

INF99X: Sample Course

- **Download Latest Student Handbook and AllFiles Content**
`/home/ll/Azure_clone/Azure_new/PL-100-Microsoft-Power-Platform-App-Maker/../../releases/latest`
- **Are you a MCT?** - Have a look at our [GitHub User Guide for MCTs](#)
- **Need to manually build the lab instructions?** - Instructions are available in the [MicrosoftLearning/Docker-Build](#) repository

What are we doing?

- To support this course, we will need to make frequent updates to the course content to keep it current with the Azure services used in the course. We are publishing the lab instructions and lab files on GitHub to allow for open contributions between the course authors and MCTs to keep the content current with changes in the Azure platform.
- We hope that this brings a sense of collaboration to the labs like we've never had before - when Azure changes and you find it first during a live delivery, go ahead and make an enhancement right in the lab source. Help your fellow MCTs.

How should I use these files relative to the released MOC files?

- The instructor handbook and PowerPoints are still going to be your primary source for teaching the course content.
- These files on GitHub are designed to be used in conjunction with the student handbook, but are in GitHub as a central repository so MCTs and course authors can have a shared source for the latest lab files.
- It will be recommended that for every delivery, trainers check GitHub for any changes that may have been made to support the latest Azure services, and get the latest files for their delivery.

What about changes to the student handbook?

- We will review the student handbook on a quarterly basis and update through the normal MOC release channels as needed.

How do I contribute?

- Any MCT can submit a pull request to the code or content in the GitHub repro, Microsoft and the course author will triage and include content and lab code changes as needed.
- You can submit bugs, changes, improvement and ideas. Find a new Azure feature before we have? Submit a new demo!

Notes

Classroom Materials

It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only.

title: Online Hosted Instructions permalink: index.html layout: home

Content Directory

Hyperlinks to each of the lab exercises and demos are listed below.

Labs

```
{% assign labs = site.pages | where_exp:"page", "page.url contains  
'/Instructions/Labs'" %} | Module | Lab || --- | --- | {% for activity in labs %}|  
{ { activity.lab.module } } | {{ activity.lab.title }}.{% if activity.lab.type %}-  
{{ activity.lab.type }}.{% endif %} | {% endfor %}
```

Demos

```
{% assign demos = site.pages | where_exp:"page", "page.url contains  
'/Instructions/Demos'" %} | Module | Demo || --- | --- | {%- for activity in  
demos %}| {{ activity.demo.module }} | .{{ activity.demo.title }}. | {%- endfor  
%}
```

Trainer Guide

Environments

Students will need one Common Data Service (CDS) environment each to complete the course work. Environments can be provisioned as trial environments by the students during class or they can be provisioned prior to the delivery with credentials handed over to the students prior to the first exercise.

The environments need no specific configuration considerations, a plain CDS *without any sample data* is all that is needed.

Course Flow

PL-100 is broken down into two different delivery formats.

- Days 1 & 2: PowerPoint presentations, instructor led demos, quick hands-on-labs.
- Day 3: Scenario driven hands on labs.

Day 1 and 2

Most of the content delivered on days 1 and 2 is based on the content available on Microsoft Learn. Some topics that don't currently have a lot of coverage on Microsoft Learn have had some additional slides added to fill in any gaps. Microsoft Learn labs are sprinkled in throughout both days to act as a supplement PPT decks and instructor led demonstrations.

The PowerPoint decks specifically callout when to have the students do the hands-on-labs and which ones to have them do. Some of the Microsoft Learn labs build off each other, and some of them are stand alone, it is a good idea to familiarize yourself with the Microsoft Learn labs. This will also help you determine which ones could be skipped during class time in the event that you are falling behind.

Day 3

Day 3 is a full day of hands-on-labs where students will be building a company 311 solutions. At the beginning of the day, you will present the students with the solution and talk about some of the key elements in the lab. Since students will be working on labs all day, you should check back with them periodically to keep them engaged. After each lab is completed, check back, and talk about some of the discussion points related to that lab as well as introduce the tasks and expectations for the next lab.

It is highly recommended that you walk through and complete the complete list of Day 3 labs as a student prior to the training delivery.

- Familiarity with the exercises will make the delivery more efficient.
- Solution components built during this exercise can be used as needed for your demos. While demos are not required, they are a very effective delivery tool and will provide students a visual change from just listening

to power points. You should plan to do as many as you feel comfortable with on the topics you know best.

Days 1 & 2 Agenda

Below you will find a general agenda outline that you can use while delivering. The timings on the agenda are estimates based on the content that is being presented. Your actual delivery time may vary depending on your audience, questions, delivery style, etc. The estimated times include any instructor led demonstrations as well as any Microsoft Learn practice exercises students will be doing.

Day 1

8:00 ♦♦♦ 8:30♦♦	Welcome & Overview♦♦
8:30 ♦♦♦ 9:00	M01♦♦♦♦♦ Power Platform Introduction♦♦♦♦
9:00 ♦♦♦ 9:30	M01 ♦♦♦ Getting Started with the Power Platform
9:30 ♦♦♦ 9:45♦♦♦♦♦	Break♦♦ M02♦♦♦♦♦♦♦♦ Data Model & Model Driven Application♦♦?
9:45 ♦♦♦ 10:45	<ul style="list-style-type: none">• Intro the Common Data Service♦♦• Approach♦♦• Building Blocks♦♦• Security♦♦• Practice Exercises
10:45 ♦♦♦ 11:00	Break M02♦♦♦♦♦♦♦♦ Data Model & Model Driven Application♦♦?(Cont)
11:00 ♦♦♦ 12:00	<ul style="list-style-type: none">• Intro to Solutions• Entities• Fields• Practice Exercises
12:00 ♦♦♦ 1:00♦♦♦♦♦	LUNCH♦♦

Day 1

M02 Data Model & Model Driven Application (Cont.)

1:00 ? ? ? 2:00

- Relationships
 - Intro to BPF
 - Building Model Apps
 - Practice Exercises

2:00 ? ? ? 2:15

Break

M03? ? ? ? ? ? ? Canvas Apps? ?

2:15 ? ? ? 3:30 ? ?

- Building Power Apps
 - Screens and Controls
 - Navigation
 - Practice Exercises

3:30 ? ? ? 3:45 ? ?

Break? ?

M03 ? ? ? Canvas App UI ? ?

3:45 ? ? ? ? ? 4:30 ? ?

- Building a UI elements
 - Working with Components
 - App Management
 - Practice Exercises

4:30? ? ? ? ? ? 5:00? ? End of day wrap up and Q&A? ?

Day 2

M04 ◊◊◊ Automate Business Processes ◊◊

8:00 ? ? ? 10:00 ? ?

- Get Started with power Automate ???
 - What is Power Automate
 - Building Flows Practice Exercises ???

10:00 ♦?♦?♦? 10:15♦?♦?

Break ? ?

Day 1

M04 ◊◊◊ Automate Business Processes◊◊

10:15 ◊◊◊ 12:00

- Working with Power Automate Admin Center
- Building Complex Flows◊◊ Practice Exercises
- Power Automate Security and Governance◊◊

12:00 ◊◊◊ 1:00◊◊

LUNCH◊◊◊◊◊

M05◊◊◊◊◊◊◊◊ Create and Use Analytics◊◊

1:00 ◊◊◊ 2:00◊◊

- Getting Started with Power BI◊◊◊
- Getting Data with power BI◊◊
- Practice Exercise

2:00 ◊◊◊ 2:15◊◊

Break◊◊

M06◊◊◊◊◊◊◊◊ Getting Started with AI Builder

2:15 ◊◊◊ 3:00◊◊

- AI Builder Overview
- Capabilities
- Model Types
- Using Models

3:00 ◊◊◊ 3:15◊◊◊◊◊

Break◊◊

M08◊◊◊◊◊◊◊◊ Testing and Documentation

3:15 ◊◊◊ 4:00◊◊

- Testing apps
- Documentation

4:00 ◊◊◊ 4:30◊◊

End of Day wrap up and Next Steps

Days 3 Agenda

Below you will find a general agenda outline for Day 3 that you can use while delivering. The timings on the agenda are estimates based average time students take when working on labs. Depending on your audience, these times could be shorter or longer. It is important that you engage with your students often, especially during a virtual delivery to ensure you are keeping everyone together.

Day 3◆◆?

Scenario Overview◆◆?

8:00 ◆◆◆ 8:15◆◆?

- Introduce overall scenario
- Call out discussion points
- Point students to lab resources

Lab 01 ◆◆◆ Design Solution◆◆?

8:15 ◆◆◆ 9:15◆◆?

- Introduce the design scenario
- Explain why this is important
- Provide students with 45 mins to 1 hour to work on lab

9:15 ◆◆◆ 9:30◆◆?

Break◆◆?

Lab 02 ◆◆◆ Data Model and Model App◆◆?

9:30 ◆◆◆ 10:45

- Review Discussion Points from Lab 01
- Introduce the Lab 02 scenario
- Call out any potential pain points to be aware of
- Provide students with 45 mins to 1 hour to work on lab

10:45 ◆◆◆
11:00◆◆?

Break◆◆?

Lab 03 ◆◆◆ Canvas Apps◆◆?

11:00 ◆◆◆ 12:00

- Review Discussion Points from Lab 02
- Introduce the Lab 03 scenario
- Call out any potential pain points to be aware of
- Provide the rest of the morning to work on Lab 03

Day 3◆◆

11:45 ◆◆◆ 1:00◆◆ Lunch◆◆

Lab 04 ◆◆◆ BPF & Business Rules◆◆

1:00 ◆◆◆ 1:45

- Review Discussion Points from Lab 03
- Introduce the Lab 04 scenario
- Call out any potential pain points to be aware of
- Provide students with 30 to 45 mins to work on lab

1:45 ◆◆◆ 2:00◆◆ Break◆◆

Lab 05 ◆◆◆ Power Automate◆◆

2:00 ◆◆◆ 2:45

- Review Discussion Points from Lab 04
- Introduce the Lab 05 scenario
- Call out any potential pain points to be aware of
- Provide students with 30 to 45 mins to work on lab

2:45 ◆◆◆ 3:00◆◆ Break◆◆

Lab 06 ◆◆◆ Power BI

3:00 ◆◆◆ 3:45

- Review Discussion Points from Lab 05
- Introduce the Lab 06 scenario
- Call out any potential pain points to be aware of
- Provide students with 30 to 45 mins to work on lab

3:45 ◆◆◆ 4:00◆◆ Break◆◆

Lab 07 ◆◆◆ Test and Deploy

4:00 ◆◆◆ 4:45

- Review Discussion Points from Lab 06
- Introduce the Lab 07 scenario
- Call out any potential pain points to be aware of
- Provide students with 30 to 45 mins to work on lab

4:45 ◆◆◆
5:00◆◆◆◆◆

Course Wrap up / Evaluations / Questions◆◆

Day 3 Hands on Labs Overview

Below, you will find the scenario that students will be walking through as part of day three. Make sure that you have taken time to be familiar with the scenario.

Scenario

You are an employee at Lamna Healthcare Company and work in the billing department.

As you were walking out to your car from work you noticed the company digital sign by the exit was still welcoming participants of the 2019 convention. You would tell someone, but you have no idea who to tell. So often you find something in your company that should be fixed but have no way to tell anyone.

As you were driving home sitting in traffic you had an idea. What if there was an app you could use to report things like this. You could report a location, a category and even a picture of the problem so someone could easily locate and fix it. The only problem was if you got people to report things who would be listening.

The next day at work you met with your friend in the facility department and shared your idea. She was excited too because today there is no coordinated way, she can get problems like these reported and fixed. She explained how often they involve different departments and must be routed and followed up to find out the status. She had looked at commercial options, but they had been too complex and inflexible, overengineered, and expensive. She did however give you an example of a 311-system used by cities for their citizens to report problems. Based on that example, you decided you would call it Company 311

[!IMPORTANT] The intent hands-on-labs on day three is to help the students to better understand how to build a complete solution using the Power Platform. They will be building components typical for Power Platform solutions. There will be dependencies. Students will need to perform the labs in order to ensure any required components are available moving forward. Labs two and three will involve importing solutions. Sample data imported during the lab makes building and testing the apps easier, more realistic, and more relatable.

Trainers should clearly explain the objective of every lab and outline high-level solution components and techniques. Students then should be encouraged to design and build their own solutions without following step by step instructions for every exercise.

Microsoft Teams

The world has changed rapidly in the recent months and organizations need tools to work collaboratively yet remotely. Microsoft Teams provides a powerful platform for collaboration and communication, and also provides an app maker with additional options when deploying apps and building business processes. Consider ways some of your business apps and processes could be elevated by integrating with Microsoft Teams.

To help students better understand Microsoft Teams as a communication and collaboration mechanism, some of the labs include additional exercises highlighting integration of the Power Platform components in Microsoft Teams.

Components

Completed components are available for *all* exercises. They include canvas apps, model-driven app, power automation flows, Power BI pbix file, sample data and data import map, and complete solution exported as both managed and unmanaged. Components shouldn't be used as a shortcut to exercise completion. Instead, they are useful as a demo tool and as a general point of reference.

Labs

Lab 01 Design the Solution

Application designers need to be able to take requirements, requests, wants, and needs and turn them into a tangible solution. Based on requirements gathering, app designers should be able to identify gaps and requirements for specific ideas. They should be able to map requirements to the Power Platform and determine what is needed for a data model. Clearly defining requirements and building a solid data model is a foundation of the apps built on the platform. In this lab, students gather requirements from interviews, and other artifacts. They will be walked through how to create a data model to support the requirements and import sample data.

Topics to discuss:

- What tools would you use?
- How do you approach modeling?
- Where do you start?
- Relationships
- Good practices around implementing data model in CDS
- Bringing Common Data Model entities (e.g. contacts) what to include.
- Dev/test sample data sets how much is enough. Good practices around building dev/test data sets.

Lab 02 Data model and model-driven app

In this lab students will be implementing the data model that was discussed in the previous lab. In addition, they will be model-driven app that will be used by anyone fixing problems or managing the overall effort. This is the not application that would be used to report the problem. That is a canvas app that will be built in the next lab.

Topics you can discuss

- Model-driven apps as part of a solution

- Model-driven vs canvas Power Apps. Target audience and app objectives.
- What makes a good sitemap?
- What to include into an app for a given entity?
- Apps & roles, security in CDS, UI trimming in model-driven apps
- Adjusting data model vs adjusting the model-driven app

Things to watch out for:

Some students may have questions on the creation of entities. Make sure to encourage students to read the complete instructions and not just focus on the pictures. Focusing on the images only may result in missing a step that will be important later.

Some students may struggle with importing the demo data solution. Make sure you have walked through it a couple of times to ensure you can easily field any questions that they might have.

Lab 03 Canvas app

This lab has the students building the canvas application that employees will be able to use to report issues they encounter. The canvas app will include a screen that displays reported issues, and a screen where users can report issues. The hand-on-lab consists of three exercises.

- Exercise 1: Import Company Components  Students will import a SharedComponents solution that includes some prebuilt components that they will use for the header and footer of the app.
- Exercise 2: Create App and Layout  Students will create the main and issue reporting screens. They will use Galleries and Forms to control how people interact with the app.
- Exercise 3: Submit a new report  Students will use the canvas app that they created to report a new issue and test the functionality.

[!IMPORTANT] Students will need to have completed all the exercises in Lab 2 prior to working on this lab.

Topics to discuss

- Tablet vs phone form factor. Responsive layouts (available soon in canvas).
- How does the target audience impact the app design.
- How knowledge of Excel can benefit app makers (traditional procedural programming vs properties and expression syntax of Power Apps)
- Which connector to use. How CDS connector is now internal to canvas Power Apps.
- App checker    new metrics every day. Make sure to retest the app.
- Accessibility    what makes a good accessible app

Things to watch out for

Component Library Issue

At the time the course was created, there was a glitch with imported components libraries from solutions not showing up as component libraries. The lab was modified to include instructions for opening the app that includes the components and saving it with a slightly different name. This will ensure that it saves as a component library and that they can be added correctly to the app.

Formula Error

In Exercise 1 > Task 3 > Step 20 students are instructed to copy and paste a formula into the canvas apps OnStart event. If students are copying and pasting the text from a PDF or Markdown, they might encounter the text not populating correctly. Many of the Double Quotes items such as Label and Icon could be replaced with single quotes. This will cause the formula to error out.

Some students may struggle with importing the components library solution. Make sure you have walked through it a couple of times to ensure you can easily field any questions that they might have.

Lab 04 Business Process Flows and Business Rules

In this lab, students will create a Business Process Flow that can be used to help guide responsible parties in resolving any issues reported. Students will also create a Business Rule that prevents closing the Reported item if no resolution is provided.

[!IMPORTANT] Students need to complete the exercises in Lab 02 prior to working on this lab. The business rules and business process flows build on the items data model created in Lab 02.

Topics to discuss

- How else might Business Rules be leveraged as part of this process.
- What if you did not want to start the business process with a Reported item? What if you wanted to make more generic?
- Scope: Talk through how scope of a business rule can impact which rule you choose and how it is interacted with by users.

Things to watch out for

This lab does not have any real issues that come up. The biggest issue is this lab would be if people had not completed all the necessary steps in the previous labs.

Lab 05 Power Automate

In this lab, students will use Power Automate flows to automate various parts of the Company 311 solution. ♦♦? The following have been identified as requirements you must implement to complete the project:♦♦?

- Escalation, approval, and execution process for urgent maintenance issues♦♦?
- Notify reporting user about the issue status changes♦♦?
- How to use a business rule to implement logic.

Keeping in mind the requirements above, students will add additional fields that will be used to support the desired automation. Students will leverage both the CDS connector and approvals to handle the escalation of items as needed.

[!IMPORTANT] Students need to complete the exercises in both Lab 02 and Lab 04 prior to working on this lab. The approval flows that you will be creating build on the data model created in lab 02 and the additional fields added in lab 04.

Topics to discuss

- Flows in solutions
- CDS connectors (standard vs current environment)
- Other automation means in Power Platform (real-time workflow, UI automation)
- Human interaction with flows - approval processes
- Using adaptive cards instead of approval emails
- Testing the processes during the build
- Sharing the flows
- Monitoring running processes

Items to watch out for

Adding **Wait for an approval** action sometimes generates the following error message:

Http request failed with unhandled exception of type 'NullReferenceException' and message: 'Object reference not set to an instance of an object.'

This message can be ignored.

There are no any other real issues that occur during this lab. The biggest item is that students begin to drift a little towards the end of exercise three. Many of the steps are similar and start to run together.

Lab 06 Power BI

In this lab, students will build a Power BI dashboard that visualizes data related to reporting items. Students will use Power BI desktop to build a dashboard and report elements that can be viewed in Power BI.

[!IMPORTANT] Students need to complete the exercises in Lab 02 prior to working on this lab. The visual elements that they will create in this lab, use the data model created in lab 02.

Topics to discuss

- Who is the target audience of the report?
- How will the audience consume the report? Typical device? Location?
- How to start the design; how to move beyond the blank canvas
- Connect to data sources, build and refine data model

- Do you have sufficient data to visualize?
- What are the possible characteristics you can use to analyze data about the visits?
- Publishing and sharing the reports
- Accessibility in Power BI when building interactive reports

Items to watch out for

Students will need to have Power BI desktop installed on their local machine to build data sets and transform data. Students that do not have Power BI Desktop installed should download it from the Microsoft downloads page: <https://aka.ms/pbiSingleInstaller>.

[!IMPORTANT] While students can install it from the Windows Store, it is not recommended to have them install from there. They will likely receive permission errors.

Some students may not be able to download the application at all because of permission restrictions on their machine. In those instances, the .pbix file that students create is included. Students can load that into Power BI and build the report in Power BI service (<https://app.powerbi.com>). These steps are outlined in **Exercise 5**.

Lab 07 Model-driven app

In this lab students will learn how to deploy solutions to another environment. This lab lets students better understand additional item they should consider when designing application and deploying them, such as security, compatibility testing.

Topics to discuss

- Who should be able to access the different apps?
- Model-driven vs canvas Power Apps. (How do you control access?)
- Where are you deploying the applications to?

- Apps & roles, security in CDS, UI trimming in model-driven apps

lab: title: 'Lab: Validate lab environment' module: 'Module 0: Course introduction'

[!NOTE] Effective November 2020: - Common Data Service has been renamed to Microsoft Dataverse. [Learn more](#) - Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. [Learn more](#)

This content will be updated soon to reflect the latest terminology.

Module 0: Course introduction

Practice Lab ◉◉◉ Validate lab environment

Scenario

In this Module 0 lab, you will acquire a Power Platform trial tenant and access the Power Platform admin center. In the admin center, we will create an individual environment for configuration during the course.

Exercise 1 ♦♦♦ Acquire your Power Platform trial tenant

1. Copy your **Microsoft 365 credentials** from the Authorized Lab Hoster.
2. Navigate to and click **Start free**.
3. Under **Work email**, enter the email address from your Microsoft 365 credentials.
4. You see a prompt that you have an existing account with Microsoft.
Select **Sign in**.
5. Enter the password provided by the Authorized Lab Hoster.
6. Select **Yes** to stay signed in.
7. Click the **Start** button to begin building your Power Platform trial tenant.

Exercise 2 - Create your environment

In this exercise, you will be create your **Practice** environment that you will do the majority of your lab work in.

Task 1 ♦♦♦ Create environment

1. Access <https://admin.Powerplatform.microsoft.com> and log in with your Microsoft 365 credentials if prompted again.
2. Select **Environments** and click **+New**.
 - For **Name**, enter **[my initials] Practice**. (Example: AJ Practice.)
 - For **Type**, select **Trial**.
 - Change the toggle on **Create a database for this environment?** to **Yes**.
 - Leave all other selections as default and click **Next**.
 - On the next tab, leave all selections to default and click **Next** again.
 - On the next tab, leave all selections to default and click **Save**.
3. Your **Practice** environment should now show in the list of Environments.
4. Your environment may take a few minutes to provision. Refresh the page if needed. When your environment is prepared, select your **Practice** environment by clicking on the ellipses next to its name to expand the drop down menu and select **Settings**.

3. Explore the different areas in Settings that you are interested in but do not make any changes yet.

lab: title: 'Lab: Design the solution' module: 'Module 1: Introduction to Power Platform'

[!NOTE] Effective November 2020: - Common Data Service has been renamed to Microsoft Dataverse. [Learn more](#) - Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. [Learn more](#)

This content will be updated soon to reflect the latest terminology.

Lab 01: Design the solution

In this lab you will be shaping your idea into something that can be implemented on the Power Platform. As part of this you will meet with other people in your organization to get more clarity on how your idea could be implemented. Using this information, you will identify what applications and automations need to be built.

What you will learn

- How to identify gaps and requirements for an idea
- How to map a problem domain to the Power Platform
- How to determine required Tables for a data model

High-level lab steps

- Exercise 1 - Scenario overview
- Exercise 2 - Extract needs from interview with a co-worker and facility staff
- Exercise 3 - Design a data model
- Exercise 4 - Identify apps and automation that are needed
- Exercise 5 - User story, mockup the app UI

Detailed steps

Exercise 1: Scenario Overview

In this exercise, you will get up to speed on the scenario you will be building in this series of labs.

Task 1: Read the scenario

Read the following scenario and make note of any key points you think might be important later.

You are an employee at Lamna Healthcare Company and work in the billing department.

As you were walking out to your car from work you noticed the company digital sign by the exit was still welcoming participants of the 2019 convention. You would tell someone, but you have no idea who to tell. So often you find something in your company that should be fixed but have no way to tell anyone.

As you were driving home sitting in traffic you had an idea. What if there was an app you could use to report things like this? You could report a location, a category and even a picture of the problem so someone could easily locate and fix it. The only problem was if you got people to report things who would be listening?

The next day at work you met with your friend in the facility department and shared your idea. She was excited too because today there is no coordinated way she can get problems like these reported and fixed. She explained how often they involve different departments and must be routed and followed up to find out the status. She had looked at commercial options, but they had been too complex and inflexible, overengineered, and expensive. She did however give you an example of a 311-system used by cities for their citizens to report problems. Based on that example, you decided you would call it Company 311.

Exercise 2: Extract needs from interviews

In this exercise, you review the text from an interview you had with a couple of colleagues. In each of these interviews you shared your idea for a Company 311 solution and got feedback from your coworkers. You should use this information to shape the solution design.

Task 1: Interview #1

Review the following discussion with your coworker and take notes of any of the key things you learned from the interaction. This Interview is from a coworker in your same Department that you are friends with.

You: I wanted to get your thoughts on the idea for a Company 311 that we talked about. I am thinking of starting to build that app. What do you think we need to include when we submit a problem report?

Coworker: Well you have to capture where the problem is - we have so many buildings. What department should fix it? Oh, and a picture.

You: Do you think building is enough to identify where the problem is?

Coworker: Maybe allow them to describe where in the building...

You: After you report a problem what do you expect to happen?

Coworker: Them to fix it of course!

You: No, I mean in the app, what do you see after you click submit problem?

Coworker: I want to know that someone got it and it's being worked on and when fixed. Actually, not always but most of the time...maybe let me choose to be notified?

You: So perhaps a list of all your items submitted?

Coworker: Yeah that would be great!

You: perfect, I will let you know when you can try the app!

After you complete reading this and have your notes compare them to our notes in the next task to see if you missed anything.

Task 2: Interview #1 Notes

In this task, you will compare your notes from Interview #1 with our notes.

The following are our notes from Interview #1

- Need to be able to pick a building for each problem
- Need to capture which department needs to fix it when you submit the problem
- Need a photo of the problem
- Need a freeform text description of the problem location within the building
- Need a way to indicate if you want to be notified when completed
- Need to see all the problems you submitted and their status

Task 3: Interview #2

Review the following discussion with your coworker and take notes of any of the key things you learned from the interaction. This Interview is from a coworker in facilities management that you are friends with. You believe most of the problem reports will be handled by them.

You: I wanted to get your thoughts on the idea for a Company 311 that we talked about. I am thinking of starting to build that app. What do you think needs to be included when people submit a problem report?

Coworker: As much details as possible, a photo would be nice. Oftentimes we get reports that are just a very vague indication of the problem that if we had a picture it would be 1000 times clearer.

You: What do you think about allowing them to pick which department they think will fix the problem?

Coworker: Now that is funny! Most people have no idea who fixes it and think it is just magic. I would suggest that people just submit the problem report without a department and then one of our facilities people would assign the department it needs to address the problem.

You: Perfect! Do you fix all the problems that get reported?

Coworker: Many of them are duplicates and do not get fixed; others will cost too much and must get manager approval. If they are not approved they don't get fixed.

You: How do you do that approval today?

Coworker: If I get one that I think is going to be expensive I have to try to track down the manager for approval, and sometimes if I can't get it right away it gets set aside till I remember.

You: Ok, so if we could include approval that might help. I will let you know when you can try the app!

After you complete reading this and have your notes compare them to our notes in the next task to see if you missed anything.

Task 4: Interview #2 Notes

In this task, you will compare your notes from Interview #2 with our notes.

The following are our notes from Interview #2

- Having a photo would be helpful
- Department should not be provided by user but assigned after submitting
- Approval required over specific amount, would be helpful to automate

Exercise 3: Design a data model

In this exercise, you will create the data model to support the apps you will be building.

Task 1: Evaluate what you know already about the data

In this task, you will be evaluating the information you already collected about your proposed solution and trying to identify what data Tables are needed and how they are related. If you want, you can do this task concurrently with the next task where you draw the data model.

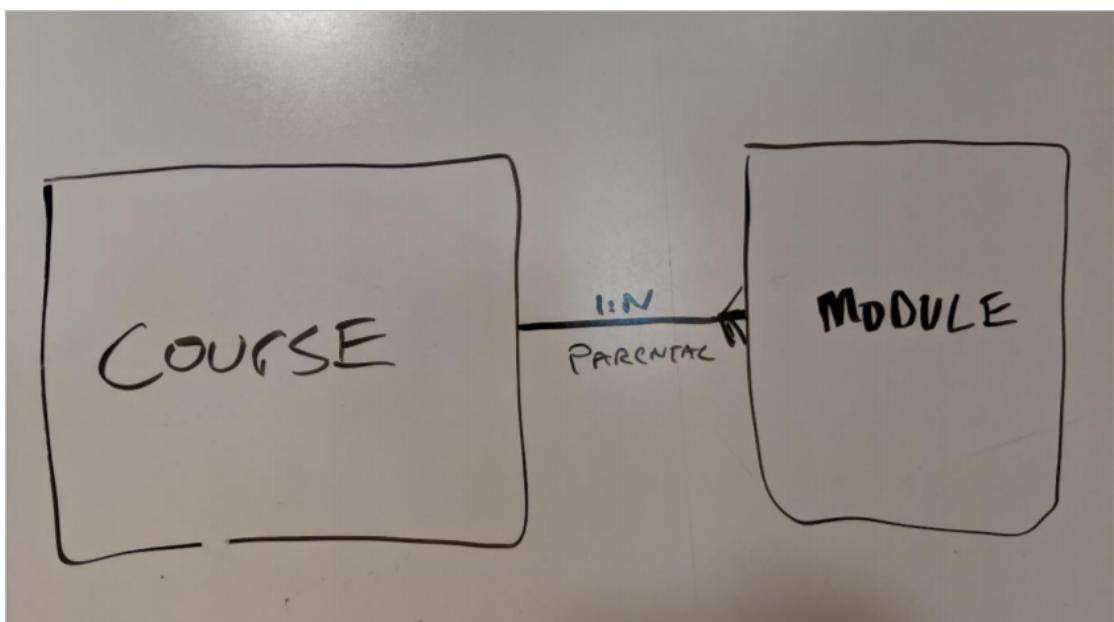
- Identify the main data that will be managed by the solution. This will typically become one or two Tables and will be the focus of the app you build. Other data is typically related to and supports these Tables.

- Identify related Tables needed to support your scenario.
- Identify how Tables should be connected using relationships.
- Evaluate what should be Columns and what should be Tables. For example, how should the photo be stored or the location within the building?

Task 2: Draw a draft data model

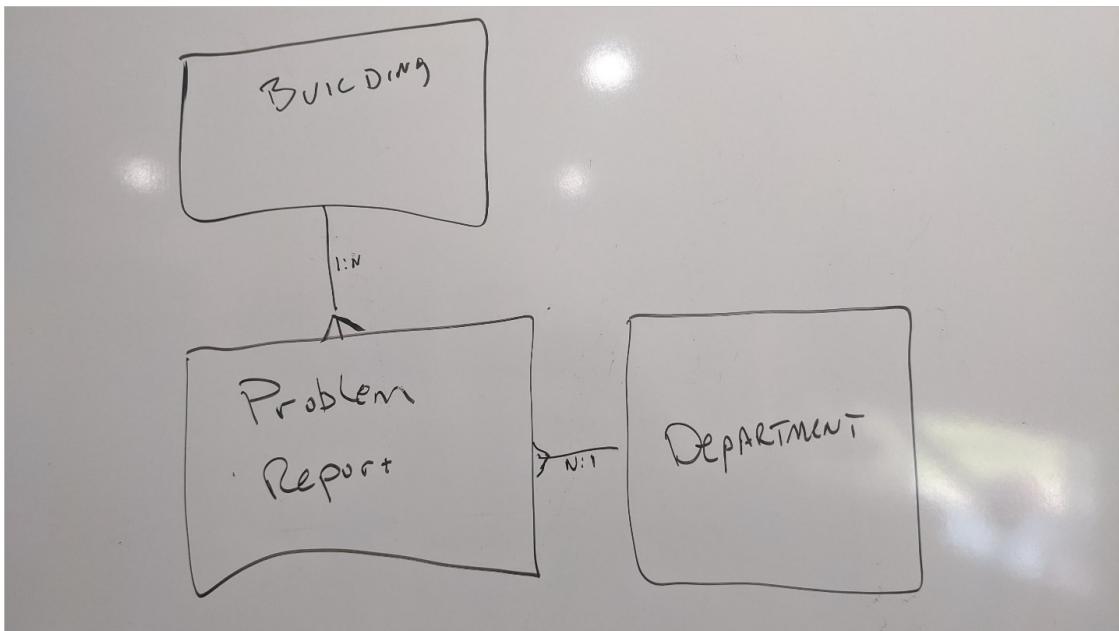
Use whatever tools you have available; you can use a whiteboard, Visio, PowerPoint, OneNote, or you can even use it piece of paper and pen. The goal here is not to be picture perfect but to allow you to think through what the data model should look like and possibly share it with others and get their ideas. This data model will typically be your guide when you are creating the Tables in the maker portal. You could, of course, just start creating the Tables in the portal, but creating a diagram helps ensure it's more carefully planned out.

1. Draw your data model, including relationships and any relationship behaviors. Your drawing should look like the following example, except yours should be for your Company 311 solution.



Task 3: Compare data models

1. Compare the data model you created in the previous task with the one we prepared. If there are significant differences you should discuss those with your instructor.



Exercise 4: Identify apps and automation needed

In this exercise, you will be looking at the information you collected and deciding what apps and automations are required to implement the solution. The goal is not to identify every feature of the application or automation but to identify if you need one app or ten apps, and what style app they should be.

Task 1: Evaluate what apps are needed

In this task you are going to look at how the users interact with the applications and decide if you need one or multiple applications and what style they're going to be, i.e. canvas or model-driven. There is no single right answer to how to accomplish this, but by asking the right questions you can design a better solution for your users. As you go through the following steps make some notes about your Company 311 solution.

1. Identify who will be using the app.
2. Identify how each set of users will be accessing the app. Will it be mostly from mobile device or desktop?
3. Of the overall functionality you are going to provide, are there specific subsets that some users use all the time?
4. Is there any device usage that would lend itself to one type of application versus another?

5. Model-driven apps are great for data management. Is there any functionality that would lend itself more to a model-driven app?
6. Considering your answers to the above questions and make notes of how many apps you will be building, the type of app, and what each app will do and who it will be used by.

Task 2: Compare your notes on apps

In this task, you should compare your notes from the previous task with our prepared notes. If there are big differences you should discuss them with your instructor.

1. Identify who will be using the app:
 - Group 1 ♦?♦? Any employee in the company
 - Group 2 - Facilities staff and anyone in the different departments that fix problems
2. For each set of users will they be accessing it mostly from for mobile device or desktop?
 - Group 1 ♦?♦? Probably mostly on their mobile devices
 - Group 2 - Mostly on their desktop but sometimes on mobile
3. Of the overall functionality you are going to provide are there specific subsets that some users use all the time?
 - Group 1 ♦?♦? Most important functionality is submitting a problem report, they do not do anything to manage the list of buildings or departments
 - Group 2 - Most important functionality is routing and resolving problem reports and managing the reference data associated with buildings and apartments
4. Is there any device usage that would lend itself to one type of application versus another?
 - Easy use of camera or photo upload from mobile device
5. Is any of the functionality more data management that would lend itself more to a model-driven app?

- The management of reference data for buildings and apartments would be easy to do in a model driven app.
 - The routing and assignment of problem reports to different users would be easy to handle in a model driven app.
6. Considering your answers to the above questions make notes of how many apps you will be building, the type of app, and what each app will do and how it will be used by.
- App 1 ♦♦♦ Company 311 - This will be a canvas application used to submit new problem reports and see a list of any problem reports submitted.
 - App 2 ♦♦♦ Company 311 Admin - This will be a model driven application used by all those that route and resolve problem reports. this application will also manage all the reference data like buildings and department lists.

Exercise 5: User story, app UI mockup

In this exercise, you will review a user story that describes a user interacting with the app to submit a problem report.

Task 1: User Story

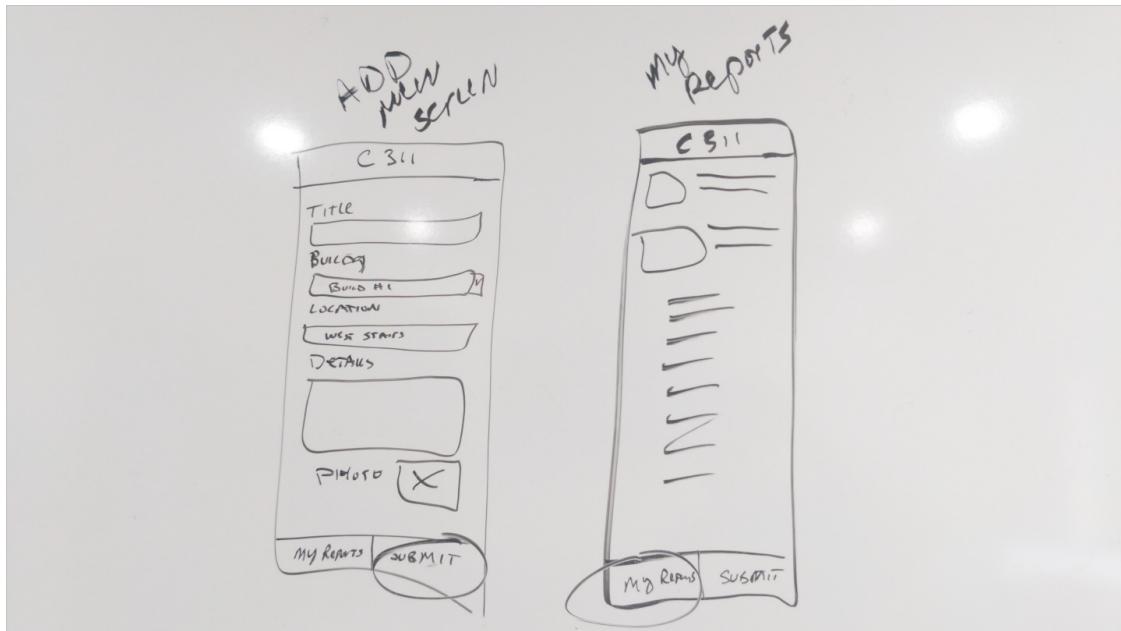
Review the following user story:

As a user I want to be able to quickly open the app and submit a problem report. I should be able to pick a building, give a location that describes where the problem is. The app should allow me to provide one line title and details of the problem. I should be able to optionally provide a photo. I should be able to easily switch over and see the list of problems that I already submitted and their status.

1. Using any of the tools you have available, such as a whiteboard, Visio, OneNote or even a piece of paper and a pen, draw a mockup of the user interface to satisfy the above user story.
2. After you have completed your drawing of the mockup go ahead to the next task and compare it to the one we provide.

Task 2: Compare mockup

The following is an example UI mockup showing both the add new item and My reports list. There is no single answer to what this has to look like and there are possibly many examples you could come up with. The goal for UI markup is to be able to quickly demonstrate what you want to build and show it to somebody and without having to actually build it. Depending on the tool you used you can often make changes quickly as you evolve the mockup. The markup is used to help you build the actual applications screens quicker with less rework.



lab: title: 'Lab: Data model and model-driven app' module: 'Module 3: Building model-driven apps'

[!NOTE] Effective November 2020: - Common Data Service has been renamed to Microsoft Dataverse. [Learn more](#) - Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. [Learn more](#)

This content will be updated soon to reflect the latest terminology.

Lab 02: Data model and model-driven app

In this lab you will be implementing the data model for the  solution and building a model-driven app that will be used by anyone fixing problems or managing  the overall effort.

What you will learn

- Create Tables, Columns and relationships
- Create a model-driven app
- Create a site map
- Create and configure Table forms
- Create and configure Table views

High-level lab steps

- Exercise 1 Create publisher and solution
 - Data Model
 - Building
 - Department
 - Problem Report
- Exercise 2 Implement the data model
 - Data Model
- Exercise 3 Configure forms and views
- Exercise 4 Compose a basic model-driven app
- Exercise 5 Input data and refine some views, import some problem reports

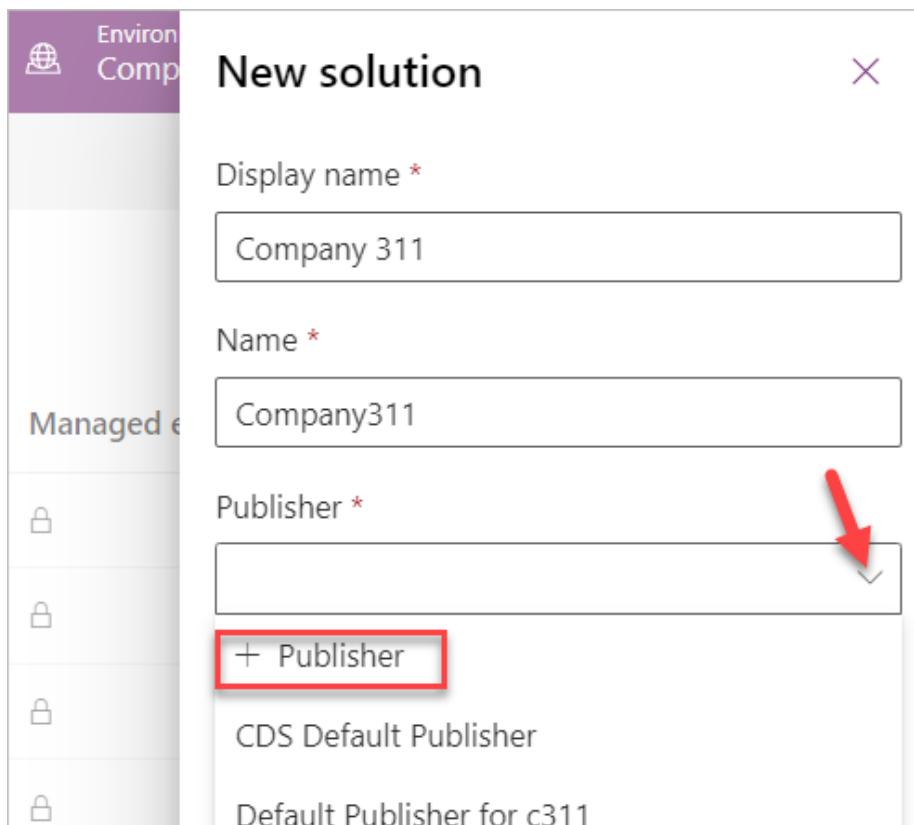
Detailed steps

Exercise 1: Create publisher and solution

In this exercise, you will create a custom solution publisher and a solution. This custom solution will be used in all the labs for this course to keep all the assets together.

Task 1: Create publisher and solution

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the practice environment you created.
2. Select **Solutions** and click **+ New solution**.
3. Enter **Company 311** for **Display name**.
4. Click on the **Publisher** dropdown and select **+ Publisher**.



1. Enter **Lamma Healthcare** for **Display name**, **lh** for **Prefix**, and click **Save and Close**.

File Save and Close

Publisher: New Publisher

Information

General

Display Name * Name *

Description

Set the prefix name for custom entities and fields

Prefix * Option Value Prefix *

Name Preview

Contact Details

1. Click **Done** on the popup.
2. Click on the **Publisher** dropdown again and select the **Lamna Healthcare** publisher you created.
3. Click **Create**.

New solution

Display name *

Name *

Publisher *

Edit publisher

Version *

More options

Create **Cancel**

1. You should now see the solution you created in the solution list.

Solutions				
Display name	Name	Created ↓	Version	
Company 311	... Company311	7/30/2020	1.0.0.0	
Power Apps Checker Base	... msdyn_PowerAppsC...	7/25/2020	1.2.0.22	

Exercise 2: Implement data model

In this exercise, you will create Tables, Columns, and the relationships you identified when you designed the data model for the Company 311 app.

Task 1: Create Tables

1. In the [Power Apps maker portal](#) select **Solutions** and click to open the **Company 311** solution you created in Exercise 1.
2. Click **+ New** and select **Table**.
3. Enter **Building** for **Display name** and click **Create**.

New table

X

Display name *

Building

Plural display name *

Buildings

Name * ⓘ

Ih_ Building

Primary Name Column ⓘ

Display name *

Name

Name * ⓘ

Ih_ Name

Enable attachments (including notes and files)

More settings ▾

Activate Windows

Go to Settings to activate Windows.

Create

Cancel

1. Go back to the solution by clicking on the solution name.

New table

X

Display name *

Problem Report

Plural display name *

Problem Reports

Name * ⓘ

Ih_ProblemReport

Primary Name Column ⓘ

Display name *

Title

Name * ⓘ

Ih_Title

Enable attachments (including notes and files)

More settings ▾

Activate Windows

Go to Settings to activate Windows.

Create

Cancel

1. Click + New and select **Table** again.

2. Enter Department for **Display name** and click **Done**.

New table

Display name *

Plural display name *

Name * ⓘ

Primary Name Column ⓘ

Display name *

Name * ⓘ

Enable attachments (including notes and files)

More settings ▾

Activate Windows

Go to Settings to activate Windows.

Create **Cancel**

1. Go back to the solution by clicking on the solution name.
2. Click **+ New** and select **Table** one more time.
3. Enter **Problem Report** for **Display name**, change the **Primary Column Display name** to **Title**, and click **More settings**.

New table

X

Display name *

Problem Report

Plural display name *

Problem Reports

Name * ⓘ

Ih_ProblemReport

Primary Name Column ⓘ

Display name *

Title

Name * ⓘ

Ih_Title

Enable attachments (including notes and files)

More settings ▾

Activate Windows

Go to Settings to activate Windows.

Create

Cancel

1. Click to expand the **Collaboration** section.

Fewer settings ^

> Description

> Entity type and ownership

> Collaboration

> Create and update settings

1. Check the **Enable queues** checkbox and click **Done**. Enabling queues allows Problem Report Rows to be associated with one or more queues to help facilitate routing to the different departments.

Auto create access teams ⓘ

Enable queues + ⓘ

Automatically move records to the owner's default queue when a record is created or assigned

+ After you enable this option it cannot be disabled

> Create and update settings

> Offline

Done **Cancel**

1. Click **Okay** on the **Confirm changes** popup.

Confirm changes

The following options cannot be disabled after you enable them:

- Enable queues

Don't show me this again

Okay

Cancel

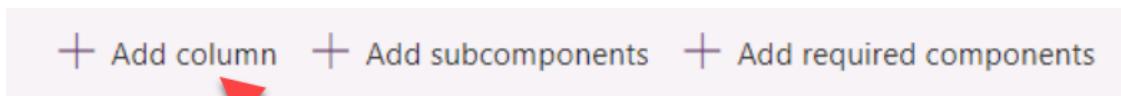
Task 2: Add Columns

In this task, you will add Columns to the Problem Report Table.

1. Navigate to the [Power Apps maker portal](#) page and make sure you are in the correct environment.
2. Select **Solutions** and click to open the **Company 311** solution you created in exercise 1.
3. Locate and click to open the **Problem Report** Table.

Solutions > Company 311	
Display name	Name
Building	... lh_building
Department	... lh_department
Problem Report	... lh_problemreport

1. Select the **Columns** tab and click **+ Add Column**.



1. Enter **Location** for **Display name**, select **Text** for **Data type**, and click **Advanced options**.

The screenshot shows the 'Location' column configuration dialog. It includes fields for Display name (Location), Name (lh_Location), Data type (Text), Required status (Optional), and Searchable checkbox (checked). There is also a 'Calculated or Rollup' section with an 'Add' button. The 'Description' and 'Advanced options' sections are also visible. A red arrow points to the 'Advanced options' link at the bottom right of the dialog.

1. Change **Max length** to **150** and click **Done**.

Max length *

IME mode

Done **Cancel**

1. Click **+ Add Column** again.
2. Enter **Details** for **Display name**, select **Multiline text** for **Data type**, make the Column **Required**, and click **Done**.

