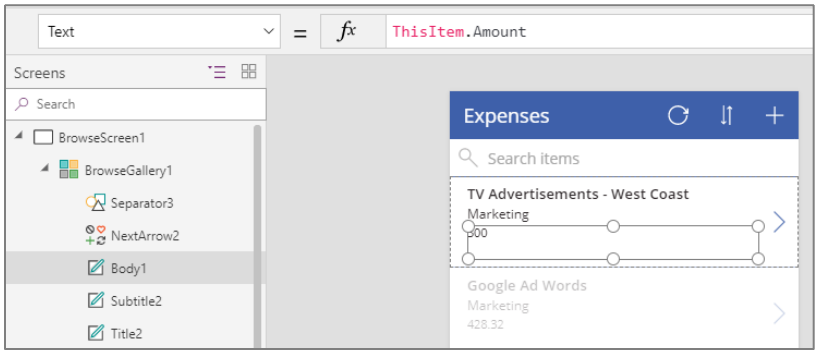
d) Try a different search by typing the word “Coffee” into the search box.



e) Once you have tested the search functionality, stop the app by clicking the button with the **X** icon at the top right.



4. Configure the formatting of the expense **Amount** field.
   a) Select the textbox named **Body1** which displays the **Amount** field for each expense. You should be able to see that the **Text** value of this textbox currently configured with a formula which is **ThisItem.Amount**.

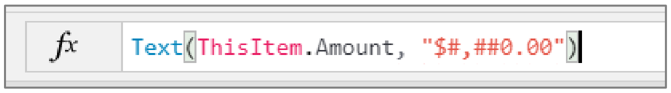


```
ThisItem.Amount
```

   b) Update the **Text** property of the **Body1** textbox with the following formula.

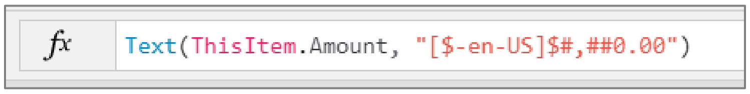
```
Text(ThisItem.Amount, "$#,##0.00")
```

   c) When you update the formula, it will initially match the following screenshot.



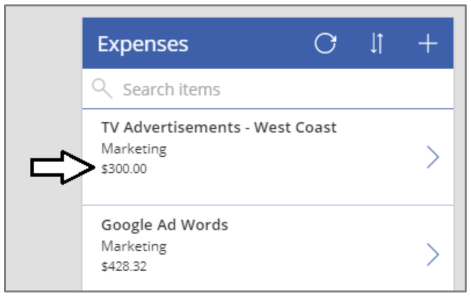
```
Text(ThisItem.Amount, "$#,##0.00")
```

   d) Note that after you update the formula, PowerApps Studio will automatically update the formula to include **[$-en-US]**.

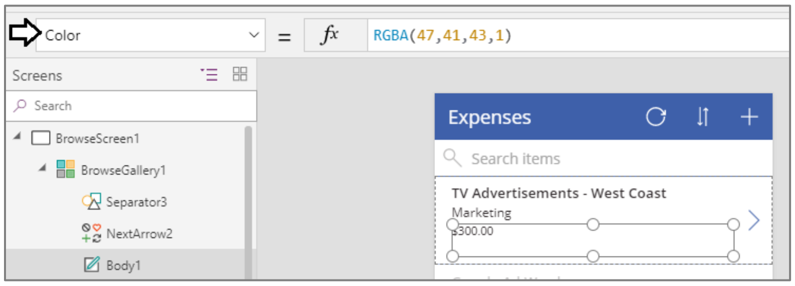


```
Text(ThisItem.Amount, "[$-en-US]$#,##0.00")
```

   e) The **Amount** field should now display its value with currency formatting.



5. Configure the **Color** property of **Body1** to display **Amount** values in red when they are $500 or greater.
   a) With the **Body1** control selected, use the property drop down to display the **Color** property.

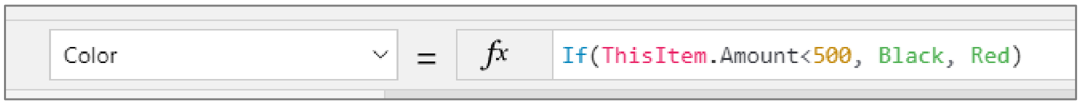


```
RGBA(47,41,43,1)
```

b) Update the **Color** property for **Body1** with the following formula.

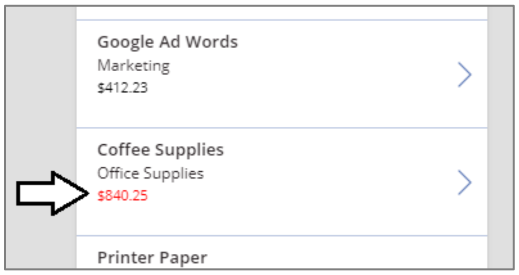
```
If(ThisItem.Amount<500, Black, Red)
```

c) The formula bar should match the following screenshot.



Color = fx If(ThisItem.Amount<500, Black, Red)

d) You should now see that **Amount** values of $500 or greater are displayed with a red font.



