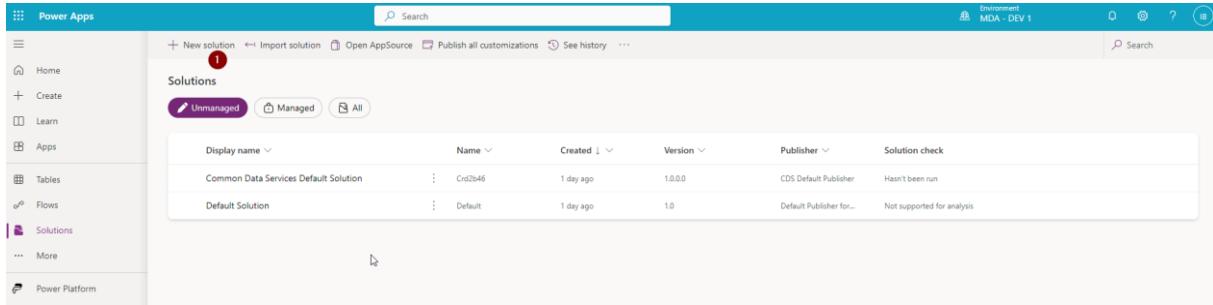


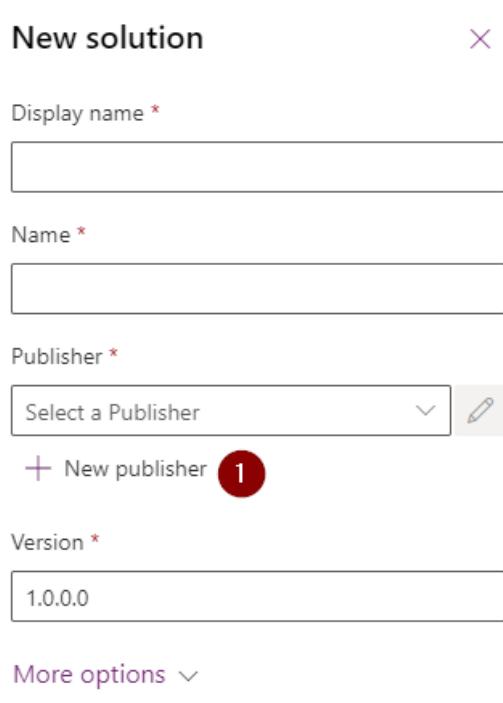
Once we're done with the Environment setup, we can enter the path of adventures, fun and glory, and start creating our Events manager solution.

## CREATING NEW SOLUTION AND NEW PUBLISHER



The screenshot shows the 'Solutions' section of the Power Apps portal. On the left, there's a sidebar with options like Home, Create, Learn, Apps, Tables, Flows, Solutions, More, and Power Platform. The main area has a search bar at the top. Below it, there are buttons for 'New solution' (highlighted with a red circle), 'Import solution', 'Open AppSource', 'Publish all customizations', 'See history', and more. A table lists existing solutions: 'Common Data Services Default Solution' (Name: Crd2s46, Created: 1 day ago, Version: 1.0.0.0, Publisher: CDS Default Publisher, Solution check: Hasn't been run) and 'Default Solution' (Name: Default, Created: 1 day ago, Version: 1.0, Publisher: Default Publisher for..., Solution check: Not supported for analysis). There are also tabs for 'Unmanaged' (selected), 'Managed', and 'All'.

1. Click on the **New solution** button.
2. A new window will open.



The 'New solution' dialog box is open. It contains fields for 'Display name \*' (empty), 'Name \*' (empty), 'Publisher \*' (dropdown menu 'Select a Publisher' with a pencil icon), 'Version \*' (text input '1.0.0.0'), and a 'More options' dropdown. A red circle highlights the '+ New publisher' button next to the 'Select a Publisher' dropdown.

If you haven't created Publisher already, click on the New publisher button.

## New publisher

X

Publishers indicate who developed associated solutions. [Learn more](#)

[Properties](#) [Contact](#)

Display name \*

Name \*

Description

Prefix \*

Choice value prefix \*

Preview of new object name

\_Object

---

[Save](#)

[Cancel](#)

Enter the Publisher data.