Details

Display name *

Name * ⓘ

Data type * ⓘ

Multiline Text

Required *

Required

Searchable ⓘ

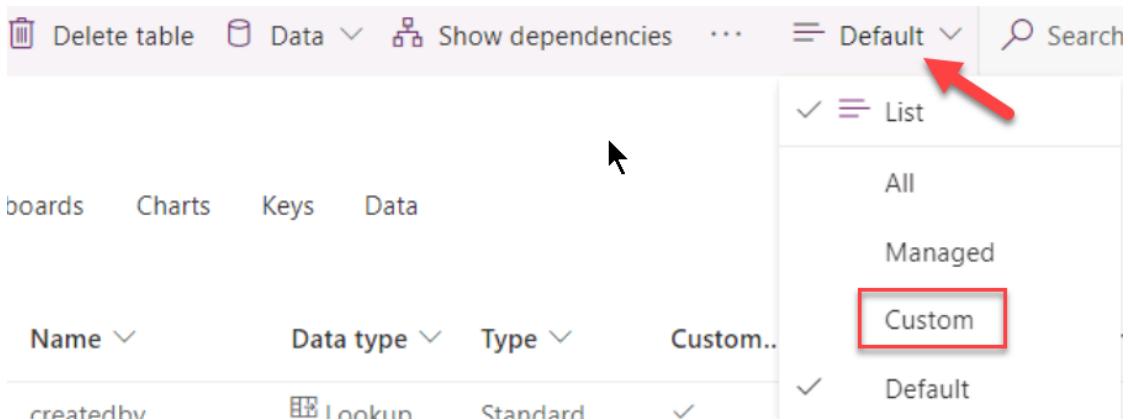
Description ⓘ

Advanced options

Done **Cancel**

The screenshot shows the 'Details' column configuration dialog. It includes fields for Display name, Name, Data type (set to Multiline Text), Required status, and Searchable checkbox. There is also a Description field and an Advanced options section. At the bottom are Done and Cancel buttons.

1. Click **+ Add Column** again.
2. Enter **Photo** for **Display name**, select **Image** for **Data type**, and click **Done**.
3. Click **+ Add Column**.
4. Enter **Resolution** for **Display name**, select **Multiline text** for **Data type**, and click **Done**.
5. Click **+ Add Column**.
6. Enter **Resolved On** for **Display name**, select **Date and time** for **Data type**, and click **Done**.
7. Click **Default Filter** and select Custom



1. You should now see the 5 new Columns you created. Click **Save Table**.

Solutions > Company 311 > Problem Report

Fields

Display name ↑	Name ↓	Data type ↓	Type ↓	Custom... ↓	Required ↓	Searchable... ↓
Details	lh_details	Multi-line Text	Custom	✓	Required	✓
Location	lh_location	Text	Custom	✓	Optional	✓
Photo	lh_photo	Image	Custom	✓	Optional	
Resolution	lh_resolution	Multi-line Text	Custom	✓	Optional	✓
Resolved on	lh_resolvedon	Date and Time	Custom	✓	Optional	✓
Title <small>Primary Name Field</small>	lh_title	Text	Custom	✓	Required	✓

Don't see the items you're looking for? Reset the filter above to see more.

Save Entity

1. Go back to the solution by clicking on the solution name.
2. Click **Publish all customizations** and wait for the publishing to complete.
3. Do not navigate away from this page until all customizations have been published successfully.

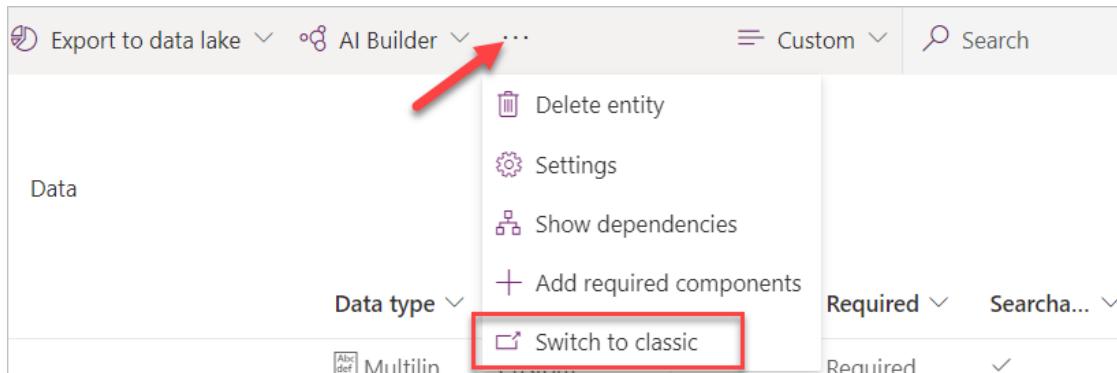
Task 3: Edit status reason Choice

In this task, you will edit the status reason Column of the problem report Table.

1. Make sure you are in the **Company 311** solution.
2. Click to open the **Problem Report** Table.
3. Click on the **More commands** button and select **Switch to classic**.

[!NOTE]

You are switching to classic because the modern solution explorer does not support editing status reason yet but will in the future.



1. Select **Fields** and double click to open the **Status Reason** Column.

A screenshot of the Dynamics 365 classic Fields page for the 'Problem Report' entity. The left sidebar shows navigation options like 'Common', 'Information', 'Forms', 'Views', 'Charts', and 'Fields'. 'Fields' is selected and highlighted with a red box. The main area shows a table of fields with columns for 'Name', 'Schema Name', 'Display Name...', and 'Type'. The 'Status' field is shown with its schema name 'statecode' and type 'Status'. The 'Status Reason' field is shown with its schema name 'statuscode' and type 'Status Reason'. A red arrow points from the text above to the 'Status Reason' column header.

1. Make sure **Active** is selected for **Status** and double click to open the **Active** option.

A screenshot of the Dynamics 365 classic Active options dialog for the 'Status' field. The dialog shows a table with two rows: one for 'Status' and one for 'Active'. The 'Status' row has a dropdown menu with 'Active' selected. The 'Active' row has a dropdown menu with 'Active' selected. A red box highlights the 'Status' row, and another red arrow points from the text above to the 'Active' label in the 'Status' row's dropdown. To the right of the table is a vertical toolbar with buttons for 'Move Up', 'Move Down', 'Edit', 'Add', and 'Delete'.

1. Change the **Label** value to **New** and click **OK**.

Modify List Value

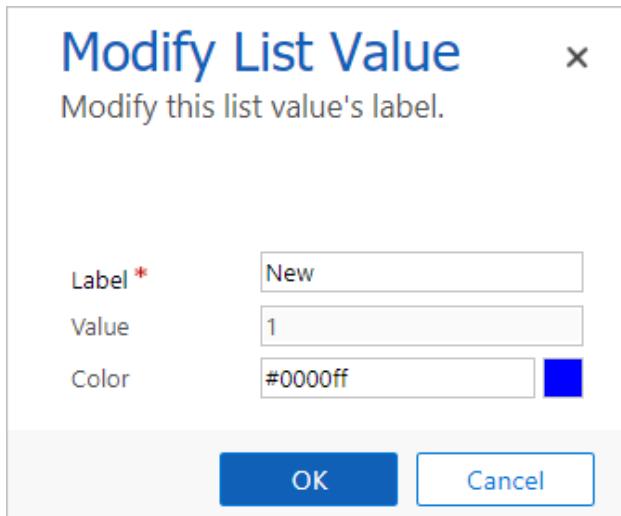
Modify this list value's label.

Label *

Value

Color #0000ff

OK **Cancel**



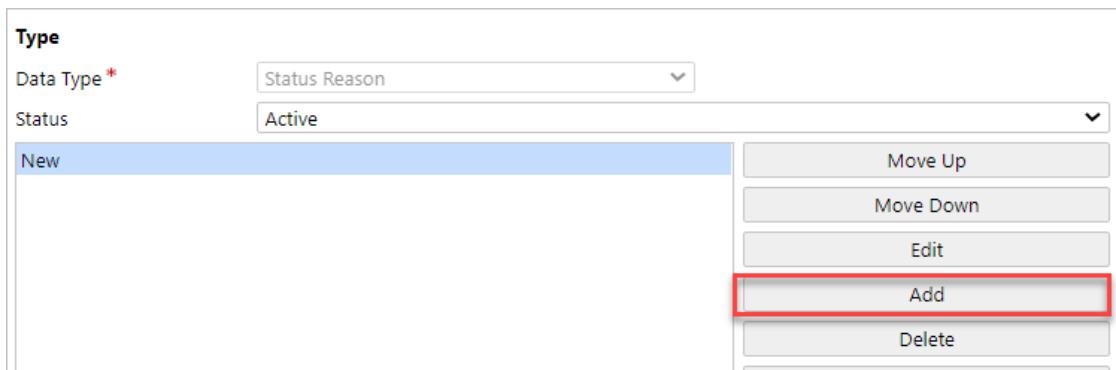
1. Click **Add**.

Type

Data Type *

Status

New	Move Up
	Move Down
	Edit
	Add
	Delete



1. Enter Assigned for Label and click OK.

Add List Value

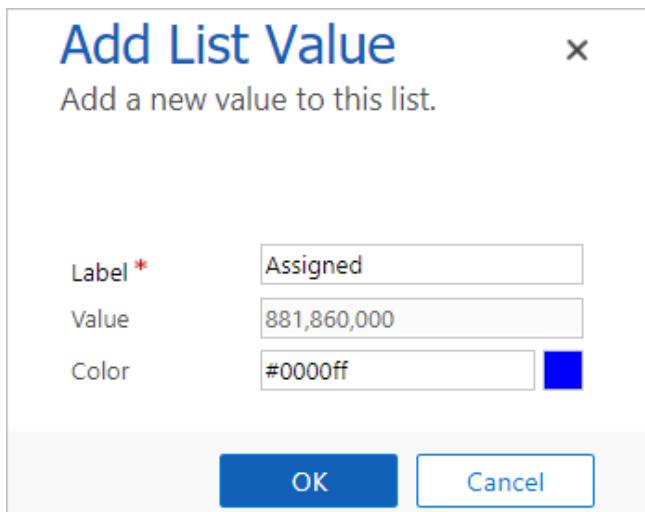
Add a new value to this list.

Label *

Value

Color #0000ff

OK **Cancel**

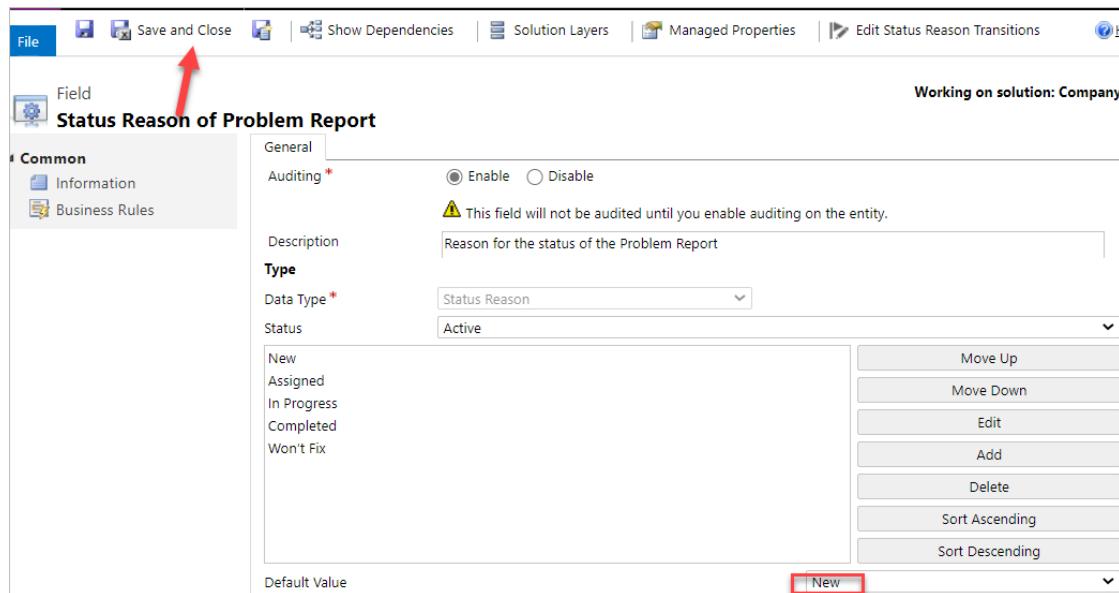


1. Click **Add** again.

2. Enter **In Progress** for **Label** and click **OK**.

3. Click **Add** again.

4. Enter **Completed** for **Label** and click **OK**.
5. Click **Add** one more time.
6. Enter **Won't Fix** for **Label** and click **OK**.
7. You should now have 5 options. Select **New** for **Default Value** and click **Save and Close**.



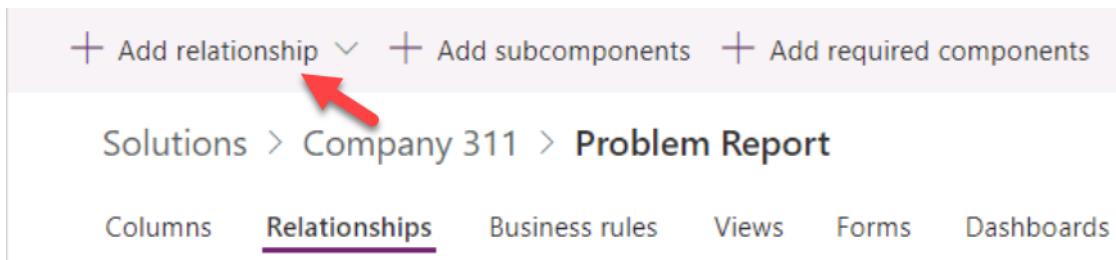
1. Click **Publish** and wait for the publishing to complete.
2. Click Save and Close to close the classic editor.
3. You should now be back on the **Power Apps Maker** portal.

The screenshot shows the Power Apps Maker portal. The navigation bar includes 'Add field', 'Add subcomponents', 'Edit data in Excel', 'Get data', and other options. Below, the 'Solutions > Company 311 > Problem Report' path is shown. The 'Fields' tab is selected. A table displays a single field: 'Details' (Display name: 'Details', Name: 'lh_details').

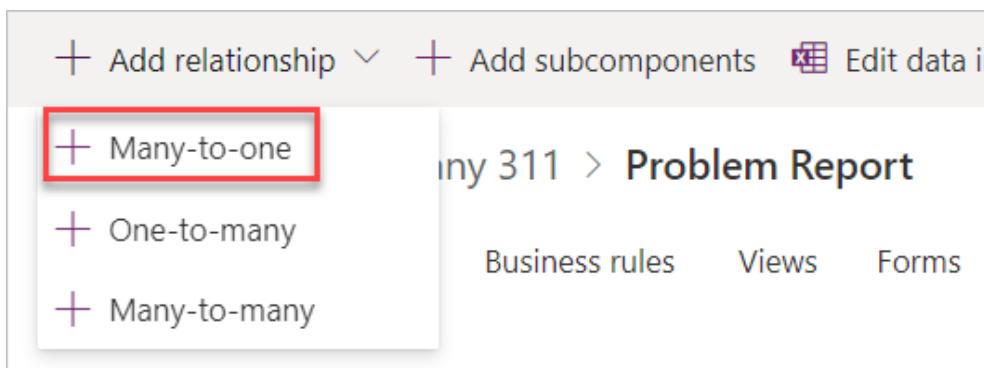
Task 4: Relationships

In this task, you will create many to one relationships between the problem report Table and the building and department Tables.

1. Make sure you are in the **Problem Report** Table.
2. Select the **Relationships** tab and click **+ Add relationship**.



1. Select **Many-to-one**.



1. Select **Building** for **Related (One) Table** and click **Done**.

Many-to-one

Choose the **Related entity** to which to create your relationship lookup. [Learn more](#)

Current (Many)	Related (One)
Entity *	Entity *
Problem Report	* — 1 Building
Lookup field display name *	Building
Lookup field name *	Ih_Building

> General

> Advanced options

Done **Cancel**

1. Click + **Add relationship** again.
2. Select **Many-to-one**.
3. Select **Department** for **Related (One) Table** and click **Done**.
4. Click **Save Table**.
5. Go back to the solution by clicking on the solution name.
6. Click **Publish all customizations** and wait for the publishing to complete.

Exercise 3: Configure form and views

In this exercise, you will configure form and views for the problem report Table.

Task 1: Configure form

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select Solutions and click to open the **Company 311** solution.

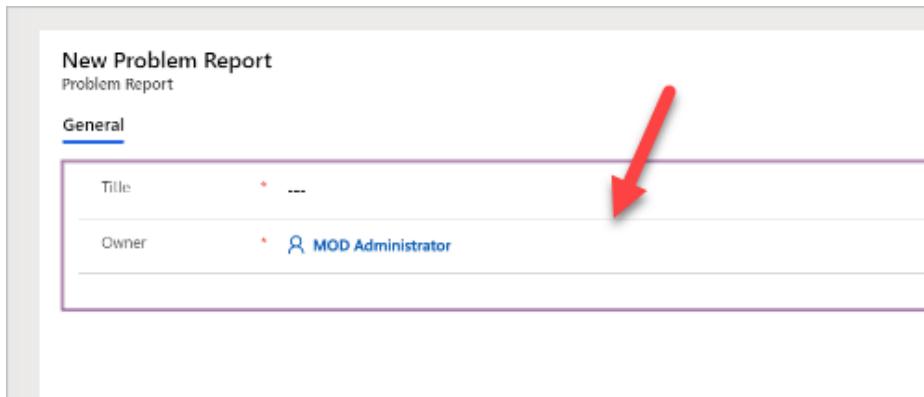
3. Locate and click to open the **Problem Report** Table.
4. Select the **Forms** tab and click to open the **Main** form.

Solutions > Company 311 > **Problem Report**

Columns Relationships Business rules Views **Forms** Dashboards Charts Keys Data

Name ↑ ↴	Form type ↴
Information	... Main
Information	... QuickViewForm
Information	... Card

1. Use the Zoom control at the bottom of the form to make the form large enough for you to work easily. Select the form section.



1. Go to the **Properties** pane, change the **Label** to **Problem details**, and enter **section_problem_report** for **Name**.

Problem details >

Section

Properties

Display options

Section label *

Problem details

Name * ⓘ

section_problem_report

Hide label

Lock section ⓘ

1. While you still have the section selected, go to the **Columns** pane, and click on the **Building** Column. The Building Column will be added to the form.

The screenshot shows the 'Table columns' pane from the SharePoint ribbon. On the left, there's a search bar and a 'Default' dropdown. Below it, there are two buttons: '+ New table column' and 'Show only unused table columns' (which is checked). At the bottom, there are two columns: 'Building' and 'Created By'. The 'Building' column is highlighted with a red box and has a red arrow pointing to its name. To the right of the pane, a preview of the 'New Problem Report' form is shown, featuring a 'General' section with fields for 'Title' and 'Owner'.

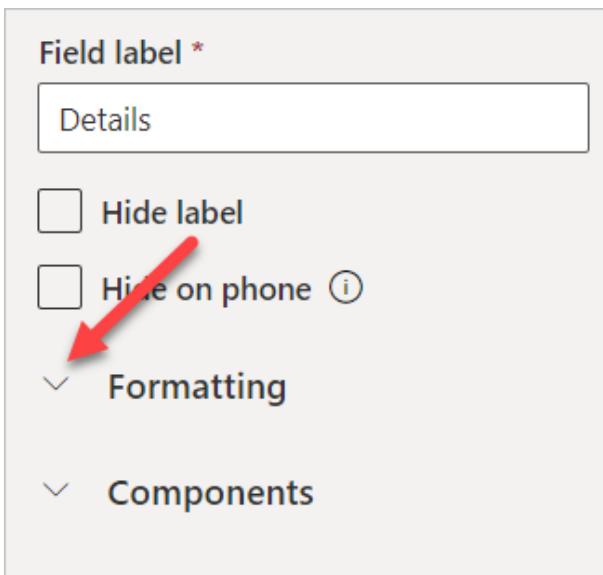
1. Add the **Details**, and **Photo** Columns to the form.
2. Your form should now look like the image below. Select the **Details** Column.

New Problem Report
Problem Report

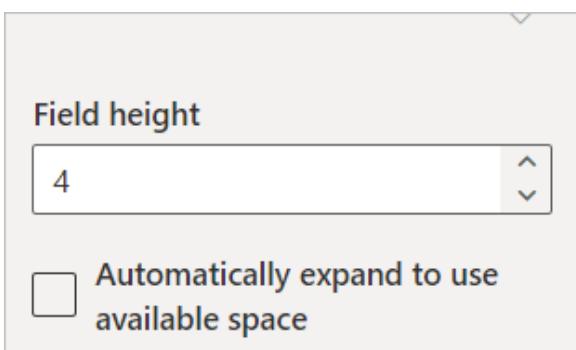
General

Title	---
Owner	* MOD Administrator
Building	---
Details	---
Photo	This record hasn't been created yet. To enable this content, create this record

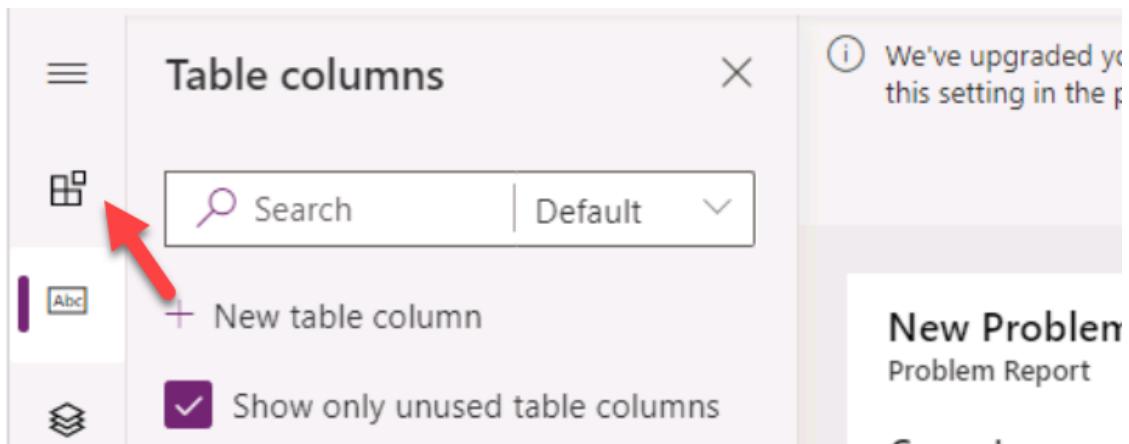
1. Go to the **Properties** pane and click to expand the **Formatting** section.



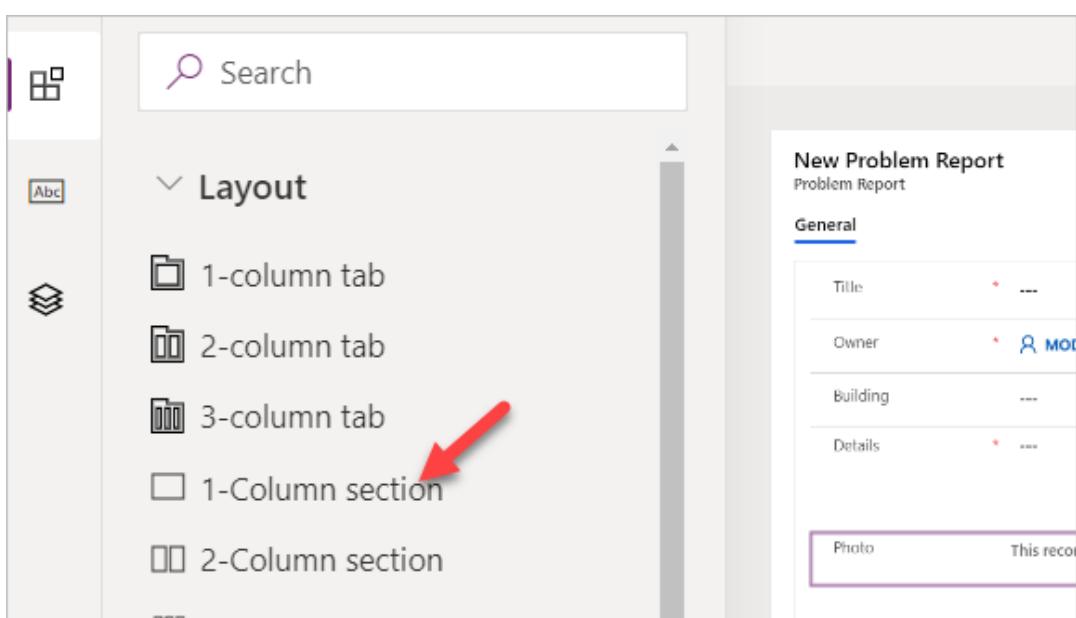
1. Change the **Column height** to **4**.



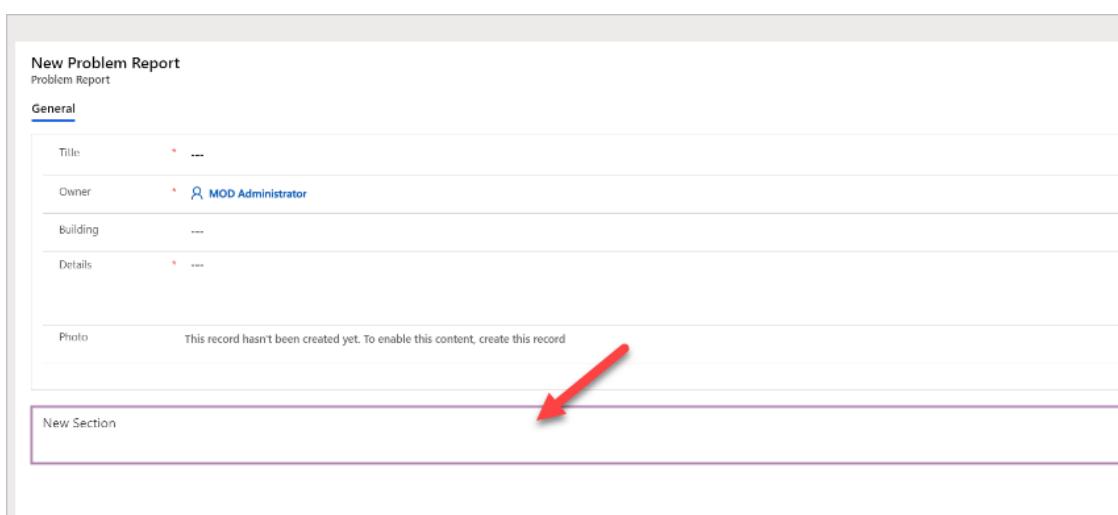
1. Select the **Components** from the toolbar.



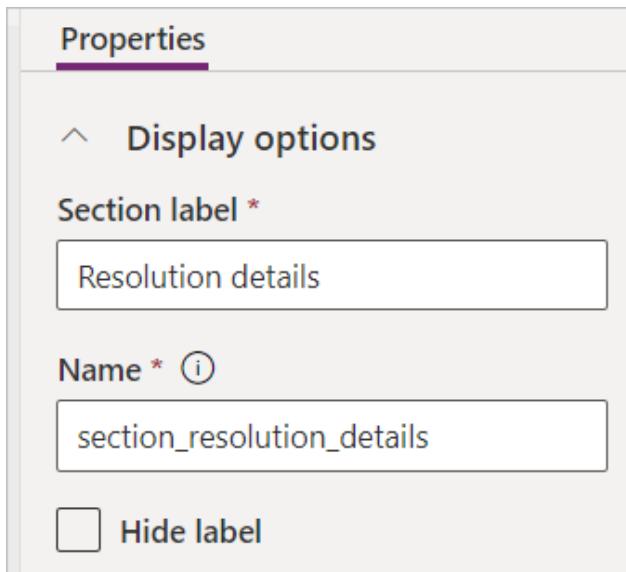
1. Select 1-Column section.



1. A new section should be added to the form. Select the new section.



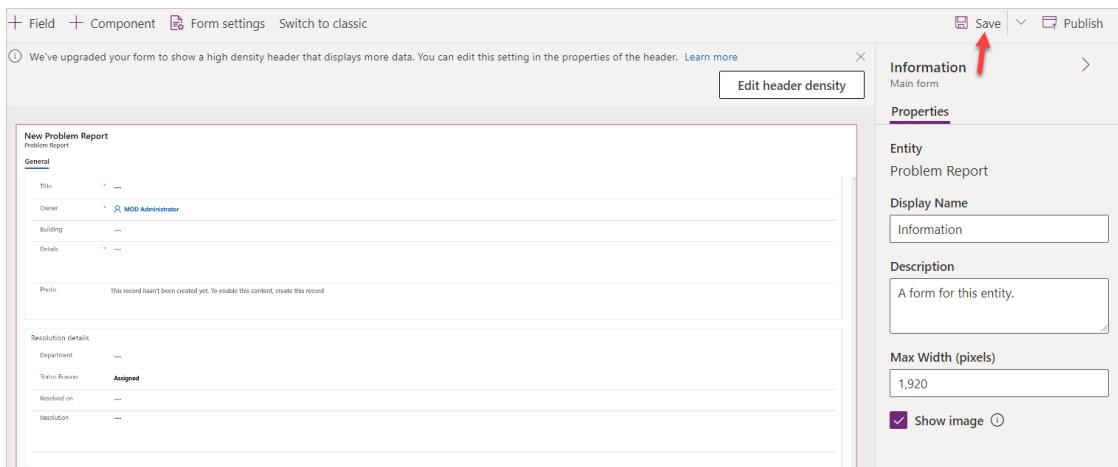
1. Go to the **Properties** pane, change the **Section label** to **Resolution details**, and enter **section_resolution_details** for **Name**.



1. Select **Table columns** from the toolbar.
2. Add **Department**, **Status Reason**, **Resolved on**, and **Resolution** Columns to the **Resolution details** section.

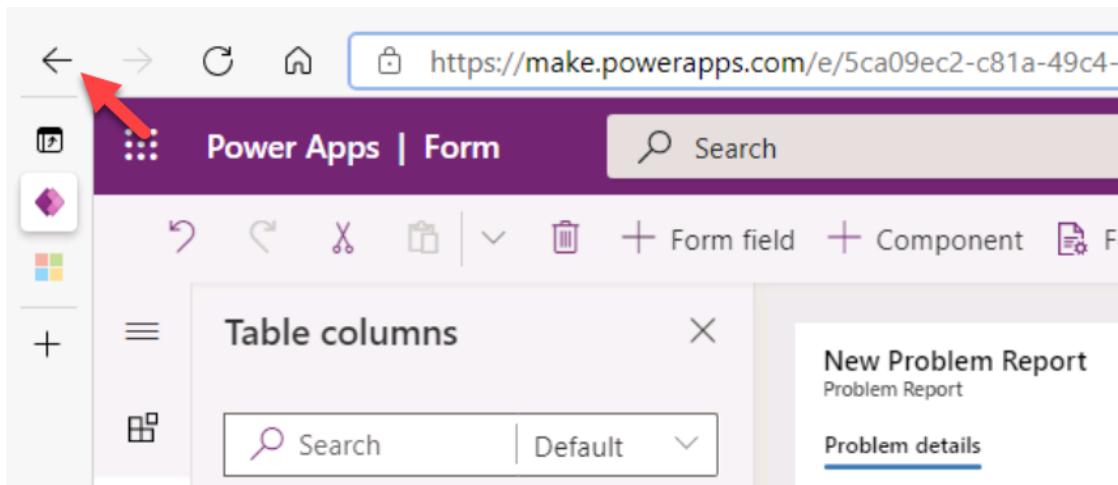
The form has a section titled 'Resolution details' containing four columns: 'Department' (value: ---), 'Status Reason' (value: Assigned), 'Resolved on' (value: ---), and 'Resolution' (value: ---). The 'Resolution' column is highlighted with a purple border.

1. Select the **Resolution** Column.
2. Go to the **Properties** pane and click to expand the **Formatting** section.
3. Change the **Column height** to **4**.
4. Your form should now look like the image below. Click **Save**.



1. Click **Publish** and wait for the publishing to complete.

2. Click on the <- Back button.



1. You should now be back to the Table.

Task 2: Edit view

1. Select the **Views** tab and click to open the **Active Problem Reports** view.

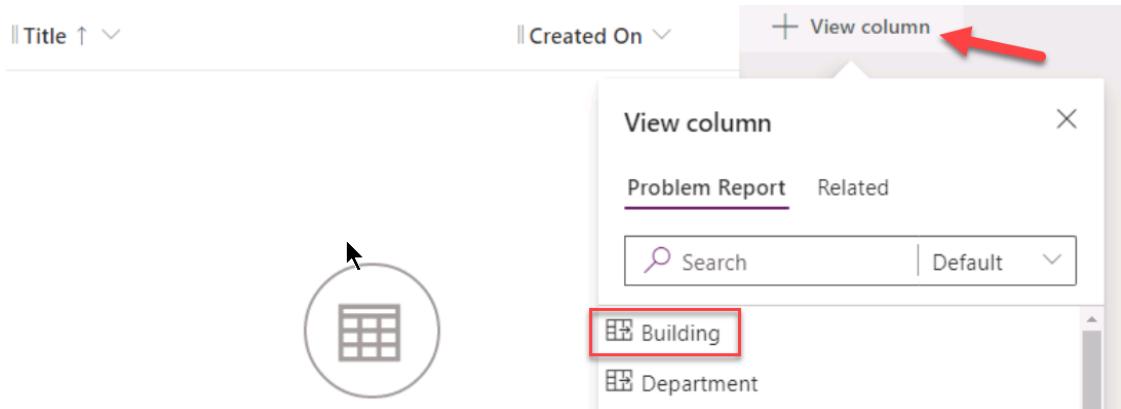
Solutions > Company 311 > **Problem Report**

Columns Relationships Business rules **Views** Forms Dashboards Charts Keys

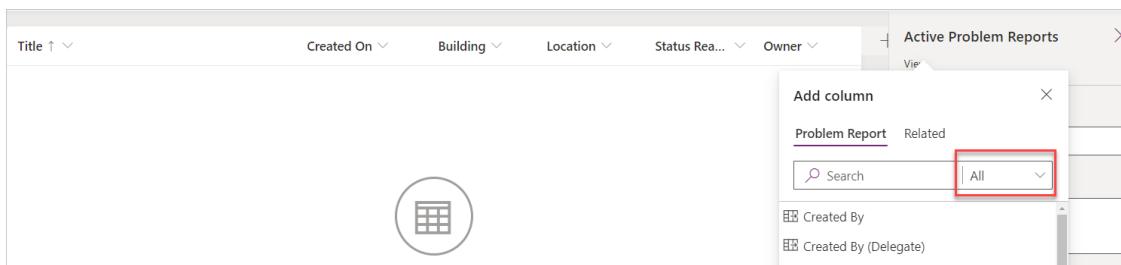
Name ↑ ↓	View type
Active Problem Reports	... Public View Def
Inactive Problem Reports	... Public View

A red arrow points to the 'Active Problem Reports' view entry in the list.

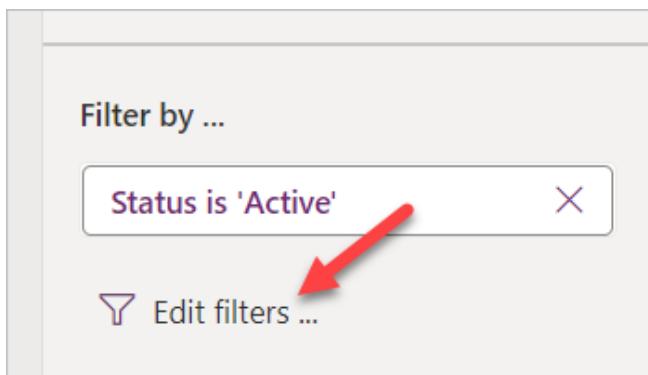
1. Click **+ View column** and select **Building** to add the **Building** column to the view.



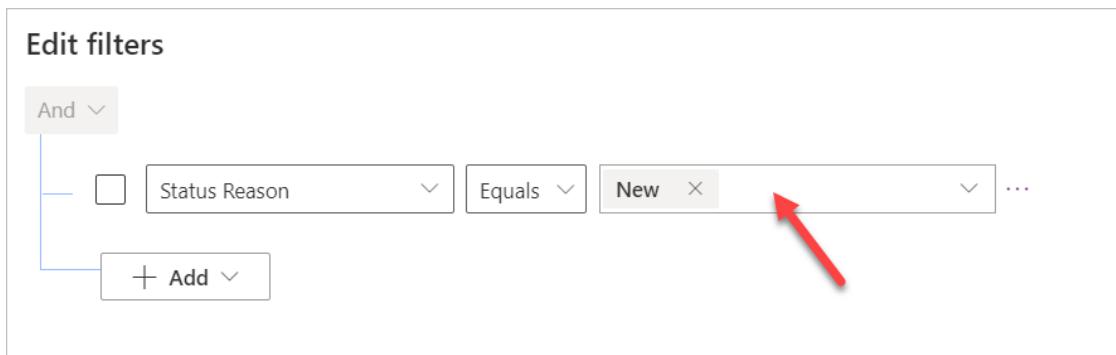
1. Add **Location**, **Status Reason**, and **Owner** columns to the view. You will have to change the column filter to All when adding status reason and owner columns.



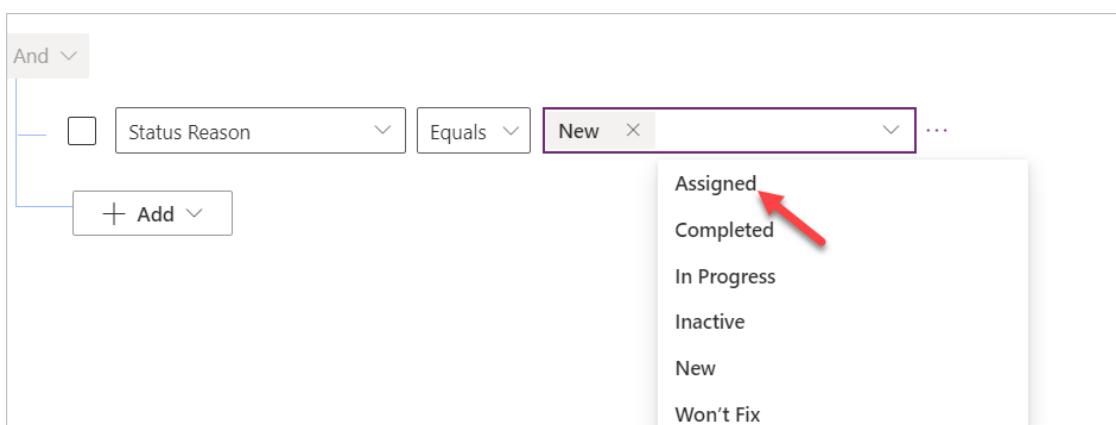
1. Go to the view properties pane and click **Edit filters**.



1. Set the first filter to **Status Reason Equals New**.
2. Click on the Column where **New** is selected.

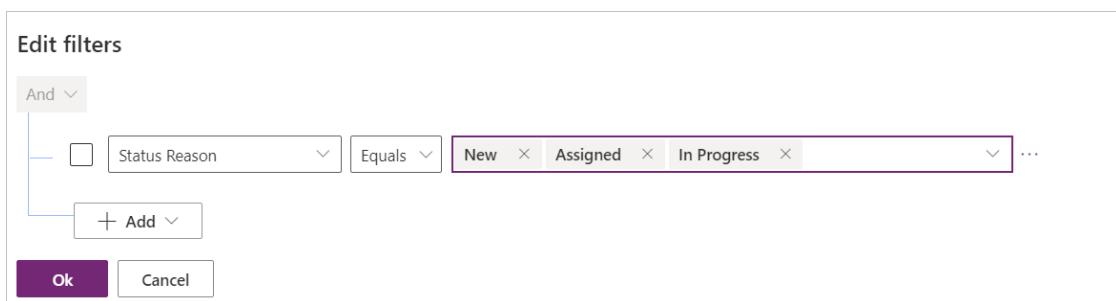


1. Select **Assigned**.



1. Click on the column again and select **In progress**.

2. The filter should now look like the image below. Click **OK**.

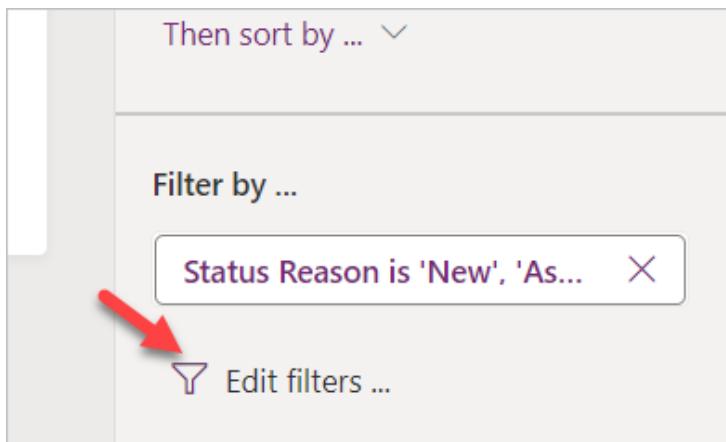


1. Click **Save**.

Task 3: Create view from existing

In this task, you will create a new view from the Active Problem Reports view.

1. Click **Edit filters**.



1. Remove **In Progress** from the filter.

The screenshot shows the 'Edit filters' dialog with an 'And' condition. It contains three filter items: 'Status Reason Equals New', 'Status Reason Equals Assigned', and 'Status Reason Equals In Progress'. A red arrow points to the 'In Progress' item.

1. Remove **Assigned** and **New** values from the filter.

2. Select **Completed**.

The screenshot shows the 'Edit filters' dialog after removing 'Assigned' and 'New'. The 'Status Reason Equals' condition now has a dropdown menu open, listing 'Assigned', 'Completed', 'In Progress', 'Inactive', 'New', and 'Won't Fix'. A red arrow points to the 'Completed' option.

1. Add **Won't Fix** and **Inactive** values to filter.

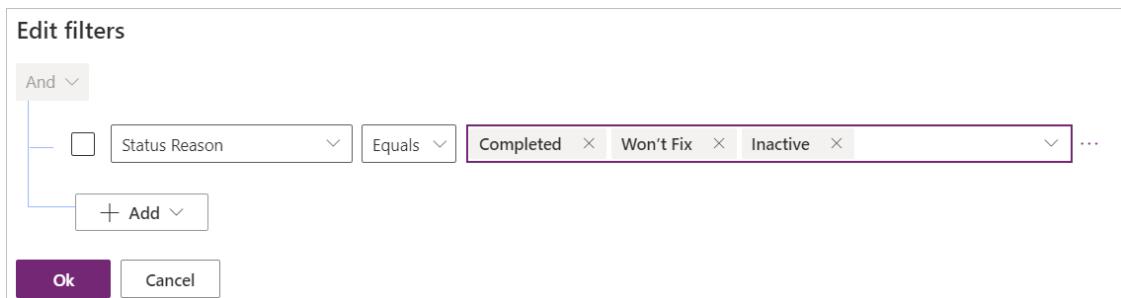
2. The filter should now look like the image below. Click **OK**.

Edit filters

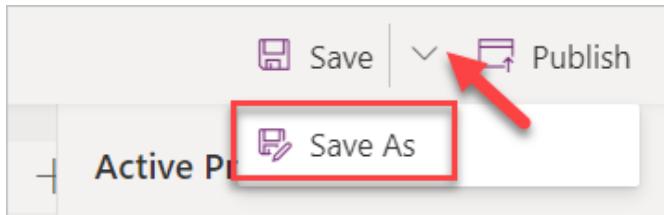
And Status Reason Equals Completed × Won't Fix × Inactive × ...

+ Add

Ok Cancel



1. Click on the chevron button next to the save button and select **Save As**.



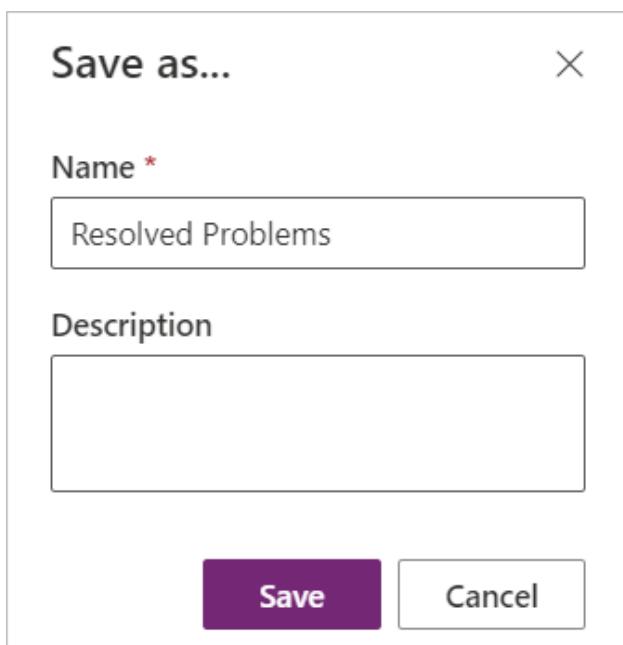
1. Enter **Resolved Problems** for **Name** and click **Save**.

Save as...

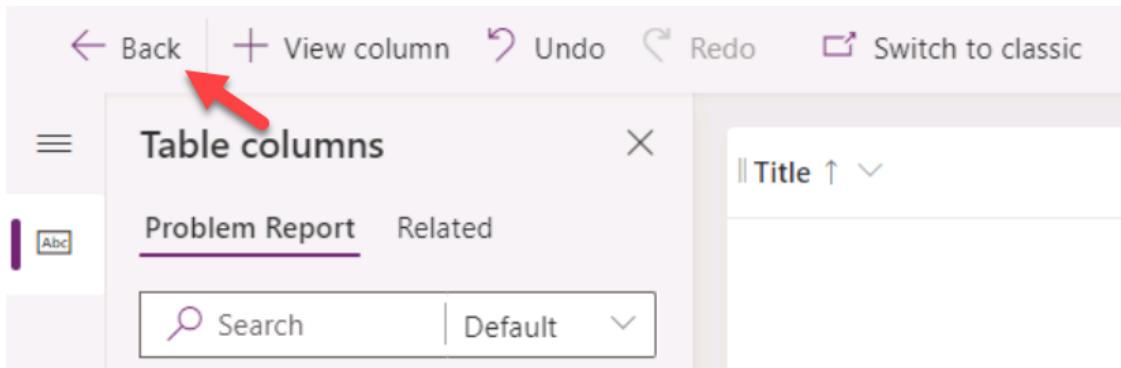
Name *

Description

Save Cancel



1. Click on the <-Back Button.



1. Go to the solution by clicking on the solution name.

Solutions > Company 311 > Problem Report

Columns Relationships Business rules Views Forms Dashboards Ch

Name ↑ ▾

1. Click **Publish all customizations** and wait for the publishing to complete.

+ New ▾ + Add existing ▾ Delete Export **Publish all customizations** ...

Solutions > Company 311

Display name ▾	Name
----------------	------

Exercise 4: Compose model-driven application

In this exercise, you will create model-driven application.

Task 1: Create new model-driven application

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select Solutions and click to open the **Company 311** solution.
3. Click **+ New | App | Model-driven app**.