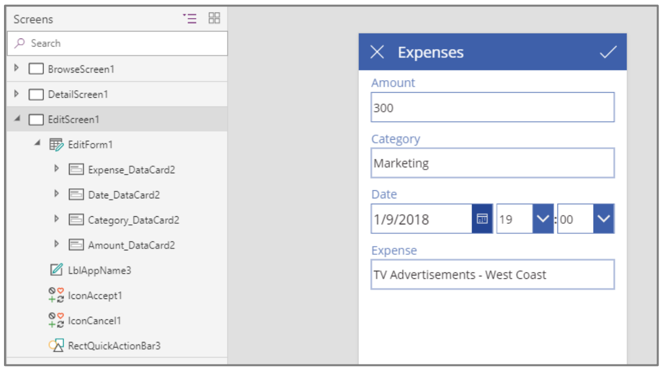
Google Ad Words
Marketing
$412.23

Coffee Supplies
Office Supplies
$840.25

Printer Paper

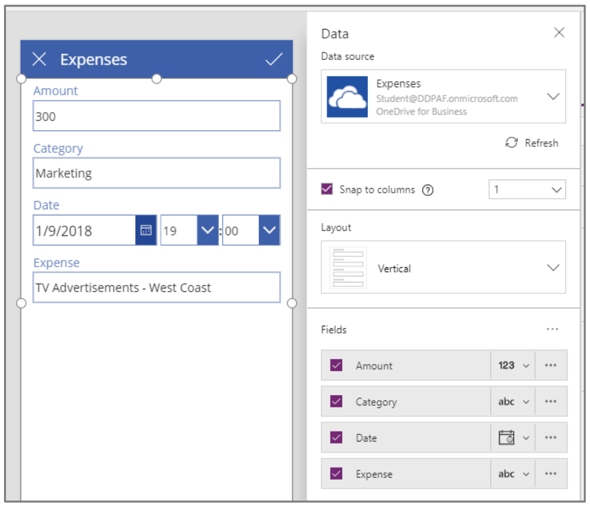
6. Modify the edit screen to streamline data entry for new expenses.

a) Using the left navigation, move to the edit screen.



Amount: 300
Category: Marketing
Date: 1/9/2018 19 : 00
Expense: TV Advertisements - West Coast

b) Display the **Data** pane so you can see the Fields collection of the edit form. At this point, the fields are sorted alphabetically.

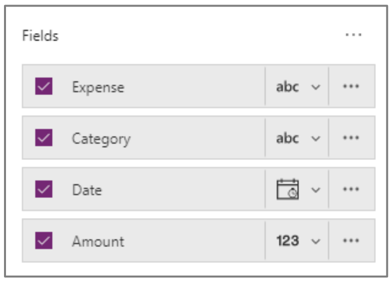


Amount: 300
Category: Marketing
Date: 1/9/2018 19 : 00
Expense: TV Advertisements - West Coast

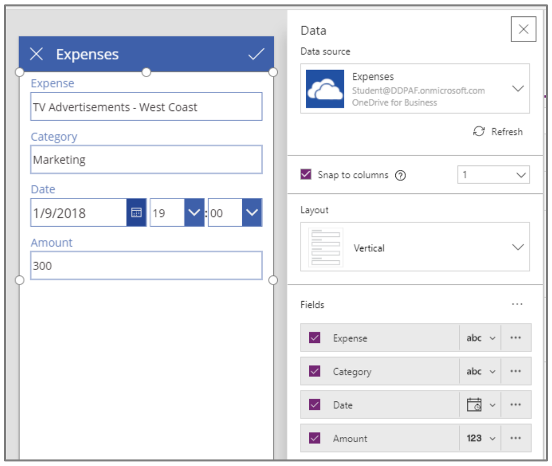
Snap to columns: ✓ 1
Layout: Vertical

Fields:
- ✓ Amount 123
- ✓ Category abc
- ✓ Date
- ✓ Expense abc

c) Using the mouse, rearrange the fields by moving **Expense** to the top followed by **Category**, **Date** and then **Amount**.



d) The edit screen should now display its fields using the new sort order.

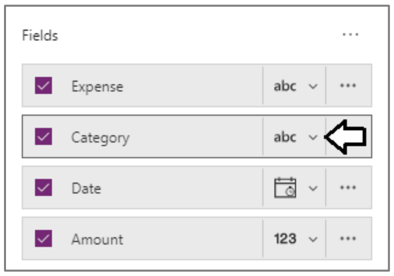


Expense
TV Advertisements - West Coast
Category
Marketing
Date
1/9/2018 19 00
Amount
300

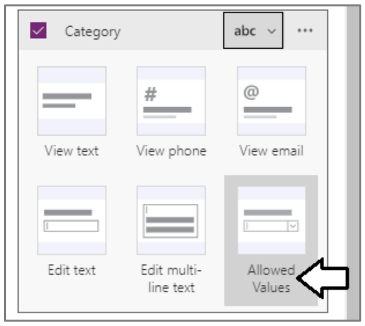
Expenses
Student@DDPAF.onmicrosoft.com
OneDrive for Business

7. Update the data card for the **Category** field to provide a dropdown list with allowed values.

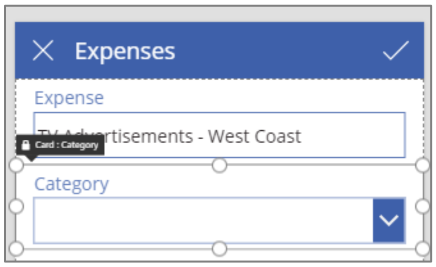
a) Drop down the menu with the **abc** icon to the right of the **Category** field.