## New solution

X

Display name \*

Events manager

1

Name \*

Eventsmanager

2

Publisher \*

baltic (baltic)

3

▼



+ New publisher

Version \*

1.0.0.0

More options ▾

4

Create

Cancel

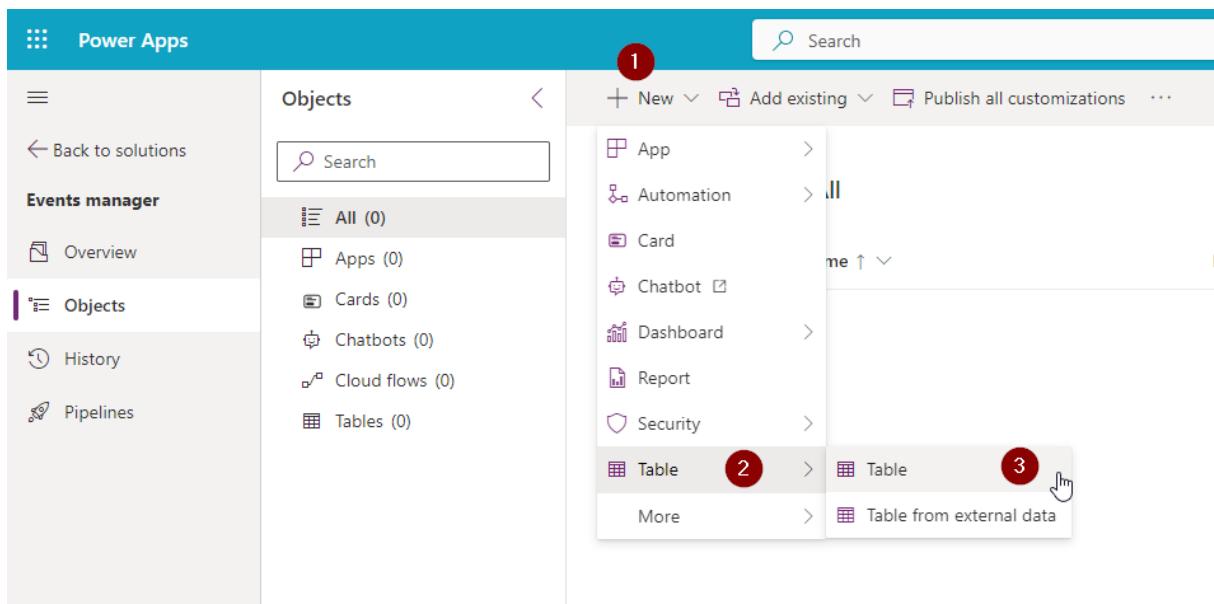
Once you're done with creating the Publisher, go back to solution creation.

1. Enter Name.
2. Enter Display name (it derived automatically from solution's name, but you can change it here).
3. Select the Publisher.
4. Click on the Create button.

And there you go – you just created your first solution!

Unfortunately (or fortunately if you like to make your hands dirty) that's not enough 😞

## CREATING NEW TABLE



Open solution you just created. Click on:

1. New.
2. Table.
3. Table.

The pop-up will arise, making your heart beating faster.

## New table

X

Use tables to hold and organize your data. Previously called entities  
[Learn more](#)

Properties Primary column

Display name \*

Event

1

Plural name \*

Events

2

Description

Enable attachments (including notes and files) <sup>1</sup>

Advanced options ▾

3

Save

Cancel

Enter:

1. Display name.
2. Plural name (unless it doesn't make any sense, better leave it as it is – derived from Display name).
3. Click on the Save button.