1. Enter **Company 311 Admin** for name and click **Done**.

Create a New App

Name :* Company 311 Admin

Unique Name :* lh_Company311Admin

Description:

Icon: Use Default Image

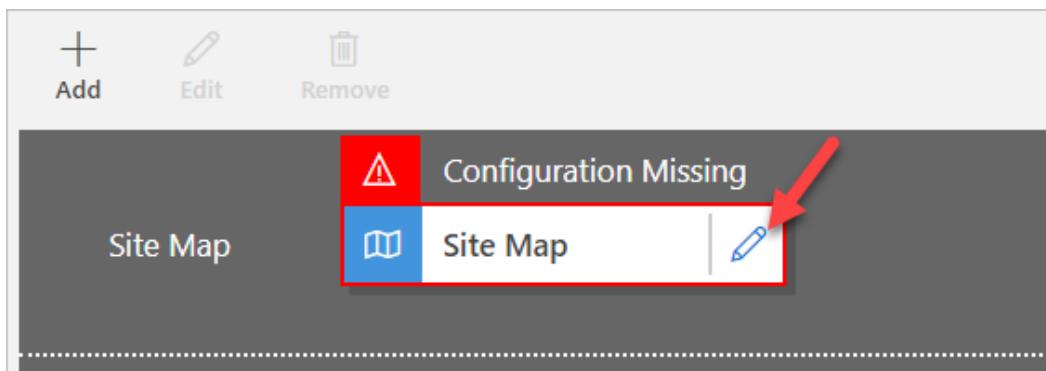
App Tile:

Company 311 Admin

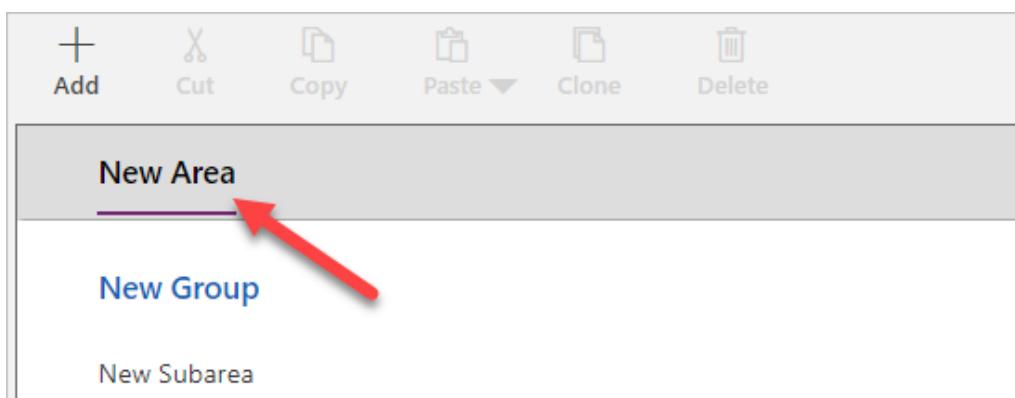
Done Cancel

This is a screenshot of the 'Create a New App' dialog. It shows fields for Name (Company 311 Admin), Unique Name (lh_Company311Admin), Description, and Icon (Use Default Image). An app tile preview is shown on the right. Buttons for Done and Cancel are at the top right.

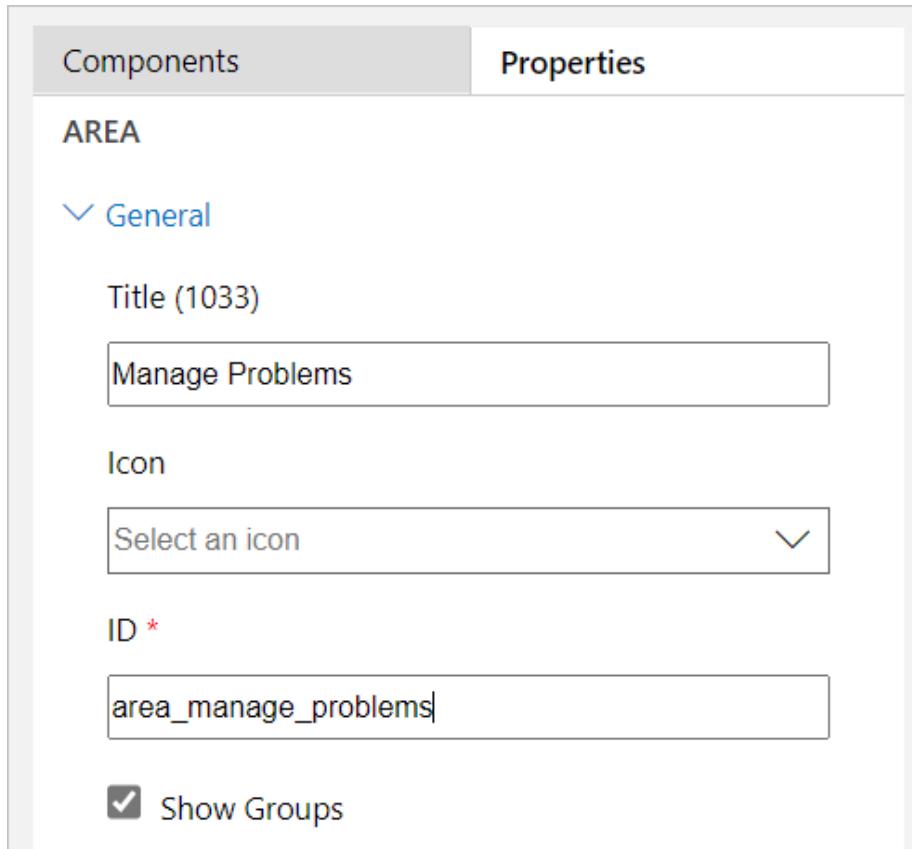
1. Click **Edit Site Map**.



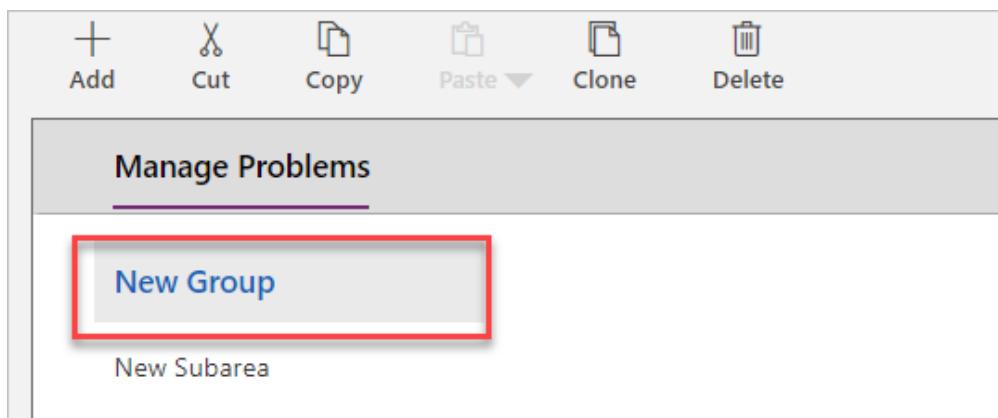
1. Select the **New Area**.



1. Go to the **Properties** pane, enter **Manage Problems** for **Title**, and enter **area_manage_problems** for **ID**.



1. Select the **New Group**.



1. Go to the **Properties**, enter **Problems** for **Title**, and enter **group_problems** for **ID**.

Components Properties

GROUP

General

Title (1033)

Problems

ID *

group_problems

Advanced

This screenshot shows the 'Properties' pane in a software application. At the top, there are tabs for 'Components' and 'Properties'. Below the tabs, the word 'GROUP' is displayed. Under 'GROUP', there is a section titled 'General' with a blue downward arrow icon. Inside the 'General' section, there are two fields: 'Title (1033)' containing the text 'Problems' and 'ID *' containing the text 'group_problems'. Below the 'General' section, there is another section titled 'Advanced' with a blue right-pointing arrow icon.

1. Select the New Subarea.

Add Cut Copy Paste ▾ Clone Delete

Manage Problems

Problems

New Subarea

This screenshot shows the 'Manage Problems' interface. At the top, there is a toolbar with buttons for 'Add' (plus sign), 'Cut' (scissors), 'Copy' (copy icon), 'Paste' (paste icon with a dropdown arrow), 'Clone' (clone icon), and 'Delete' (trash can icon). Below the toolbar, the title 'Manage Problems' is displayed. Underneath the title, there is a section titled 'Problems'. Within the 'Problems' section, there is a button labeled 'New Subarea' which is highlighted with a red rectangular border.

1. Go to the **Properties** pane, select **Table** for **Type**, and select **Problem Report** for **Table**.

Components Properties

SUB AREA

General

Type

Entity

Entity *

Problem Report

URL

1. Click **+ Add**.

Manage Problems

Problems

Problem Reports

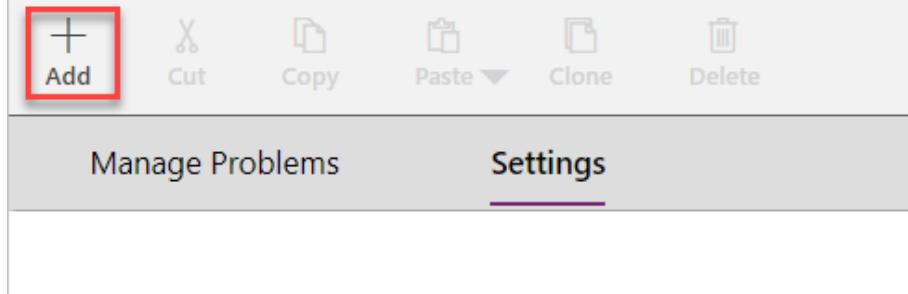
1. Select **Area**.

2. Select the **New Area** you just added.

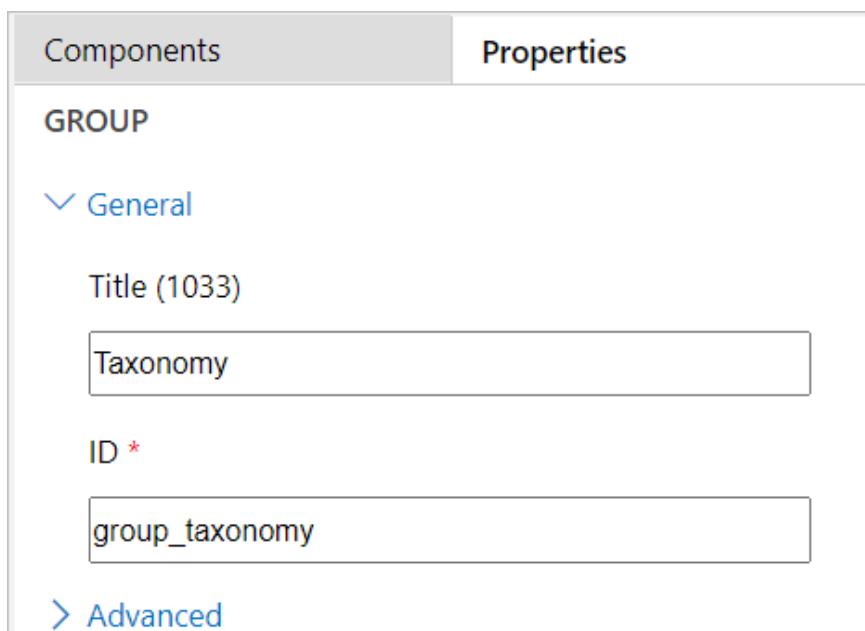
Manage Problems

New Area

1. Go to the **Properties** pane, enter **Settings** for **Title**, and enter **area_settings** for **ID**.
2. Select the **Settings** area and click **+ Add**.



1. Select **Group**.
2. Select the **New Group** you just added.
3. Go to the **Properties** pane, enter **Taxonomy** for **Title**, and enter **group_taxonomy** for **ID**.



1. Select the **Taxonomy** group you just added and select the **Components** pane.

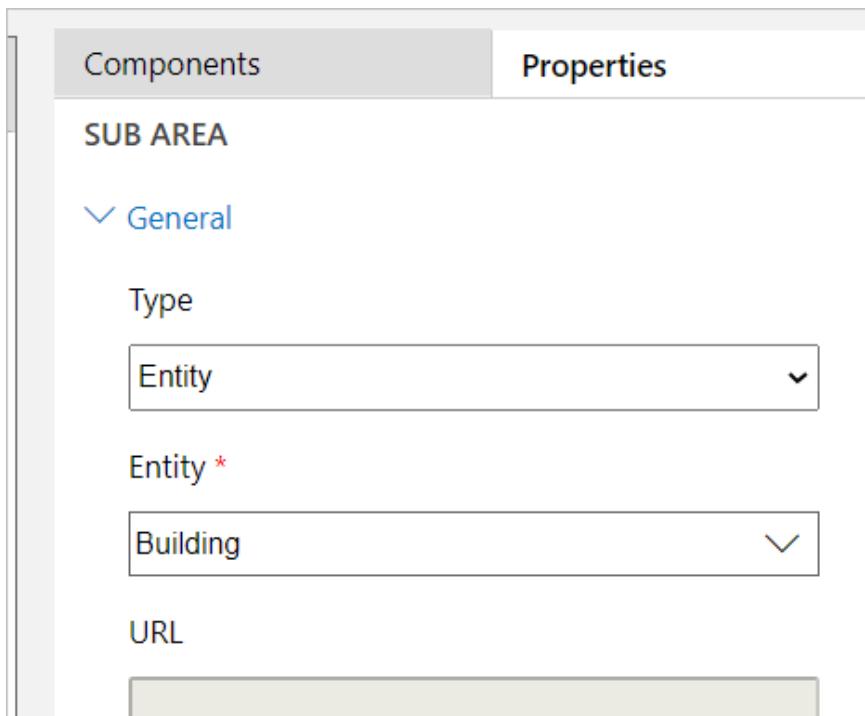
A screenshot of a software interface titled "Taxonomy". The top navigation bar includes "Add", "Cut", "Copy", "Paste", "Clone", and "Delete" buttons. Below the navigation is a "Manage Problems" section and a "Settings" tab, which is currently active. On the left, there's a "Taxonomy" section. To the right, the "Components" tab is selected, indicated by a red arrow. The "Properties" tab is also visible. Under the "Components" tab, there's a "GROUP" section with a "General" subsection. It shows a "Title (1033)" field containing "Taxonomy" and an "ID *" field.

1. Drag **Subarea** and drop it under the **Taxonomy** group.

A screenshot of the same software interface. The "Components" tab is still selected. In the center, there's a "Taxonomy" group with a "Subarea" component being dragged from the "Components" pane. A dashed red arrow indicates the drag operation. The "Properties" tab is visible on the right. The "Components" pane contains three items: "Area", "Group", and another "Subarea" component.

1. Select the **New Subarea**.

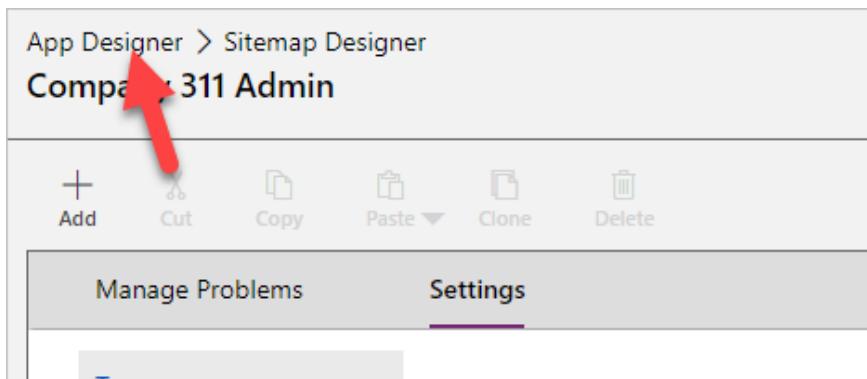
2. Go to the **Properties** pane, select **Table** for **Type**, and select **Building** for **Table**.



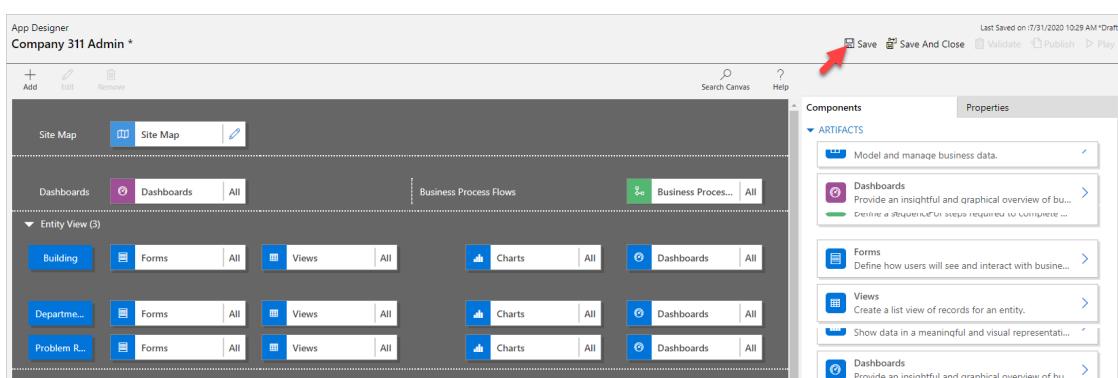
1. Add another **Subarea** component to the **Taxonomy** group.
2. Select the **New Subarea**.
3. Go to the **Properties** pane, select **Table** for **Type**, and select **Department** for **Table**.
4. The sitemap should now look like the image below. Click **Save** to save the sitemap.

The screenshot shows the 'Sitemap Designer' interface. On the left, there's a sidebar with 'Manage Problems' and 'Settings' tabs, and a list of categories: 'Taxonomy' (selected), 'Buildings', and 'Departments'. On the right, the 'Components' pane shows a 'SUB AREA' component with its 'General' properties. The 'Type' dropdown is set to 'Entity', and the 'Entity *' dropdown is set to 'Department'. The top right of the screen has a toolbar with 'Save' and 'Save And Close' buttons, with a red arrow pointing to the 'Save' button.

1. Click **Publish** to publish the sitemap and wait for the publishing to complete.
2. Go back to the **App Designer**.



1. The application should now have three Tables. Click **Save** to save the application.



1. Click **Publish** to publish the application and wait for the publishing to complete.
2. Close the app designer browser tab or window.
3. Click **Done**.

Currently creating a new model-driven app

When you're done creating the new model-driven app, click Done below to return to the page. This will refresh the page and fetch your changes.

Done

Exercise 5: Input data

In this exercise, you will input data.

Task 1: Input data

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select **Apps** and click to open the **Company 311 Admin** application you created.

The screenshot shows the Power Apps maker portal interface. On the left, there's a navigation bar with options like Home, Learn, Apps (which is selected), Create, Data, Flows, and Chatbots. The main area is titled 'Apps' and shows a table with two rows. The columns are 'Name' (with a red arrow pointing to it) and 'Modified'. The first row is 'Company 311 Admin' (modified 6 min ago) and the second is 'Solution Health Hub' (modified 6 d ago).

Name	Modified
Company 311 Admin	6 min ago
Solution Health Hub	6 d ago

1. Click **Change area**.

The screenshot shows the 'Active Problem Reports' screen. The left sidebar has sections for Home, Recent, Pinned, and Problems, with 'Problem Reports' selected. The main area is titled 'Active Problem Reports' and shows a table with a single row: '0 - 0 of 0 (0 selected)'. At the bottom left, there's a blue button labeled 'Manage Problems' with a red arrow pointing to it.

1. Select **Settings** area.
2. Select **Departments** and click **+ New**.

The screenshot shows the 'Active Departments' list page. The left sidebar includes 'Home', 'Recent', 'Pinned', 'Taxonomy', 'Buildings', and 'Departments'. The main area displays a table with columns for Name, Owner, and Status. The top navigation bar features 'Show Chart', '+ New' (highlighted with a red arrow), 'Delete', 'Refresh', and 'Email a Link'. A search bar at the top right allows filtering by Name.

1. Enter **Facility Maintenance** for Name and click **Save**.

The screenshot shows the 'New Department' form. The 'General' tab is active. The 'Name' field contains 'Facility Maintenance' with a red asterisk indicating it is required. The 'Owner' field is set to 'MOD Administrator'. Other tabs like 'Locations' and 'Employees' are visible but inactive.

1. Click **+ New** again.

2. Enter **Human Resources** for Name and click **Save**.

3. Click **+ New** one more time.

4. Enter **Marketing** for Name and click **Save**.

5. Select **Departments**.

6. You should now have three department Rows. Select **Buildings**.

The screenshot shows the 'Active Departments' page. On the left, there is a sidebar titled 'Taxonomy' with two items: 'Buildings' and 'Departments'. A red arrow points to the 'Buildings' item. The main content area lists three departments: 'Facility Maintenance', 'Human Resources', and 'Marketing'.

1. Click **+ New**.
2. Enter **San Francisco Main Campus** for Name and click **Save & Close**.
3. Click **+ New** again.
4. Enter **London Paddington** for Name and click **Save & Close**.
5. You should now have two building Rows. Click **Change area**.

The screenshot shows the 'Active Buildings' page. On the left, there is a sidebar titled 'Taxonomy' with 'Buildings' selected, indicated by a blue highlight. A red arrow points to the 'Settings' button at the bottom left of the page. The main content area lists two buildings: 'London Paddington' and 'San Francisco Main Campus'. At the bottom, there is a navigation bar with buttons for 'All', '#', A, B, C, D, E, F, G, and H, and a status message '1 - 2 of 2 (0 selected)'.

1. Select **Manage Problems**.
2. Click **+ New**.

The screenshot shows the 'Active Problem Reports' section of a CRM interface. On the left, there's a navigation bar with 'Home', 'Recent', and 'Pinned' items, followed by 'Problems' and 'Problem Reports'. Below this is a 'Manage Problems' button. The main area has a header 'Active Problem Reports' with sorting options for 'Title ↑' and 'Created On ↓'. A navigation bar at the bottom includes 'All', '#', and letters A through I. At the bottom right, it says '0 - 0 of 0 (0 selected)'.

1. Enter **Broken door** for **Title**, select **San Francisco Main Campus** for **Building**, enter **The main entrance door will not open all the way** for **Details**, and click **Save**

The screenshot shows the 'New Problem Report' form under the 'General' tab. It contains the following fields:

Title	* Broken door
Owner	* MOD Administrator
Building	San Francisco Main Campus
Details	* The main entrance door will not open all the way.
Photo	This record hasn't been created yet. To enable this content, create this record
Resolution details	

1. Click on the **Photo** Column.

BD Broken door
Problem Report

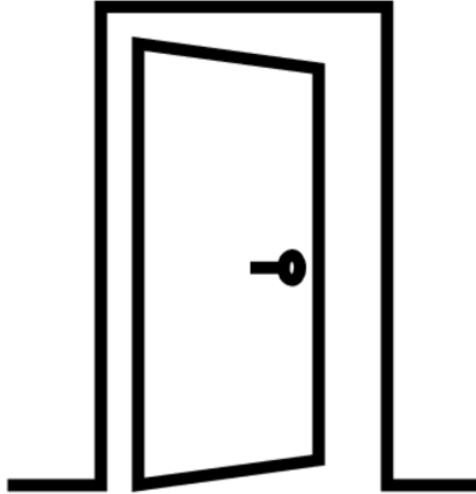
General Related

Title	* Broken door
Owner	* MOD Administrator
Building	San Francisco Main Campus
Details	* The main entrance door will not open all the way.
Photo	<div style="border: 1px solid #ccc; padding: 10px; text-align: center;"> Upload an image  </div>

1. Select an image from your device. The sample imaged displayed below can be found here: F:\Instructions\Labs\02\media\image89.png
2. The image should now show on the form.

Details * The main entrance door will not open all the way.

Photo



1. Click **Save & Close**.
2. Close the browser tab.

Exercise 6: Import data

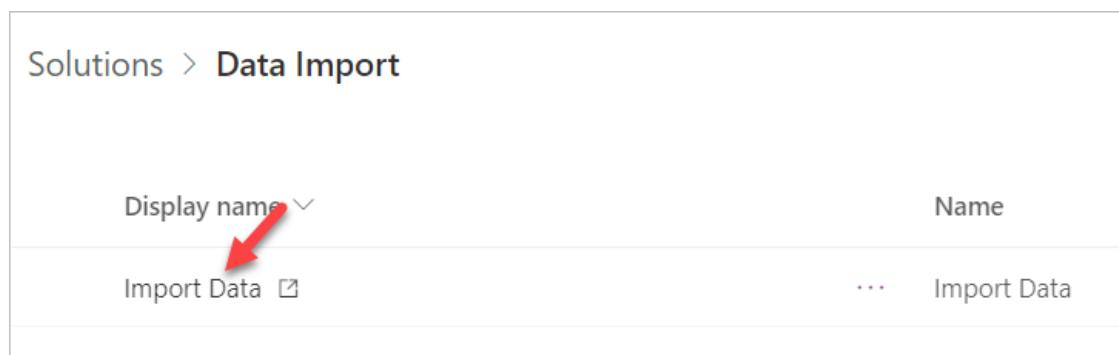
In this exercise, you will import sample data into your environment. Rows are imported by a Power Automate flow that you will first import using a solution.

Task 1: Import solution

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select **Solutions** and click **Import**.
3. Click **Choose File**.
4. Select the **DataImport.zip** solution file located in the lab resources folder and click **Open**.
5. Click **Next**.
6. Click **Import** and wait for the message **Solution "Data Import" imported successfully.** to appear.
7. Click **Publish all customizations** and wait for the publishing to complete.

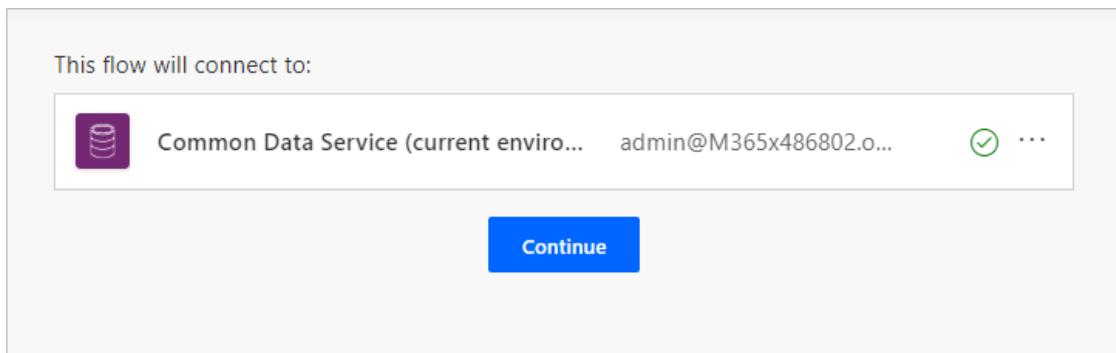
Task 2: Review and run flow

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select **Solutions** and click to open the **Data Import** solution you imported.
3. Click to open the **Import Data** flow. Click the **Get Started** button on the **Welcome to Power Automate** window.

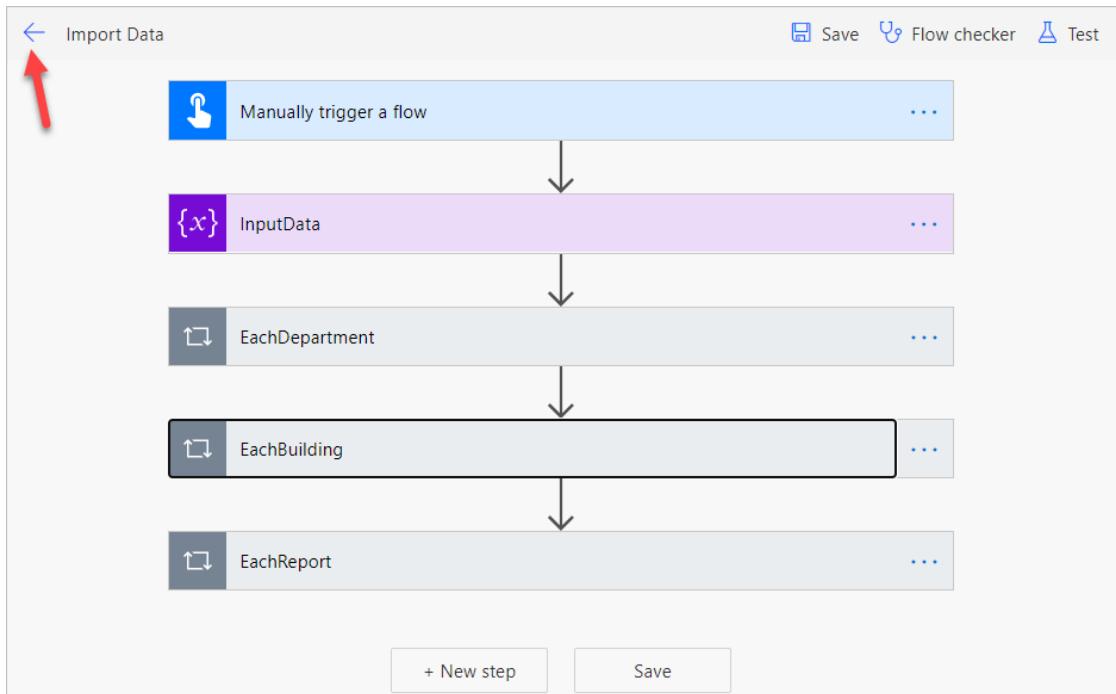


1. Click **Edit**.

2. Click **Continue**.



1. Click to expand the **Input Data** step.
2. Review the Json text in the value Column. This is the data that will be imported into your environment
3. Expand the **Each Department** for each control
4. Expand and review the **Upsert Department** step.
5. Expand and review the rest of the steps.
6. Click **Save** to save the flow.
7. Click on the button and go back to the flow details page.



1. Click **Run**.

2. Click **Run flow**.
3. Click **Done**.
4. Wait for the flow run to complete. Click on the **Refresh** button to check if the flow run completed successfully.

Start	Durati...	Status
Aug 9, 09:22 PM (1 min ago)	00:01:11	Succeeded

1. Close the flow editor browser window or tab.
2. Click **Done** on the popup

Task 3: Review imported data

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select **Apps** and click to open the **Company 311 Admin** application.
3. Select Problem Reports and change the view to My Reports. You should see at least three new Rows.

Created On	Building	Location	Status
7/17/2020 6:0...	Alpine Ski Hou...	floor 4, uni...	New
7/4/2020 6:00 ...	Fabrikam Resid...	Building g...	Com
8/2/2020 6:00 ...	Fabrikam Resid...	Left side o...	In Pr

1. Click to open one of the **Problem Report** Rows.

2. Click on the **Building** lookup and make sure building Rows were imported.

General Related

Title	* Leaking toilet															
Owner	* MOD Administrator															
Building	Alpine Ski House <input type="button" value="X"/> <input type="button" value="Search"/>															
Details	<table border="1"><thead><tr><th></th><th>Buildings</th><th>Recent records</th></tr></thead><tbody><tr><td></td><td>Contoso Suites 8/9/2020 9:23 PM</td><td></td></tr><tr><td></td><td>Fabrikam Residences 8/9/2020 9:23 PM</td><td></td></tr><tr><td></td><td>Fourth Coffee House 8/9/2020 9:23 PM</td><td></td></tr><tr><td></td><td>London Paddington 7/31/2020 11:28 AM</td><td></td></tr></tbody></table>		Buildings	Recent records		Contoso Suites 8/9/2020 9:23 PM			Fabrikam Residences 8/9/2020 9:23 PM			Fourth Coffee House 8/9/2020 9:23 PM			London Paddington 7/31/2020 11:28 AM	
	Buildings	Recent records														
	Contoso Suites 8/9/2020 9:23 PM															
	Fabrikam Residences 8/9/2020 9:23 PM															
	Fourth Coffee House 8/9/2020 9:23 PM															
	London Paddington 7/31/2020 11:28 AM															
Photo																

1. Scroll down and click on the **Department** lookup.
2. Make sure the department Rows got imported.

Bonus exercise

- Deal with problem report assignment within a department.

lab: title: 'Lab: Canvas app' module: 'Module 4: Building canvas apps'

[!NOTE] Effective November 2020: - Common Data Service has been renamed to Microsoft Dataverse. [Learn more](#) - Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. [Learn more](#)

This content will be updated soon to reflect the latest terminology.

Lab 03: Canvas app

In this module you will design and build a canvas app for the company employees to submit problem reports.

What you will learn

- Import and use a pre-built component library
- Create a Power Apps canvas app
- Connect to a data source
- Filter data
- Create data Rows
- Use images with data Rows
- Embed canvas Power App into Microsoft Teams

High-level lab steps

- Import company components
- Create app and layout main screen (including list of my items)
- Submit New Report
- Test
- Embed canvas app in Microsoft Teams

Prerequisites

- Must have completed **Lab 02: Data model and model-driven app**

Detailed steps

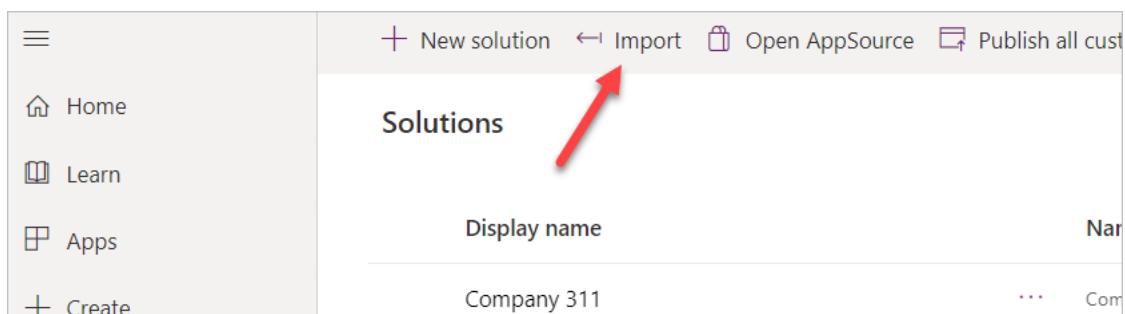
Exercise 1: Create canvas application

In this exercise, you will import a solution with shared components, create a view for the problem report Table, and create a canvas application.

Task 1: Import component library solution

In this task, you will import the shared components solution into your environment. This shared component library was built by another team at your company.

1. Navigate to the [Power Apps maker portal](#) page and make sure you are in the correct environment.
2. Select **Solutions** and click **Import**.



1. Click **Choose File**.
2. Go to the course resources folder, select the **Shared components** solution, and click **Open**.
3. Click **Next**.
4. Click **Import**.
5. Click **Publish All Customizations** and wait for the publishing to complete.
6. Click **Close**.
7. You should now see the **Shared Components** solution you imported. Click to open the **Shared Components** solution you imported.

8. The solution should have one item in it, **Lamna Healthcare Shared Components**.

Solutions > Shared Components	
Display name	Name
Lamna Healthcare Shared Components	... lh_lamnahealthcaresharedcomponents_77ba0

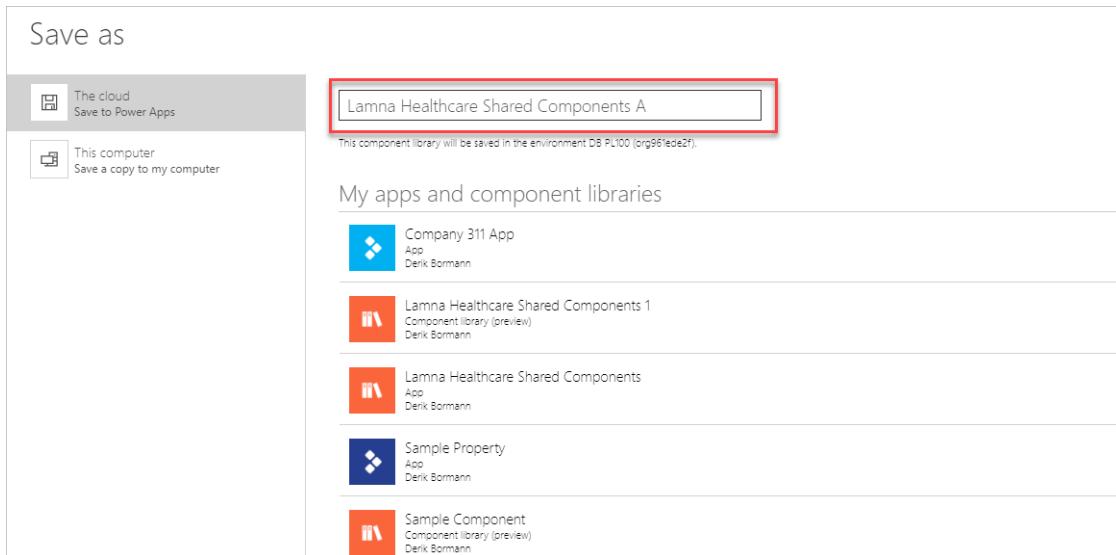
[!IMPORTANT] There is an issue where importing the app as part of a solution may not add it to your components library. The following steps are designed to resolve the issue.

1. Navigate to **Apps**, Select the **Lamna Healthcare Shared Components App**.
2. Click the **Edit Icon** to edit the app.

The screenshot shows the Microsoft Teams Apps page. At the top, there is a navigation bar with icons for New app, Edit, Play, Share, Export package (preview), Add to Teams, and Monitor. Below the navigation bar, the title 'Apps' is displayed. Under the 'Apps' tab, there is a list of apps. One app, 'Lamna Healthcare Shared Components', is selected and highlighted with a purple background. A context menu is open next to the selected app, with a red arrow pointing to the 'Edit' option. The context menu also includes 'Play', 'Share', 'Export package (preview)', and 'Add to Teams'. The list of apps includes:

Name	Modified
Company 311 App	1 min ago
Lamna Healthcare Shared Components	53 min ago
Sample Property	
Company 311 Admin	
Real Estate	
Asset Checkout	

1. After the app opens, click **File > Save As**.
2. Save the app as **Lamna Healthcare Share Components A**.



1. Click **OK**.
2. Close the **Lamna Healthcare Shared Components** tab in your browser.

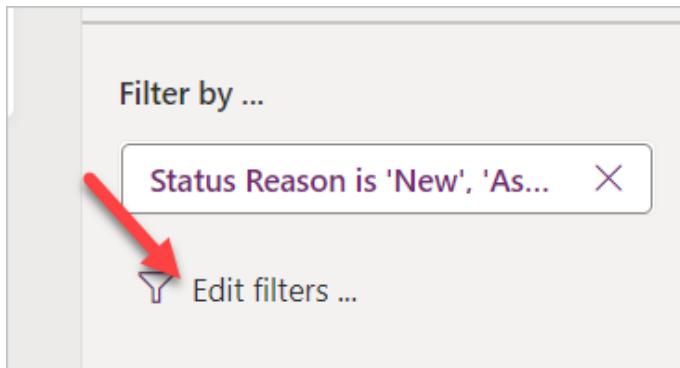
Task 2: Create view

In this task, you will create a view that will show the current user's problem reports. Later you will use this view with the filter function in the canvas app.

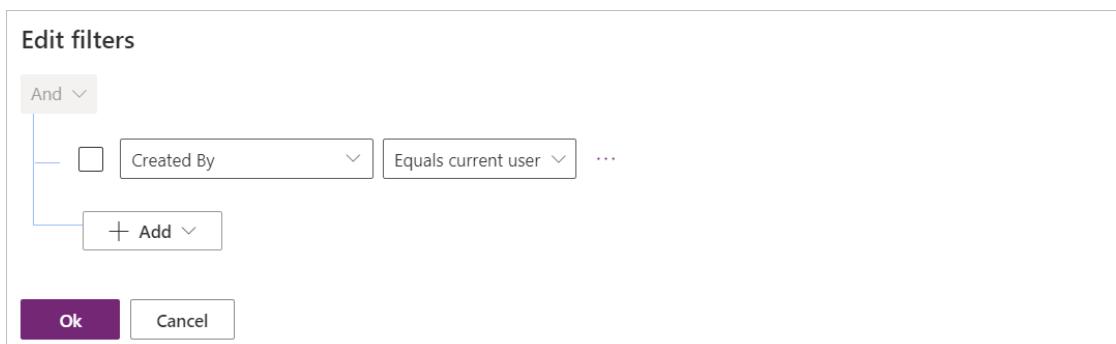
1. Navigate to the [Power Apps maker portal](#) page and make sure you are in the correct environment.
2. Select **Solutions** and click to open the **Company 311** solution.
3. Locate and click to open the **Problem Reports** Table.
4. Select the **Views** tab and click to open the **Active Problem Reports** view.

Solutions > Company 311 > Problem Report								
Fields	Relationships	Business rules	Views	Forms	Dashboards	Charts	Keys	Data
Name ↑ ▾					View type ▾			
					...	Public View	Default	
Active Problem Reports					...	Public View		
Inactive Problem Reports					...	Public View		

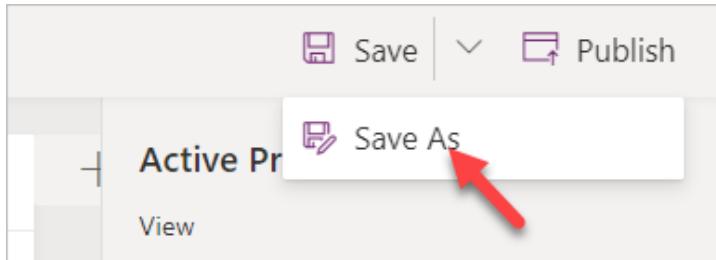
1. Click **Edit filters**.



1. Change the filter to **Created By Equals current user** and click **OK**.



1. Click on the chevron button next to the Save button and select **Save As**.



1. Enter **My Reports** for **Name** and click **Save**.

2. Click **Publish** and wait for the publishing to complete.

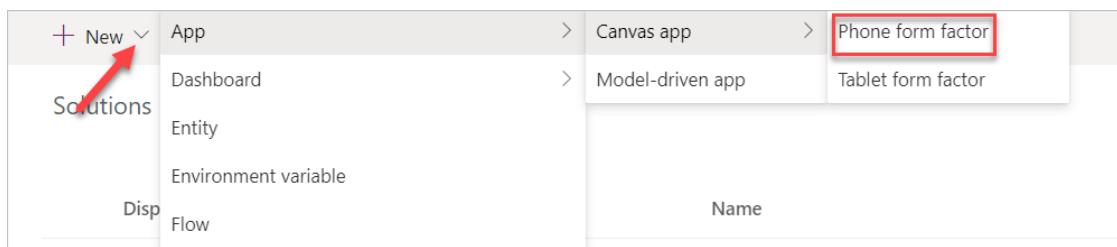
3. Click on the **Back** button.

Task 3: Create the user application

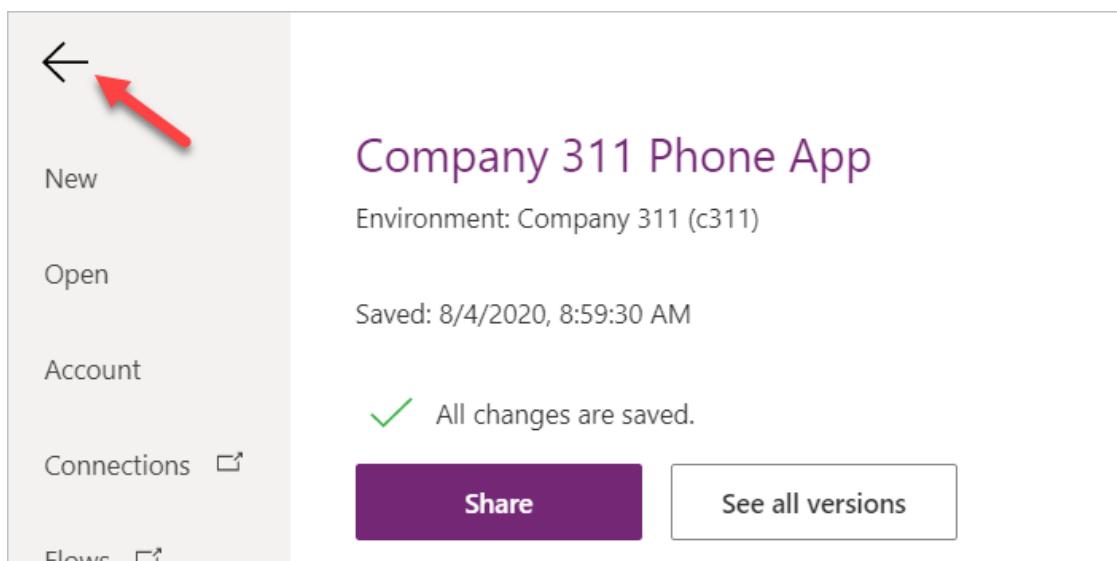
In this task, you will create a canvas application using the phone form factor.

1. Navigate to the [Power Apps maker portal](#) page and make sure you are in the correct environment.
2. Select **Solutions** and click to open the **Company 311** solution.

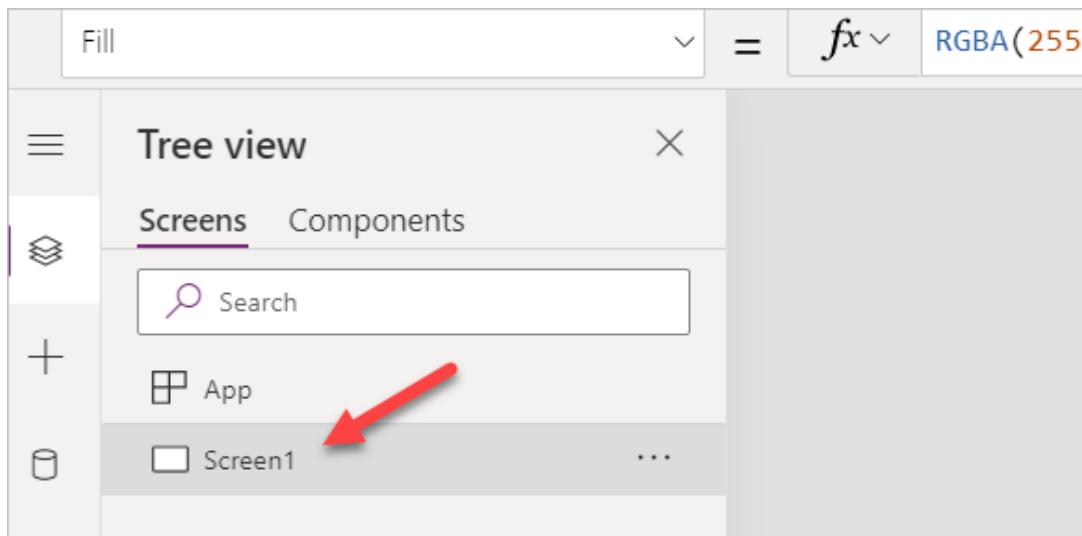
3. Click + New | App |Canvas app | Phone form factor.



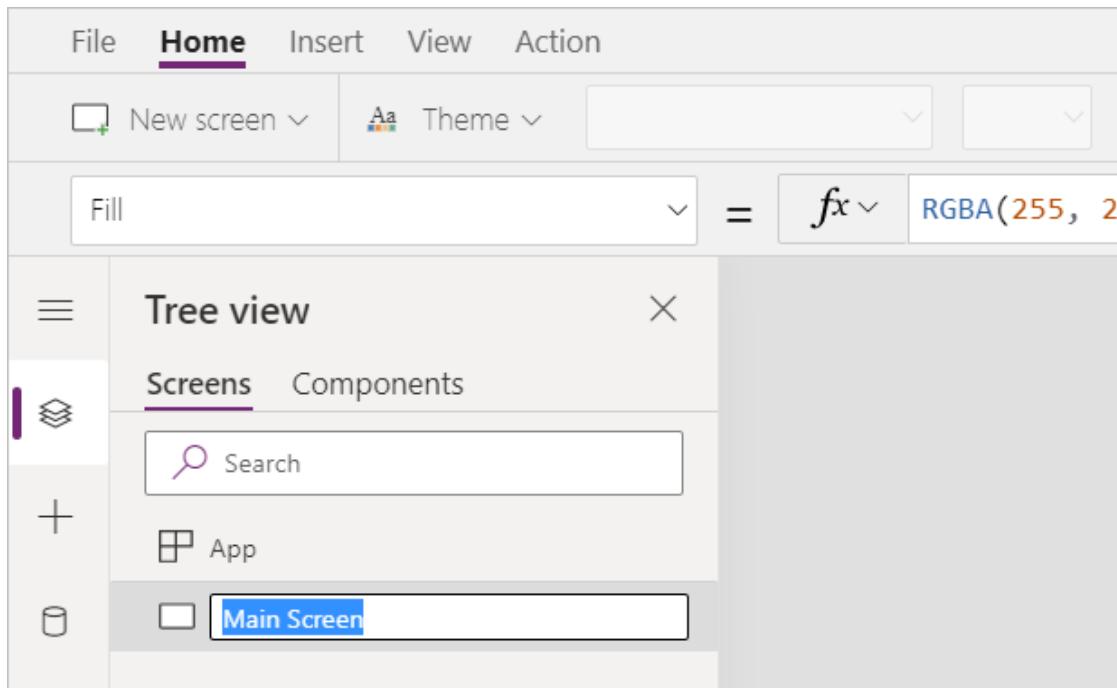
1. Click **Skip** on the welcome screen.
2. Select your **Region/Country** and click **Get started**.
3. Click **File** and select **Save**.
4. Enter **Company 311 Phone App** for name and click **Save**.
5. Click on the back button.