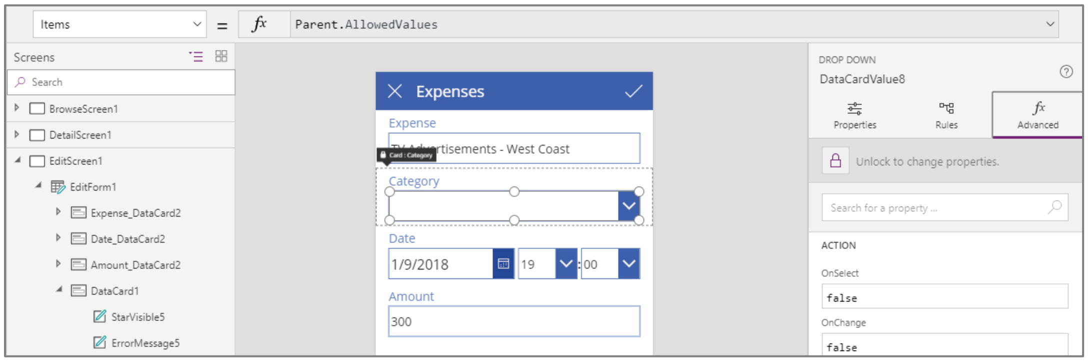
b) Select a control type of **Allowed Values**.



c) The control which displays the **Category** field should change to a dropdown menu.

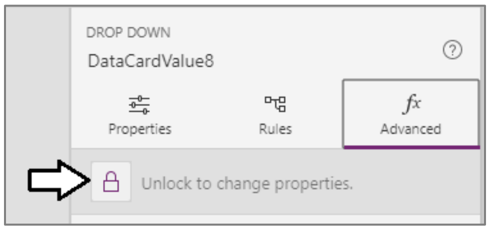


d) Select the dropdown menu and examine the **Items** property in the formula bar.


Items = *fx* `Parent.AllowedValues`

You will notice that the formula bar is read-only for the Items property because the data card is locked by default.

e) In the Advanced pane, click the Unlock to change properties button.

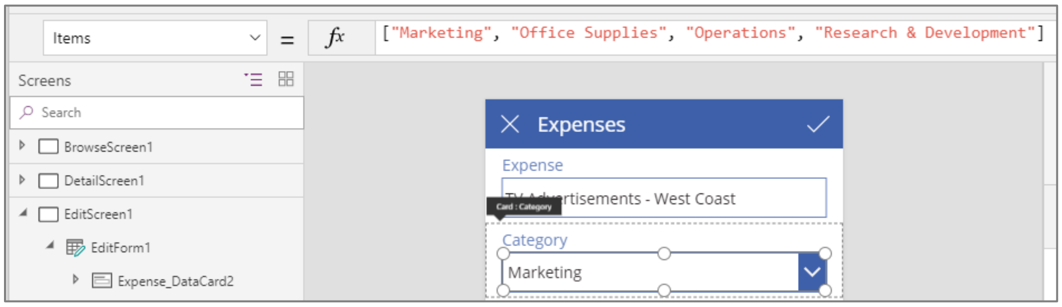


Note that the user interface experience might seem a bit strange when you click the **Unlock to change properties** button. At first it seems like nothing is happening. However, after a few seconds you should see that he **Items** property become editable.

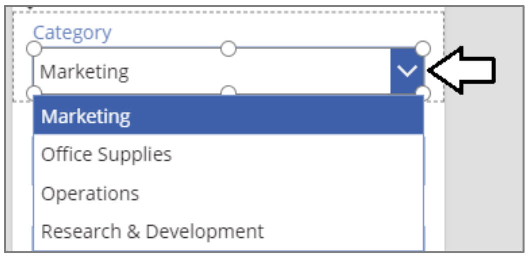
f) Update the **Items** property of the dropdown list with the following formula.

```
["Marketing", "Office Supplies", "Operations", "Research & Development"]
```

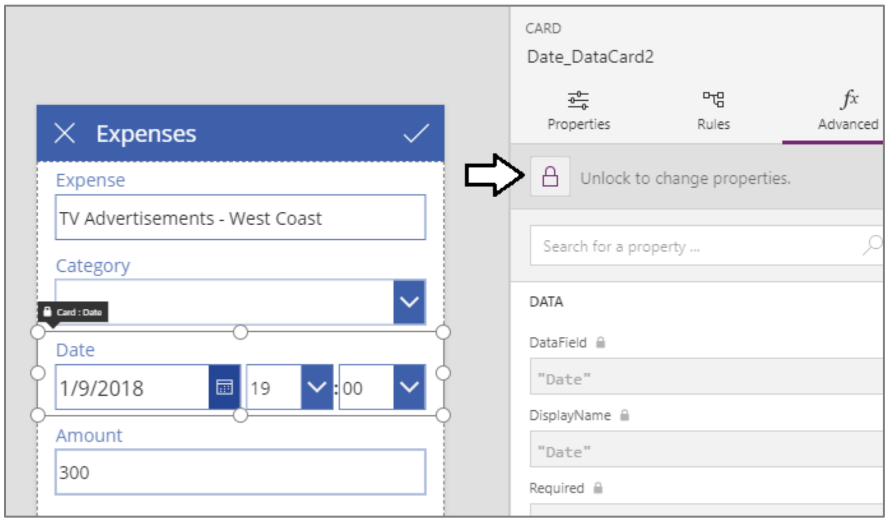
g) The formula bar should match the following screenshot.