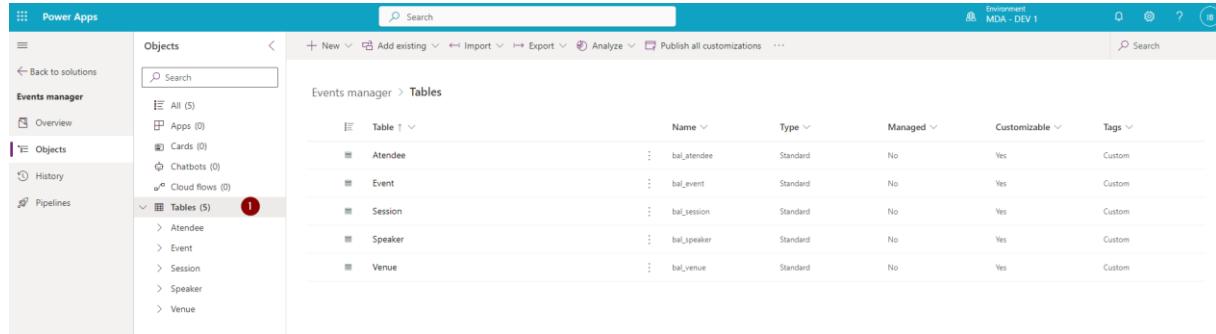
That's the minimum effort. If you consider yourself a maximizer, you can also go to Primary column tab and change primary column name (default name of primary column name is... Name! What a strange world we live in!).

## CREATING COLUMNS

If you studied history of art briefly, you probably know that in ancient Greece there were three types of columns: Doric, Ionic and Corinthian.

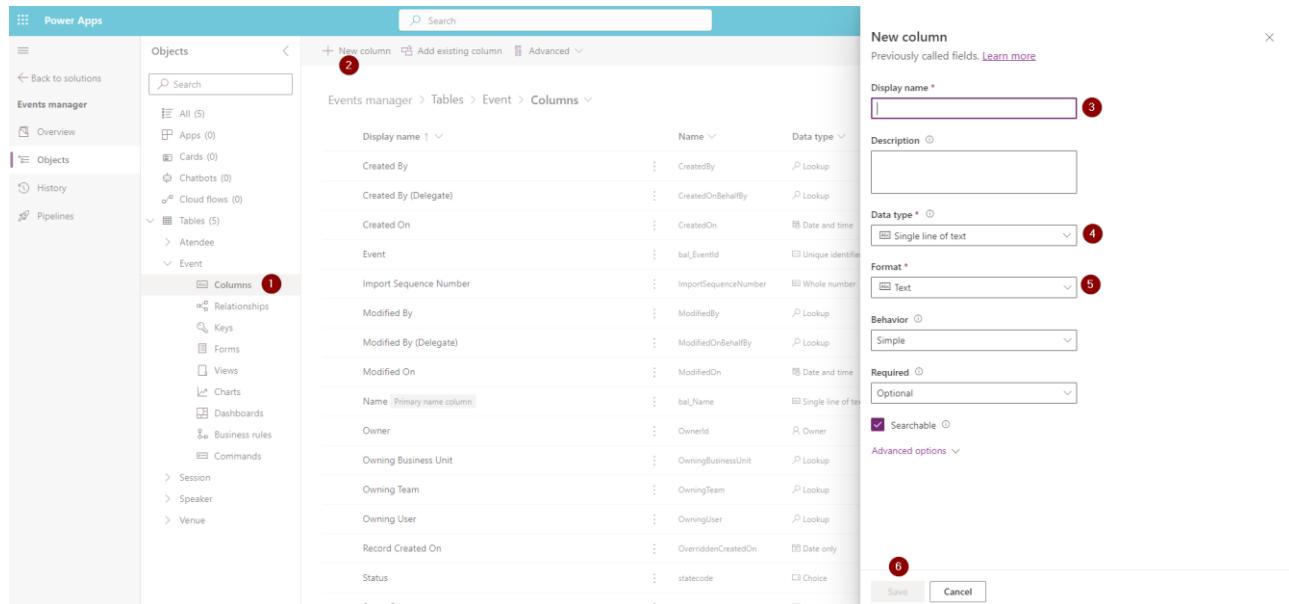
But it's the Dataverse, and not the ancient Greece we live in! So, there's much more. You can find their description here: [Types of columns](#).

Let's create some right now.