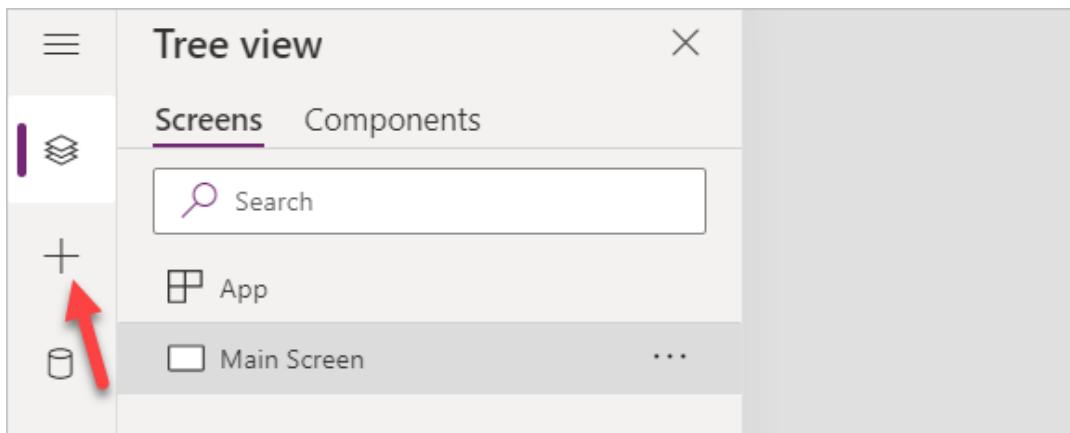
1. Go to the Tree view and double click **Screen1**.



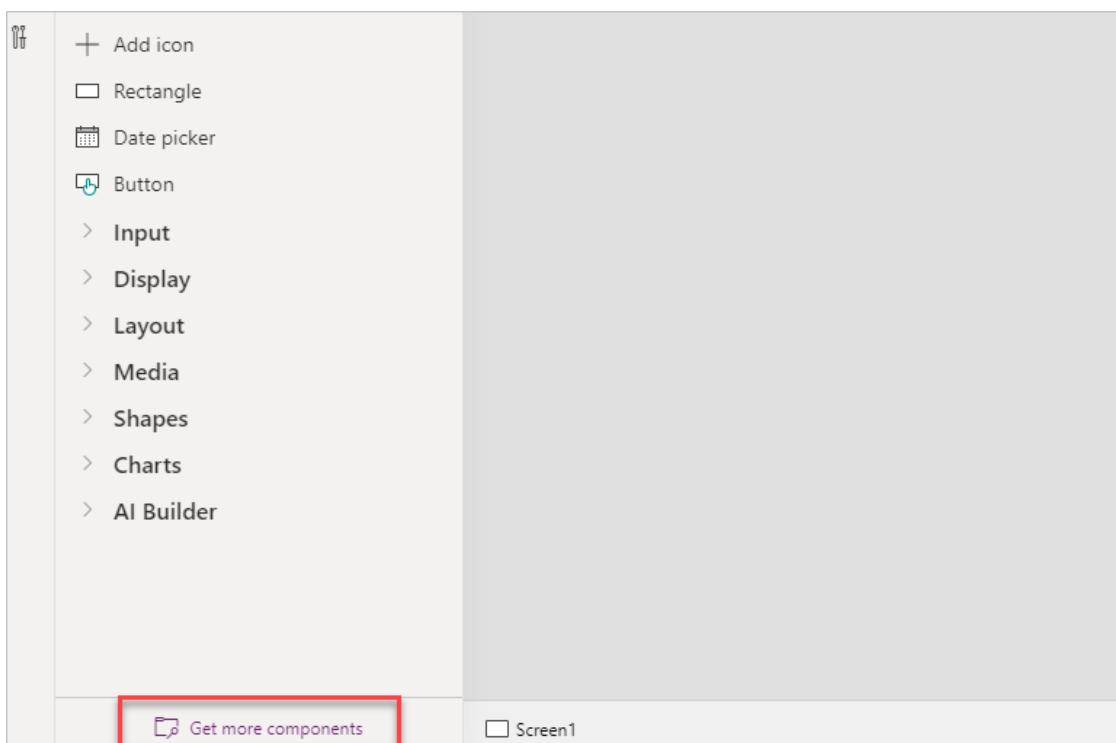
1. Rename the screen **Main Screen**. It's always a good idea to give your screens a meaningful name.



1. Select the **Main Screen** and click **Insert**.



1. Select **Get more Components**.



1. Expand the **Lamna Healthcare Shared Components A Library**, and select **Header** and **Tab Control**.

Import components

Import components created by other people in your environment. After importing, components will show in the Insert pane. [Learn more](#)

Refresh | Search

Lamna Healthcare Shared Components A
Last published on 8/12/20 by Derik Bormann. 4 component(s) **Select all**

Preloader

Dialog

Header

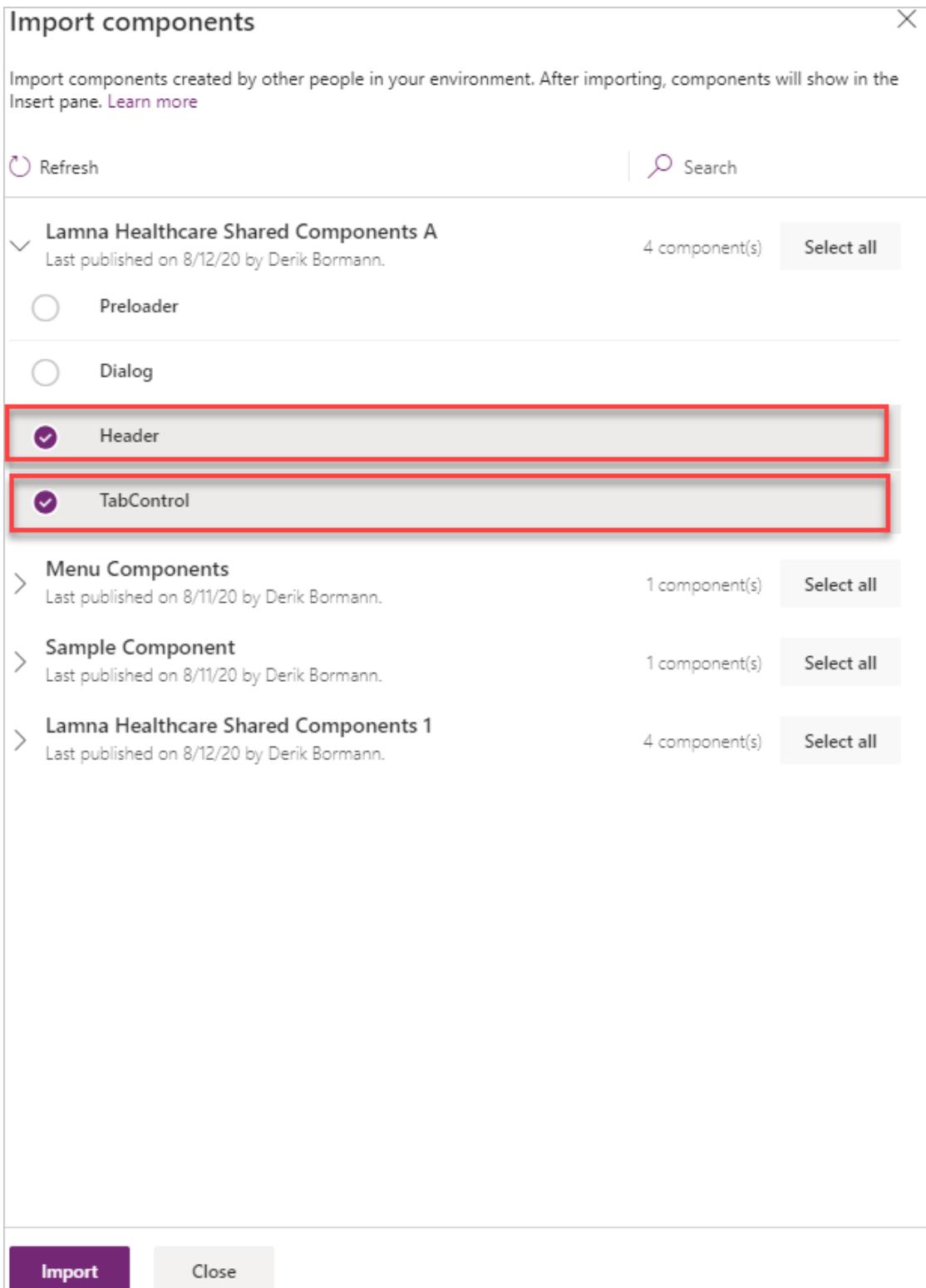
TabControl

Menu Components
Last published on 8/11/20 by Derik Bormann. 1 component(s) **Select all**

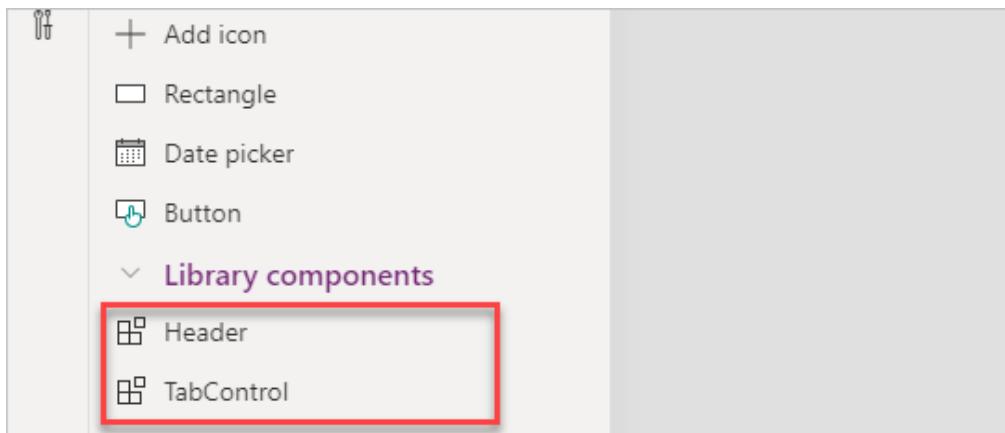
Sample Component
Last published on 8/11/20 by Derik Bormann. 1 component(s) **Select all**

Lamna Healthcare Shared Components 1
Last published on 8/12/20 by Derik Bormann. 4 component(s) **Select all**

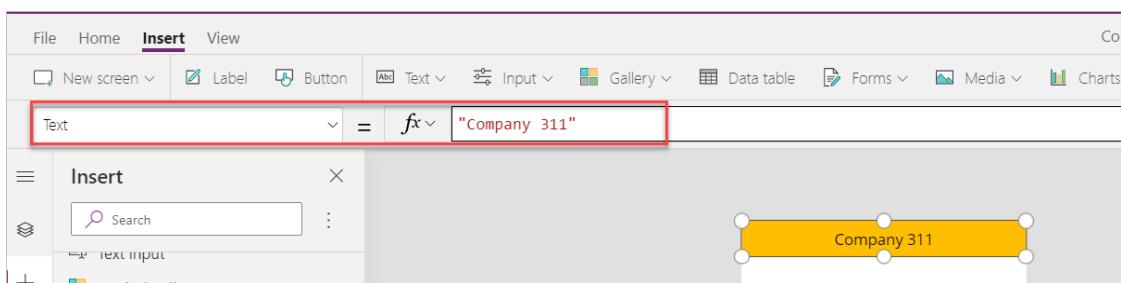
Import **Close**



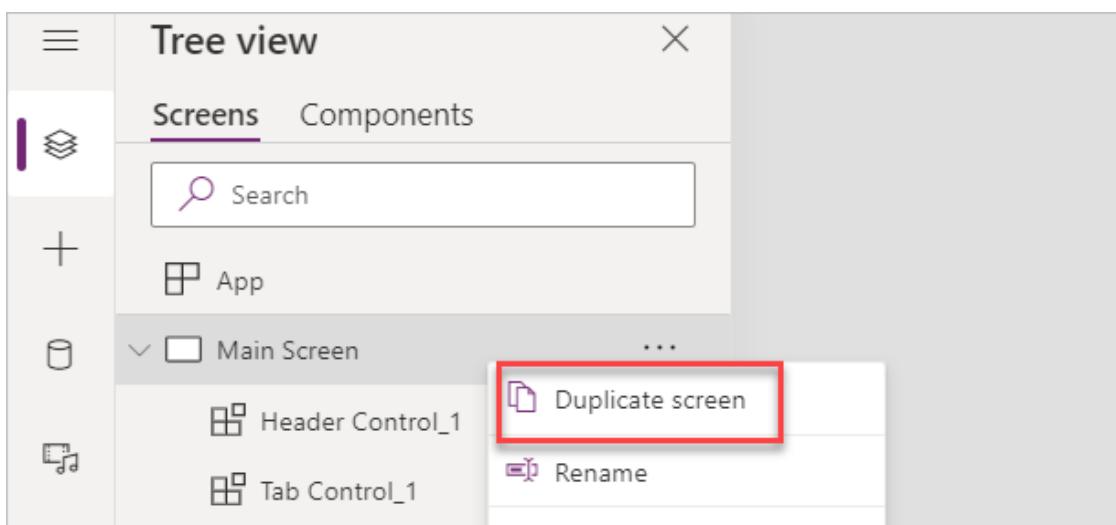
1. Click **Import**.
2. Expand **Library components**, select **Header Control** and **TabControl**.
These are both components from the library you imported earlier in the lab.



1. Move the **Tab Control** to the bottom of the screen and the **Header Control** to the top of the screens.
2. Select the **Header Control** and change the **Text** value to "**Company 311**".



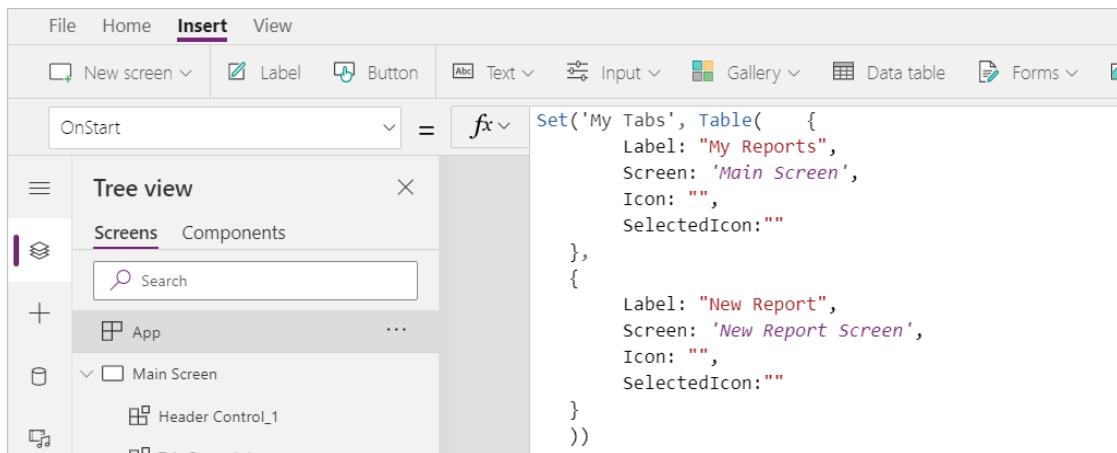
1. Right click on the Main Screen and select Duplicate screen.



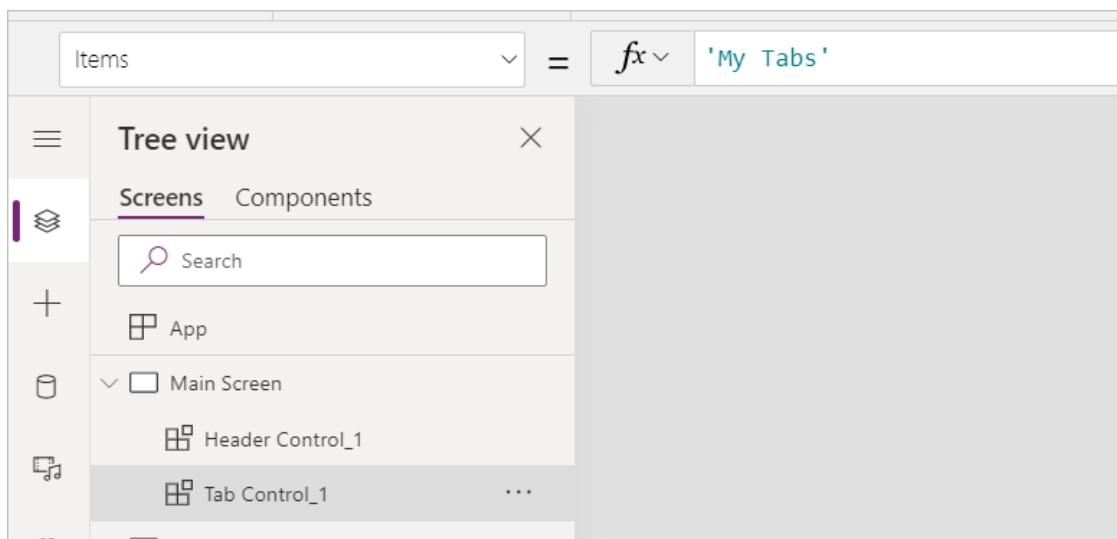
1. Rename the new screen **New Reports Screen**.
2. Select the **Tree view**, select **App** and change the **OnStart** value to the formula below. This formula will create a new variable named **My Tabs** and set it to a table of tab items.

```
javascript Set('My Tabs', Table( { Label: "My Reports", Screen: 'Main Screen', Icon: "", SelectedIcon:"" }, { Label: "New Report", Screen: 'New Reports Screen', Icon: "", SelectedIcon:"" } ))
```

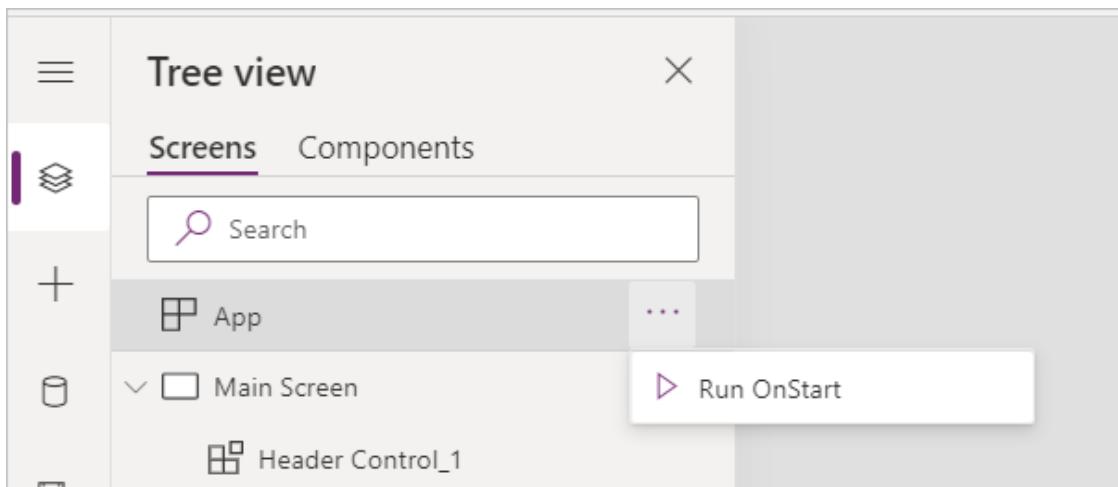
[!IMPORTANT] When expressions are copied, the quotes and double quotes are sometimes replaced with their "smart" counterparts which are not valid in formulas. If you copy and paste the expression above, make sure the resulting formula does not contain any errors.



1. Select the **Tab Control** in the **Main Screen** and change the **Items** value to **'My Tabs'**.



1. Change the **SelectedColor** value to **WhiteSmoke**.
2. Select the **Tab Control** inside the **New Report Screen** and set the Item value to **'My Tabs'**.
3. Change the **SelectedColor** value to **WhiteSmoke**.
4. Click on the **Run OnStart** button of the **App** and select **Run OnStart**.



1. Your tabs should now show the two tabs you added.



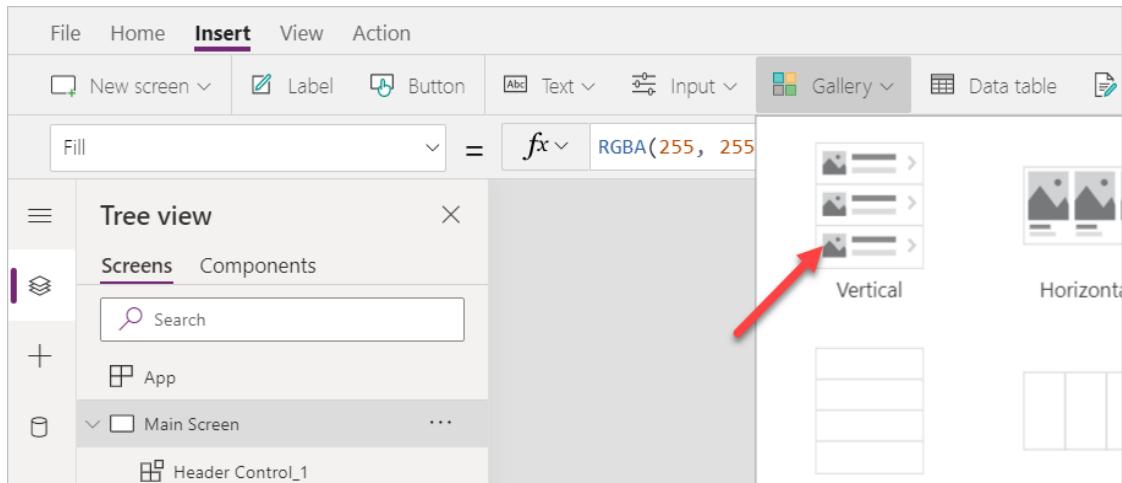
1. Do not navigate away from this page.

Exercise 2: My reports

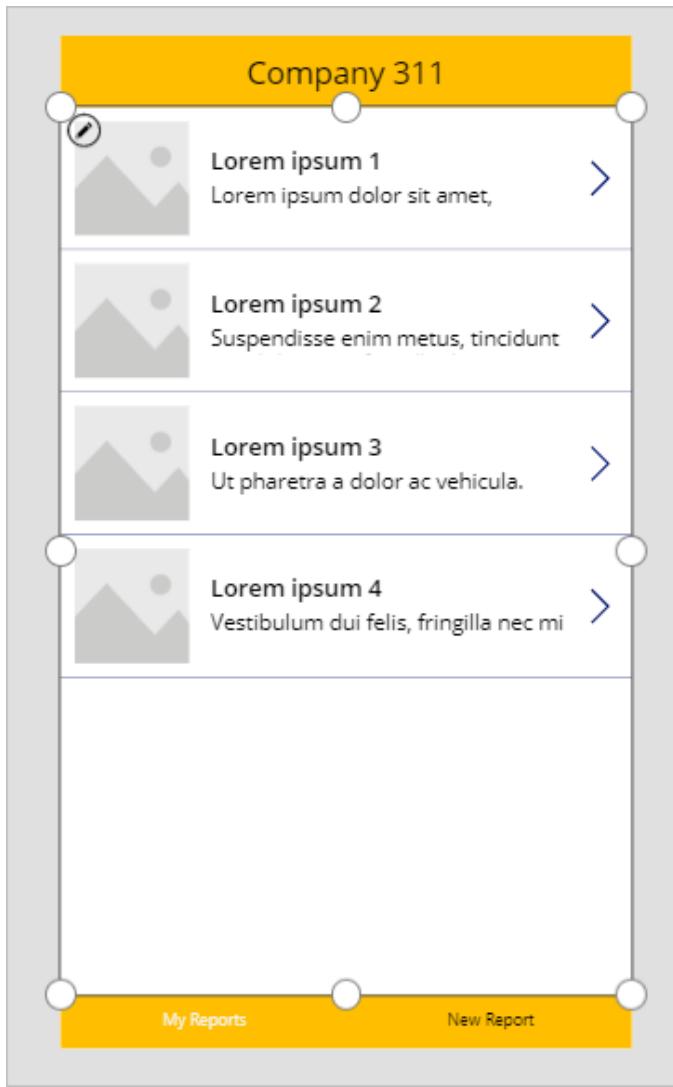
In this exercise, you will add a gallery that will show reports created by the current logged in user.

Task 1: Add gallery

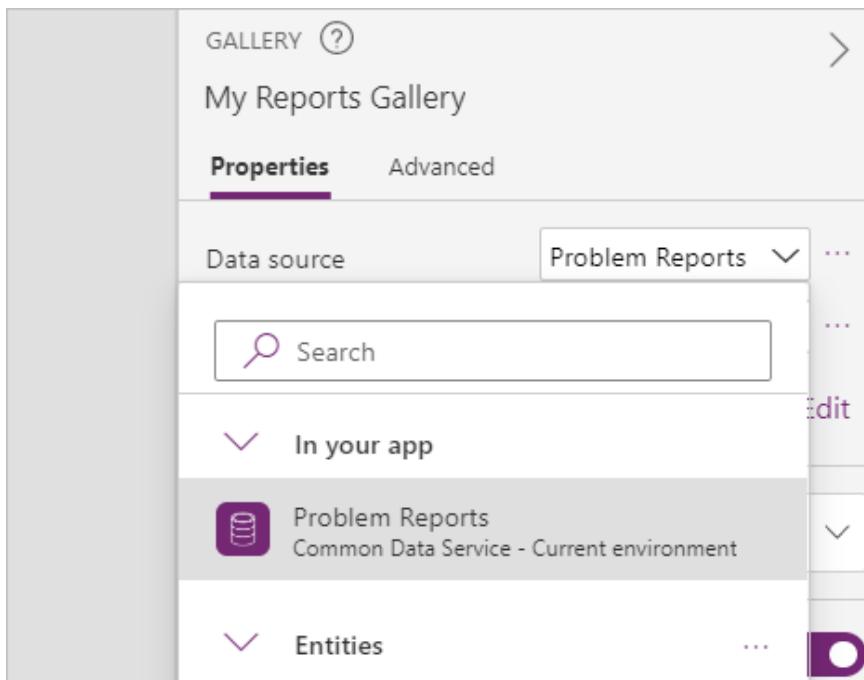
1. Select the **Main Screen**, go to the **Insert** tab, click **Gallery**, and select **Vertical**.



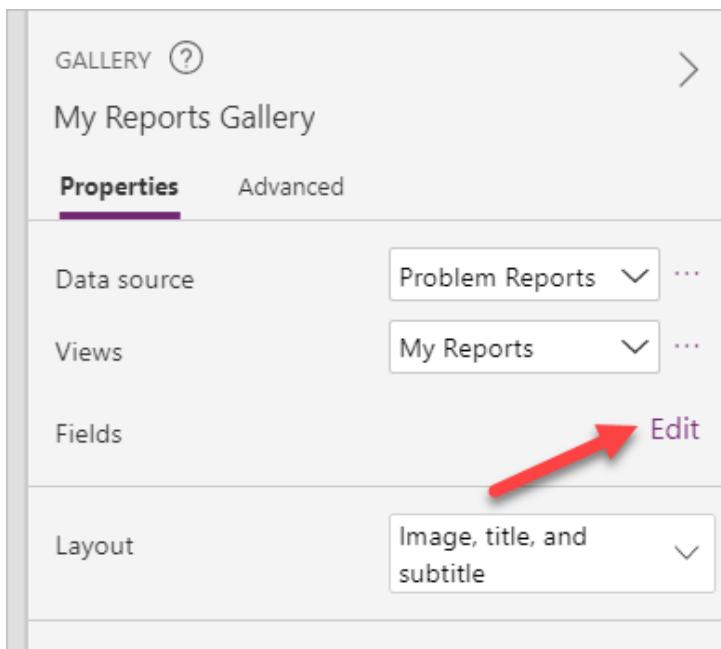
1. Rename the new gallery **My Reports Gallery**.
2. Resize and reposition **My Reports Gallery** and make sure the screen looks like the image below.



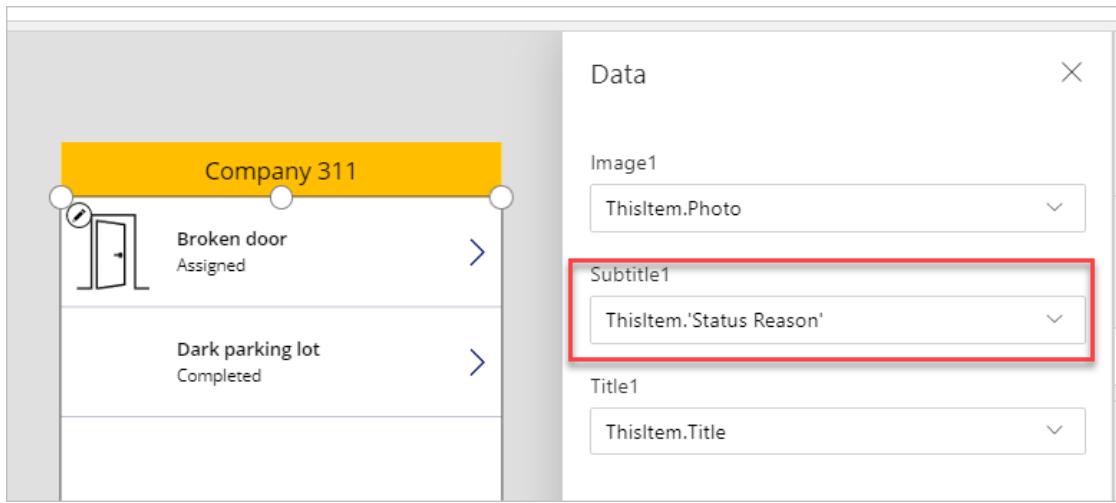
1. Select **My Reports Gallery**, go to the **Properties** pane, and select **Problem Reports** for **Data Source**. If you do not see Problem Reports, click See all Tables.



1. Select the **My Reports** view you created for **View**.
2. Click **Edit Columns**.



1. Change Subtitle1 to **statuscode**. This is the Status Reason Column.



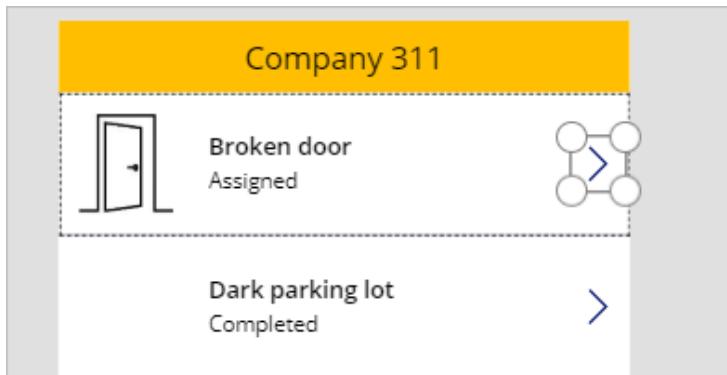
1. Click File and then click Save.
2. Click on the Back button.
3. Do not navigate away from this page.

Exercise 3: Allow removing reports

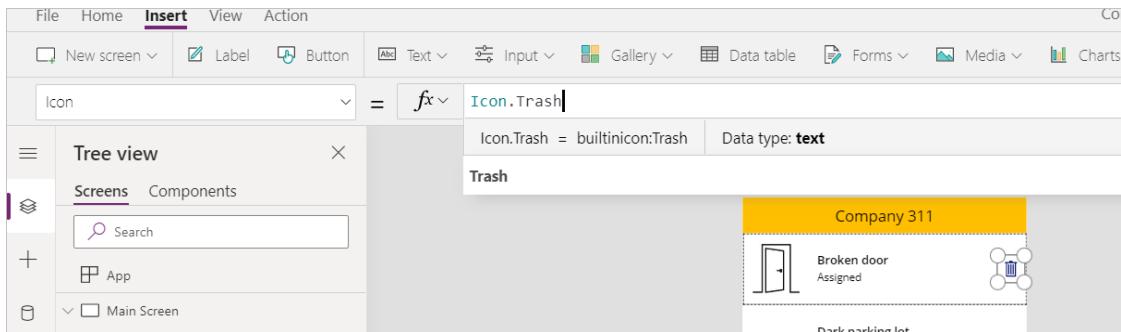
In this exercise, you will allow unassigned reports to be removed. This will allow users to easily remove any accidental reports.

Task 1: Allow remove

1. Expand the **My Reports Gallery**.
2. Select the **Icon** inside the **My Reports Gallery**.

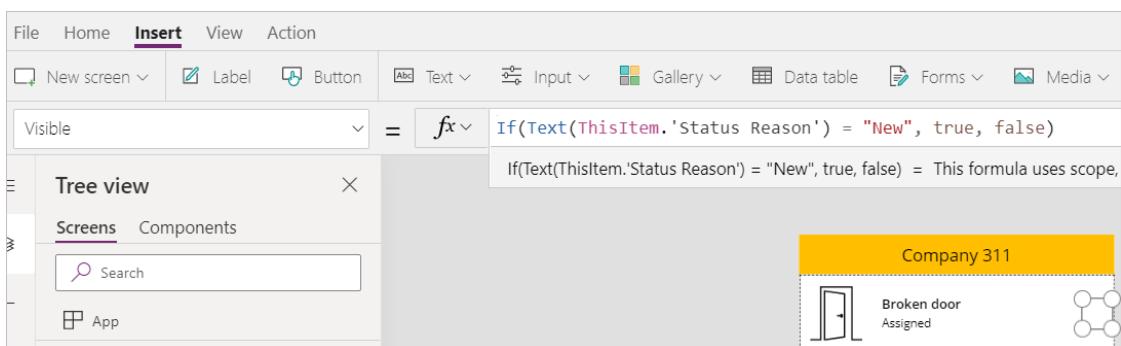


1. Change the **Icon** value to **Icon.Trash**.



1. Change the **Visible** value to the formula below. This formula will hide the icon if the status reason is not New.

```
If(Text(ThisItem.'Status Reason') = "New", true, false)
```



1. Make sure you still have the icon selected. Change the **OnSelect** value to the formula below. This formula will remove item from the data source.

```
Remove('Problem Reports', ThisItem)
```

1. Click **File** and then click **Save**.

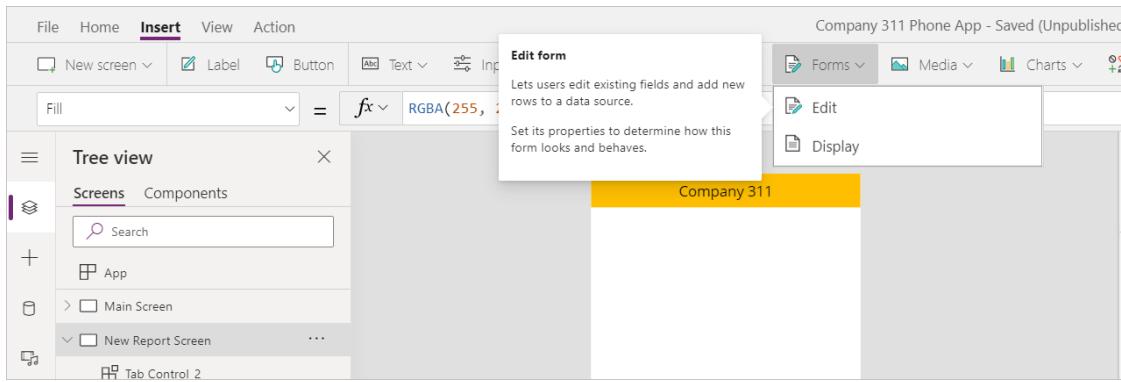
2. Do not navigate away from this page.

Exercise 4: Add new report

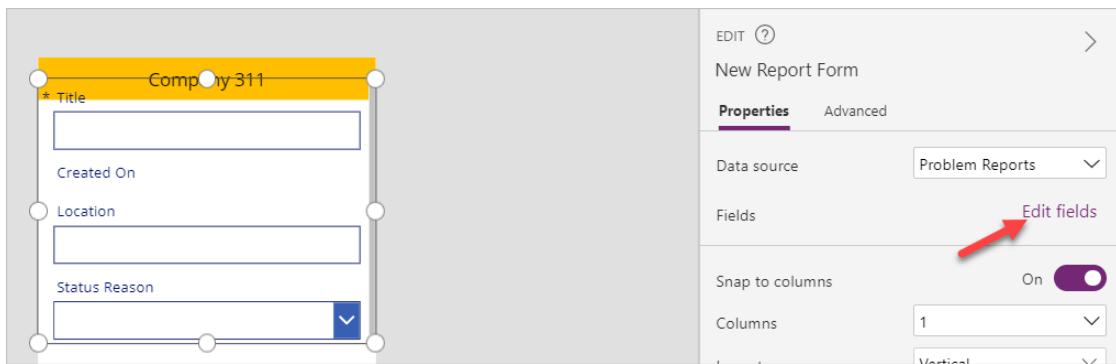
In this exercise, you will add a form to submit new problem reports.

Task 1: Add new report form

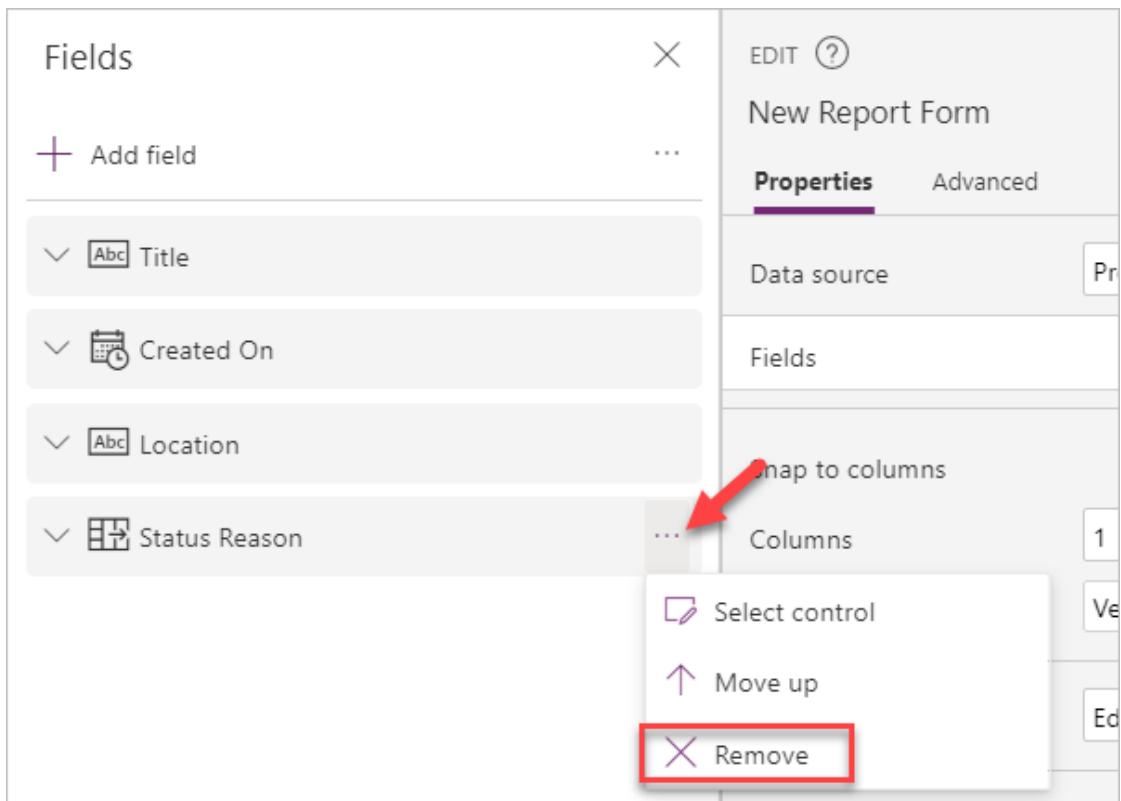
1. Select the **New Report Screen**, go to the **Insert** tab, click **Form**, and select **Edit**.



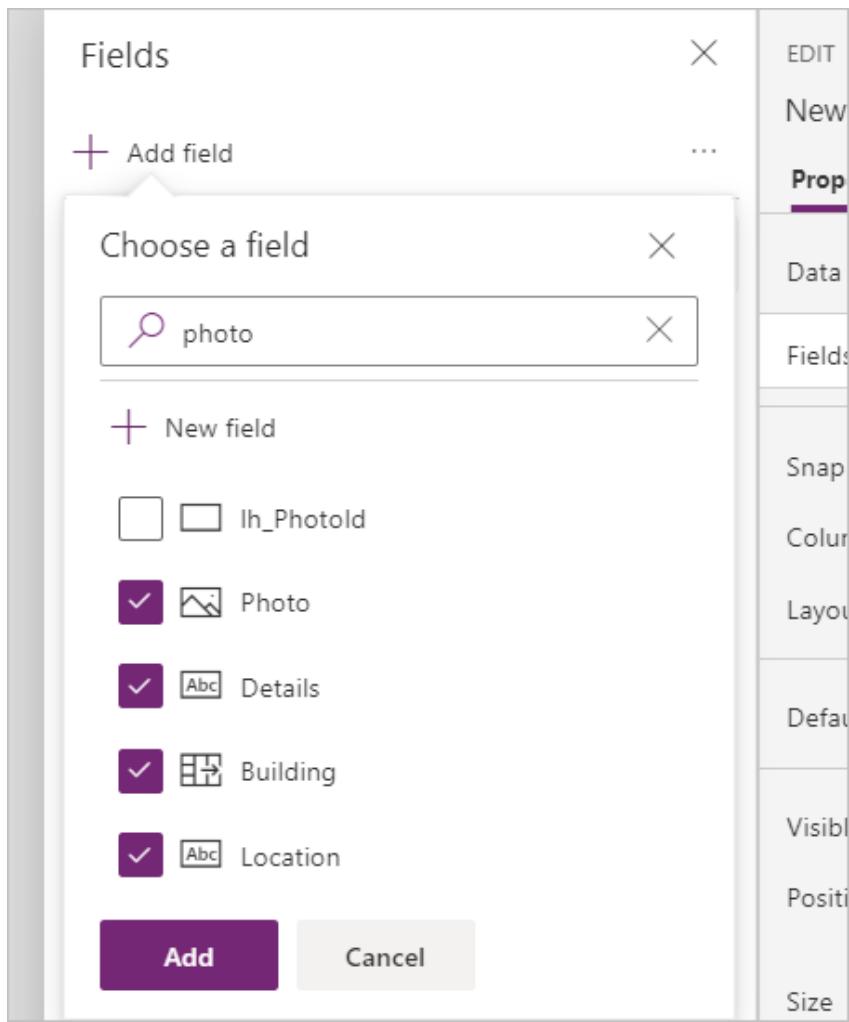
1. Rename the form **New Report Form**.
2. Select **New Report Form**, go to the **Properties** pane, and select **Problem Report** for Data source.
3. Click **Edit Columns**.



1. Remove the **Status Reason** Column.



1. Remove the **Created On** Column.
2. Remove the **Location** Column.
3. Click **+ Add Column**.
4. Select **Details, Building, Department**, and **Photo**, and then click **Add**.



1. Resize and reposition the form so it takes most of the page and leave enough room for a button in the bottom.

Company 311

* Title

* Details

Building

Find items ▾

Department

Find items ▾

Photo

Tap or click to add a picture

My Reports New Report

1. Select the **New Report Screen**.
2. Go to the **Insert** tab and select **Button**.
3. Rename the button **Submit Report**.
4. Place the button below the form and make it stretch across the screen
5. Change the **Submit Report** button text to **Submit**
6. Select the Submit Report button and change the **OnSelect** value to the formula below. This formula will create a new Row and clear the form when the button is clicked.

```
SubmitForm('New Report Form'); NewForm('New Report Form')
```

1. Select the **New Report Form**.

2. Change the **OnSuccess** value to the formula below. This formula will show a notification after the new Row gets created.

```
Notify("Created new problem report Row")
```

1. Select the **New Report Screen**.

2. Set the **OnVisible** value to the formula below. This formula will create a new form when the screen becomes visible.

```
NewForm('New Report Form')
```

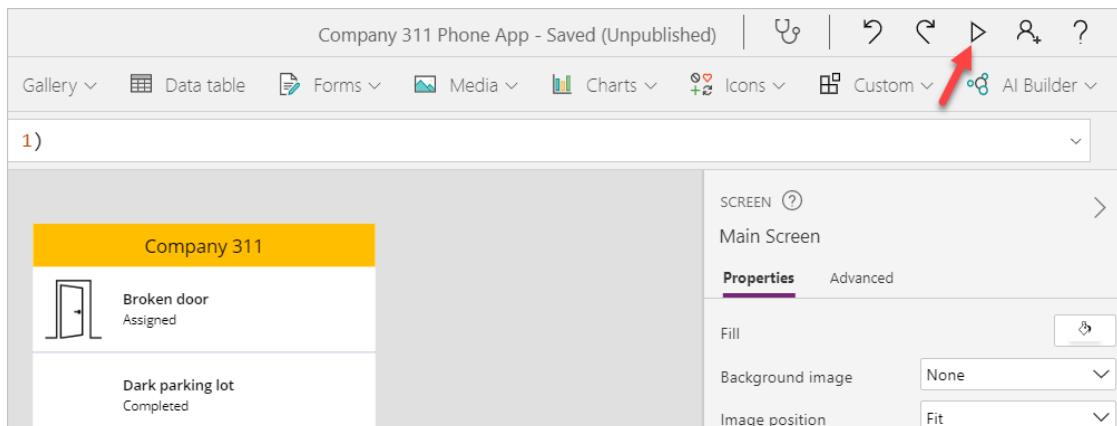
1. Click **File** and then click **Save**.

Exercise 5: Test the application

In this exercise, you will test the canvas application you created by submitting a problem report.

Task 1: Test application

1. Select the **Main Screen** and click **Preview the app**.



1. The application should load, and the list should show all the reports you created.

Company 311



Broken door

Assigned

Dark parking lot

Completed

[My Reports](#)

[New Report](#)

1. Select the **New Report** tab.

2. The **New Report Form** should load. Fill out the form and click on the **Photo Column**.
3. Select an image.
4. Click **Submit**
5. The Row should get created successfully and you should see the success message.