Items = *fx* `["Marketing", "Office Supplies", "Operations", "Research & Development"]`

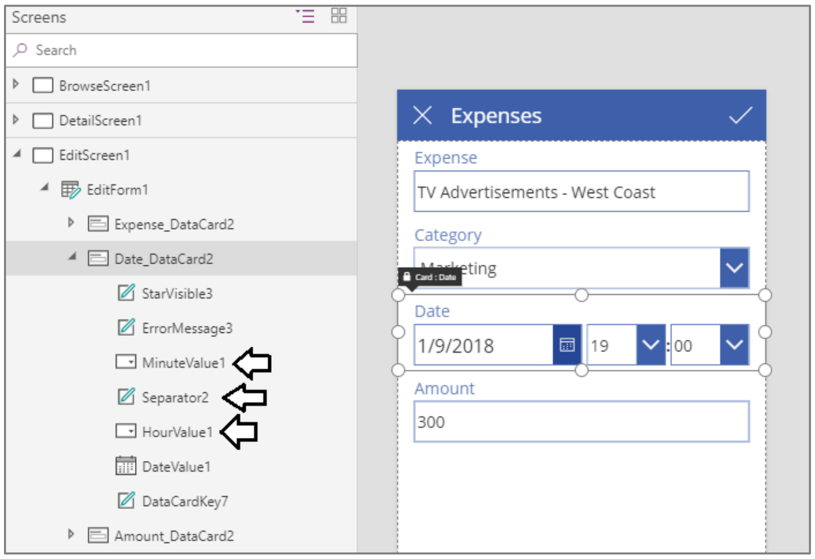
h) You should be able to test the dropdown list and verify that it provides four allowed values.



8. Update the data card for the **Date** field to make it a date-only.
   a) Show the **Data** pane if it is not already showing.
   b) Select the data card for the **Date** field.
   c) In the **Advanced** pane, click the **Unlock to change properties** button for the data card for the **Date** field.

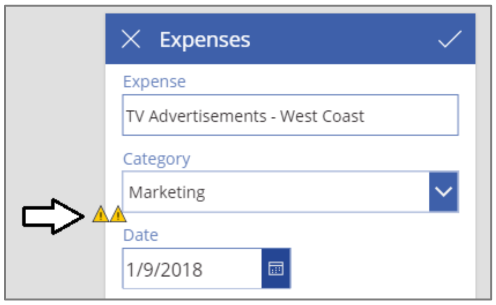


   d) Using the left navigation, select and delete the controls named **MinuteValue1**, **Seperator2** and **HourValue1**.

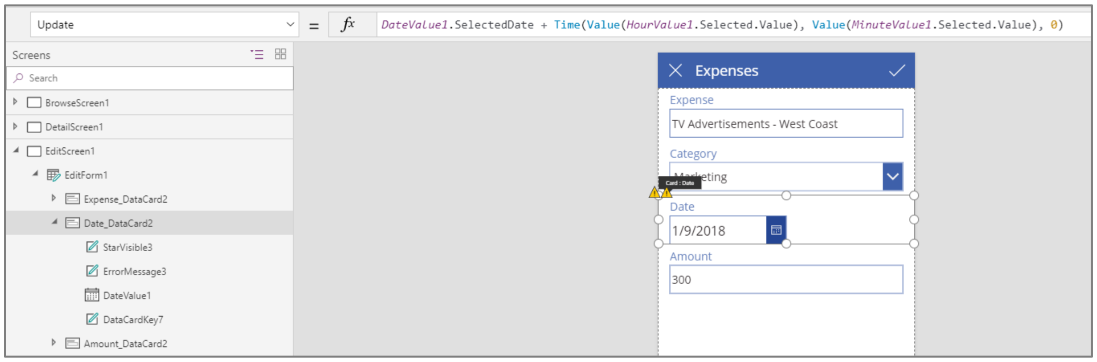


   e) After deleting **MinuteValue1** and **HourValue1**, you will notice formula errors due to referencing deleted controls.

f) Click on the error icon with the yellow triangle to the left.



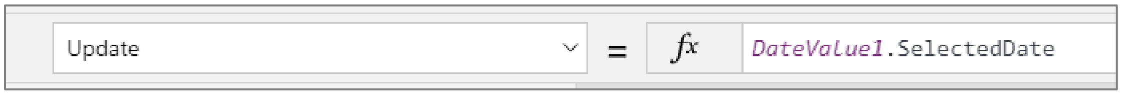
g) At this point, you should see the formula for the **Update** property in the formula bar.