The screenshot shows the Microsoft Power Apps interface. On the left, there's a navigation sidebar with 'Events manager' selected. Under 'Objects', the 'Tables' section is expanded, showing five tables: Atendee, Event, Session, Speaker, and Venue. Each table has a small red circle with the number '1' next to it, likely indicating a step or note.

1. Go to Tables (you see that I've already created some additional ones) and select Event table.



The screenshot shows the 'Event' table's columns configuration. In the 'Columns' section of the left sidebar, 'Columns' is selected, and a red circle with '1' is next to it. A 'New column' dialog is open on the right side of the screen. The dialog fields are as follows:

- Display name:
- Description:
- Data type:
- Format:
- Behavior:
- Required:  Searchable
- Advanced options:

At the bottom of the dialog, there are 'Save' and 'Cancel' buttons, with a red circle labeled '6' next to the 'Save' button.

1. Click on Columns.
2. Select New column. A side pane will open on the right.
3. Enter column's Display name.
4. Select Data type from the dropdown (if you followed the link above you should be familiar with column types. If not, why am I even writing this manual?).
5. Select Format.
6. Click on the Save button.

You can also define some Advanced options, Behavior and whether the column is Required. You can, but you don't have to.

## CREATE LOOKUP COLUMN

Is lookup column a special column? Aren't we all special?

A Lookup column is a column which allows us to create a relationship with another table. The relationship with that table is n:1 relationship. Below is an example. In the Attendee table I created an Event lookup meaning that each Attendee is assigned to a specific Event.

How to create a lookup column?

The screenshot shows the Microsoft Power Apps interface for the 'Events manager'. On the left, there's a navigation sidebar with 'Objects', 'History', and 'Pipelines' sections. Under 'Objects', 'Tables' is expanded, and 'Attendee' is selected. Within 'Attendee', 'Columns' is selected, indicated by a red circle with the number 1. A modal window titled 'New column' is open over the table list. The 'Display name' field contains 'Event' (marked with a red circle 3). The 'Data type' dropdown is set to 'Lookup' (marked with a red circle 4). The 'Related table' dropdown is set to 'Event'. The 'Schema name' field contains 'bal\_Event' and the 'Relationship name' field contains 'bal\_attendee\_Event\_bal\_event'. At the bottom right of the modal are 'Save' and 'Cancel' buttons, with 'Save' being highlighted (marked with a red circle 5).

1. Select the table and go to Columns.
2. Click on New column.
3. Select column Display name.
4. Set Data type to Lookup and select related table.
5. Click on the Save button.

Now you can go to the Relationships section of that table.

The screenshot shows the 'Relationships' section for the 'Attendee' table. The left sidebar shows 'Relationships' selected under 'Attendee' (marked with a red circle 1). A specific relationship named 'Event' (marked with a red circle 2) is highlighted. The table lists various relationships with their details: 'Base Record ID' (Duplicate Record, One-to-many), 'Created By' (User, Many-to-one), 'Created By (Delegate)' (User, Many-to-one), 'Duplicate Record ID' (Duplicate Record, One-to-many), 'Entity instance' (Field Sharing, One-to-many), 'Event' (Event, Many-to-one), 'Modified By' (User, Many-to-one), 'Modified By (Delegate)' (User, Many-to-one), 'Name' (Bulk Delete Failure, One-to-many), 'Object Id' (User Entity Instance Data, One-to-many), 'Owner' (Owner, Many-to-one), 'Owning Business Unit' (Business Unit, Many-to-one), 'Owning Team' (Team, Many-to-one), 'Owning User' (User, Many-to-one), 'Record' (Sync Error, One-to-many), and 'Regarding' (System Job, One-to-many). The 'Managed' and 'Customizable' columns show values like 'No', 'Yes', or 'Yes'.