Created new problem report record

Company 311

* Title

* Details

1. Select the **My Reports** tab.
2. You should see the new report you created. Click **Delete** to test the delete.

	Broken door Won't Fix
	Dark parking lot Completed
	Paw prints on the second floor New

1. The Row should be deleted and removed from the list.

Company 311	
	Broken door Assigned
	Dark parking lot Completed

1. Close the application.

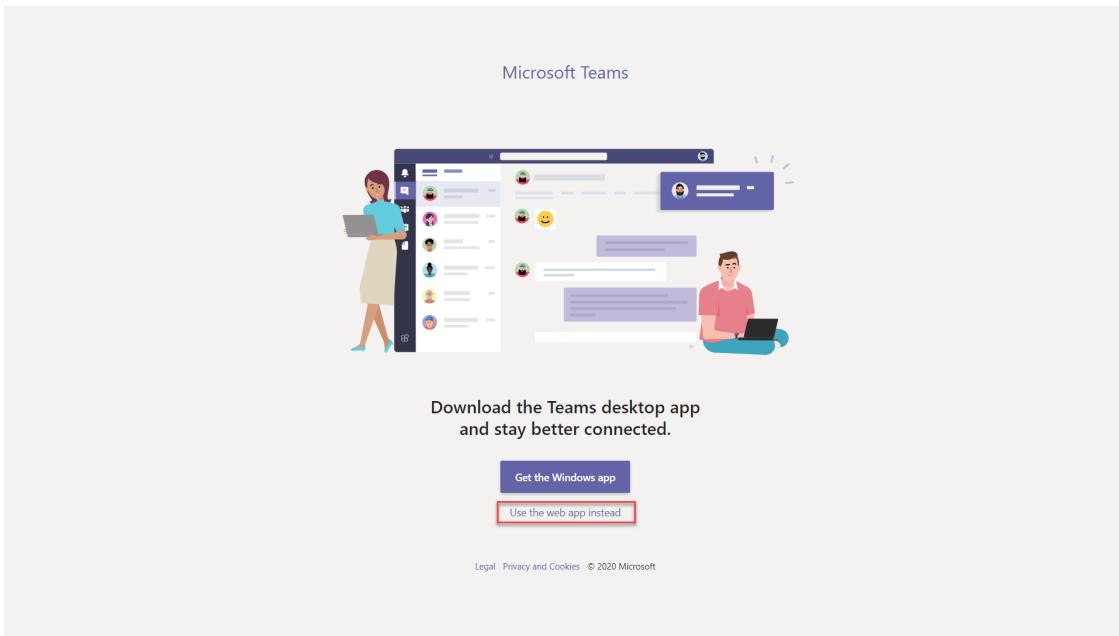
Exercise 6: Embed canvas app in Microsoft Teams

In this exercise, you will add the Company 311 Phone App that you created earlier, to Microsoft Teams as a way for staff to be able to log issues directly within Teams.

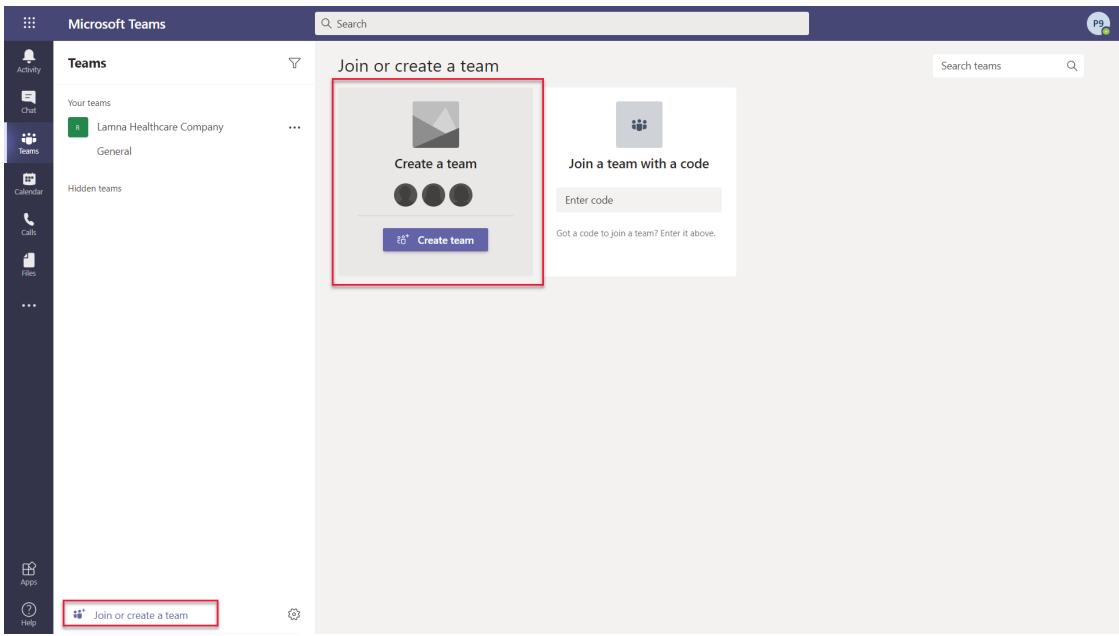
Task 1: Setup Company 311 Team

In this task you will setup a Microsoft Teams team for the Lamna Healthcare Company, if you have not done so previously.

1. Navigate to [Microsoft Teams](#) and sign in with the same credentials you have been using previously.
2. Select **Use the web app instead** on the welcome screen.



1. When the Microsoft Teams window opens, dismiss the welcome messages.
2. On the bottom left corner, choose **Join or create a team**.
3. Select **Create a team**.

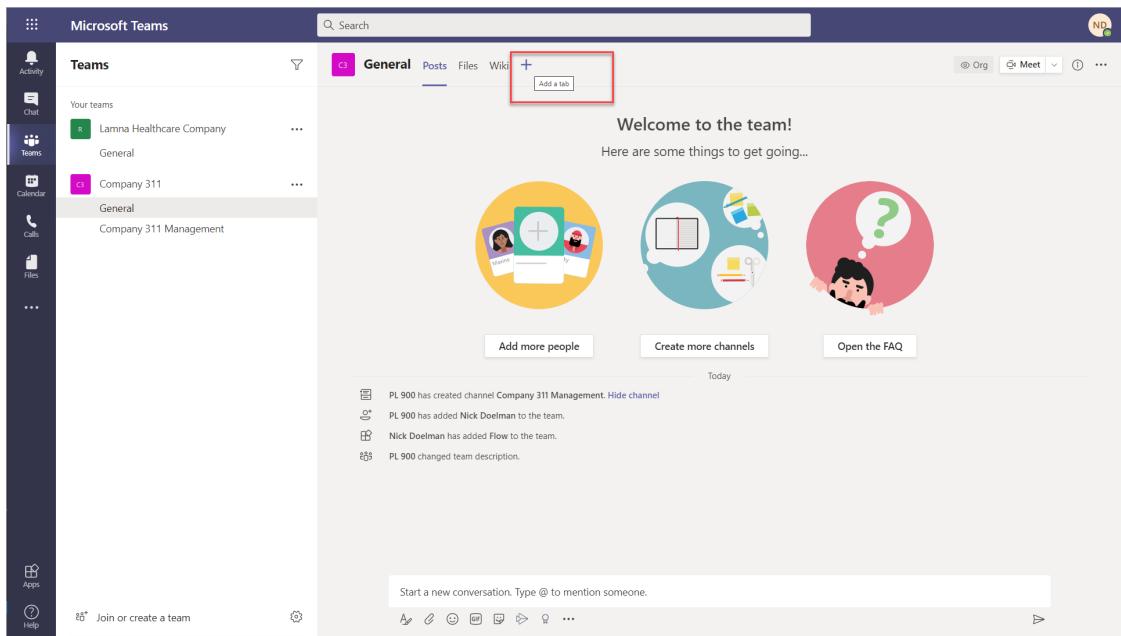


1. Press **Build a team from scratch**.
2. Select **Public**.
3. For the Team name choose **Company 311** and select **Create**.

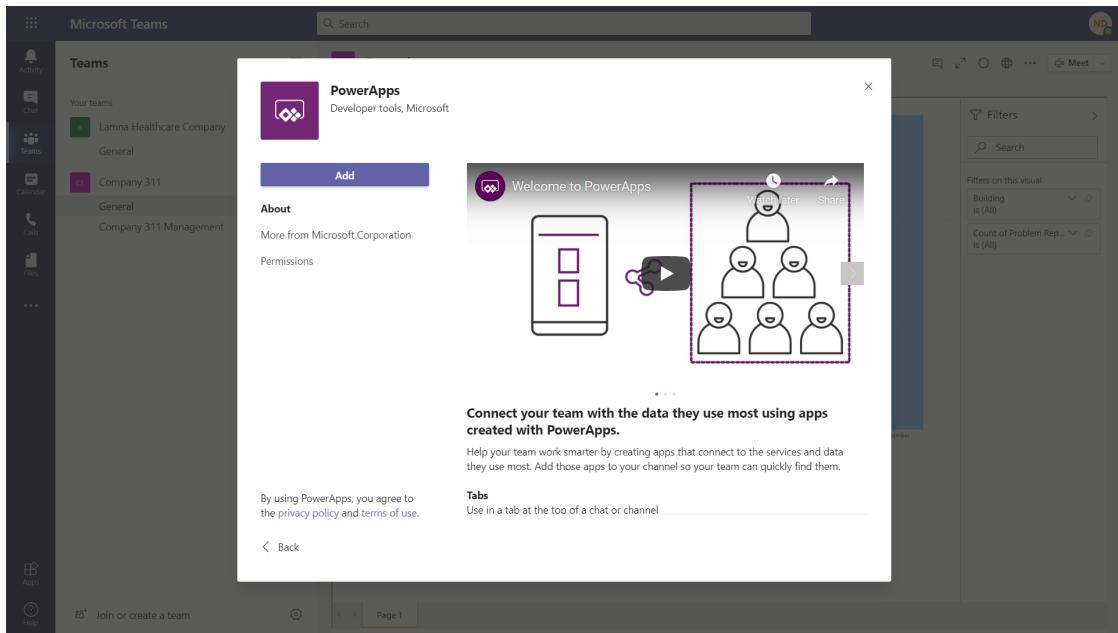
4. Select **Skip** adding members to Company 311.

Task 2: Add canvas app to Teams

1. Navigate to [Microsoft Teams](#)
2. Select the **General** channel of the **Company 311** team.
3. On the top of the page, press the **+** symbol to add a new tab.



1. Search for **power** and select **PowerApps** from the results.
2. Press **Add** to add Power Apps to Teams

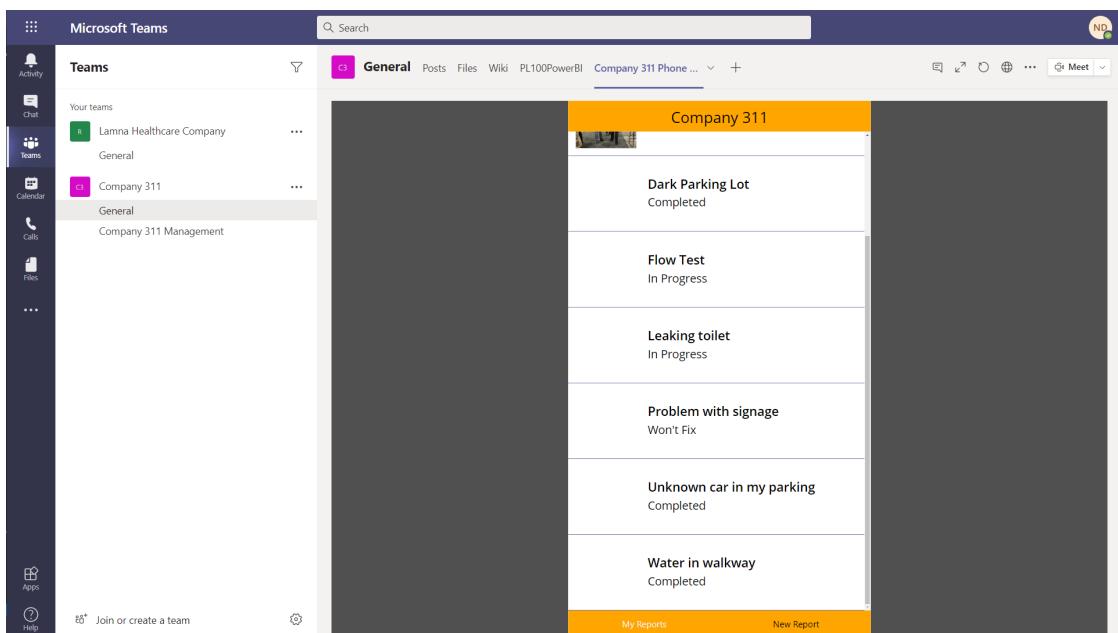


1. Select the **Company 311 Phone App** that you created earlier in this lab.

[!IMPORTANT] If you do not see the app you need to go back to the app editor and publish the app

1. Press **Save**

2. The **Company 311** app should now appear on a tab in Microsoft Teams.



lab: title: 'Lab: Business Process Flows and Business Rules' module: 'Module 3: Building model-driven apps'

[!NOTE] Effective November 2020: - Common Data Service has been renamed to Microsoft Dataverse. [Learn more](#) - Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. [Learn more](#)

This content will be updated soon to reflect the latest terminology.

Lab 04: Business Process Flows and Business Rules

In this lab you will enhance the data model and improve the app behavior by adding a business process flow and a business rule.

What you will learn

- How to identify stages in a Business Process Flow (BPF)
- How to create and use a BPF
- How to use a business rule to implement logic

High-level lab steps

- Exercise 1 Create BPF lifecycle of problem report
 - Route
 - Fix
 - Resolved
- Exercise 2 Business rule to not allow close without resolution

Prerequisites

- Must have completed **Lab 02: Data model and model-driven app**

Detailed steps

Exercise 1: Create business process flow

In this exercise, you will create a business process flow for the problem report Table.

Task 1: Customize Table

In this task, you will add a lookup Column to the problem report Table.

1. Navigate to the [Power Apps maker portal](#) page and make sure you are in the correct environment.
2. Select **Solutions** and click to open the **Company 311** solution.
3. Locate and click to open the **Problem Report** Table.
4. Make sure you have the **Columns** tab and click **+ Add Column**.
5. Enter **Assign to** for **Display name**, select **Lookup** for **Data type**, select **User** for **Related table**, and click **Done**.

Assign to

Display name *

Assign to

Name * ⓘ

Ih_ Assignto

Data type * ⓘ

Lookup

Related table *

User

Required * ⓘ

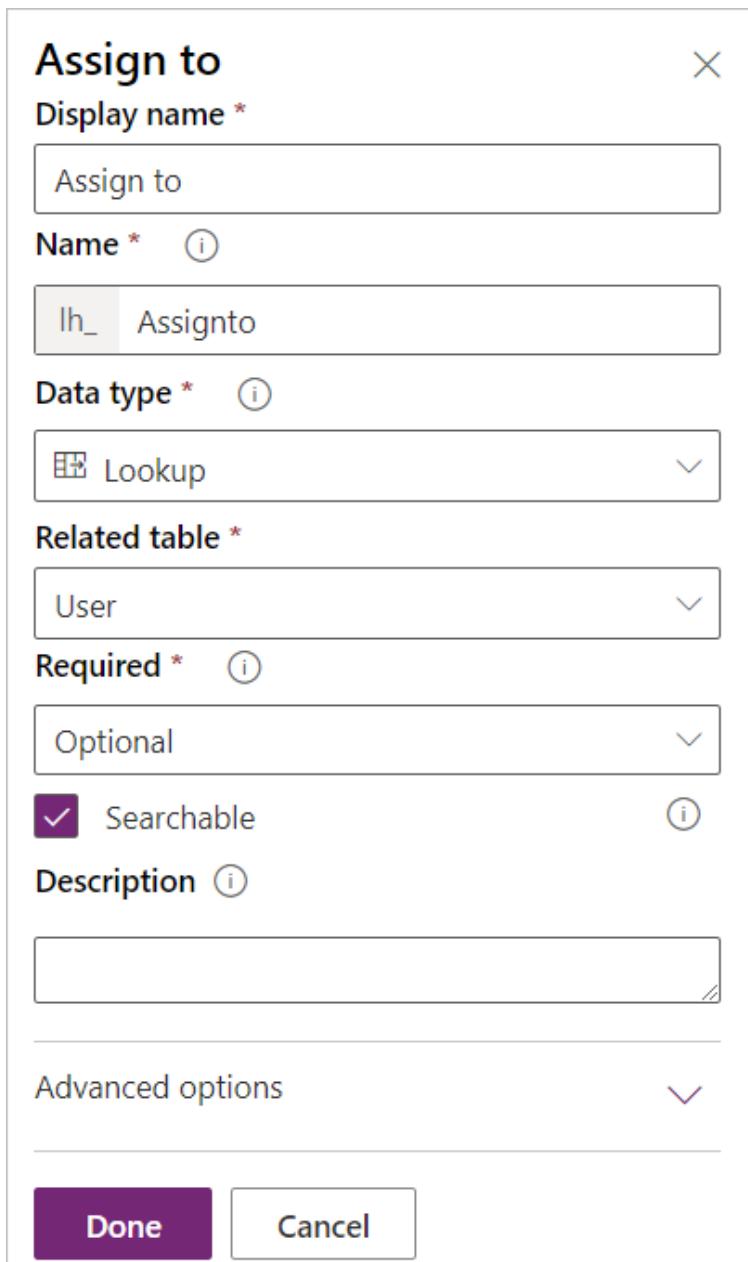
Optional

Searchable ⓘ

Description ⓘ

Advanced options

Done **Cancel**

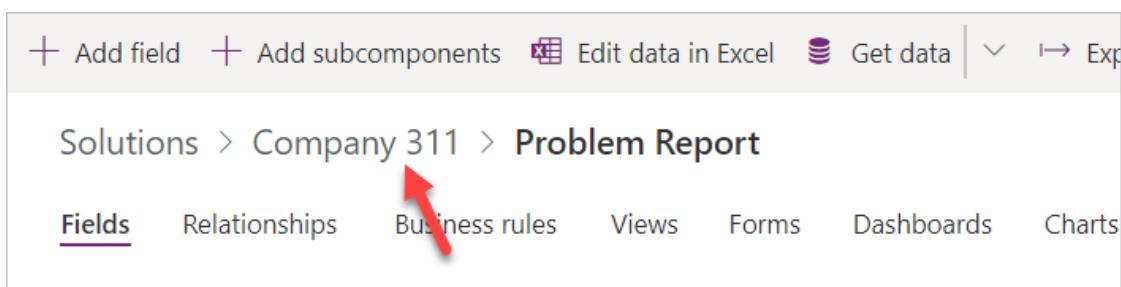


1. Click **Save Table**.

2. Go back to the solution by clicking on the solution name.

Solutions > Company 311 > **Problem Report**

Fields Relationships Business rules Views Forms Dashboards Charts

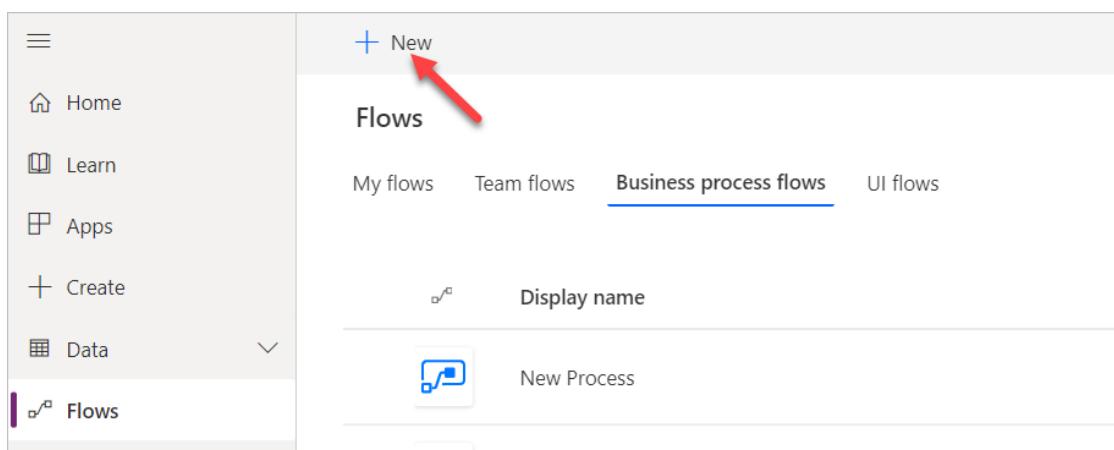


1. Click **Publish all customizations** and wait for the publishing to complete.

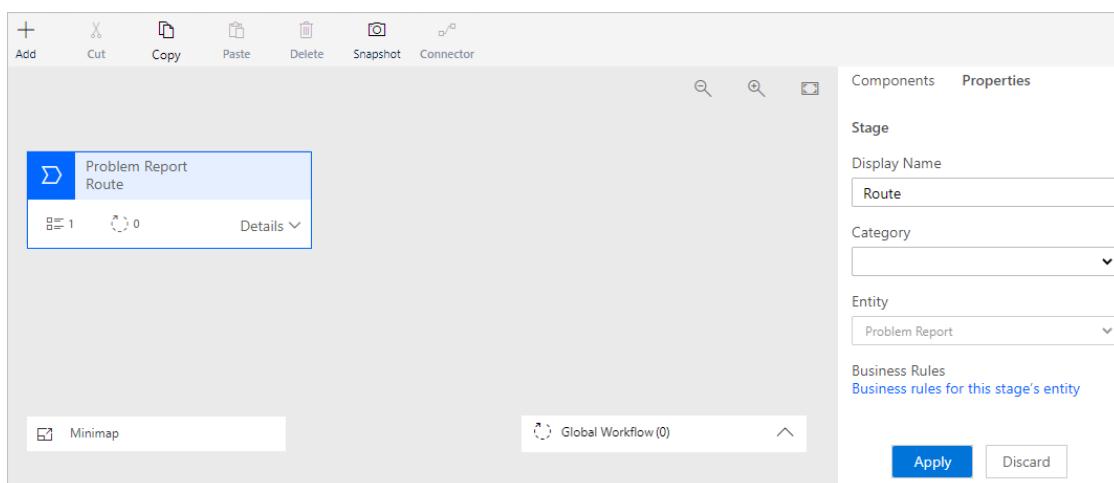
Task 2: Create business process flow

In this task, you will create a business process flow for the problem report Table.

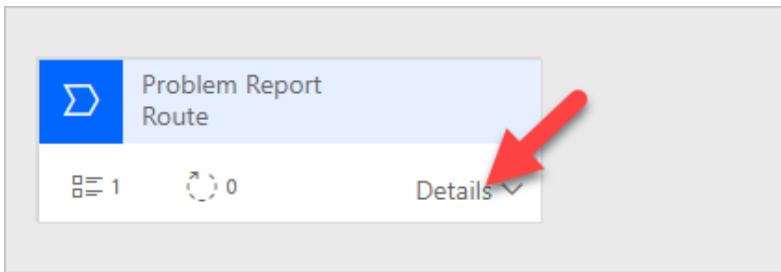
1. Navigate to the [Power Apps maker portal](#) page and make sure you are in the correct environment.
2. Select **Flows**.
3. Select the **Business process flows** tab and click **+ New**.



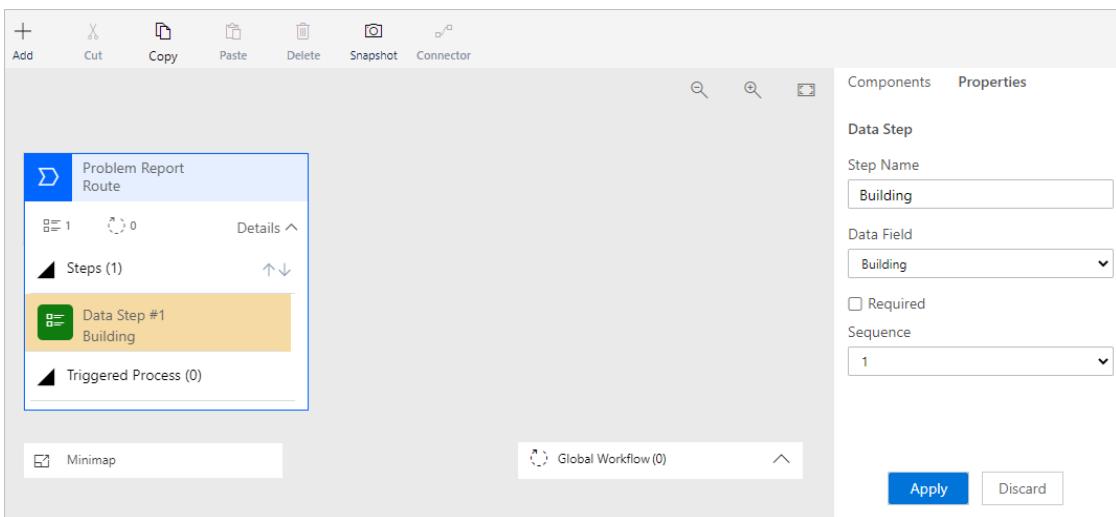
1. Enter **Problem resolution process** for **Flow Name**, select **Problem Report** for **Table**, and click **Create**.
2. Select the **New stage**, go to the **Properties** pane, change the **Display Name** to **Route**, and click **Apply**.



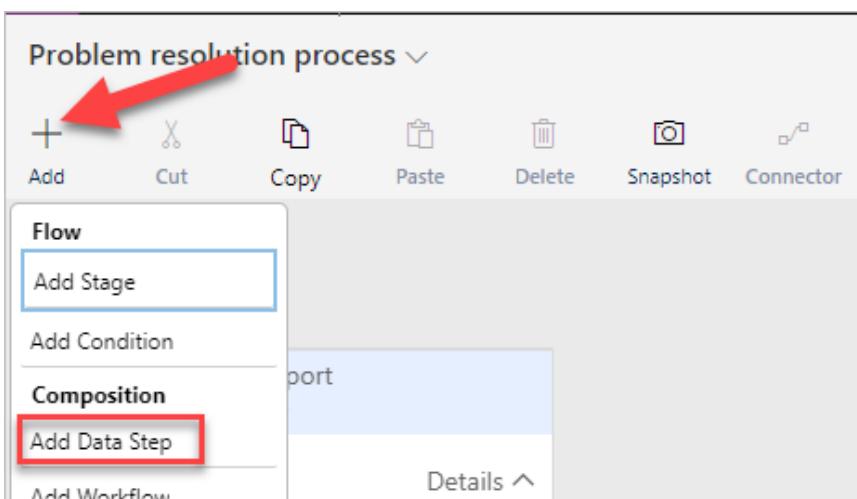
1. Expand **Details** of the **Route** stage.



1. Select **Data Step #1**, go to the **Properties** pane, select **Building** for **Data Column**, and click **Apply**.

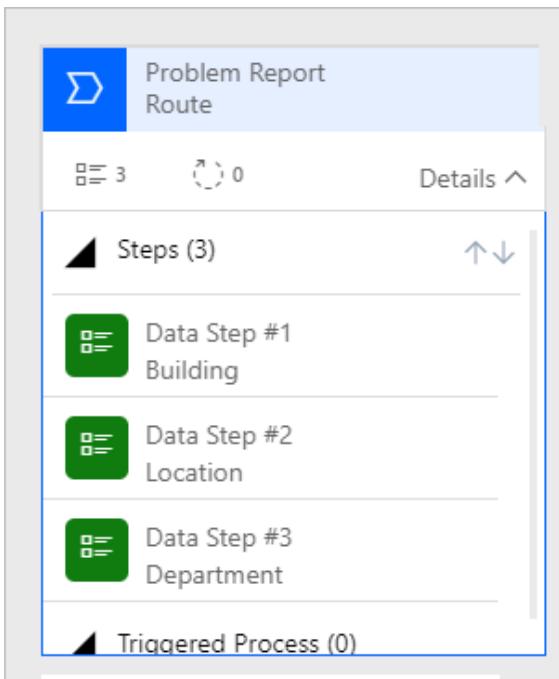


1. Click **+ Add** and select **Add Data Step**.



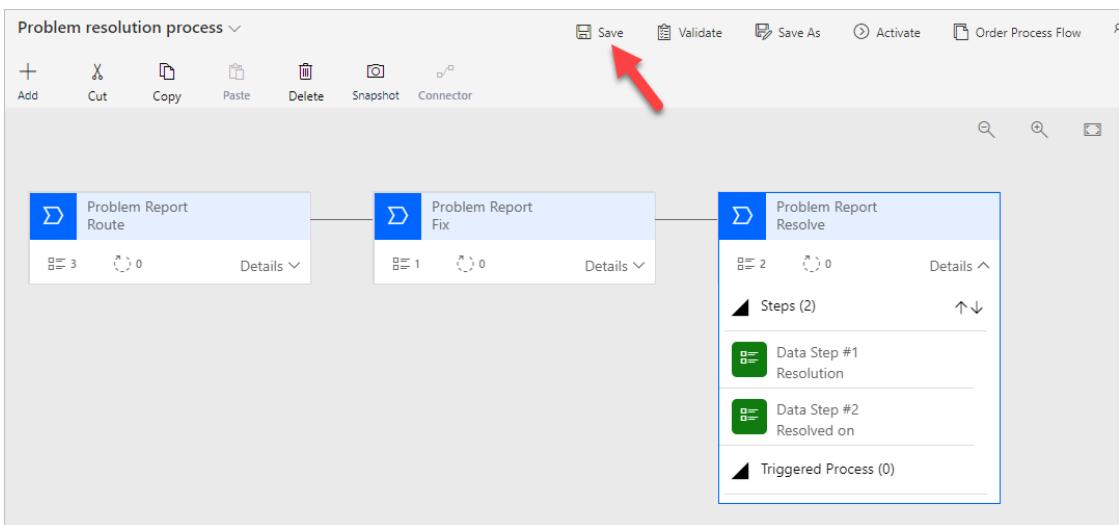
1. Add the data step below the **Building** data step.
2. Select the new data step, go to the **Properties** pane, select **Location** for **Data Column**, and click **Apply**.
3. Click **+ Add** again and select **Add Data Step**.

4. Click **+ placeholder** to add the new data step below the **Location** data step.
5. Select the new data step, go to the **Properties** pane, select **Department** for **Data Column**, and click **Apply**.
6. The **Route** stage should now look like the image below.



1. Click **+ Add** and select **Add Stage**.
2. Add the new stage after the **Route** stage.
3. Select the stage, go to the **Properties** pane, enter **Fix** for **Display Name**, and click **Apply**.
4. Expand **Details** of the **Fix** stage.
5. Select **Data Step #1** of the **Fix** stage.
6. Go to the **Properties** pane, select **Assign to** for **Data Column**, and click **Apply**.
7. Click **+ Add** and select **Add Stage**.
8. Add the new stage after the **Fix** stage.
9. Select the new stage, go to the **Properties** pane, enter **Resolve** for **Display Name**, and click **Apply**.

10. Expand **Details** of the **Resolve** stage.
11. Select **Data Step #1** of the **Resolve** stage.
12. Go to the **Properties** pane, select **Resolution** for **Data Column**, and click **Apply**.
13. Click **+ Add** and select **Add Data Step**.
14. Add the new data step below the **Resolution** data step.
15. Select the new data step, go to the **Properties** pane, select **Resolved on** for **Date Column**, and click **Apply**.
16. The Business process flow should now look like the image below. Click **Save**.



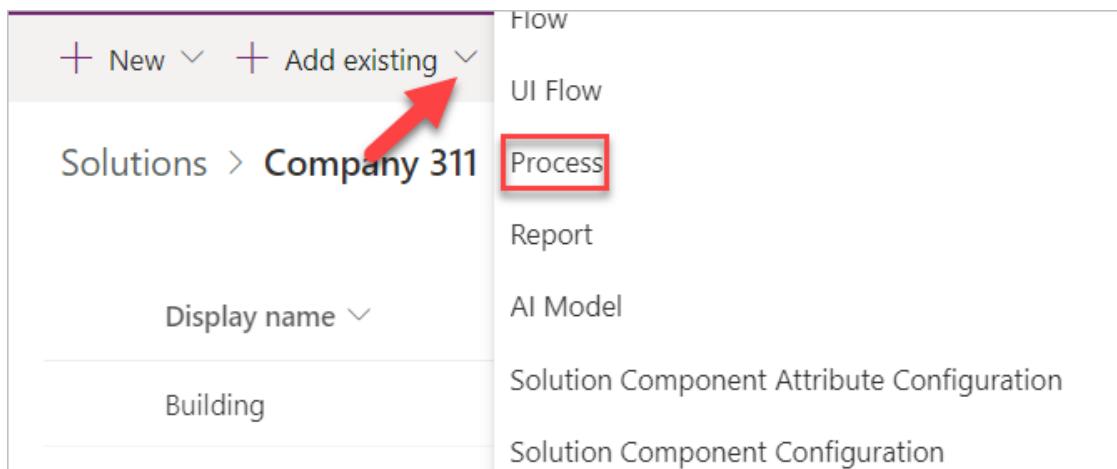
1. Click **Activate**.
2. Confirm that **Status: Active**.
3. Close the process editor browser window or tab.

Task 3: Add business process flow to solution

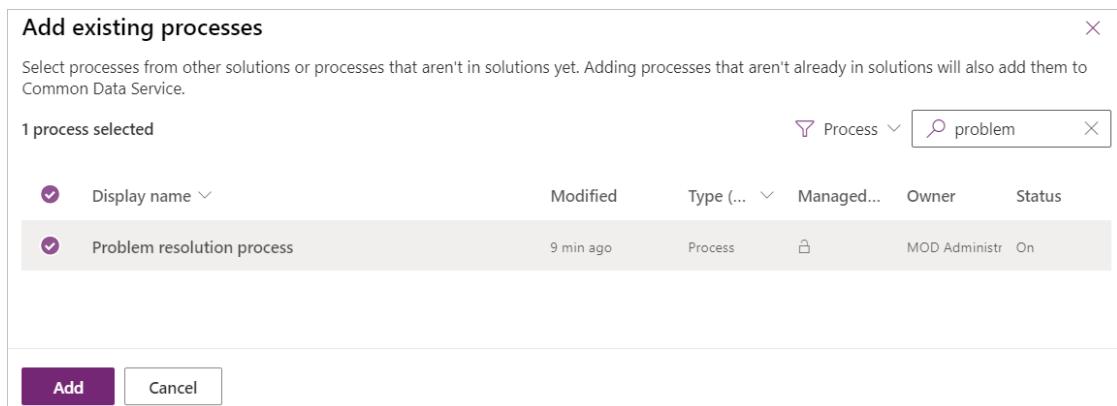
In this task, you will add the business process flow you created to the Company 311 solution.

1. Navigate to the [Power Apps maker portal](#) page and make sure you are in the correct environment.
2. Select **Solutions** and click to open the **Company 311** solution.

3. Click **+ Add existing** and select **Process**.



1. Search for problem, select **Problem resolution process**, and click **Add**.



1. Click **Publish all customizations** and wait for the publishing to complete.

Exercise 2: Create business rule

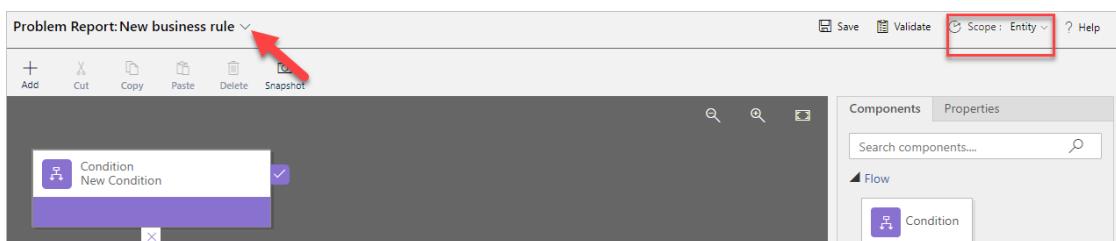
In this exercise, you will create a business rule that will block completion of problems without resolution.

Task 1: Create business rule

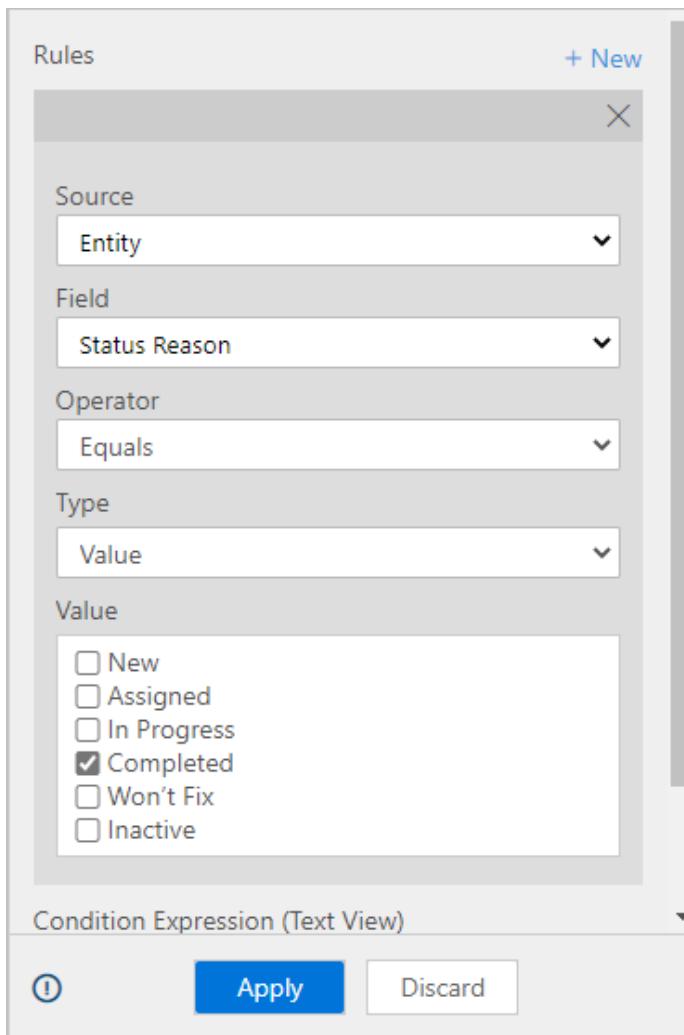
1. Navigate to the [Power Apps maker portal](#) page and make sure you are in the correct environment.
2. Select **Solutions** and click to open the **Company 311** solution.
3. Locate and click to open the **Problem Report** Table.
4. Select the **Business rules** tab and click **Add business rule**.

The screenshot shows the Dynamics 365 Business Central interface. At the top, there are several navigation links: 'Add business rule', 'Add subcomponents', 'Edit data in Excel', and 'Get data'. Below these, the breadcrumb path 'Solutions > Company 311 > Problem Report' is displayed. The main navigation bar has tabs for 'Fields', 'Relationships', 'Business rules' (which is underlined, indicating it's the active tab), 'Views', 'Forms', 'Dashboards', and 'Ch'. A search bar labeled 'Name ↑' is located below the navigation bar. The main content area is currently empty.

1. Make sure the **Scope** is set to **Table** and click **Show details**.



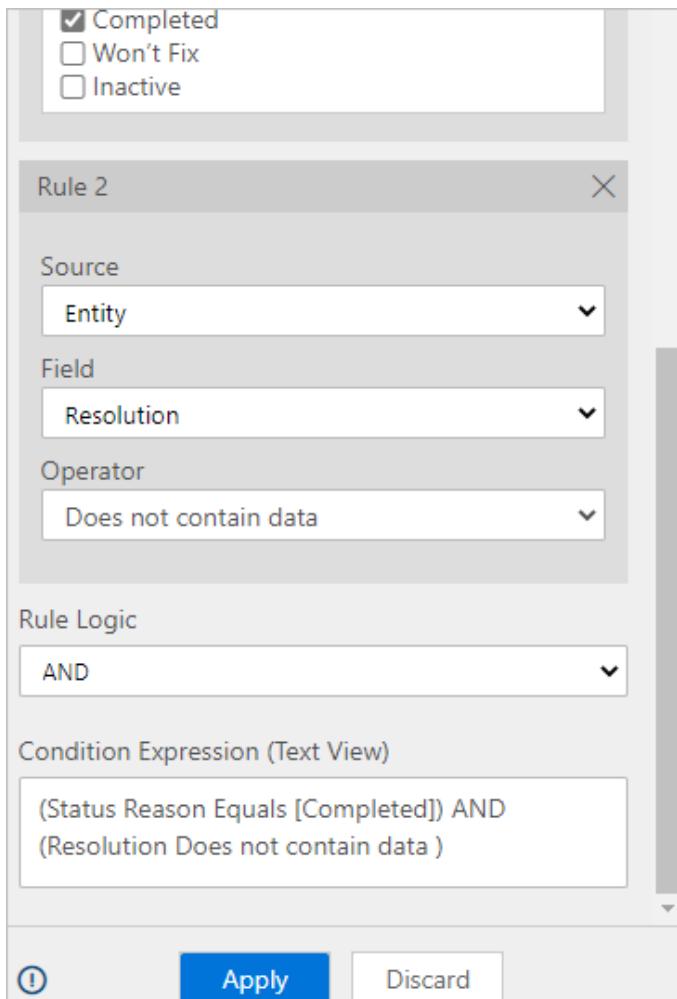
1. Change **Business rule name** to **Completion rule** and click **Hide details**.
2. Select the **Condition**.
3. Go to the **Properties** pane and change the **Display name** to **Resolution required**.
4. Scroll down to **Rule 1**, select **Status Reason** for **Column**, select **Equals** for **Operator**, select **Value** for **Type**, select **Completed** for **Value**, and click **Apply**.



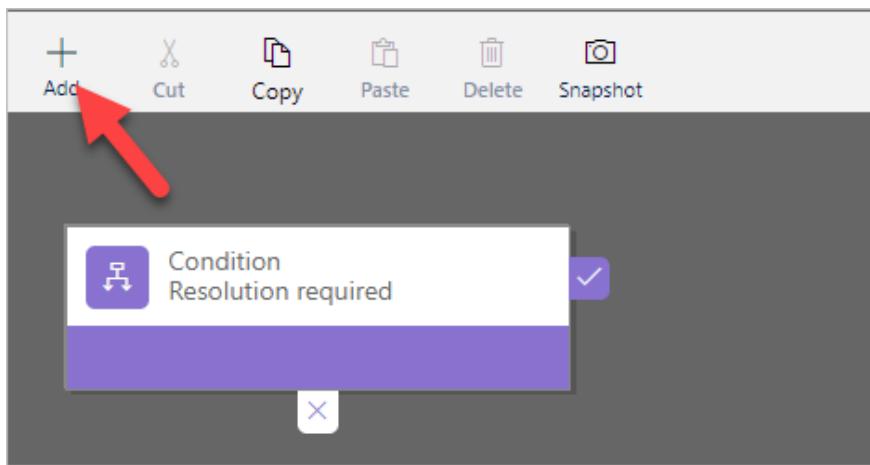
1. Click **+ New**.



1. Scroll down to **Rule 2**, select **Resolution** for **Column**, select **Does not contain data** for **Operator**, make sure **And** is selected for **Rule Logic**, and click **Apply**.

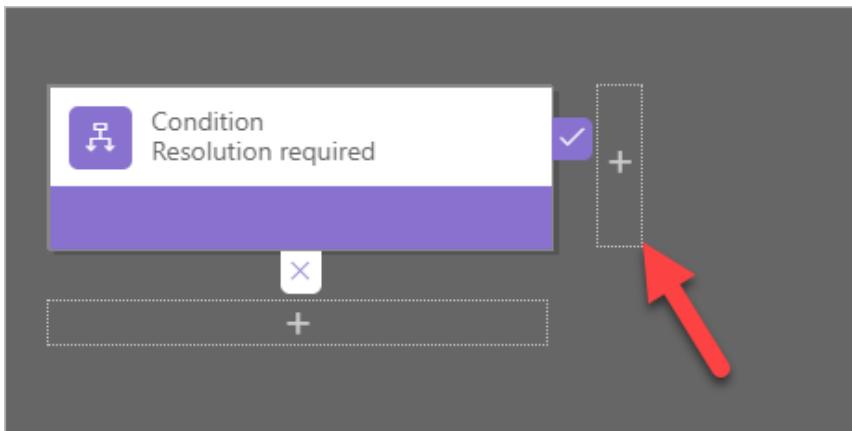


1. Click + Add.

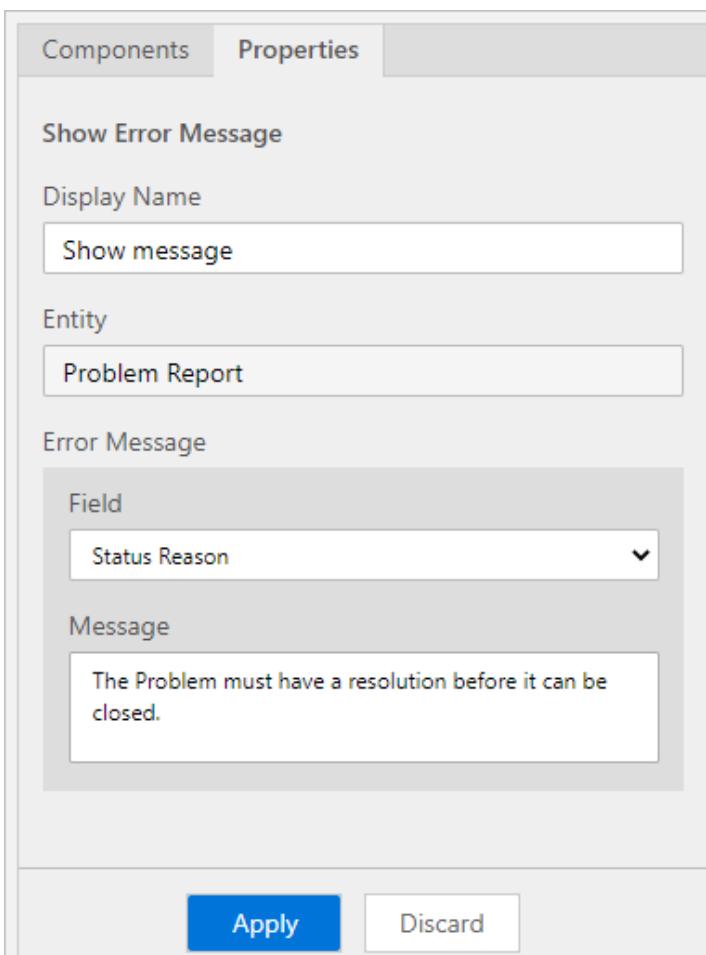


1. Select **Add show error message**.

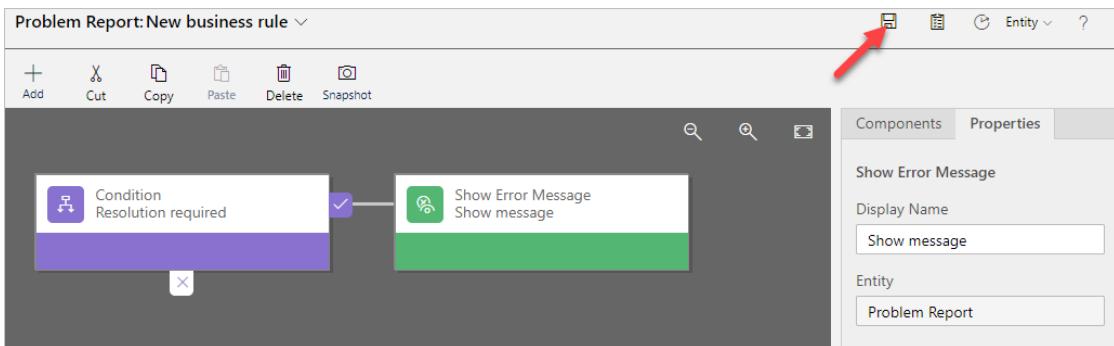
2. Add the action on the true path of the condition.