Update = DateValue1.SelectedDate + Time(Value(HourValue1.Selected.Value), Value(MinuteValue1.Selected.Value), 0)

h) Replace the existing **Update** formula with the following formula to remove references to **HourValue1** and **MinuteValue1**.

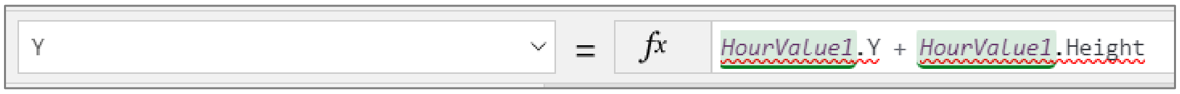
```
DateValue1.SelectedDate
```

i) The formula for the **Update** property of the data card should now appear as the formula shown in the following screenshot.

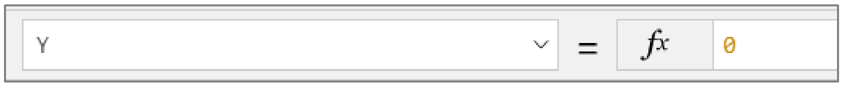

Update = DateValue1.SelectedDate

j) Click on the one remaining yellow triangle to show the other formula error.

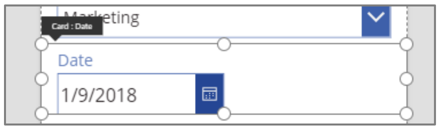
k) You should see the **Y** property of **ErrorMessage3** contains references to the deleted control named **HourValue1**.


Y = HourValue1.Y + HourValue1.Height

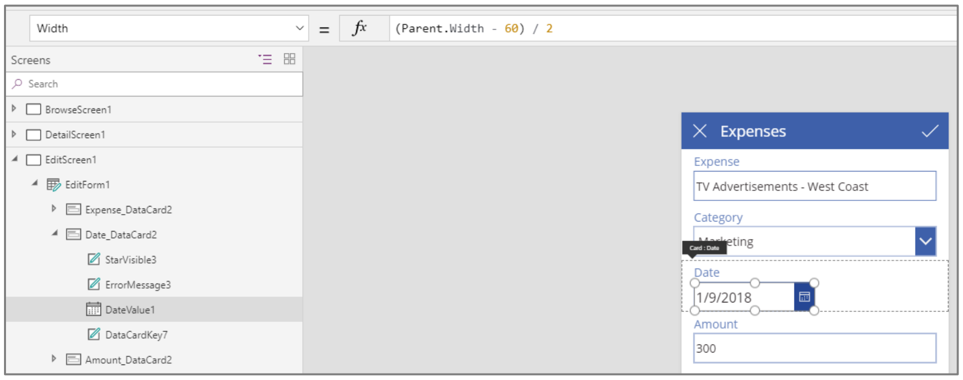
l) Replace the existing formula for the **Y** property with a value of **0** as shown in the following screenshot.


Y = 0

m) At this point, you should no longer see any error indicators.



n) Select he **DateValue1** control and examine its **Width** property.


Width = (Parent.Width - 60) / 2

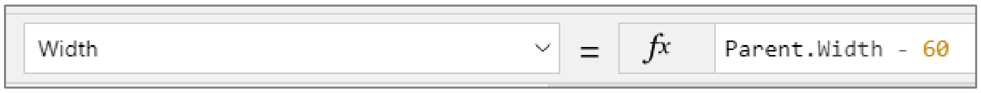
o) You should see that the formula of the **Width** property has the following value.

```
(Parent.Width – 60) / 2
```

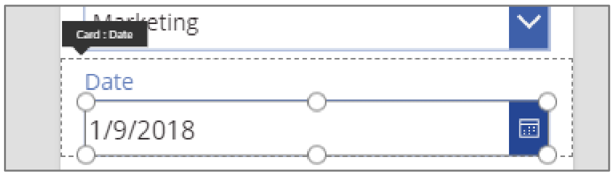
p) Update the formula of the **Width** property to the following formula.

```
Parent.Width – 60
```

q) Your formula bar should match the following screenshot.

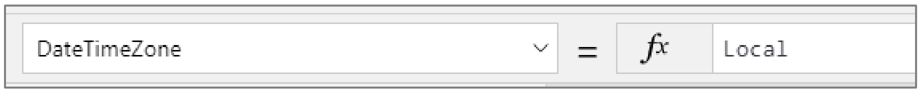

Width = Parent.Width - 60

r) The **DateValue1** control should expand to the same width of the other input controls on the edit screen.

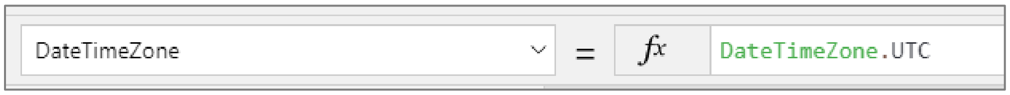


9. Update the **DataTimeZone** property of **DateValue1**.

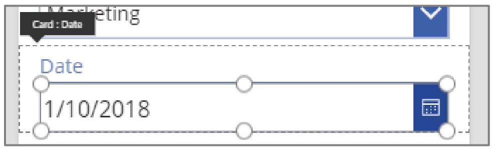
a) Inspect the **DateTimeZone** property of the **DateValue1** control. Its current value should be **Local**.