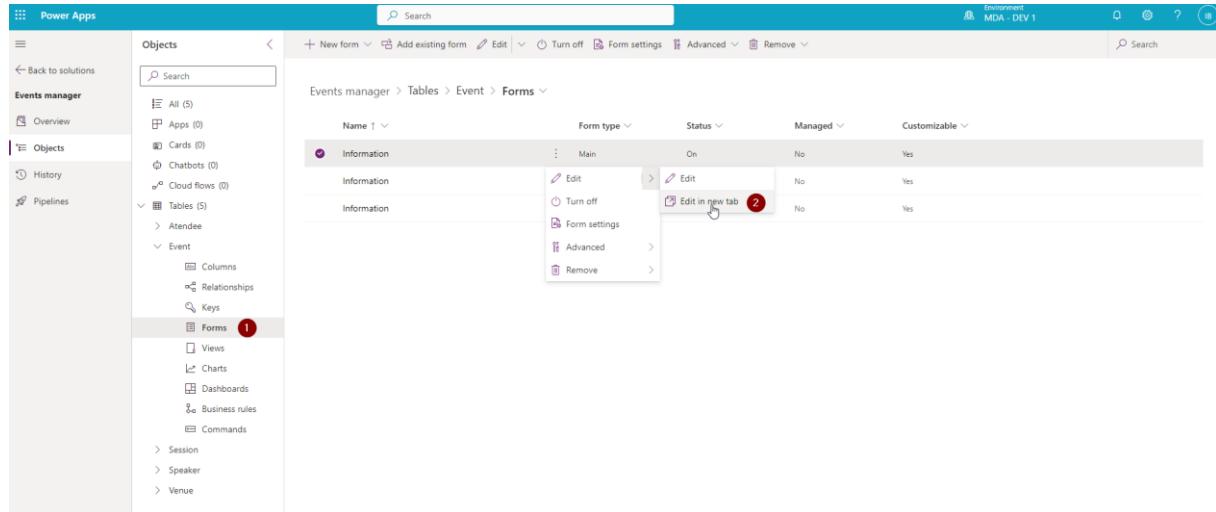
1. Click on Relationships.
2. As you can see the relationship to the Event table was automatically created.

Maybe life isn't that tough after all?

## CREATE AND EDIT FORMS

Let's create some forms. Wait, what? They're already there?!

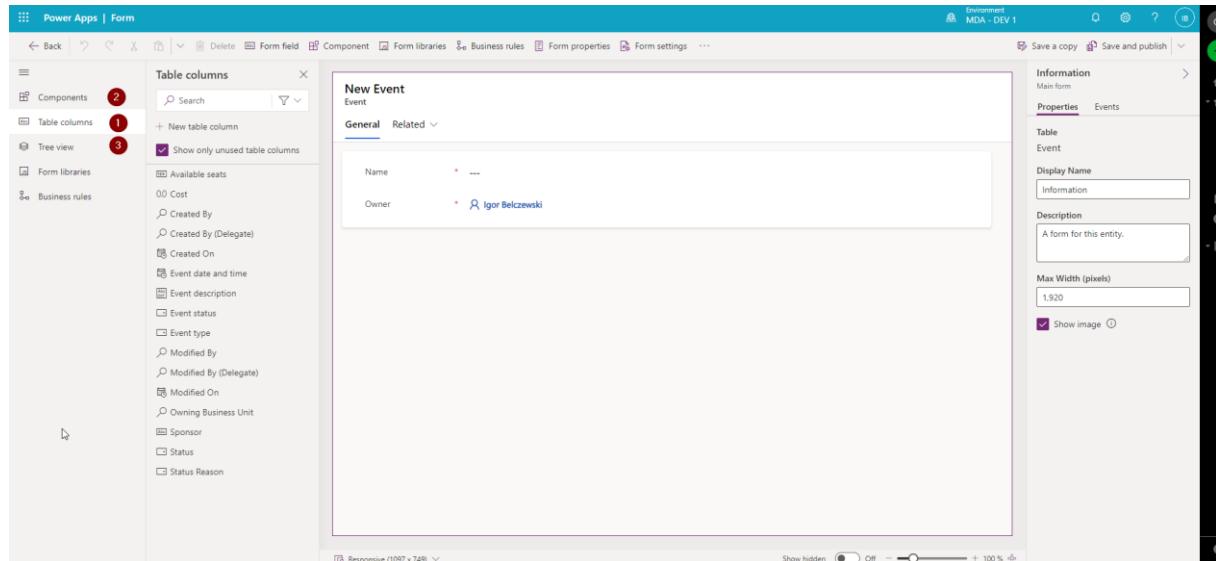
Ok, change of plans. We're gonna edit the existing form on the Event table.