1. Select the new action, go to the **Properties** pane, enter **Show message** for **Display Name**, select **Status Reason** for **Column**, enter **The Problem must have a resolution before it can be closed** for **Message**, and click **Apply**.



1. The business rule should now look like the image below. Click **Save**.



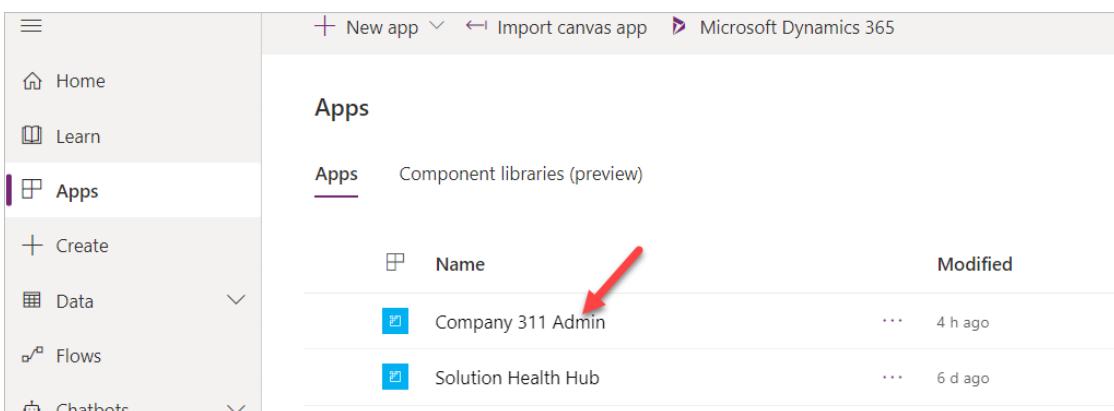
1. Click **Activate**.
2. Confirm activation.
3. Close the process editor browser window or tab.
4. Click **Done**.

Exercise 3: Test processes

In this exercise, you will test the business process flow and the business rule you created.

Task 1: Test processes

1. Navigate to the [Power Apps maker portal](#) page and make sure you are in the correct environment.
2. Select **Apps** and click to open the **Company 311 Admin** application.



1. Select **Problem Reports** and click **+ New**.
2. You should see the business process flow stages. Enter **Dark parking lot** for **Title**, select **London Paddington** for **Building**, enter **There are no lights at the north end of the parking lot**, and click **Save**.

New Problem Report

Problem resolution proc... Active for less than one mi... < Route (< 1 Min) Fix Resolve >

General

Title	* Dark parking lot
Owner	* MOD Administrator
Building	London Paddington
Details	* There are no lights at the north end of the parking lot.

Photo This record hasn't been created yet. To enable this content, create this record

1. Click on the **Route** stage.

DP Dark parking lot Problem Report

Problem resolution proc... Active for less than one mi... < Route (< 1 Min) Fix Resolve >

General Related

Title	* Dark parking lot
-------	--------------------

1. Enter **North end** for **Location**, select **Facility Maintenance** for **Department**, and select the **Fix** stage.

DP Dark parking lot Problem Report

Problem resolution proc... Active for less than one mi... < Route (< 1 Min) Fix Resolve >

General Related

Title	* Dark parking lot
Owner	* MOD Administrator
Building	London Paddington
Details	* There are no lights at the north end of the parking lot.

Active for less than one minute

- ✓ Building London Paddington
- ✓ Location North end
- ✓ Department Facility Maintenance

1. Select a user for **Assign to** and click on the **Resolve** stage.
2. Select date and time for the **Resolved on** and leave the **Resolution** value empty.
3. Scroll down to the resolution details section and select **Completed** for **Status Reason**. You should see the business rule error message.

Save Save & Close New Deactivate Delete Refresh Process Add to Queue Queue Item Details Assign :

Status Reason : The Problem must have a resolution before it can be closed.

Dark parking lot
Problem Report

Problem resolution proc... Active for 6 minutes < Route (6 Min) Fix Resolve

General Related

Resolution details

Department	<input checked="" type="checkbox"/> Facility Maintenance
Status Reason	Completed <input checked="" type="checkbox"/> The Problem must have a resolution before it can be closed.
Resolved on	7/31/2020 8:00 AM

1. Provide Resolution. The error message should go away.

Resolution details

Department	<input checked="" type="checkbox"/> Facility Maintenance
Status Reason	Completed
Resolved on	7/31/2020 8:00 AM
Resolution	Install lights on the North end of the parking lot

11. Save the Row.

lab: title: 'Lab: Power Automate' module: 'Module 5: Power Automate'

[!NOTE] Effective November 2020: - Common Data Service has been renamed to Microsoft Dataverse. [Learn more](#) - Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. [Learn more](#)

This content will be updated soon to reflect the latest terminology.

Lab 05: Power Automate

In this lab, you will create Power Automate flows to automate various parts of the Company 311 solution.

The following have been identified as requirements you must implement to complete the project:◆◆

- Escalation, approval, and execution process for urgent maintenance issues◆◆
- Notify reporting user about the issue status changes◆◆
- How to use a business rule to implement logic.

What you will learn

- How to design data Columns to support automation◆◆
- How to build a flow using Microsoft Dataverse◆◆
- How to use approvals◆◆

High-level lab steps

- Add Columns to support escalation?♦?
- Build flow to?♦?approve escalation?♦?♦?♦?
- Build flow to notify user of status change
- Build approval as an adaptive card in Microsoft Teams

Prerequisites

- Must have completed **Lab 02: Data model and model-driven app**
- Must have completed **Lab 04: Business Process Flows and Business Rules**

Things to consider before you begin

- What is the most efficient way to identify urgent maintenance issues and escalate them

Detailed steps

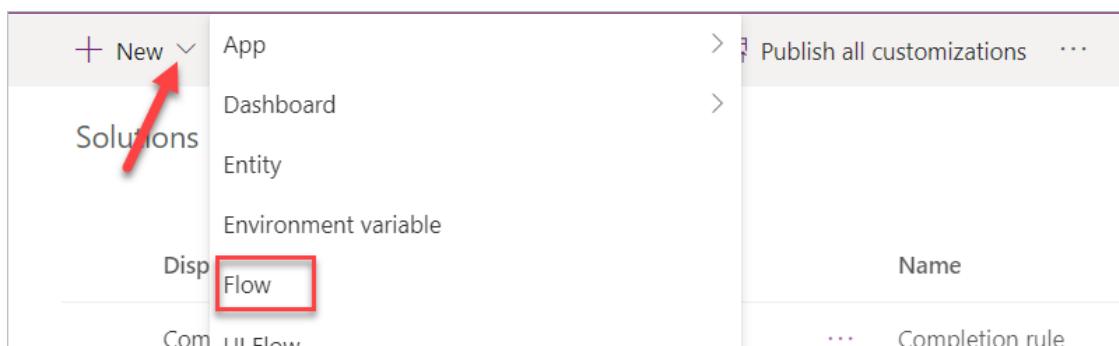
Exercise 1: Build notify flow

In this exercise, you create a flow that will notify the creator of a problem when the status changes.

Task 1: Create flow

In this task, you will create a flow that send notification when the status of problem report Row changes.

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select **Solutions** and click to open the **Company 311** solution.
3. Click **+ New** and select **Cloud Flow**.



1. Select Common Data Service. ! Select trigger - screenshot
`/home/l1/Azure_clone/Azure_new/PL-100-Microsoft-Power-Platform-App-Maker/Instructions/Labs/05/media/imageC1.png`

2. Select When a row is added, modified or deleted

The screenshot shows the Microsoft Flow interface for selecting a trigger. At the top, there's a search bar with the placeholder "When a record is created". Below it is a navigation bar with tabs: All (selected), Built-in, Standard, Premium, Custom, and My clipboard. A horizontal bar with colored segments (purple, teal, purple, blue, green, blue, blue) serves as a visual separator. Under the tabs, there are two main sections: Triggers and Actions. The Triggers section is currently active. It lists several triggers, with the first one, "When a record is created, updated or deleted" (Common Data Service (current environment)), highlighted with a red border. This trigger is labeled as PREMIUM. Below it is another trigger, "When a file is created in a folder" (SharePoint), which is not highlighted.

3. Select **Update** for Change type; **Problem Reports** for Table name; **Organization** for Scope, and click **Show advanced options**.

4. Enter **statuscode** for Column filter and **⋮** Menu button of the trigger step.

This screenshot shows the configuration details for the selected trigger. The title is "When a record is created, updated or deleted". The configuration fields are as follows:

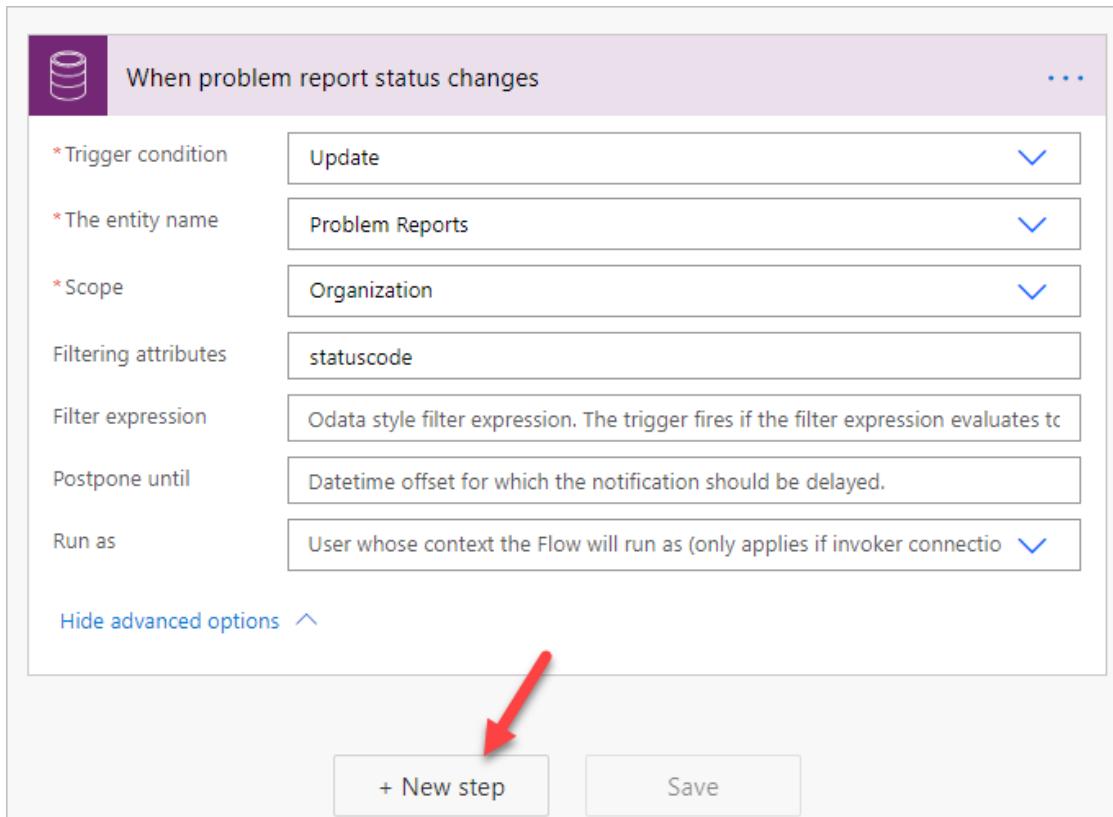
- * Trigger condition: Update
- * The entity name: Problem Reports
- * Scope: Organization
- Filtering attributes: statuscode (this field is highlighted with a red box)
- Filter expression: Odata style filter expression. The trigger fires if the filter expression evaluates to
- Postpone until: Datetime offset for which the notification should be delayed.
- Run as: User whose context the Flow will run as (only applies if invoker connection)

A red arrow points to the three-dot menu icon (⋮) located at the top right of the configuration area. Below the configuration, there's a link to "Hide advanced options".

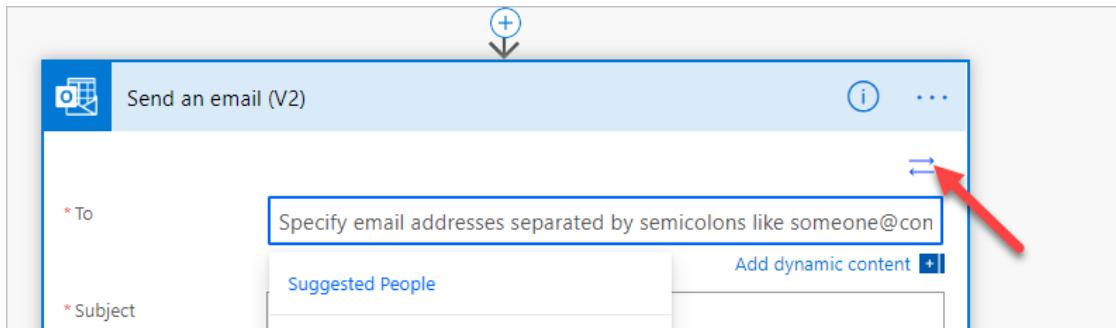
1. Select **Rename**.

2. Rename the trigger step **When problem report status changes**.

3. Click **+ New step**.

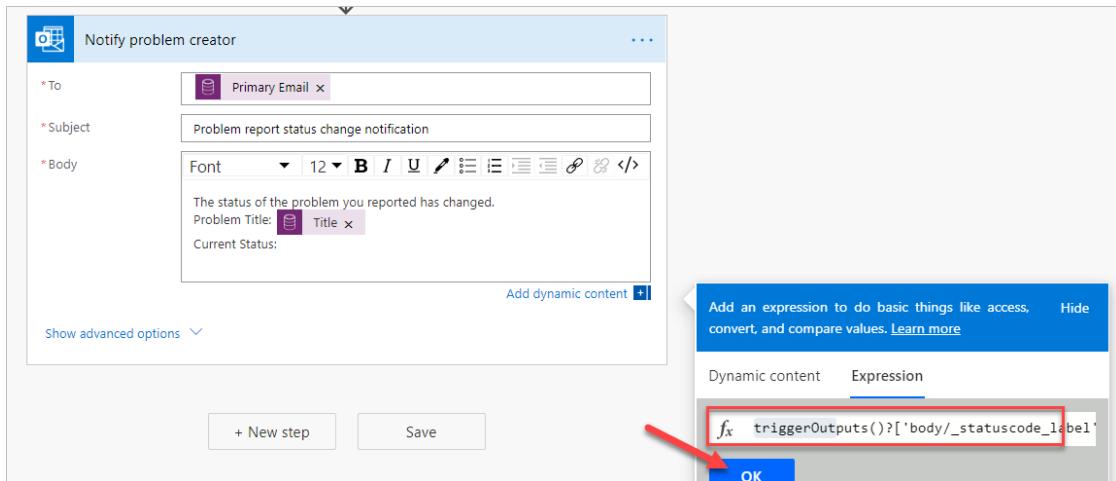


1. Select Common Data Service. Click on **Actions**, then select **Get a Row by ID**.
2. Select **Users** for **Table name**.
3. Click on the **Row ID** Column, go to the Dynamic pane, search for **created** and click once on **Created By (Value)** to add it.
4. Click **Show advanced options** of the new step.
5. Enter **internalemailaddress** for **Select columns**.
6. Click on the **☰** **Menu** button of the new step and select **Rename**.
7. Rename the step **Get problem creator**.
8. Click **+ New step**.
9. Search for **send email** and select **Send an email (V2)**.
10. Click to select the **To** Column and click **Switch to advanced mode**.
Click on this button toggles show/hide of the dynamic pane.

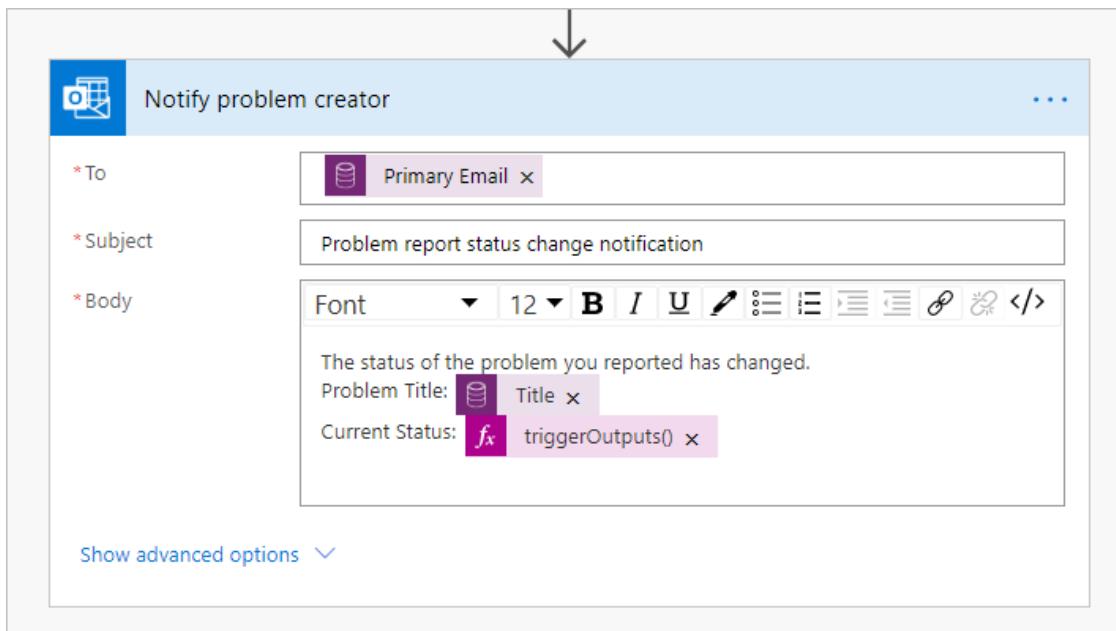


1. Select the **Primary Email** Column form the **Get problem creator** step.
2. Enter **Problem report status change notification** for **Subject**.
3. Click to select the **Body** Column.
4. Type **The status of the problem you reported has changed.** and press the **[ENTER]** key.
5. Type **Problem Title:** go to the Dynamic pane, search for **title** and select **Title**.
6. Press the **[ENTER]** key.
7. Type **Current Status:** go to the Dynamic pane, select the **Expression** tab, paste the expression below, and click **OK**. This expression will show the label of the Choice instead of the value.

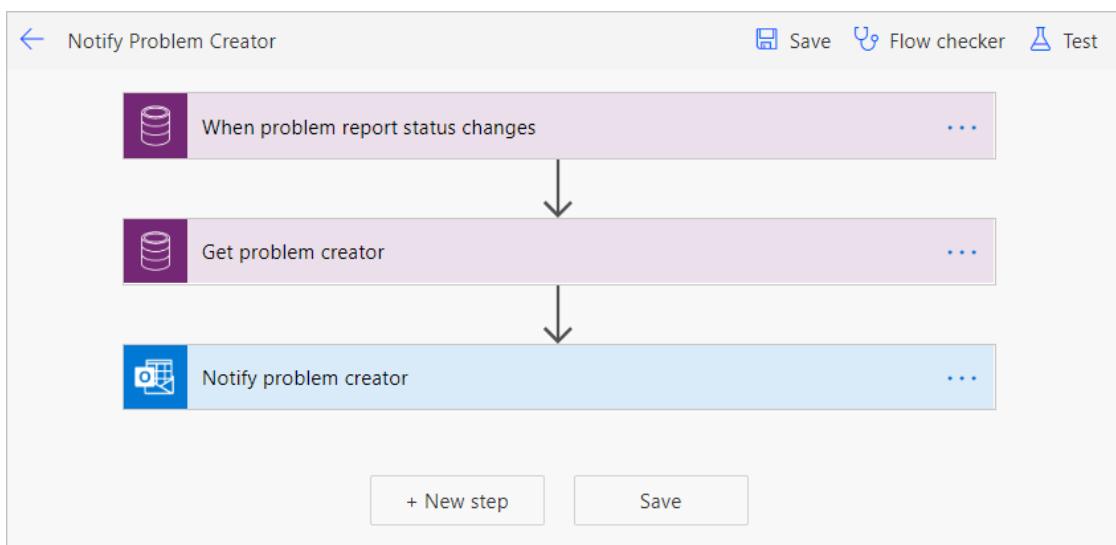
```
triggerOutputs()?'body/_statuscode_label'
```



1. Click on the **☰** **Menu** button of the new step and select **Rename**.
2. Rename the **Notify problem creator**.
3. The step should now look like the image below.



1. Scroll up change the flow name from Untitled to **Notify Problem Creator**.
2. Click **Save** to save the flow.



1. Close the flow designer browser window or tab.
2. Clock **Done** on the popup window.

Task 2: Test the flow

In this task, you will test the notify problem creator flow.

1. Make sure you are still on the [Power Apps maker portal](#) site and you are in the correct environment.

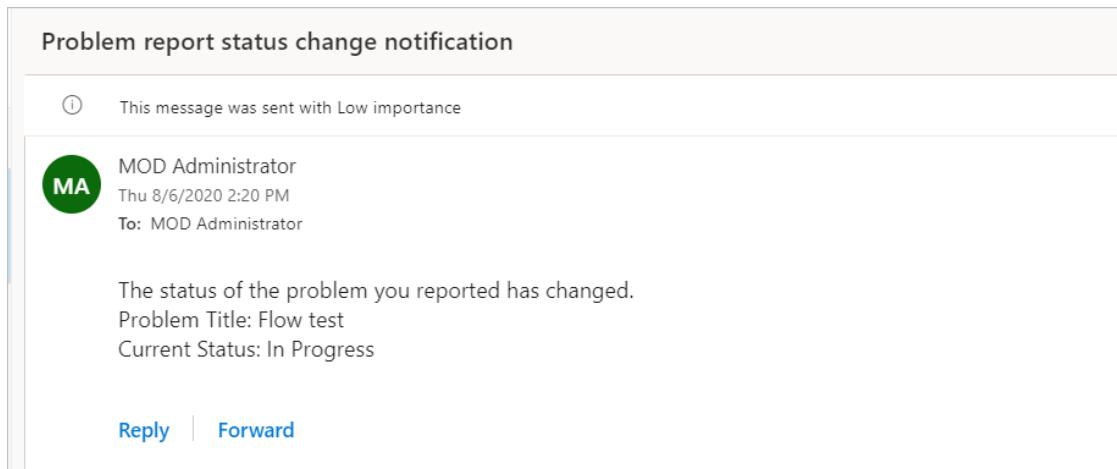
2. Select **Apps**, and then select the **Company 311 Admin** Model-driven application. Click **Play**.
3. Click **+ New**.
4. Enter **Flow test** for **Title**, select **London Paddington** for **building**, enter **This is a flow test Row** for **Details**, and click **Save**.
5. Scroll down and change the **Status Reason** value to **In Progress** and save again.
6. Close the application browser window or tab.
7. You should now be back to the [Power Apps maker portal](#)
8. Select **Solutions** and click to open the **Company 311** solution.
9. Locate and click to open the **Notify Problem Creator** flow you created.
10. You should see a succeeded flow run in the **28-day run history** section. Click to open the run.

Start	Durati...	Status
Aug 6, 03:06 PM (2 min ago)	00:00:04	Succeeded

1. All the flow steps should have a green check mark.
2. Click **App launcher** and select **Outlook**.

The screenshot shows the Microsoft App launcher interface. On the left, there's a sidebar with the title "Apps" and a list of icons for various Microsoft services: Outlook (selected and highlighted with a red box), Word, PowerPoint, SharePoint, Yammer, OneDrive, Excel, OneNote, Teams, and Dynamics 365. The main area displays a flow run history for the "Notify problem creator" flow. The first step, "When problem report status changes", completed successfully in 0s. The second step, "Get problem creator", completed successfully in 1s. The third step, "Notify problem creator", completed successfully in 3s. The overall status is "Succeeded".

1. You should get an email from the flow. Click to open the email.
2. The email should look like the image below.



Exercise 2: Build escalation flow

In this exercise, you create add two new Columns to the problem report Table and create escalation flow.

Task 1: Add Columns

In this task, you add a new Columns to the problem report Table.

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select **Solutions** and click to open the **Company 311** solution.
3. Locate and click to open the **Problem Report** Table.
4. Make sure you have the **Columns** tab selected and click **+ Add Column**.
5. Enter **Estimated Cost** for **Display name**, select **Currency** for **Data type**, and click **Done**.
6. Click the **Save Table** located on bottom right of the screen.
7. Select the **Forms** tab.
8. Click to open the **Main** form.
9. Add **Estimated Cost** Column to the form and place it below the **Status Reason** Column.

10. Add the **Assign to** Column and place it below the **Estimated Cost** Column.

11. The **Resolution details** section of the form should now look like the image below. Click **Save**.

The screenshot shows a 'Main form' for 'Information' entity. The 'Properties' pane on the right is open, showing 'Entity: Problem Report', 'Display Name: Information', 'Description: A form for this entity.', and 'Max Width (pixels): 1,920'. The main area displays fields: Photo (disabled), Department (---), Status Reason (New), Estimated Cost (---), Assign to (---), Resolved on (---), and Resolution (---). The 'Assign to' field is highlighted with a red box. The top navigation bar includes 'Field', 'Component', 'Form settings', and 'Switch to classic'.

1. Click on the **Back** button located on the top left of the screen.

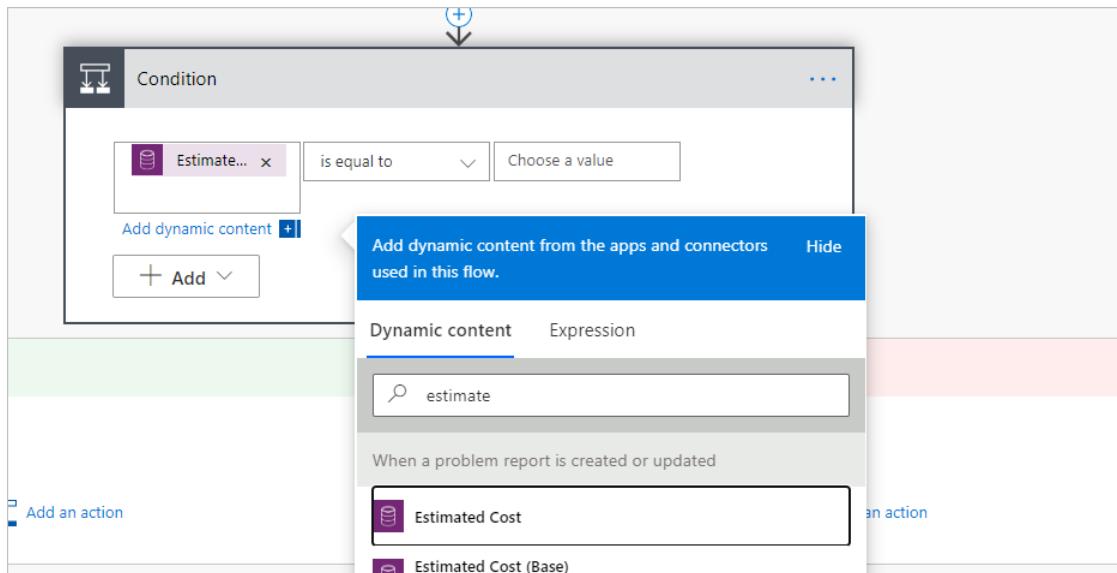
2. Select **Solution**, click **Publish all customizations**, and wait for the publishing to complete.

Task 2: Build escalation flow

In this task, you will create the escalation flow.

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select **Solutions** and click to open the **Company 311** solution.
3. Click **+ New** and select **Cloud flow**.
4. Search for **when a row is added** and select **When a row is added, modified, or deleted Microsoft Dataverse (Current environment)**.
5. Select **Create or Update** for **Change type**; select **Problem Reports** for **Table name**; select **Organization** for **Scope**, and click **Show advanced options**.
6. Enter **lh_estimatedcost** for Column filter and click **Hide advanced options**.
7. Click on the **☰** **Menu** button of the trigger step and select **Rename**.

8. Rename the trigger step **When a problem report is created or updated**.
9. Click **+ New step**.
10. Select **Condition** control.
11. Click to select the first **Choose a value** Column.
12. Go to the Dynamic content pane, search for estimated and select Estimated Cost



1. Select **is greater than** and end enter **1000**.
2. Rename the condition step **Check if cost is greater than 1000**.
3. Go to the **If yes** branch and click **Add an action**.
4. Search for **Get a row** and select **Get a row by ID Microsoft Dataverse (Current environment)**.
5. Select **Users** for **Table name**.
6. Click to select the **Item ID** Column and select **Assign to (Value)** form the **Dynamic content** pane.
7. Click **Show advanced options**.
8. Enter **internalemailaddress** for **Select columns**.
9. Click **Hide advanced option**.

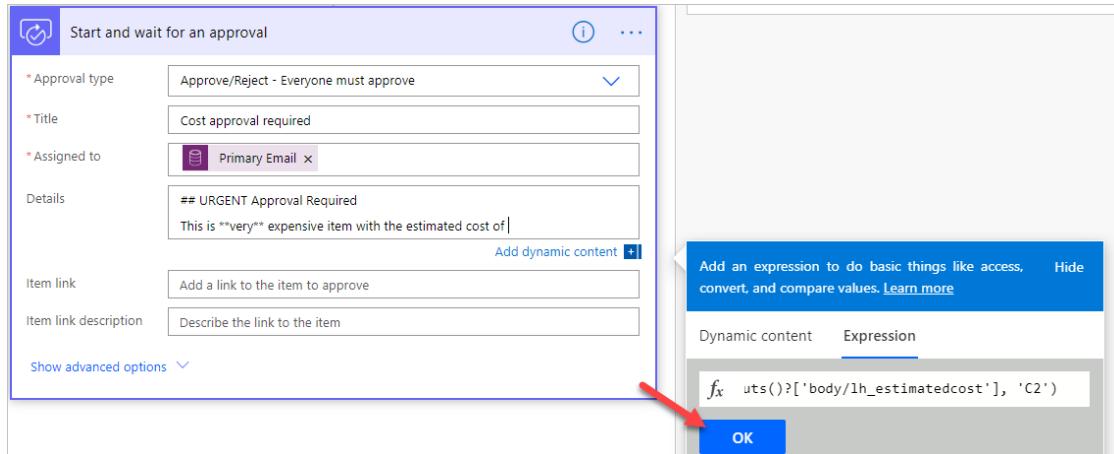
10. Rename the **Get a Row** step **Get user**.
11. Click **Add and action**.
12. Search for **approval** and select **Start and wait for an approval**.
13. Select **Approve/Reject - Everyone must approve** for **Approval type**.
14. Enter **Cost approval required** for **Title**.
15. Click to select the **Assigned to** Column.
16. Go to the **Dynamic content** pane and select **Primary Email** from the **Get user** step.
17. Paste the markdown text below in the **Details** Column.

URGENT Approval Required

This is **very** expensive item with the estimated cost of

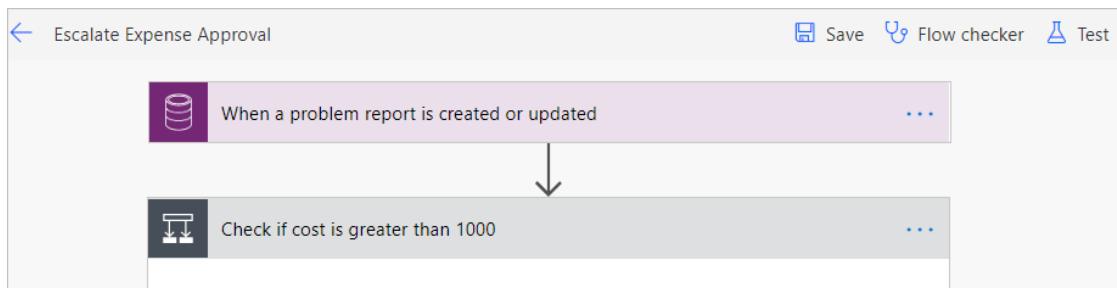
1. Place your cursor after cost of , go to the Dynamic content pane, select the Expression tab, paste the expression below, and click OK.

```
formatNumber(triggerOutputs() ? ['body/lh_estimatedcost'], 'C2')
```



1. Click **Add an action**.
2. Select **Condition** control.
3. Click to select the first **Choose a value** Column.
4. Go to the **Dynamic content** pane, search for **outcome**, and select **Outcome**.

5. Select is **equals to** and type **Reject** for value.
6. Go to the **If yes** branch and click **Add an action**.
7. Search for **update a Row** and select **Update a Row Microsoft Dataverse (Current environment)**.
8. Select **Problem Reports** for **Table name**.
9. Click to select the **Row ID** Column.
10. Go to the **Dynamic content** pane, search for **problem report**, and select **Problem Report**.
11. Click **Show advanced options**.
12. Click to select the **Resolution** Column, go to the **Dynamic content** pane, and select **Response summary**.
13. Select **Won't fix** for **Status Reason**.
14. Rename the step **Update problem report**.
15. Scroll up and rename the flow **Escalate Expense Approval**.
16. Click **Save**.



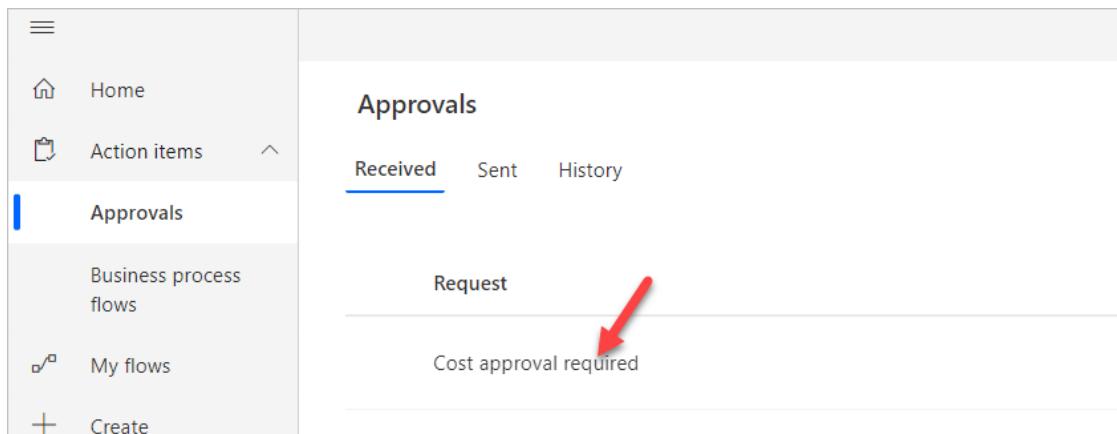
1. Close the flow designer browser window or tab.
2. Click **Done** on the popup.

Task 3: Test flow

In this task, you will test the escalation flow

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.

2. Select **Apps** and click to open the **Company 311 Admin** application.
3. Click to open one of the **Problem Report Rows**.
4. Scroll down, enter 2500 for **Estimated Cost**, assign it to yourself (for test purposes), and click **Save**.
5. Navigate to [**Power Automate**](#)
6. Expand **Action Items** and select **Approvals**.
7. You should see at least one approval in the received tab. Click to open the approval. It can take a few minutes for approvals to show up here on the first run.



1. Select **Reject**, enter **We don't have the funds for this item** for **comment**, and click **Confirm**.

Details

URGENT Approval Required

This is very expensive item with the estimated cost of \$2,500.00

Reject ▾

Add a comment (optional)

We don't have the funds for this item.

Confirm **Cancel**

1. Go back to the **Company 311 Admin** application.
2. Change the view to **My Reports** and click to open the same Row you change the estimated cost.
3. The **Status Reason** should be set to **Won't fix** and the **Resolution** should match the comment you provided.

Resolution details	
Department	---
Status Reason	Won't Fix
Estimated Cost	\$2,500.00
Assign to	 MOD Administrator
Resolved on	---
Resolution	We don't have the funds for this item

, if you have not done so previously.

Exercise 3: Send approval requests as adaptive card in Microsoft Teams

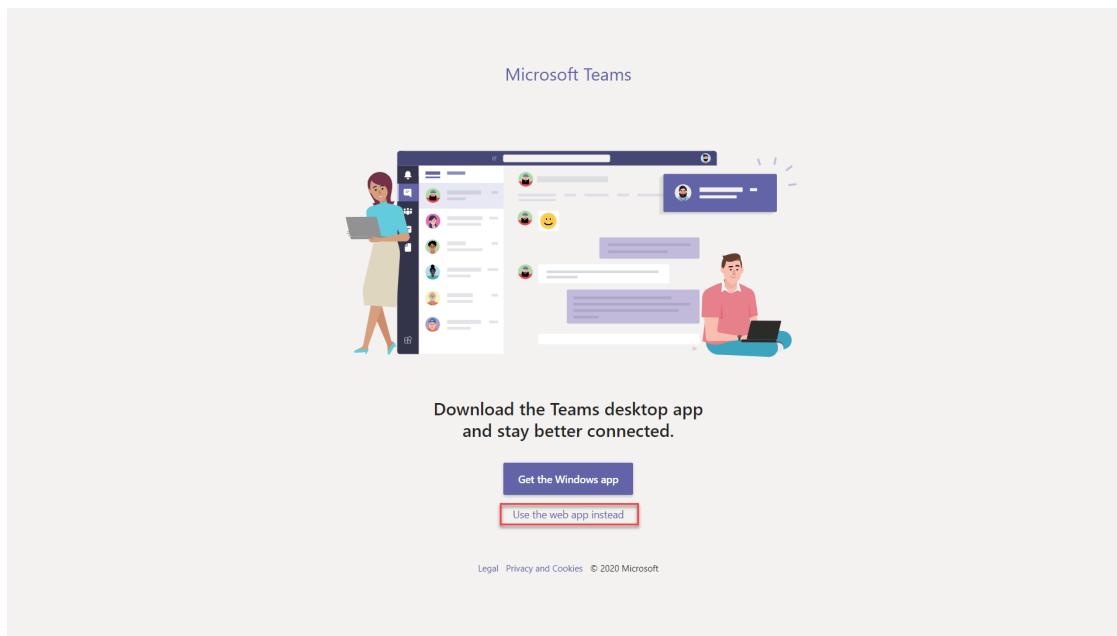
In this exercise, you will setup a team in Microsoft Teams dedicated to the Company 311 applications. You will modify the flow to send the approval request as an adaptive card in Teams chat instead of an approval message.

- Task 1: Setup Company 311 Team
- Task 2: Modify flow to send adaptive card in Teams chat
- Task 3: Test adaptive card

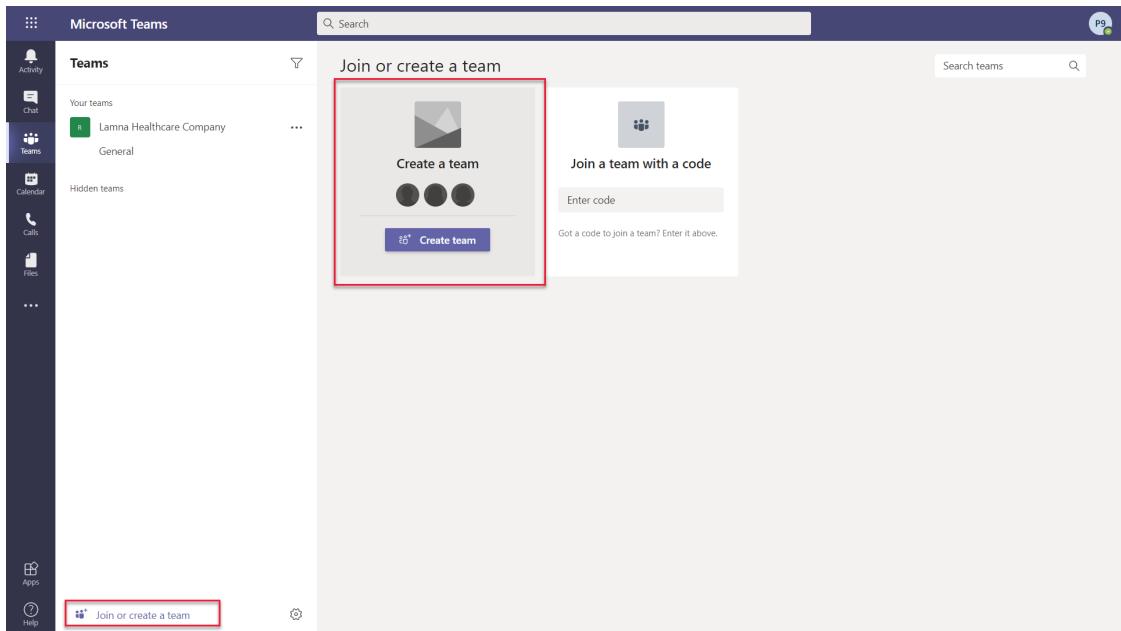
Task 1: Setup Company 311 Team

In this task you will setup a Microsoft Teams team for the Lamna Healthcare Company, if you have not done so previously.

1. Navigate to [Microsoft Teams](#) and sign in with the same credentials you have been using previously.
2. Select **Use the web app instead** on the welcome screen.



1. When the Microsoft Teams window opens, dismiss the welcome messages.
2. On the bottom left corner, choose **Join or create a team**.
3. Select **Create a team**.



1. Press **Build a team from scratch**.
2. Select **Public**.
3. For the Team name choose **Company 311** and select **Create**.
4. Select **Skip** adding members to Company 311.

Task 2: Modify flow to send adaptive card in Teams chat

In this task you will replace the approval sent by email with the adaptive card.

1. Locate **Start and wait for an approval** step created earlier in **Exercise 2, Task 2**.
2. Select ... then select **Delete**.
3. Click + between the steps to insert a new step then select **Add an action**.
4. Search for **approval** and select **Create an approval**.
5. Select **Approve/Reject - Everyone must approve** for **Approval type**.
6. Enter **Cost approval required** for **Title**.
7. Click to select the **Assigned to** Column.
8. Go to the **Dynamic content** pane and select **Primary Email** from the **Get user** step.
9. Paste the markdown text below in the **Details** Column.

{title}

{details}

This is a _very_ expensive item with the estimated cost of

1. Select **{title}** placeholder, go to the **Dynamic content** pane, locate and select **Title Column** from **When a problem report is created or updated** step.
2. Select **{details}** placeholder, go to the **Dynamic content** pane, locate and select **Details Column** from **When a problem report is created or updated** step.
3. Place your cursor after **cost of**, go to the **Dynamic content** pane, select the **Expression** tab, paste the expression below, and click OK.

```
formatNumber(triggerOutputs() ? ['body/lh_estimatedcost'], 'C2')
```

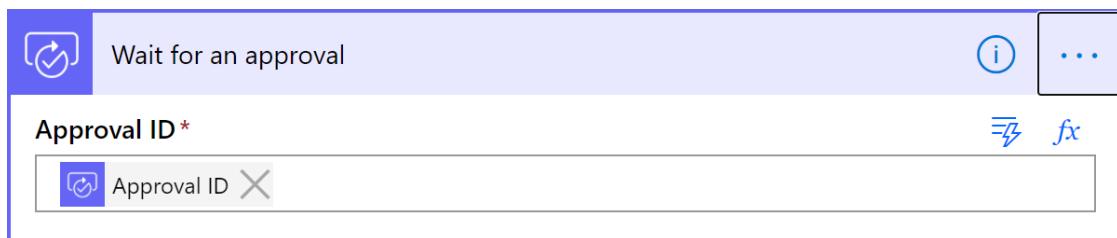
1. Your step should look like the following:

The screenshot shows the configuration of a 'Create an approval' step in Microsoft Flow. The step has the following fields:

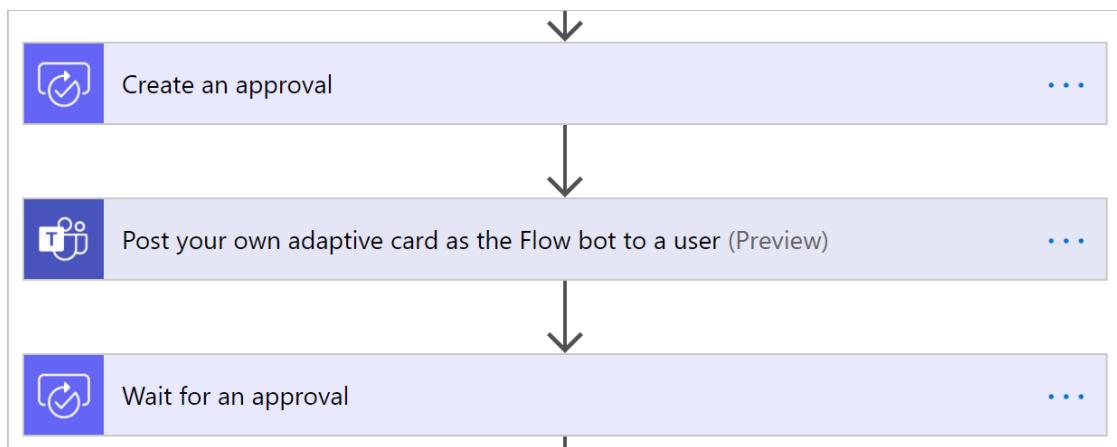
- Approval type ***: Set to "Approve/Reject - Everyone must approve".
- Title ***: Set to "Cost Approval Required".
- Assigned to ***: Set to "Primary Email".
- Details**: Contains two dynamic content blocks:
 - ** Title **: Contains "Title" and a delete icon.
 - Details: Contains "Details" and a delete icon.
 Below these blocks is a note: "This is a _very_ expensive item with the estimated cost of" followed by a formula placeholder "formatNumber(...)" and a delete icon.
- Item link**: A note: "Add a link to the item to approve".
- Item link description**: A note: "Describe the link to the item".
- Show advanced options**: A blue link at the bottom left of the step configuration area.

1. Select + then select **Add an action**.
2. Search for **teams** and select **Post your own adaptive card as the Flow bot to a user action**.

3. Click to select the **Recipient** Column.
4. Go to the **Dynamic content** pane and select **Primary Email** from the **Get user** step.
5. Click to select **Message** Column.
6. Go to the **Dynamic content** pane and select **Adaptive card** from the **Create an approval** step.
7. Select + then select **Add an action**.
8. Search for **approval** and select **Wait for an approval** action.
9. Select **Approval ID** Column.
10. Go to the **Dynamic content** pane and select **Approval ID** from the **Create an approval** step.