DateTimeZone = Local

b) Update the **DataTimeZone** property to a value of **DataTimeZone.UTC** as shown in the following screenshot.

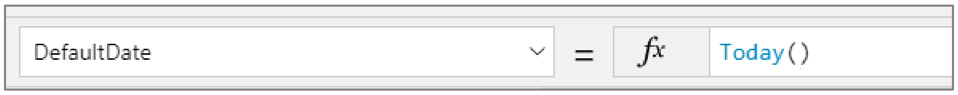

DateTimeZone = DateTimeZone.UTC

c) All the dates displayed on the edit screen should now move ahead by one day and display their proper value.



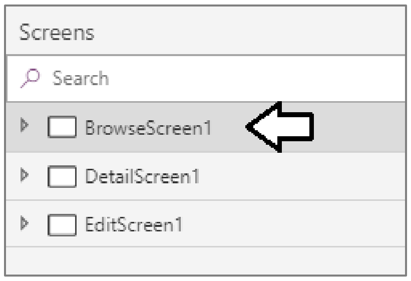
The problem with date values in the Local format is that they are offset by the difference between Greenwich Mean Time and your local time zone. For example, if you are in Eastern Daylight Time (EDT), the date of **January 10, 2018** is displayed with a 5-hour offset which is **January 9, 2018 at 7:00 PM**. By setting the **DataTimeZone** property to **UTC**, you are effectively removing the offset and the dates are displayed more accurately.

10. Configure the current day as the default value for **DateValue1**.
    a) Make sure the **DateValue1** control is selected.
    b) Inspect the **DefaultDate** property value for **DateValue1**.
    c) Update the **DefaultDate** property using the **Today()** function as shown in the following screenshot.

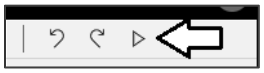


DefaultDate = *fx* Today()

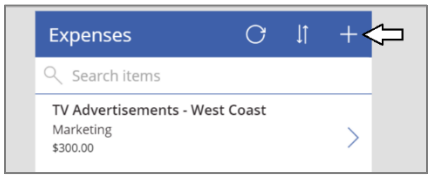
11. Test out the app by starting it and adding a new expense.
    a) Before starting the app, navigate to the screen named **BrowseScreen1**.



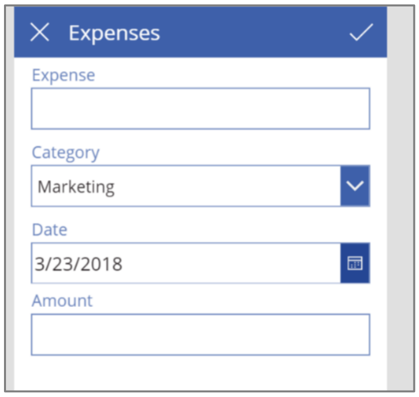
    b) Click the Start button with the arrow icon to launch the app for testing.



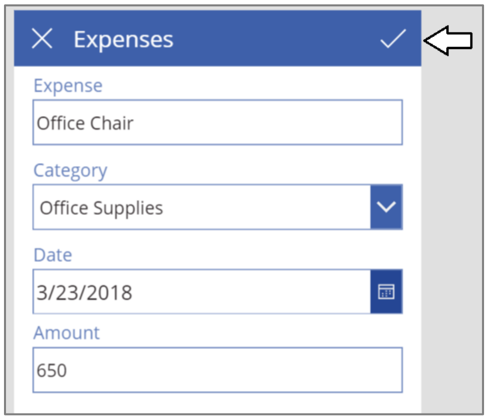
    c) When the browse screen appears, click to button with the **+** icon to add a new expense.



    d) You should now see the edit form into which you can enter a new expense.



e) Fill in the edit form for the new expense using the data shown in the following screenshot and then click the button with the checkmark icon in the upper right to save your work.



Expenses

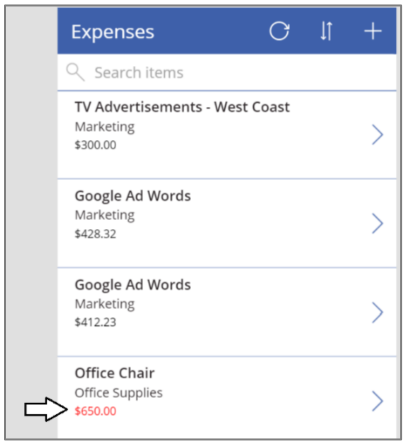
Expense: Office Chair

Category: Office Supplies

Date: 3/23/2018

Amount: 650

f) Once you have saved the new expense, you should be able to see it in the browse screen.