The screenshot shows the Power Apps interface with the 'Events manager' selected. In the left sidebar, under 'Objects', 'Forms' is highlighted with a red circle containing the number 1. The main area displays a table of forms for the 'Event' table. One row is selected, showing 'Information' as the name, 'Main' as the form type, 'On' as the status, 'No' as managed, and 'Yes' as customizable. A context menu is open over this row, with the 'Edit in new tab' option highlighted and a red circle containing the number 2.

1. Go to Forms. As you can see the guys behind the naming weren't very creative, and we ended up with three forms called Information. Fortunately, they have different types.
2. Click on the Information form of the type of Main and click Edit in a new tab (cause as we all know: the more tabs, the better).

Your eyes will be pleased to see new tab with the form editor.



The screenshot shows the Power Apps Form editor for a new 'Event'. The left sidebar has 'Table columns' selected, with three numbered circles (1, 2, 3) pointing to 'Components', 'Table columns', and 'Tree view' respectively. The main area shows a 'New Event' form with fields for 'Name' and 'Owner'. The 'Owner' field has 'Igor Belczewski' selected. On the right, the 'Properties' pane is open, showing 'Information' as the display name and a description: 'A form for this entity.' It also includes settings for 'Max Width (pixels)' (set to 1920) and a 'Show image' checkbox. The bottom of the screen shows a responsive view at 1097x749 and a zoom control at 100%.

Here you can:

1. Add Table columns.
2. Add Components.
3. See the Tree view.

As you can see, the possibilities are almost endless. Play around and make your form great again. Just don't call me crying if you break something.

## CREATE MANY TO MANY RELATIONSHIP

If you find it hard to sleep at night don't dig into many too many relationships. Cause then you won't sleep for sure. But if you sleep like a baby, just take a look at what it is all about: [Create many-to-many relationships](#). And then create one.

The screenshot shows the Power Apps interface with the 'Events manager' selected. In the left sidebar, under 'Session', the 'Relationships' item is highlighted with a red circle containing the number 1. A modal window titled 'Many-to-many' (with a red circle containing the number 2) is open, showing the configuration for creating a relationship between the 'Session' table (Current (Many)) and the 'Event' table (Related (Many)). The 'Searchable' checkbox is checked. The 'Relationship name' field contains 'bal\_Session\_b1\_Event\_b1\_Event'. The 'Relationship table name' field contains 'bal\_Session\_b1\_Event'. The 'Done' button is visible at the bottom right of the modal.

1. Go to Relationships.
2. Click New relationship and select Many-to-many.
3. Define related table and click on Done button.
4. Sleep.