1. You now have replaced **Start and wait for an approval** step with the following:



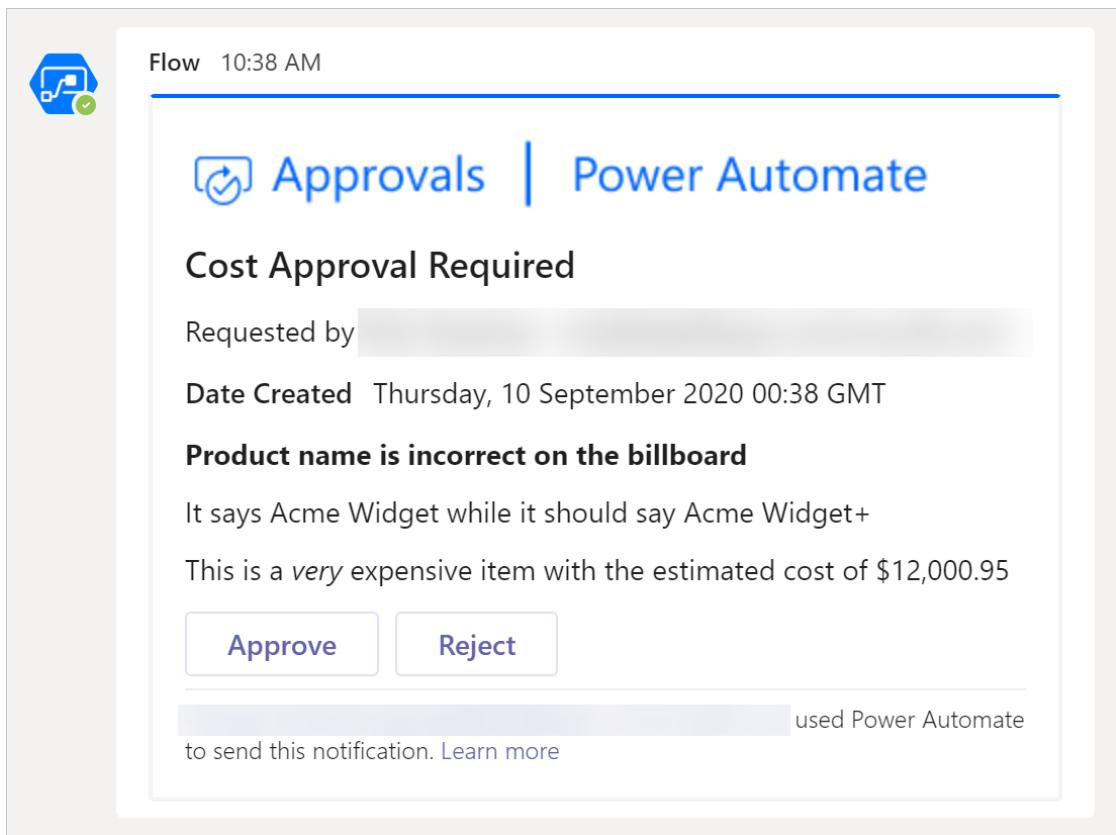
1. Expand **Condition 2** step. The left side of the condition should be empty because it was referring the step now removed.
2. Go to the **Dynamic content** pane, search for **outcome**, and select **Outcome** from **Create an approval** step.

3. Local **Update problem report** step under **If yes** branch.
4. Click **Show advanced options**.
5. Click to select the **Resolution** Column, go to the **Dynamic content** pane, and select **Response summary** from **Create an approval** step.

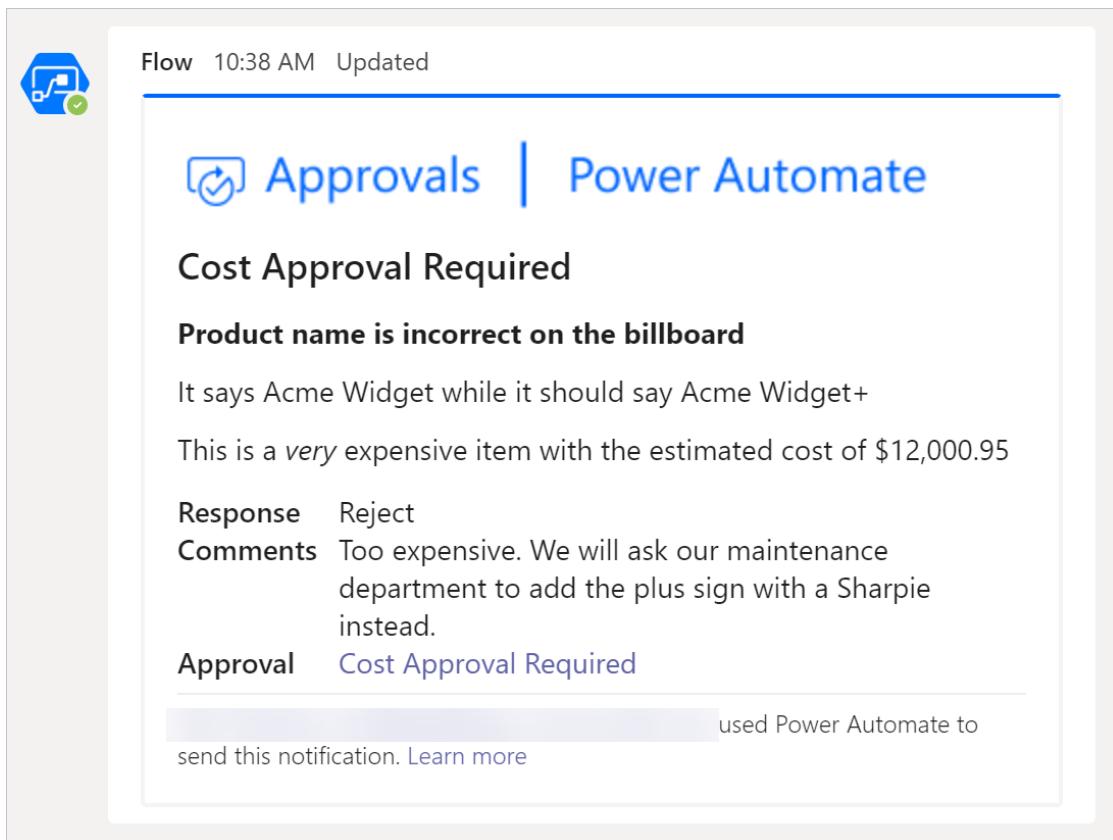
Task 3: Test flow

In this task, you will test the escalation flow with the Teams and adaptive cards.

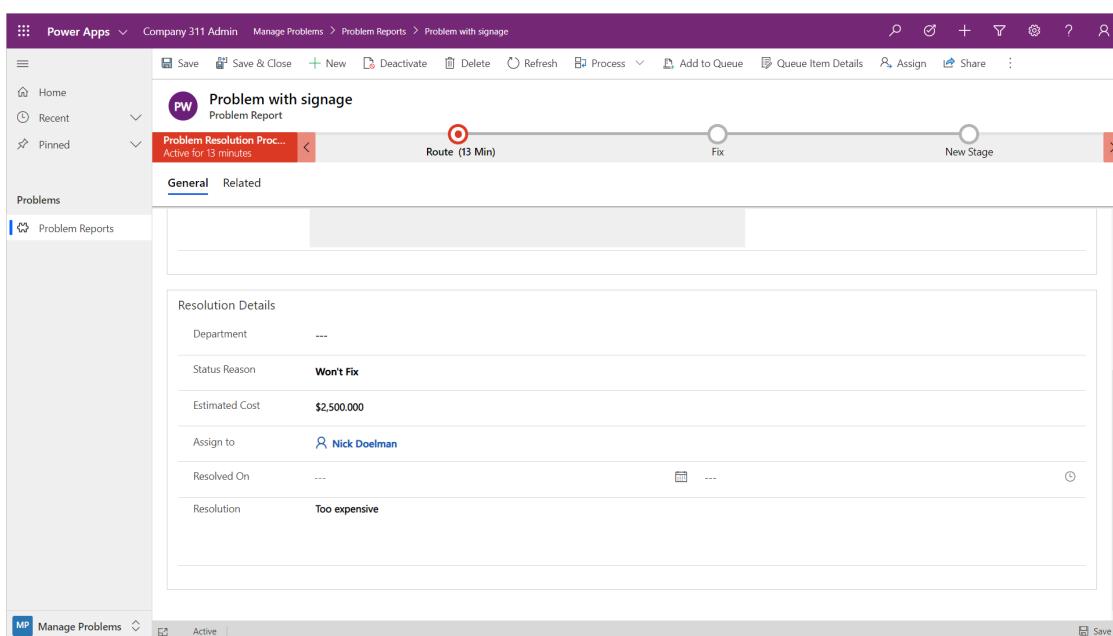
1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select **Apps** and click to open the **Company 311 Admin** application.
3. Click to open one of the **Problem Report** Rows.
4. Scroll down, enter any amount greater than **1000** for **Estimated Cost**, assign it to yourself (for test purposes), and click **Save**.
5. Navigate to [Microsoft Teams](#)
6. Select **Chat**.
7. You should see the Cost Approval Required Adaptive Card.



1. Press **Reject** button and enter a comment of your choice in the Comments area, for example **The item is too expensive.**
2. Select **Submit**. The card will become read-only.



1. Go back to the **Company 311 Admin** application.
2. Change the view to **My Reports** and click to open the same Row you change the estimated cost.
3. The **Status Reason** should be set to **Won't Fix** and the **Resolution** should match the comment you provided.



Discussion

- Would creating a bool Column for Approved/Rejected be better?
- What are the pros and cons of using Microsoft Teams over regular email?

Bonus exercises

- Add ability for the users to subscribe to the reported problems and only notify if there is a subscription.
 - Auto-subscribe creator of the problem report.
 - How to find out previous value of status reason?
 - Create your own adaptive card using [Adaptive Cards Designer](#).
-

lab: title: 'Lab: Power BI' module: 'Module 6: Power BI'

[!NOTE] Effective November 2020: - Common Data Service has been renamed to Microsoft Dataverse. [Learn more](#) - Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. [Learn more](#)

This content will be updated soon to reflect the latest terminology.

LAB 06: Power BI

In this lab, you will build a Power BI dashboard that visualizes data about problems reported by company employees.

What you will learn

- How to connect to the data
- How to refine the data model and prepare it for reporting
- How to create a Power BI visualization
- How to embed a Power BI report in Microsoft Teams

High-level lab steps

We will follow the below steps to design and create the Power BI dashboard:

- Connect to Microsoft Dataverse
- Transform the data to include user-friendly descriptions for the related Rows (lookups)
- Create and publish a report with various visualizations of the information about problem reports
- Use natural language query to build additional visualizations
- Build mobile view
- Embed the Company 311 Power BI report to Microsoft Teams

Prerequisites

- Must have completed **Lab 02: Data model and model-driven app**
- Permissions to install programs on your computer (required for Power BI Desktop installation)

Things to consider before you begin

- Who is the target audience of the report?
- How will the audience consume the report? Typical device? Location?
- Do you have sufficient data to visualize?
- What are the possible characteristics you can use to analyze data about the visits?

Detailed steps

Exercise 1: Prepare environment & data

Objective: In this exercise, you will install and configure Power BI Desktop and configure a connection to Microsoft Dataverse.

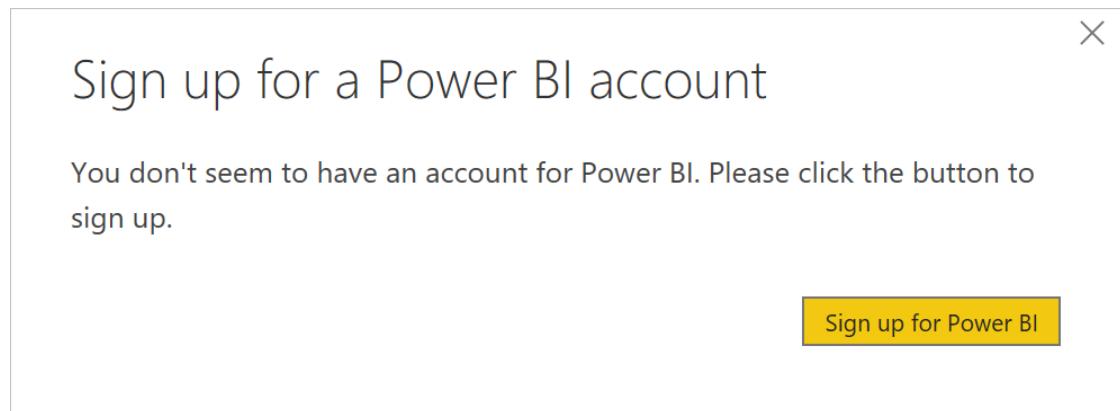
[!IMPORTANT] If you do not have required permissions to install desktop applications or experience difficulties in configuring Power BI Desktop and connecting it to the data, follow **Exercise 5: Import sample data** and then continue on **Exercise 2** but using Power BI service instead of Power BI Desktop.

Task 1: Configure Power BI Desktop

1. If you do not have Power BI Desktop installed, navigate to <https://aka.ms/pbidesktopstore> to download and install Power BI app.

[!IMPORTANT] If you experience issues installing Power BI Desktop using Microsoft Store, try standalone installer that can be downloaded from <https://aka.ms/pbiSingleInstaller>.

1. Open Power BI Desktop
2. If you were signed in into Power BI Desktop previously, select **File | Sign out**
3. Sign in if prompted or select **File | Sign in** to sign in.
4. If you're signing in for the first time you may receive the following prompt

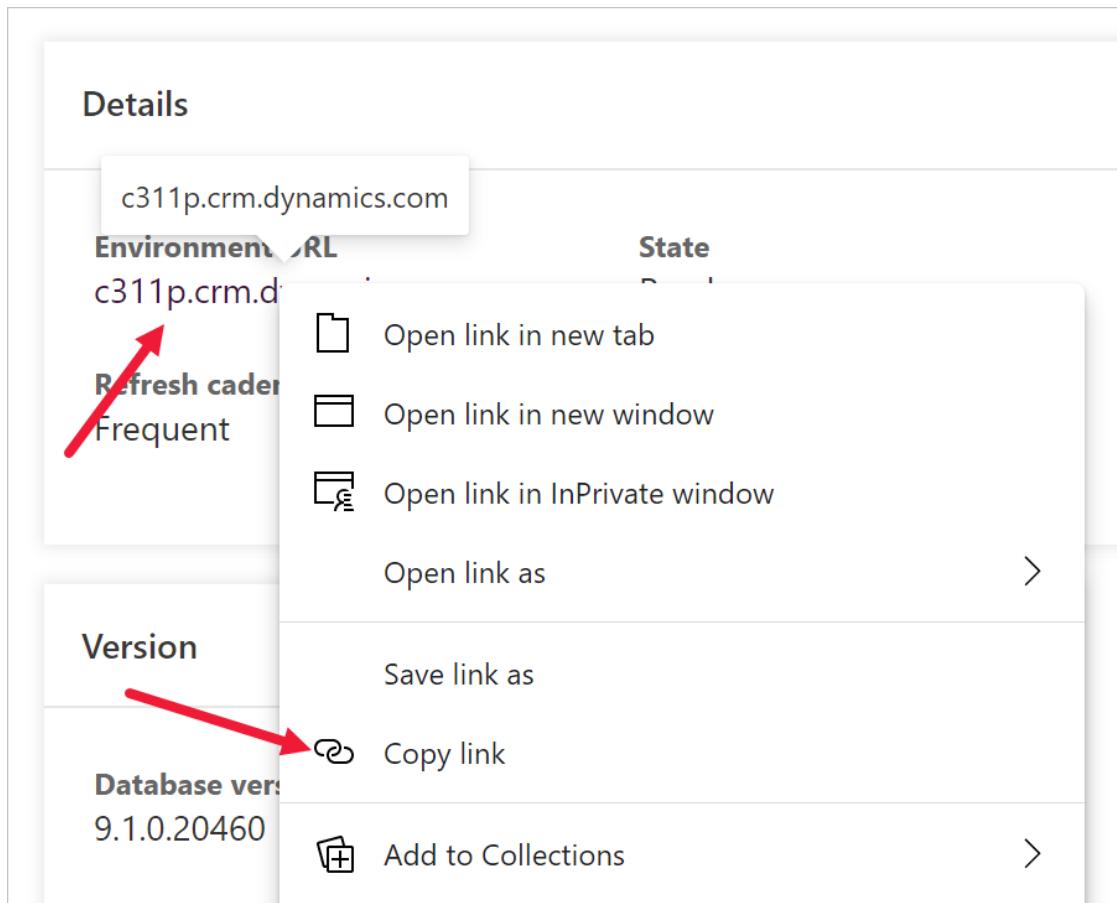


1. Select **Sign up for Power BI** and follow the prompts to complete the sign up

Task 2: Prepare Data

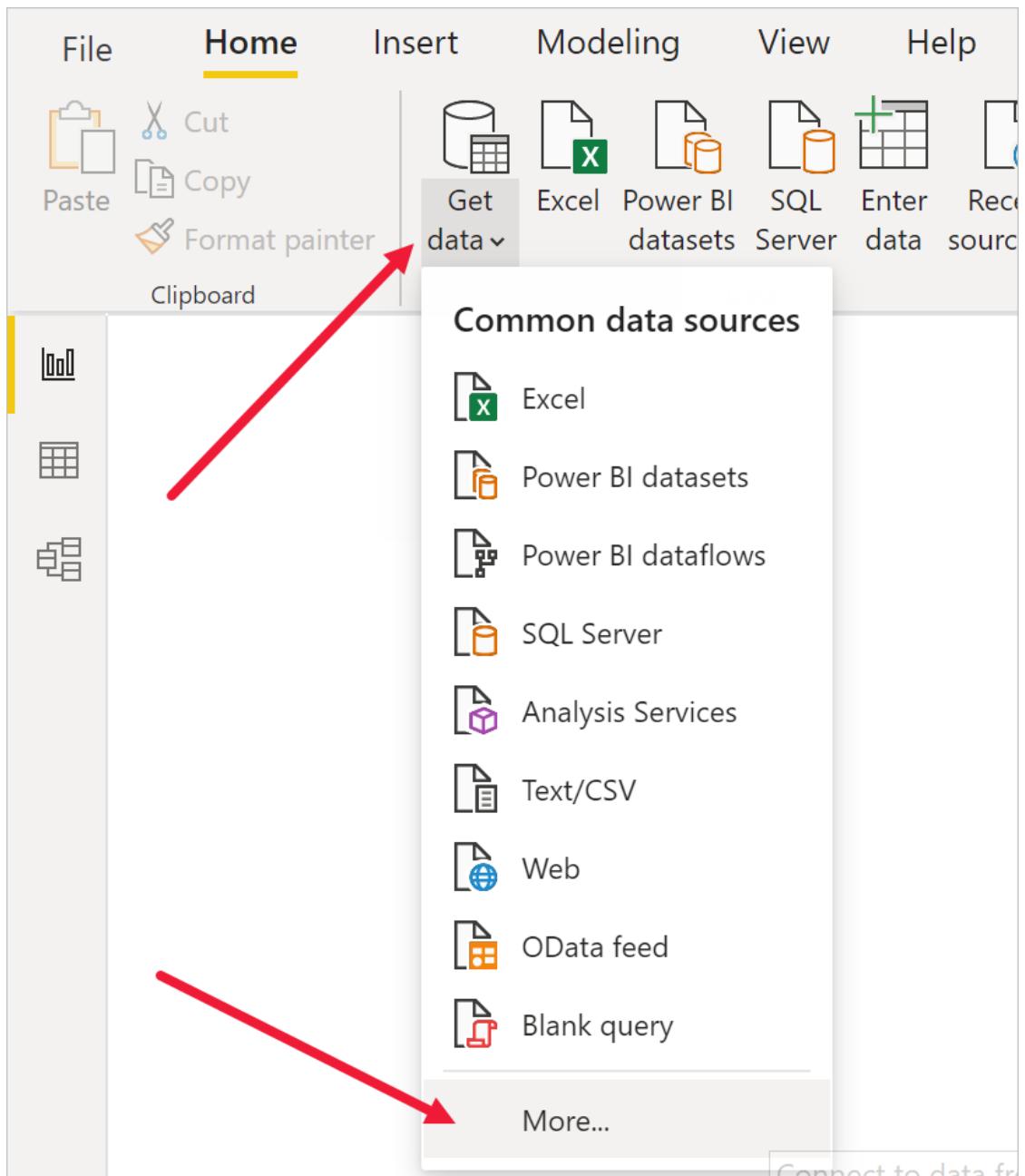
1. Find out your organization URL

- Navigate to Power Platform Admin Center at <https://aka.ms/ppac>.
- In the left navigation page, select Environments, and then click on the target environment.
- Right mouse click **Environment URL** on the **Details** panel, then select **Copy link**.

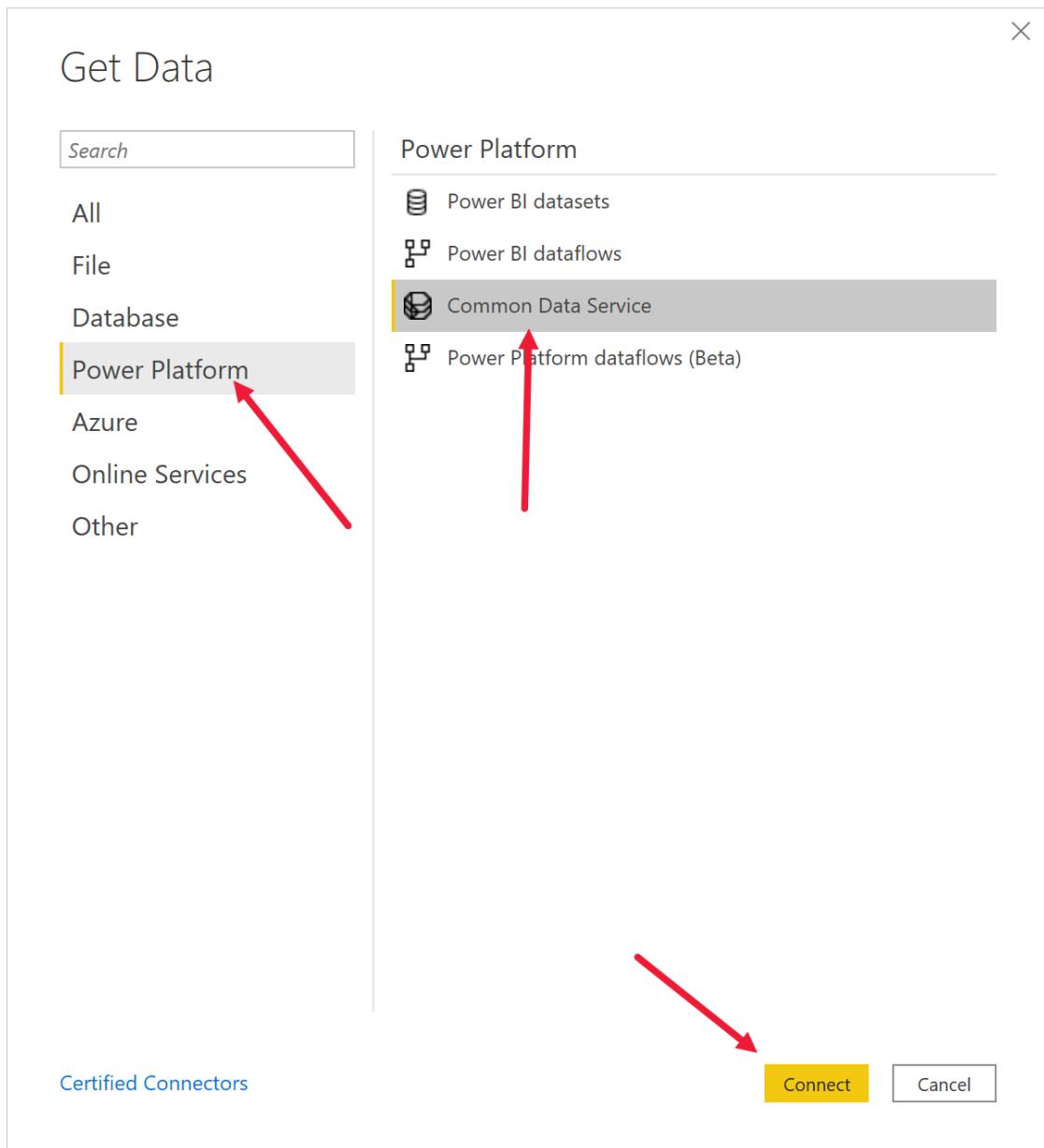


1. Switch to Power BI desktop

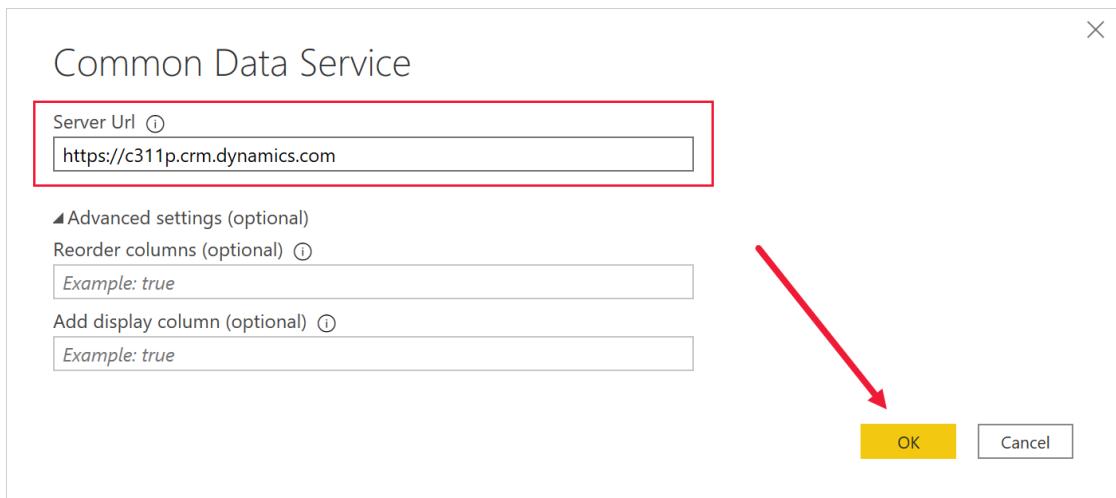
2. Select **Get data | More...**



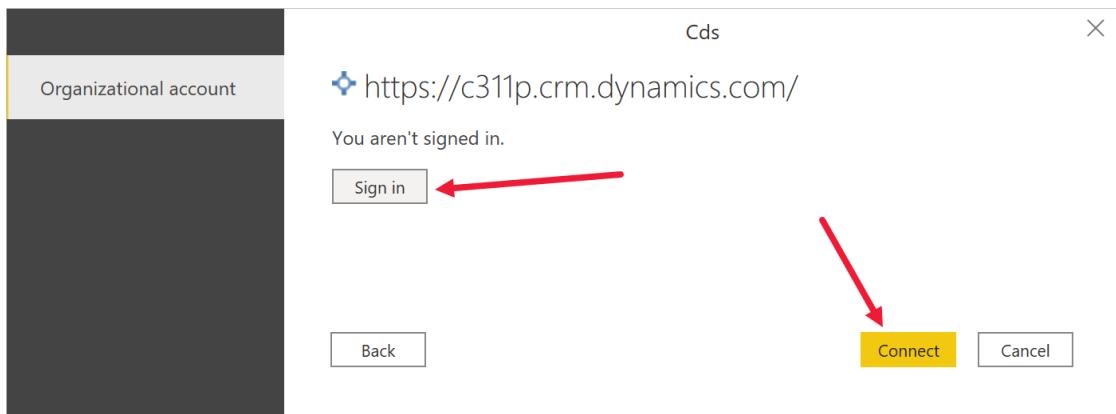
1. Select **Power Platform**, then select **Common Data Service**, and press **Connect**.



1. Type **https://** and paste the environment URL you copied earlier. Click **OK**.



1. The connection details dialog will open up. If you are not signed in, press **Sign in** and follow the prompts to sign in. Press **Connect**.



1. Expand **Entities** node, select **lh_Building**, **lh_Department**, **lh_ProblemReport** Tables, press **Load**. Wait until the load is complete.

Navigator

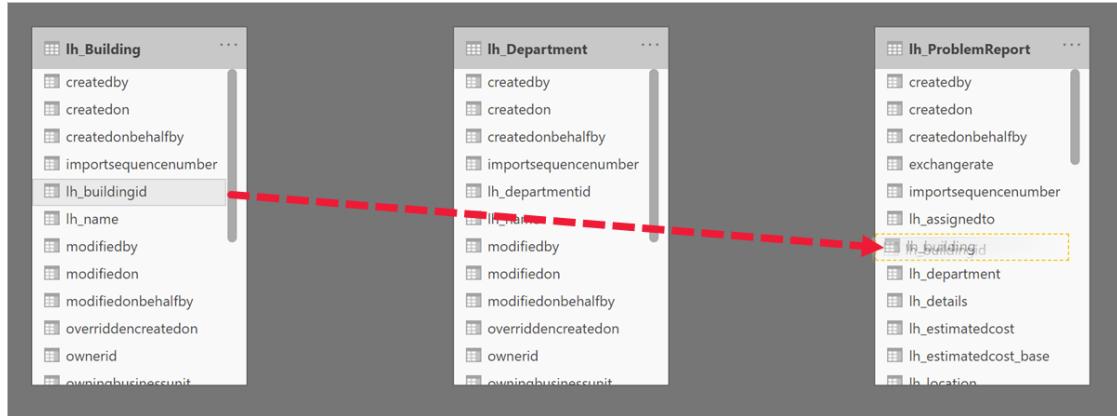
createdby	createdon	createdonbehalfby
20badf32-aad6-4619-93c5-76ba1ca1a19c	29/07/2020 6:09:49 PM +00:00	
d150cbf1-13d1-ea11-a812-000d3a563be2	29/07/2020 8:53:27 PM +00:00	
20badf32-aad6-4619-93c5-76ba1ca1a19c	31/07/2020 8:40:43 PM +00:00	
20badf32-aad6-4619-93c5-76ba1ca1a19c	31/07/2020 8:39:25 PM +00:00	
ef9635c0-13d1-ea11-a812-000d3a563be2	31/07/2020 1:37:53 AM +00:00	
d150cbf1-13d1-ea11-a812-000d3a563be2	3/08/2020 10:48:22 AM +00:00	
d150cbf1-13d1-ea11-a812-000d3a563be2	7/07/2020 2:40:44 AM +00:00	
d150cbf1-13d1-ea11-a812-000d3a563be2	6/08/2020 3:04:07 PM +00:00	
d150cbf1-13d1-ea11-a812-000d3a563be2	6/07/2020 2:40:43 AM +00:00	
d150cbf1-13d1-ea11-a812-000d3a563be2	7/06/2020 2:40:41 AM +00:00	

Load **Transform Data** **Cancel**

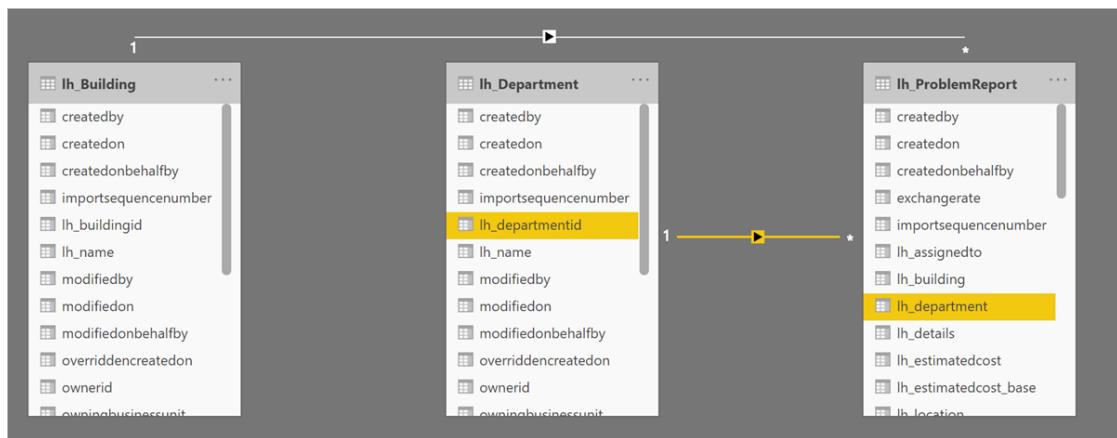
1. Click **Model** icon on the left vertical toolbar.

The screenshot shows the Microsoft Power BI desktop interface. The ribbon at the top has tabs: File, Home (which is selected and highlighted in yellow), Insert, Modeling, View, and Help. Below the ribbon, there are several icons in the 'Clipboard' group: Paste (orange clipboard), Cut (scissors), Copy (copy/paste), and Format painter. In the 'Data' group, there are icons for Get data (with a dropdown menu), Excel, Power BI datasets, SQL Server, Enter data, and Refresh source. On the far left, a vertical toolbar contains icons for Report, PivotTable, and Model. A red arrow points to the 'Model' icon (a cube icon). The main workspace is currently empty.

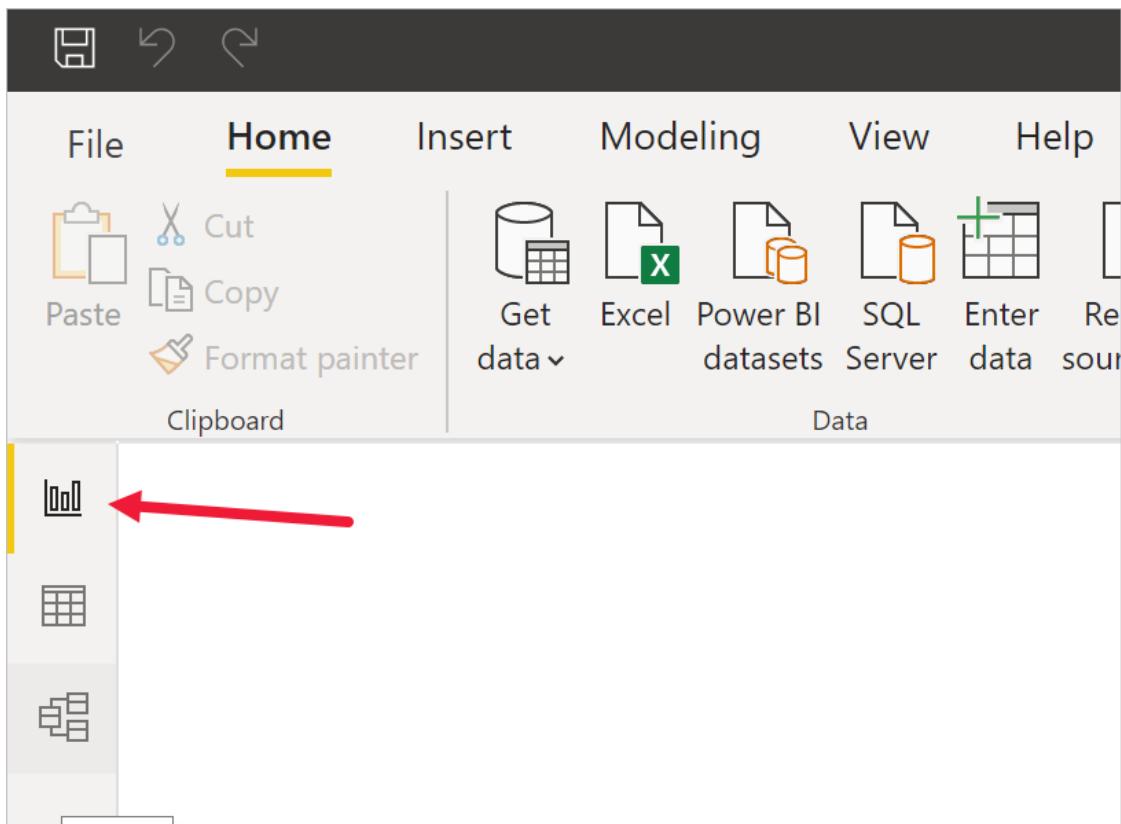
1. Drag **lh_buildingid** column from **lh_Building** table and drop it to **lh_building** column in **lh_ProblemReport** table. That will create a relationship between two Tables that Power BI will be able to use to display related data.



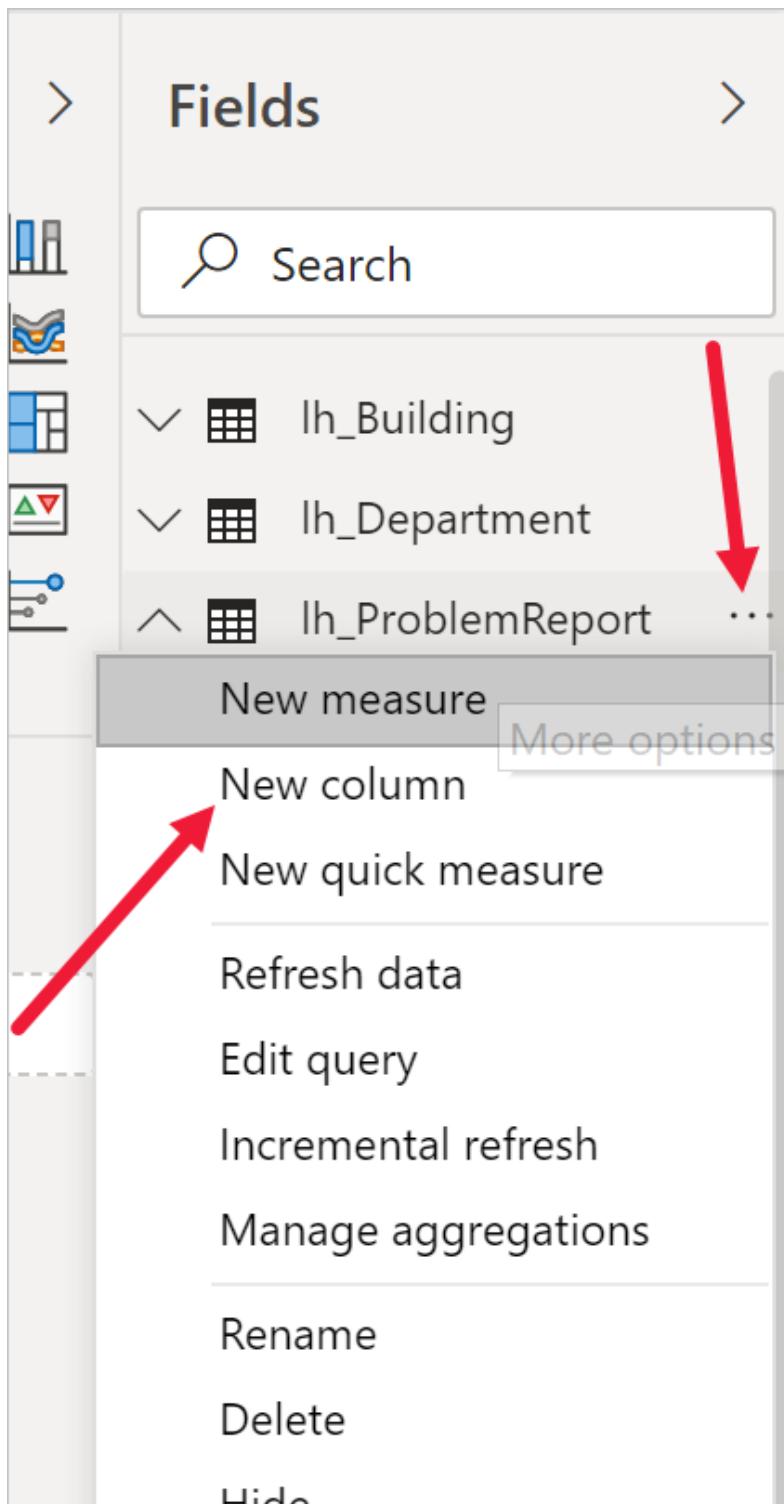
1. Repeat the previous step to drag **lh_departmentid** column and drop it into **lh_department** column in **lh_ProblemReport**. Your diagram should look like this:



1. Select **Report** icon on the left toolbar.



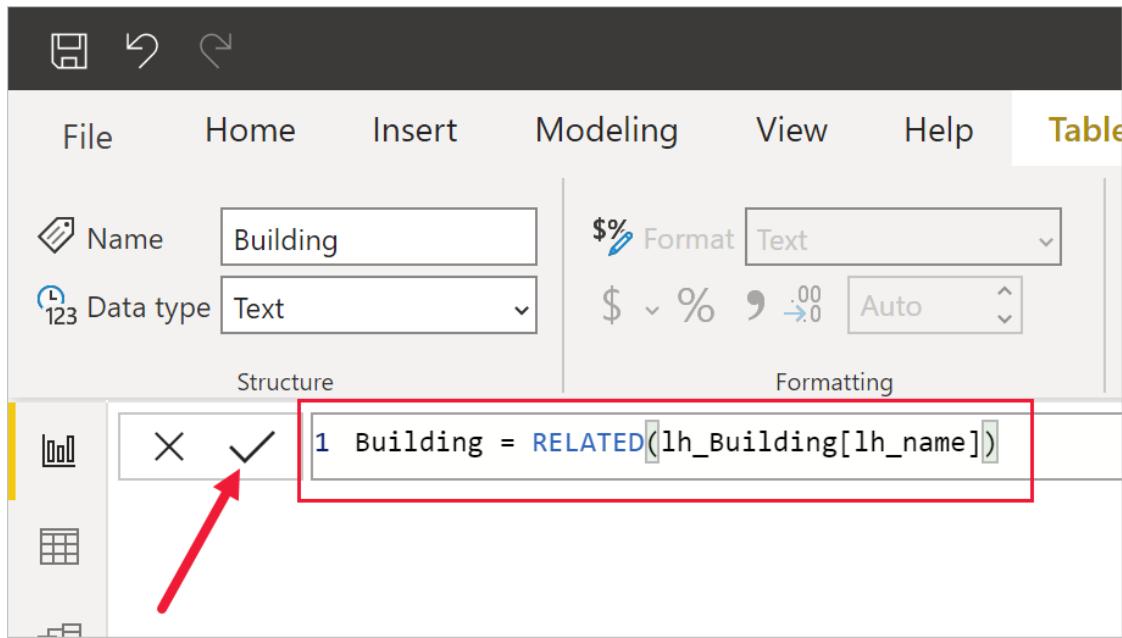
1. Expand **Ih_ProblemReports** node in the **Fields** panel.
2. Click ... and select **New Column**.



1. Complete the formula as following

```
Building = RELATED(lh_Building[lh_name])
```

and press ENTER or click checkmark button. That will add a new column with the building name into the problem report data.



1. Repeat two previous steps to add a column **Department** with the following formula

```
Department = RELATED(lh_Department[lh_name])
```

1. Click ... next to the **lh_problemreportid** column and select **Rename**. Enter **Problem Report** as the column name.
2. Click ... next to the **statuscode_display** column and select **Rename**. Enter **Status** as the column name.
3. Save work in progress by pressing **File | Save** and entering a filename of your choice.

Exercise 2: Create Power BI Report

Objective: In this exercise, you will create a Power BI report based on data from Microsoft Dataverse database.

Task 1: Create Chart and Time Visualizations

1. Press pie chart icon in the **Visualizations** panel to insert the chart.

The screenshot shows the Power BI desktop interface. On the left, the 'Visualizations' pane is open, displaying various chart and report icons. The pie chart icon is highlighted with a red arrow pointing to it. Below the pane, there are sections for 'Values' and 'Add data fields here'. On the right, the 'Fields' pane lists several data fields with checkboxes next to them.

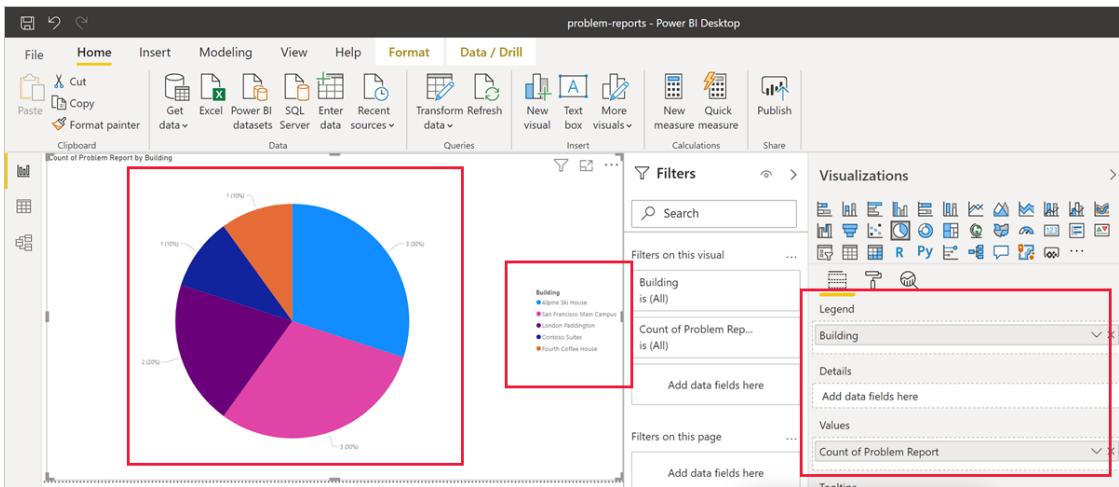
Field
<input type="checkbox"/> in_building
<input type="checkbox"/> lh_department
<input type="checkbox"/> lh_details
<input type="checkbox"/> Σ lh_estimatedcost
<input type="checkbox"/> Σ lh_estimatedcost_base
<input type="checkbox"/> lh_location

1. Drag **Building** Column and drop it into **Legend** target box.
2. Drag **Problem Report** Column and drop it into **Values** target box.

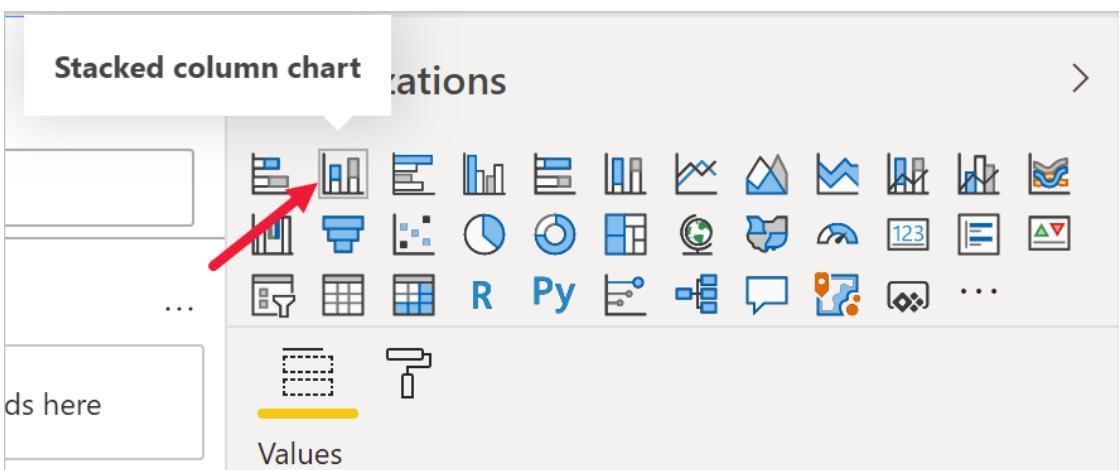
The screenshot shows the Power BI desktop interface after the steps from the previous image have been completed. The 'Visualizations' pane now shows the pie chart icon in its original position. The 'Values' section in the 'Visualizations' pane has a red dashed arrow pointing to it, indicating where the 'Problem Report' column was dropped. The 'Fields' pane remains the same as in the first screenshot.