Expenses

Search items

TV Advertisements - West Coast
Marketing
$300.00

Google Ad Words
Marketing
$428.32

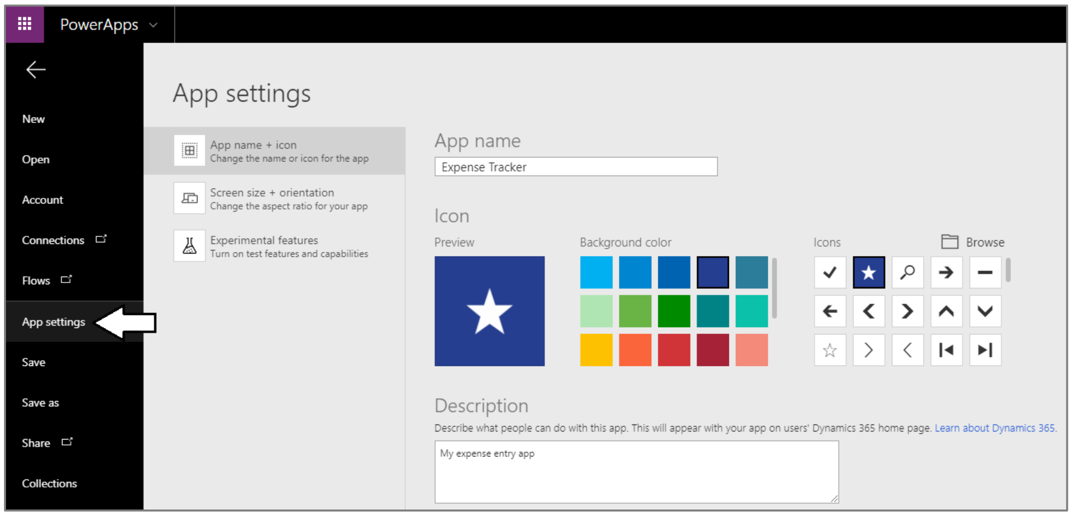
Google Ad Words
Marketing
$412.23

Office Chair
Office Supplies
$650.00

12. Save the app to the cloud.

a) Drop down the **File** menu and click the **App settings** link.

b) Name the app **Expense Tracker** and assign a color, icon and description as shown in the following screenshot.

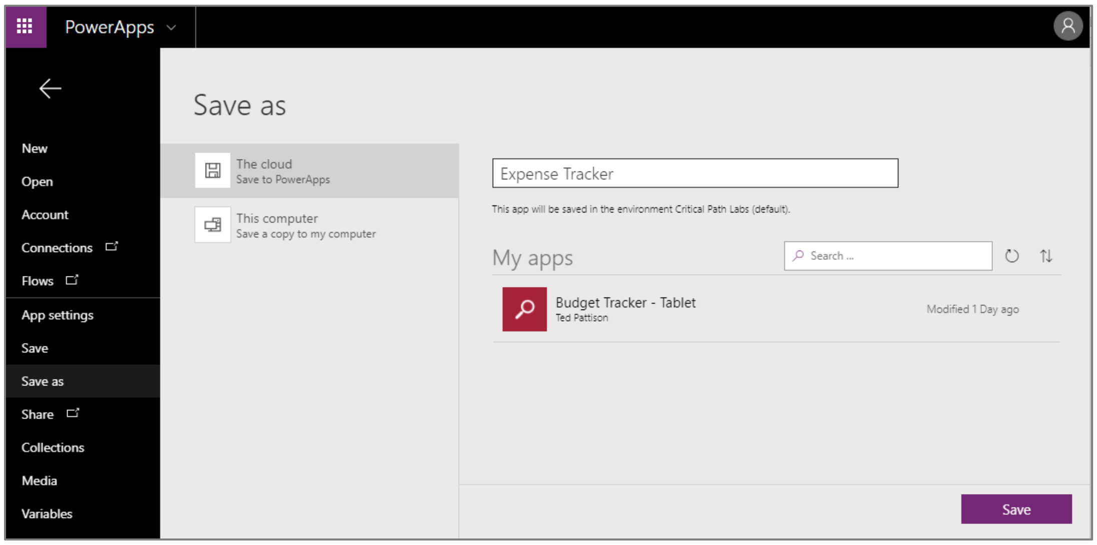


App name: Expense Tracker

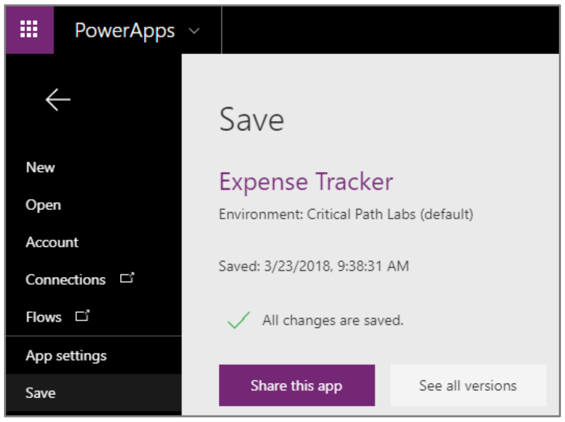
Description
Describe what people can do with this app. This will appear with your app on users' Dynamics 365 home page. Learn about Dynamics 365.

My expense entry app

c) Click the **Save** link in the left navigation and then click the **Save** button in the lower, right-hand side of the screen.



d) You should be able to confirm that your app has been saved.