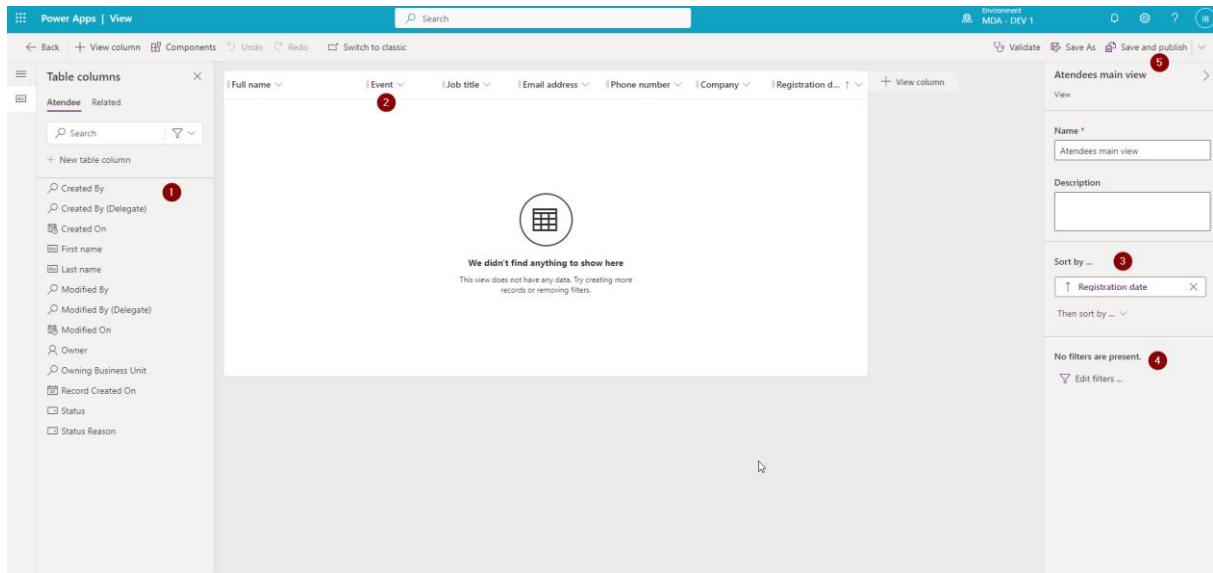
## CREATE AND EDIT VIEWS

One's point of view depends on the point where one sits. And there you are, sitting in front of your computer. Let's create some views to show them all!

The screenshot shows the Power Apps interface with the 'Events manager' selected. In the left sidebar, under 'Atendee', the 'Views' item is highlighted with a red circle containing the number 1. A modal window titled '+ New view' (with a red circle containing the number 2) is open, showing the creation of a new view for the 'Atendee' table. The table is set to 'Advanced Find View' (View type), 'On' (Status), and 'No' (Managed). The 'Customizable' column shows 'Yes' for the first two rows and 'No' for the last two. The 'Done' button is visible at the bottom right of the modal.

1. Go to Views on the Attendee table.
2. Click on New view.

View editor will show.

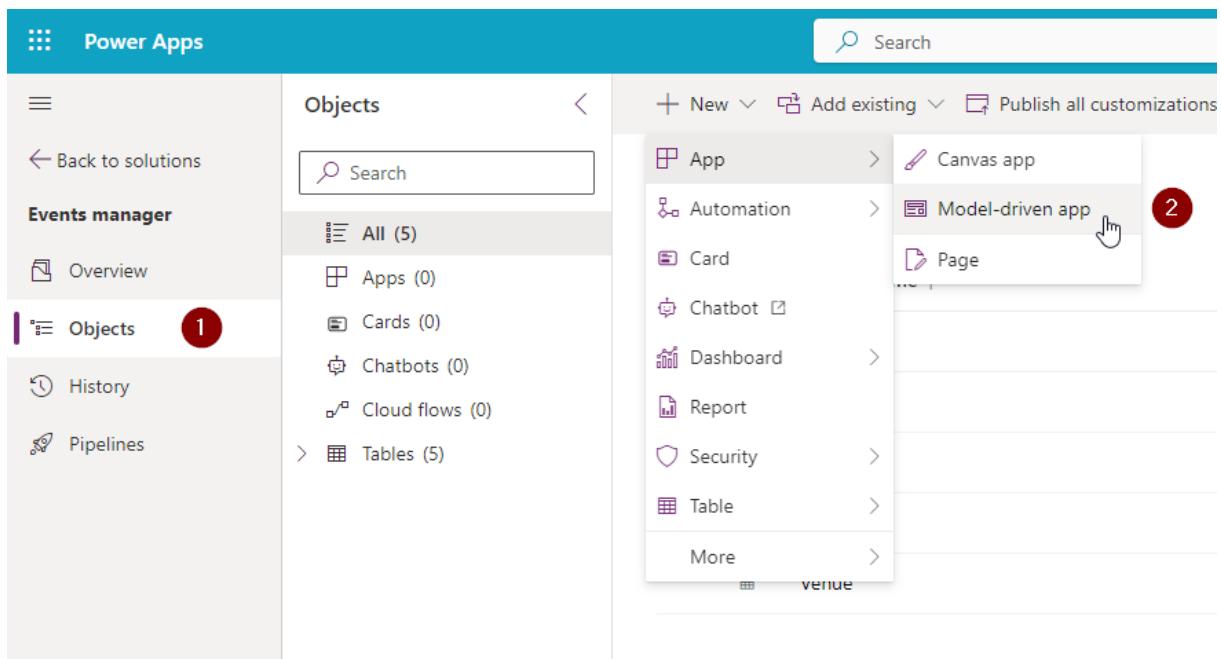


1. Add the columns you need.
2. Order them.
3. Sort them if you want.
4. Add a filter if you want.
5. Save and publish in order to save and publish.
6. Sleep again.