Field
<input type="checkbox"/> in_building
<input type="checkbox"/> lh_department
<input type="checkbox"/> lh_details
<input type="checkbox"/> Σ lh_estimatedcost
<input type="checkbox"/> Σ lh_estimatedcost_base
<input type="checkbox"/> lh_location
<input type="checkbox"/> Σ lh_photo_timestamp
<input type="checkbox"/> lh_photo_url
<input type="checkbox"/> lh_photoid
<input type="checkbox"/> lh_resolution
<input type="checkbox"/> lh_resolvedon
<input type="checkbox"/> lh_title
<input type="checkbox"/> modifiedby
<input type="checkbox"/> modifiedon
<input type="checkbox"/> modifiedonbehalfby
<input type="checkbox"/> overriddencreatedon
<input type="checkbox"/> ownerid
<input type="checkbox"/> owningbusinessunit
<input type="checkbox"/> owningteam
<input type="checkbox"/> owninguser
<input type="checkbox"/> Problem Report
<input type="checkbox"/> processid

1. Resize the pie chart using corner handles so that all chart components are visible. Your report should now look like this:

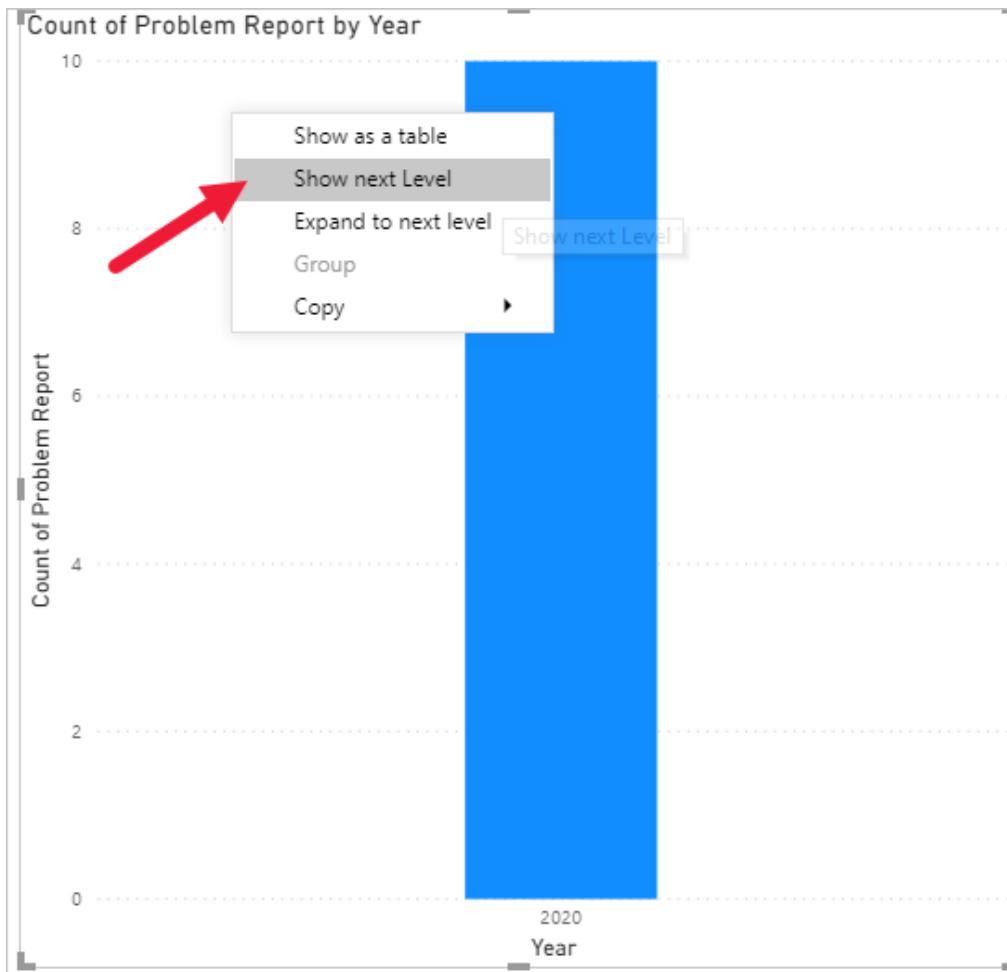


1. Click **New visual** on the Power BI ribbon then select stacked column chart in **Visualizations** pane.

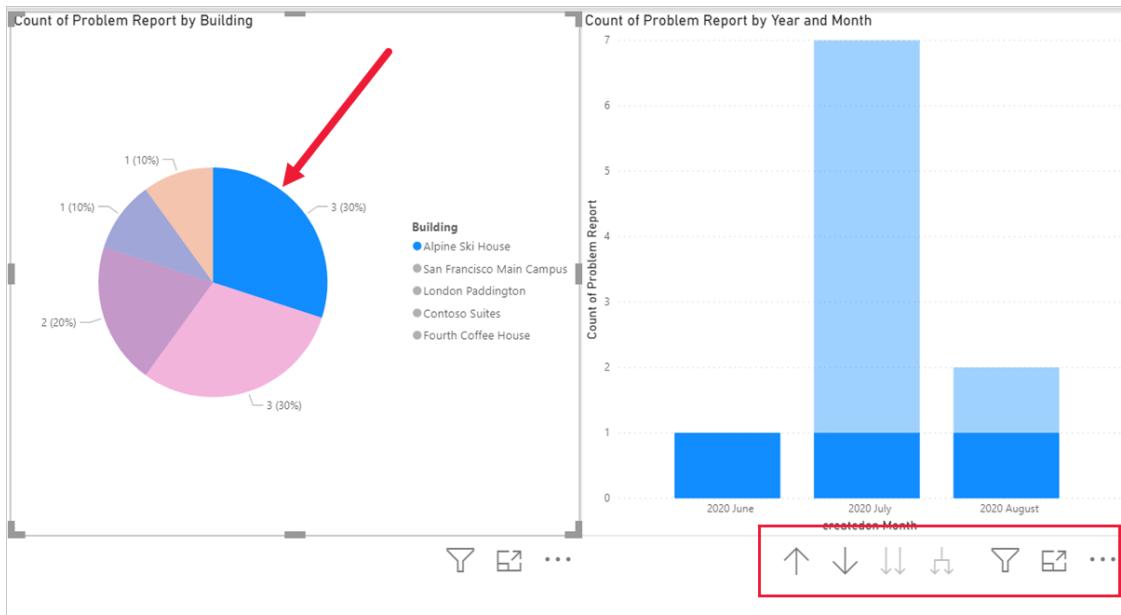


1. Drag **Problem Report** Column and drop it into **Values** target box.
2. Drag **createdon** Column and drop it into **Axis** target box.
3. Click **x** next to **Day** and **Quarter** to leave only **Year** and **Month** totals.

[!TIP] Initial stacked column chart will only display the year level. To access monthly data breakdown you need to expand report into the monthly level. The easiest way to drill down is right-button click on the report and select **Expand next level**.



1. Resize the chart as required using the corner handles.
2. Test the report interactivity:
 - o Select various building slices on the pie chart and observe changes on the time report.
 - o Select various bars on the time column chart and observe changes on the pie report.
 - o Drilldown to the month level using icons or **Data/Drill | Expand next level** ribbon command or drilldown toolbar



1. Save work in progress by pressing **File | Save**.

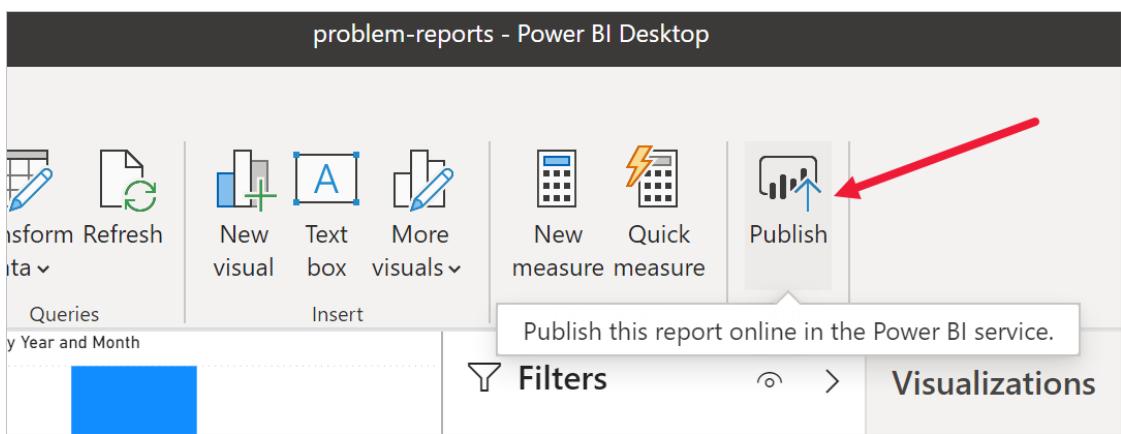
Challenge

- Replace grouping by building with grouping by **Status** column

Exercise 3: Create Power BI Dashboard

Task 1: Publish Power BI Report

1. Press **Publish** button on the ribbon.



1. Select **My workspace** as the destination, then press **Select**.
2. Wait until publishing is complete and click **Open \<name of your report>.pbix in Power BI**.

Publishing to Power BI

X

✓ Success!

[Open 'problem-reports.pbix' in Power BI](#)

[Get Quick Insights](#)



Did you know?

You can create a portrait view of your report, tailored for mobile phones.

On the **View** tab, select **Mobile Layout**. [Learn more](#)

[Got it](#)

This will open the published report in the browser.

Task 2: Create Power BI Dashboard

1. Expand **My workspace**.
2. Select the report under **Reports** heading.

 Workspaces >

 My workspace ^

Dashboards

You have no dashboards

Reports

problem-reports 

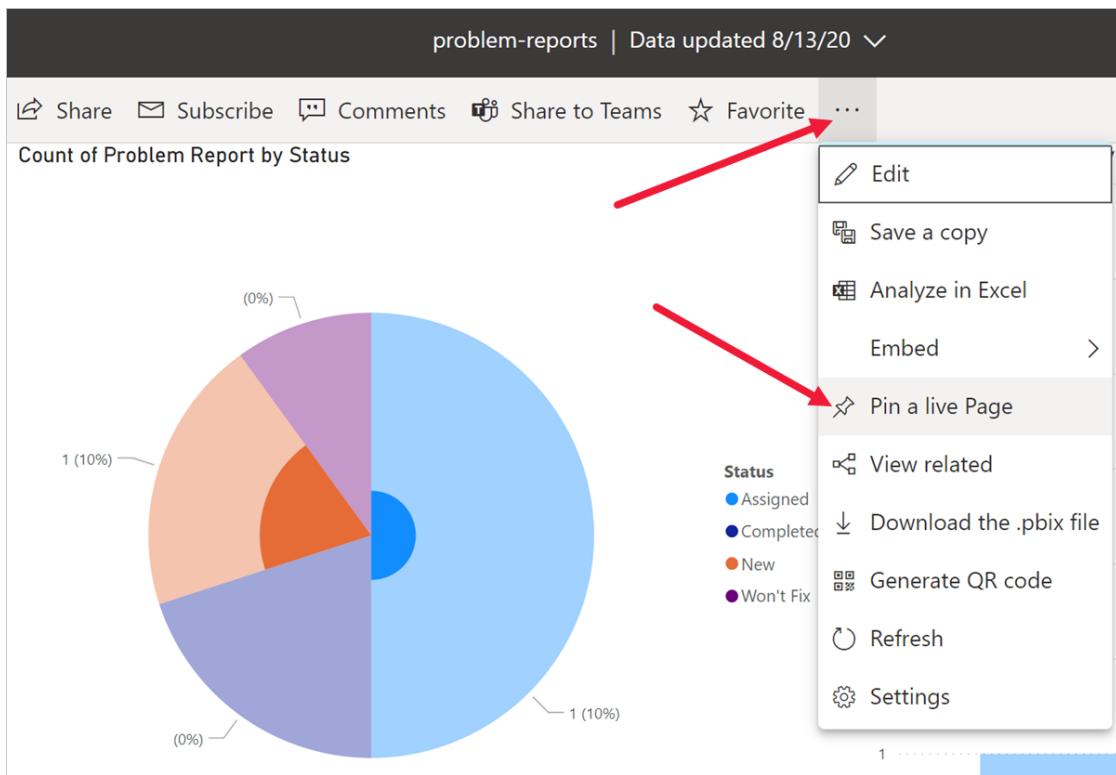
Workbooks

You have no workbooks

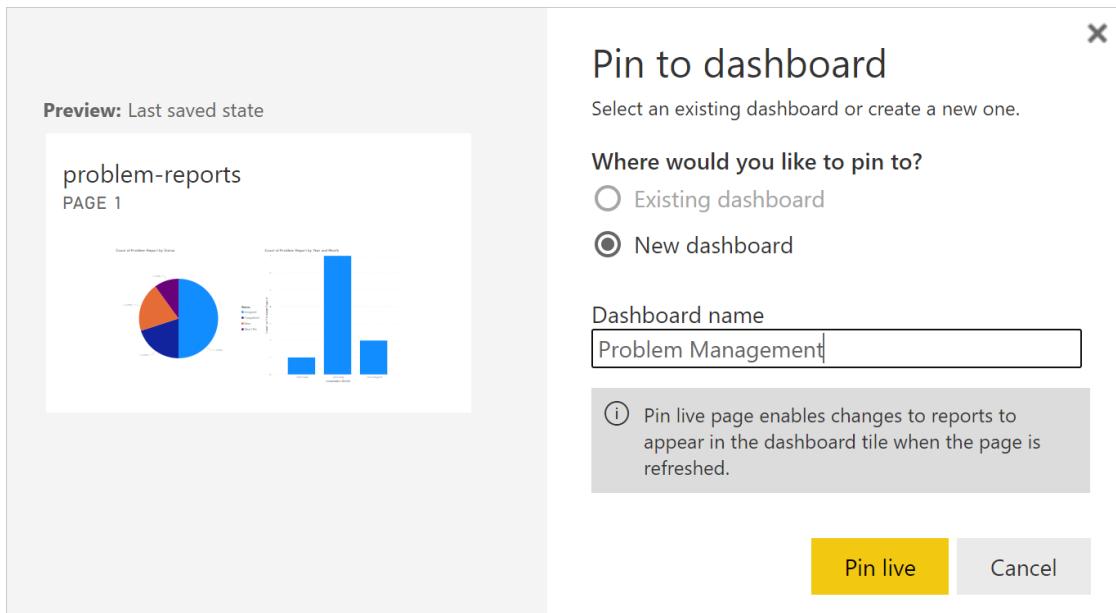
Datasets

problem-reports

1. Select **Pin a live page** on the menu. Depending on the layout you may need to press ... to show additional menu items.



1. Select **New dashboard** on **Pin to dashboard** prompt.
2. Enter **Problem Management** as a **Dashboard name**, press **Pin live**.



1. Select **My workspace** node, select **Problem Management** dashboard.
2. Test interactivity of the pie and bar charts displayed.

Task 3: Add Visualizations Using Natural Language

1. Select **Ask a question about your data** on top of the dashboard

A screenshot of the Power BI 'My workspace' interface. On the left, there's a navigation menu with options like Home, Favorites, Recent, and Apps. On the right, there's a dashboard titled 'problem-reports PAGE 1' with a chart titled 'Count of Problem Report by Status'. At the top right, there's a search bar labeled 'Ask a question about your data' with a red arrow pointing towards it.

1. Enter **funnel count of problem reports by status** in Q&A area. The funnel chart will be displayed.
2. Select **Pin visual**.

A screenshot of the Power BI interface showing a pinned funnel chart. The chart is titled 'Count of Problem Report by Status' and displays the following data:

Status	Count
Assigned	5
Completed	2
New	2
Won't Fix	1

The chart has a red border around it. Above the chart, the Q&A search bar contains the text 'funnel count of problem reports by status' with a red box around it. To the right of the chart, there's a 'Pin visual' button with a red arrow pointing towards it. A sidebar on the right shows 'FIELDS' and 'VISUALIZATIONS' sections.

1. Select **Existing dashboard**, select **Problem Management** dashboard, press **Pin**.
2. Test the behaviour by clicking on the chart to drilldown to Q&A.

Task 4: Build Mobile Phone View

1. Select the **Problem Management** dashboard from **Dashboards** area.
2. Depending on the UI version select either ... | **Mobile View** or **Web View** | **Phone View**.

3. Rearrange tiles as desired.

Edit phone view (i)

Problem Management

Count of Problem Report BY STATUS

Status	Count
Assigned	~5
Completed	~3
New	~2

problem-reports
PAGE 1

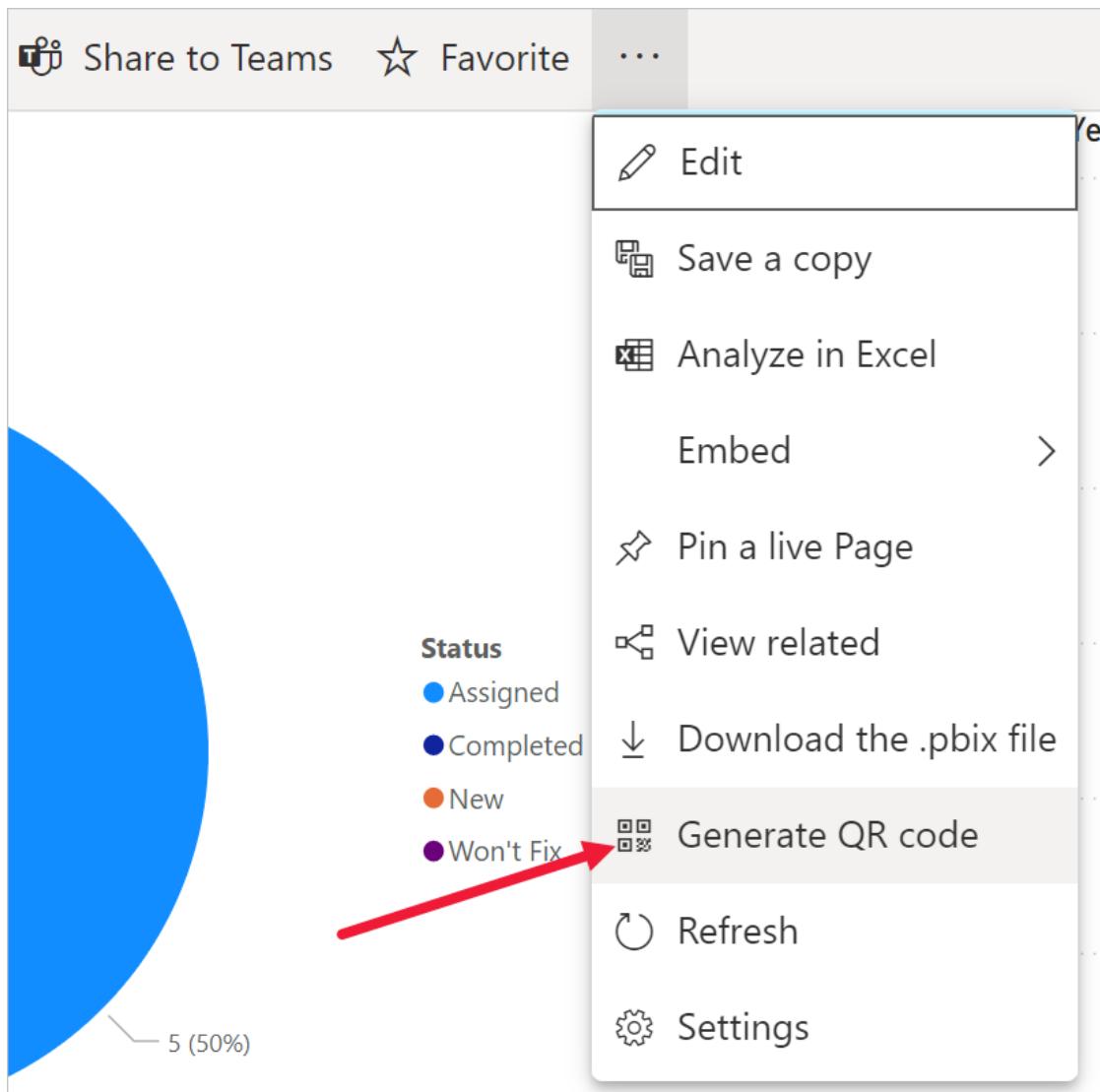
Count of Problem Report by Status

Status	Percentage
Active	~50%
Assigned	~20%
Completed	~10%
New	~20%

Count of Problem Report by Year and Month

Month	Count
2019-June	~1
2019-July	~5
2019-August	~2

1. Select your report under **My Workspace | Reports**
2. Select ... | **Generate QR Code**.



1. If you have a mobile device, scan the code using a QR scanner app available on both iOS and Android platforms.

[!NOTE] To access the dashboard and report you will have to sign in on the phone as the same user.

1. Navigate and explore reports and dashboards on a mobile device.

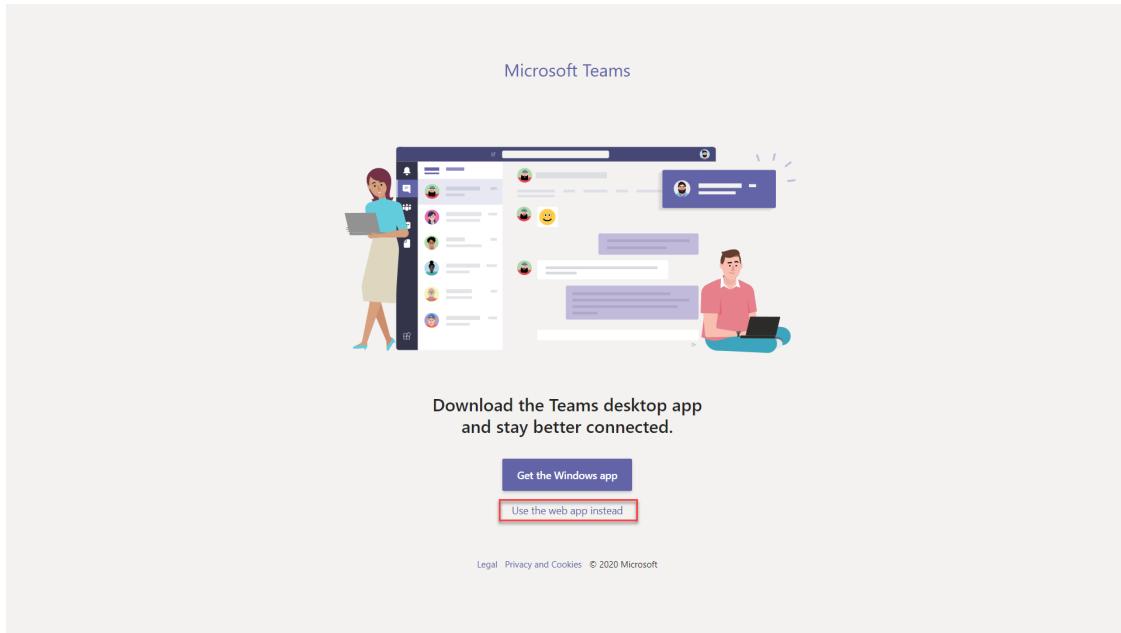
Exercise 4: Embed Power BI report in Microsoft Teams

In this exercise, you will add the Company 311 Power BI report in Microsoft Teams as a way for management and staff to be able to view the reports from directly within Teams.

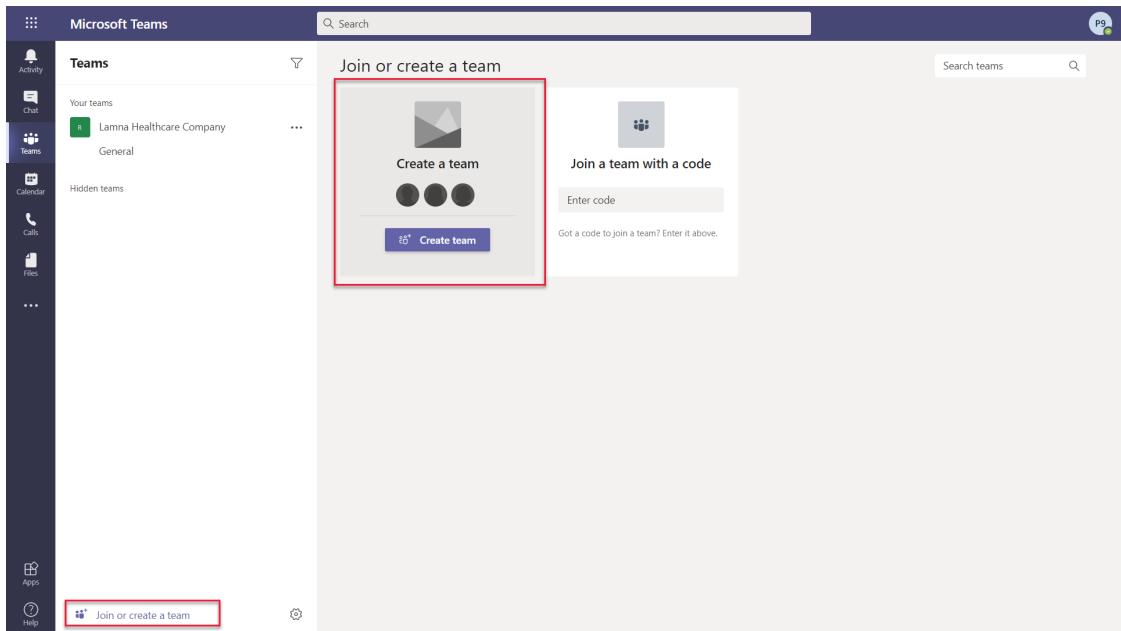
Task 1: Setup Company 311 Team

In this task you will setup a Microsoft Teams team for the Lamna Healthcare Company, if you have not done so previously.

1. Navigate to [Microsoft Teams](#) and sign in with the credentials you have been using previously.
2. Select **Use the web app instead** on the welcome screen.



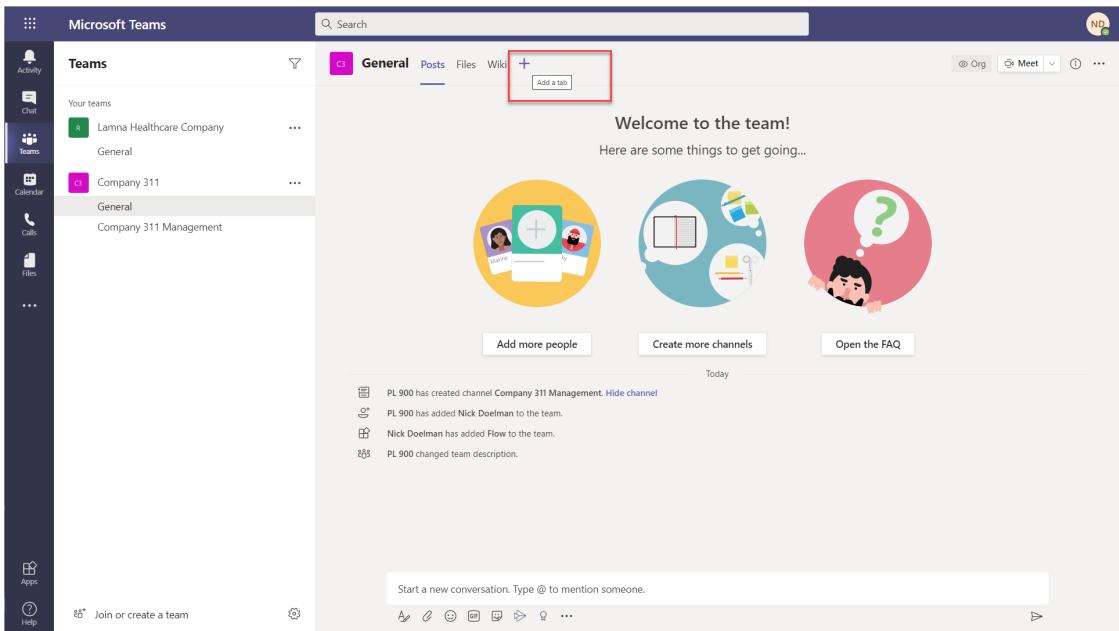
1. When the Microsoft Teams window opens, dismiss the welcome messages.
2. On the bottom left corner, choose **Join or create a team**.
3. Select **Create a team**.



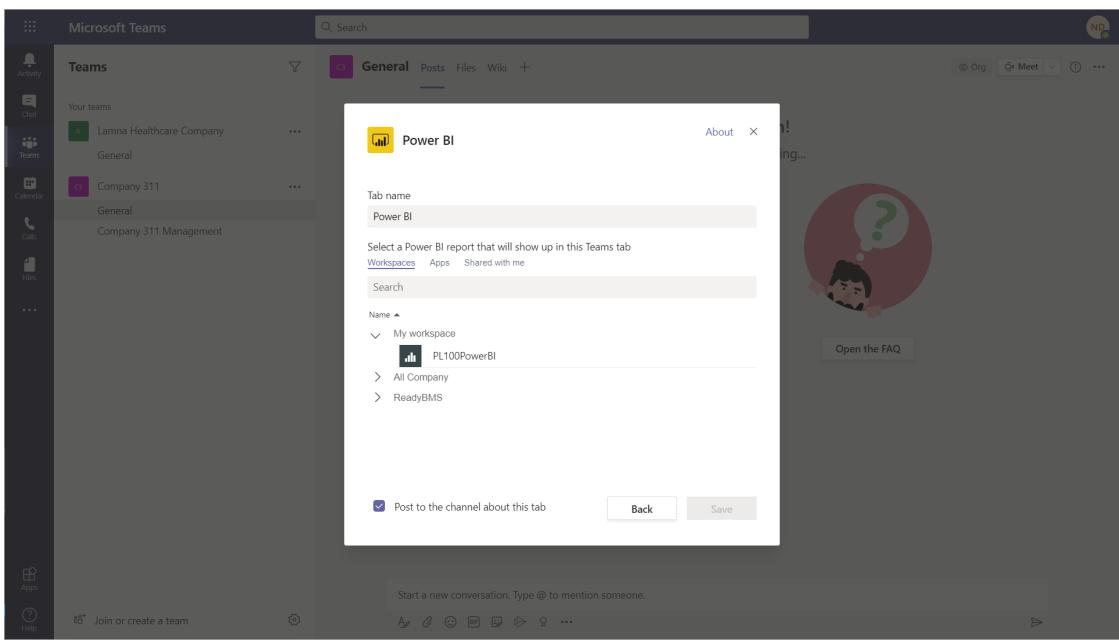
1. Press **Build a team from scratch**.
2. Select **Public**.
3. For the Team name choose **Company 311** and select **Create**.
4. Select **Skip** adding members to Company 311.

Task 2: Embed Power BI report to Teams

1. Navigate to [Microsoft Teams](#)
2. Select the **General** channel of the **Company 311** team.
3. On the top of the page, press the + symbol to add a new tab.



1. Search for **power** and select **Power BI** from the results.
2. Expand **My workspace** and select the report you created earlier in this lab.



1. Click **Save** You should now see your Power BI report in a tab in Microsoft Teams

Microsoft Teams

Search

Teams

Your teams

Lamna Healthcare Company General

Company 311 General Company 311 Management

General Posts Files Wiki PL100PowerBI +

Count of Problem Report by Building

Count of Problem Report by Month

Filters

Search

Building is (All)

Count of Problem Rep... is (All)

Count of Problem Report

July August September

Count of Problem Report

Building

- London Paddington
- Alpine Ski House
- Fabrikian Residences
- Forum Coffee House
- San Francisco Main Campus

Page 1

Challenges

- Dashboards and reports to include drilldown to individual reports with photos
- Report and analyze problem patterns and trends
- Problem resolution status visualization as a funnel

Addendum

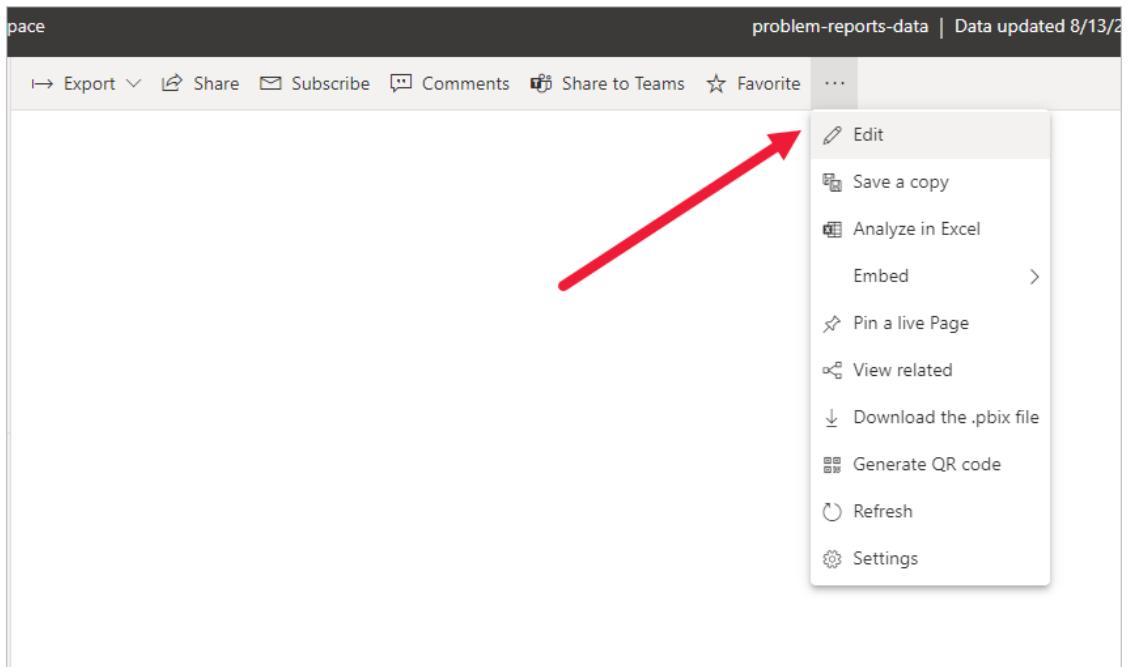
Exercise 5: Import sample data

In this exercise you will import sample data into Power BI service. That allows you to complete the lab exercises even if do not have required permissions to install desktop applications, or experience difficulties in configuring Power BI Desktop and connecting it to the data. After completion of this exercise you can resume the lab on **Exercise 2** and use Power BI service (<https://app.powerbi.com>) instead of Power BI Desktop.

1. Download [problem-reports-data.pbix](#) and save on your computer.
2. Navigate to <https://app.powerbi.com/>.
3. Click **My Workspace**.
4. Expand **+New** and select **Upload a file**.

The screenshot shows the Power BI 'My workspace' interface. On the left, there's a sidebar with navigation links like Home, Favorites, Recent, Apps, Shared with me, Learn, Workspaces, and My workspace (which is expanded). Under 'My workspace', there are sections for Dashboards (with a note: 'You have no dashboards'), Reports (with a note: 'You have no reports'), Workbooks (with a note: 'You have no workbooks'), and Datasets. On the right, there's a main area titled 'My workspace' with a user icon. A red arrow points to a 'New' button with a plus sign, which has a dropdown menu open. The dropdown menu contains several options: Report (Visualize your data), Paginated Report (Build a paginated report), Dashboard (Build a single-page data story), Dataset (Create a dataset to use in a report), Streaming dataset (Build visuals from real-time data), and Upload a file (Open a .pbix, .rdl, .xlsx, or .csv in Po...). Another red arrow points upwards from the bottom towards the 'Upload a file' option.

1. Select **Local File**.
2. Locate and select **problem-report-data.pbix** file you've downloaded earlier.
3. Once data load is complete, select **problem-reports-data** report.
4. Click ... then select **Edit**.



9. Continue on Exercise 2.

lab: title: 'Lab: Test & Deploy' module: 'Module 8: Testing & deployment'

[!NOTE] Effective November 2020: - Common Data Service has been renamed to Microsoft Dataverse. [Learn more](#) - Some terminology in Microsoft Dataverse has been updated. For example, *entity* is now *table* and *field* is now *column*. [Learn more](#)

This content will be updated soon to reflect the latest terminology.

Lab 07: Test & Deploy

In this lab you will complete solution configuration by adding security roles for the users. Then you will verify, test, and deploy your solution in the production environment.

What you will learn

- How to deploy a solution to another environment

High-level lab steps

- Exercise 1 ◉◉◉ Create security roles for users◉◉
 - Company 311 User ◉◉◉ read all on Building, user own on problem reports◉◉
 - Company 311 Admin ◉◉◉ All for Buildings, Departments, Problem reports◉◉
 - Associate Company 311 Admin role with model-driven app◉◉
- Exercise 2 ◉◉◉ Run solution checker
- Exercise 3 ◉◉◉ Use Test Studio to create Test case for submitting problem◉◉(ok to not include image)
- Exercise 4 ◉◉◉ Export and import solution◉◉

Prerequisites

- Must have completed **Lab 02: Data model and model-driven app**

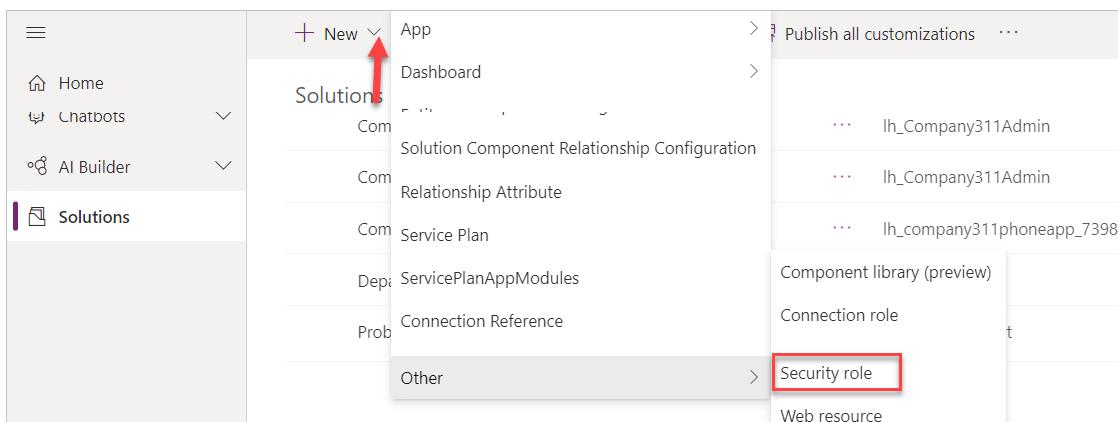
Detailed steps

Exercise 1: Create security roles

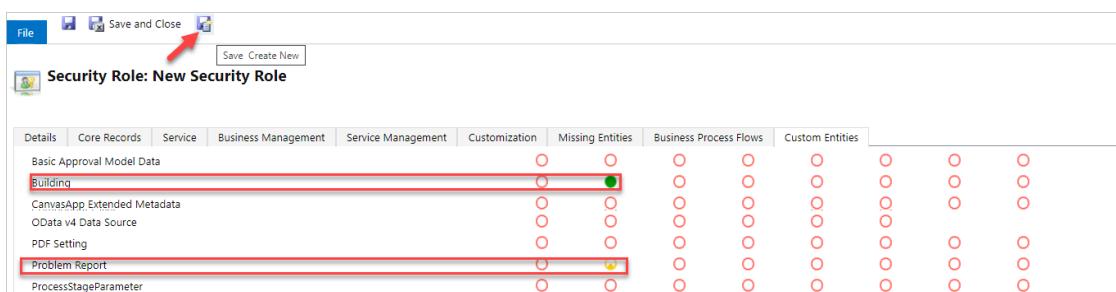
In this exercise, you create security roles for users.

Task 1: Create security roles

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select **Solutions** and click to open the **Company 311** solution.
3. Click **+ New** and select **Other | Security Role**.



1. Enter **Company 311 User** for **Role Name** and select the **Custom Tables** tab.
2. Set the read privilege of the **Building** Table to **Organization**.
3. Set the read privilege of the **Problem Report** Table to **User**.
4. The read privileges for the **Building** and **Problem Report** Tables should now look like the image below. Click **Save Create New**.



1. Enter **Company 311 Admin** for **Role Name** and select the **Custom Tables** tab
2. Set all privilege of the **Building**, **Department**, and **Problem Reports** Tables to **Organization**.

The screenshot shows the 'Security Role: New Security Role' page. At the top, there are 'File' and 'Save and Close' buttons. Below the title, there are tabs for 'Details', 'Core Records', 'Service', 'Business Management', 'Service Management', 'Customization', 'Missing Entities', 'Business Process Flows', and 'Custom Entities'. The 'Customization' and 'Custom Entities' tabs have green dots in the 'Building' row, indicating that the 'Organization' privilege is granted.

1. Select the **Customizations** tab.
2. Associate **Model-driven app** with the **Company 311 Admin** role.
3. Click **Save and Close**.
4. Click **Done** on the popup.
5. Select **Solutions**.
6. Click **Publish all customizations** and wait for the publishing to complete.

Exercise 2: Solution checker

In this exercise, you will run the solution checker on the Company 311 solution.

Task 1: Run solutions checker

1. Make sure you are still on the [Power Apps maker portal](#) site and you are in the correct environment.
2. Select **Solutions**, click **Solution checker**, and select **Run**.

Solutions

Display name

Created ↓

New solution Edit Delete Export Solution checker Show dependencies ...

Run

View results

Download results

Company 311

... Company311 7/30/2020

1. The solution checker should start checking your solution, wait for it to complete. The solution check column value will change to Results with a timestamp.
2. Select the solution again, click **Solution checker**, and select **View results**.

Solutions

Display name

New solution Edit Delete Export Solution checker Show dependencies ...

Run

View results

Download results

Company 311

... Company311 7/30/2020

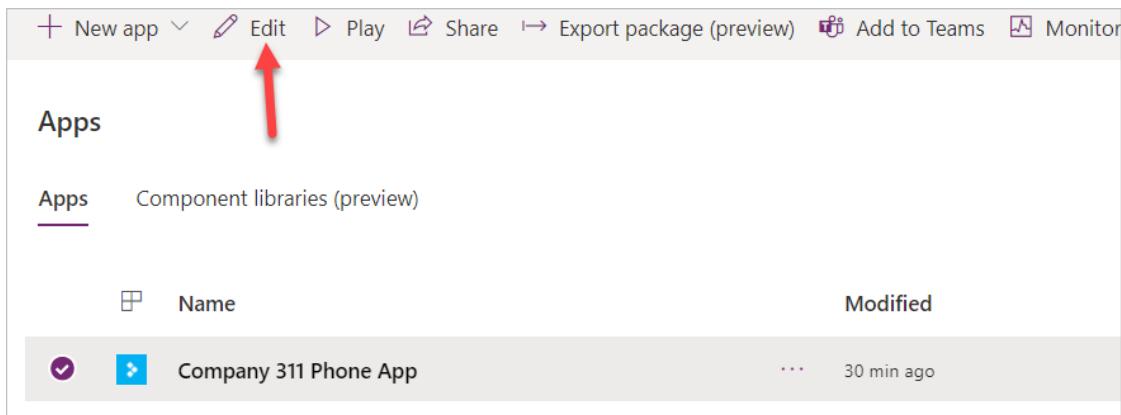
1. Review the solution check results and make sure there are no issues.

Exercise 3: Use test studio

In this exercise, you create use test studio to create test case for submitting a problem report.

Task 1: Create test case

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select **Apps**, select the **Company 311 Phone** application, and click **Edit**.



1. Click **File** and select **Settings**.
2. Select **Advanced settings**.
3. Scroll down and enable **Formula level error management**.
4. Click on the back button to go back to the app designer.

Note: Currently test studio cannot record steps inside components like the tab control we are using, you edit the App OnStart formula, so the app navigates directly to the new report screen.

5. Select the **Tree view** tab.
6. Select App and add the formula below to the existing formula.

; Navigate('New Report Screen')

A screenshot of the Power Apps formula editor. The formula being edited is 'OnStart'. The formula is:

```
Set('My Tabs', Table( {  
    Label: "My Reports",  
    Screen: 'Main Screen',  
    Icon: "",  
    SelectedIcon: ""  
},  
{  
    Label: "New Report",  
    Screen: 'New Report Screen',  
    Icon: "",  
    SelectedIcon: ""  
}); Navigate('New Report Screen')
```

 A red box highlights the line '); Navigate('New Report Screen')'.

1. Click **File** and select **Save**.
2. Click **Publish**.
3. Select **Publish this version** and wait for the publishing to complete.

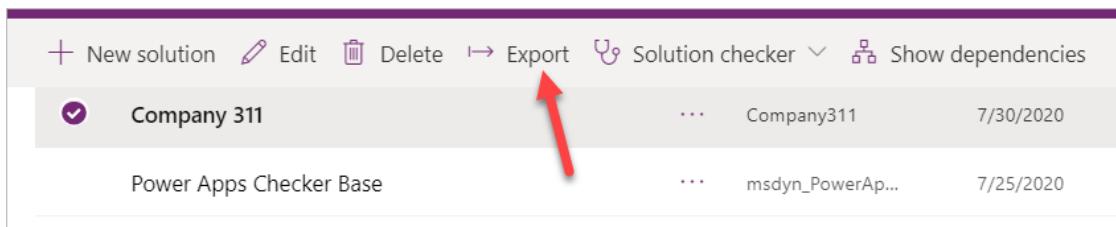
4. Click on the back button to go back to the app designer.
5. Select the **Advanced tools** tab and select **Open tests**.
6. Click on the  button of the **Case** and select **Rename**.
7. Rename the Case to **Submit problem report**.
8. Click **Record**.
9. Select the **New Report** tab.
10. Fill out the form and click **Submit**.
11. Click **Done**.
12. You should see list of the recorded steps.
13. Click **Play**.
14. Click **Publish** and wait for the publishing to complete.
15. The steps should replay correctly. Click **Done**.
16. Close the test studio browser window or tab.
17. Close the app designer browser window or tab.
18. Click Done on the popup.

Exercise 4: Import export

In this exercise, you will export the company 311 solution and import it into a new environment.

Task 1: Export solution

1. Navigate to the [Power Apps maker portal](#) and make sure you are in the correct environment.
2. Select **Solutions**, select the **Company 311** solution, and click **Export**.



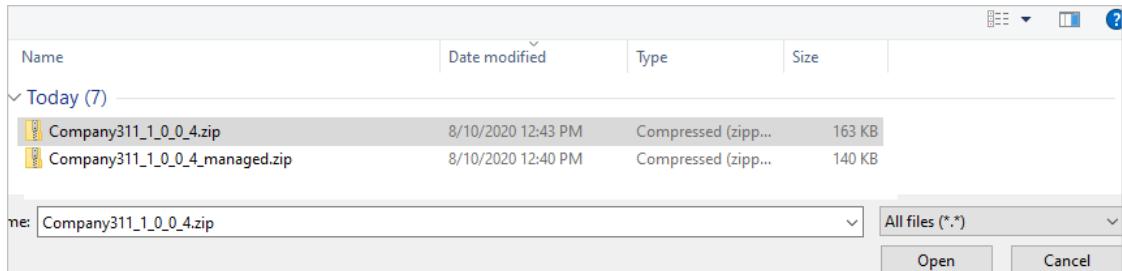
1. Click **Next**.
2. Select **Managed** and click **Export**.
3. Save the solution on your computer.
4. Click **Export** again.
5. Click **Next**.
6. Select **Unmanaged**, change the version to match the managed solution version, and click **Export**.
7. You should have the managed and unmanaged versions of the solution exported.

Task 2: Create new environment and import solution

1. Navigate to the [Power Apps admin portal](#)
2. Select environments and click **+ New**.
3. Enter **Company 311 Prod** for **Name**, select **Trial** (not subscription based), select **Yes** for **Create database for this environment**, and click **Next**.
4. Select your **Language**, select your **Currency**, and click **Save**.
5. Wait for the environment and database to be created, The State will change to **Ready** when ready. Click on the Refresh button to see the status.
6. Select **Solutions**, select the **Company 311** solution, and click **Export**.

Environments			
Environment	Type	State	Region
Company 311 Prod	Trial (30 days)	Ready	United States

1. Navigate to the [Power Apps maker portal](#) and select the environment you just created.
2. Select **Solutions** and click **Import**.
3. Click **Choose File**.
4. Select the managed solution you exported and click **Open**.



1. Click **Next**.
2. Click **Next** again.
3. Click **Import** and wait for the import to complete.
4. Click **Publish all customizations** and wait for the publishing to complete.
5. Click **Close** to close the import solution wizard.
6. Click to open the **Company 311** solution you just imported.
7. Review the components in solution.
8. Select **Apps** and make sure you have both the Canvas and Model-driven applications.
9. Click to open the **Company 311 Admin** application.
10. The application should load without issues.
11. Close the Company 311 Admin application browser window or tab.
12. Click To open the **Company 311 Phone App**.
13. The application should load without issues.
14. Close the **Company 311 Phone App** browser window or tab.

Table of Contents

It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only. ⁷

{ { activity.lab.title } }{ % if activity.lab.type % } - { { activity.lab.type } }{ % endif % }	1
{ { activity.demo.title } }	1
problem-reports-data.pbix	1