Save

Expense Tracker

Environment: Critical Path Labs (default)

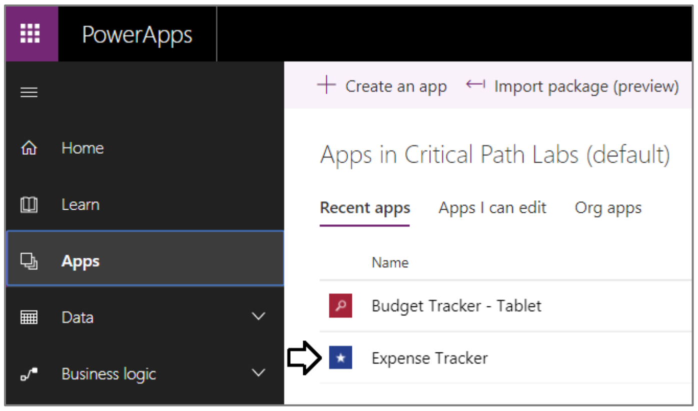
Saved: 3/23/2018, 9:38:31 AM

All changes are saved.

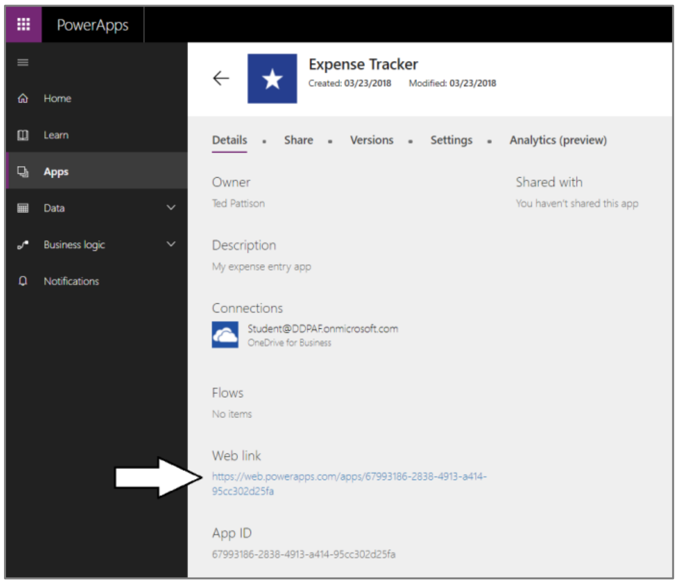
13. Examine the details of the new app.

a) Return the PowerApps Studio home page and click the **Apps** link.

b) Locate and click on the new app named **Expense Tracker** to see its details.



c) On the app details page, locate the **Web link** and click on it to launch the app.



Expense Tracker
Created: 03/23/2018 Modified: 03/23/2018

Owner
Ted Pattison

Shared with
You haven't shared this app

Description
My expense entry app

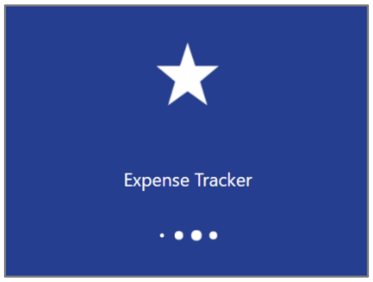
Connections
Student@DDPAF.onmicrosoft.com
OneDrive for Business

Flows
No items

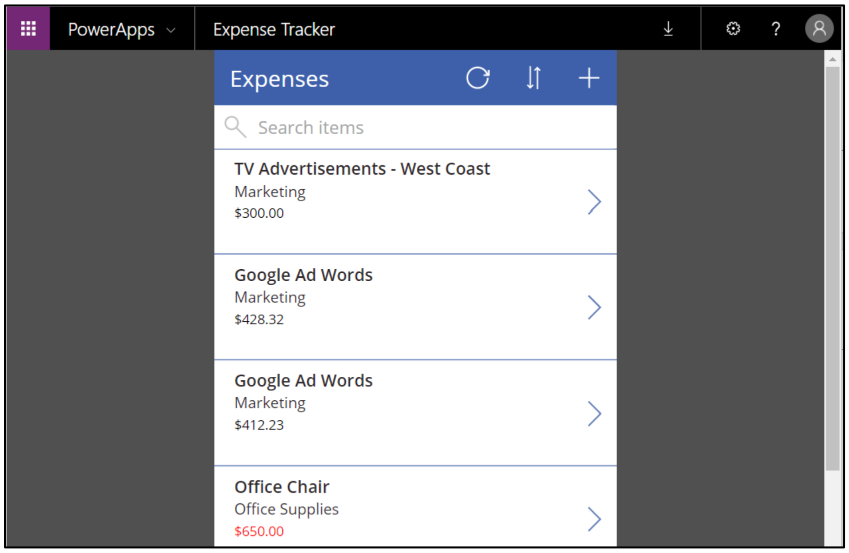
Web link
https://web.powerapps.com/apps/67993186-2838-4913-a414-95cc302d25fa

App ID
67993186-2838-4913-a414-95cc302d25fa

d) The app should start up when you click that Web link.



e) The app should now start up in the usual run mode for end users.



TV Advertisements - West Coast
Marketing
$300.00

Google Ad Words
Marketing
$428.32

Google Ad Words
Marketing
$412.23

Office Chair
Office Supplies
$650.00

Congratulations. You have now completed the first lab in this course and gone through the experience of creating two simple apps.