See, it's not that hard if you believe in yourself.

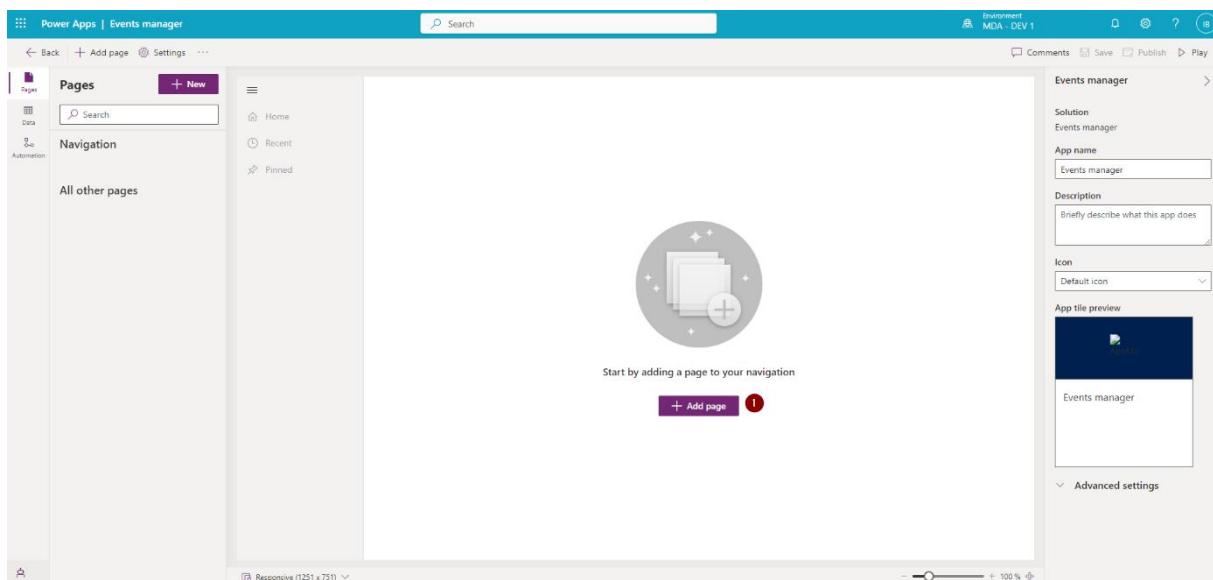
## CREATE AND EDIT NEW MODEL-DRIVEN APP

In order to create new app, go to:



1. Objects.
2. App/Model-driven app.

Our app is a bit like a book, it has pages (that's the only thing that these two have in common). You have to create the first page. In order to do so:

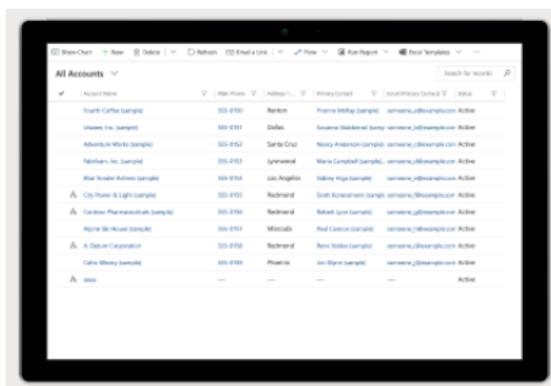


1. Click on the Add page button.

Then:

## New page

X



This option gives you two pages for each table. You get a view, which displays the rows and columns of the table. You also get a form, which lets people interact with the table by editing existing rows, or by creating new ones.

Choose content for the page

**1** Dataverse table

Dashboard

URL

Web resource

Custom page

2

Next

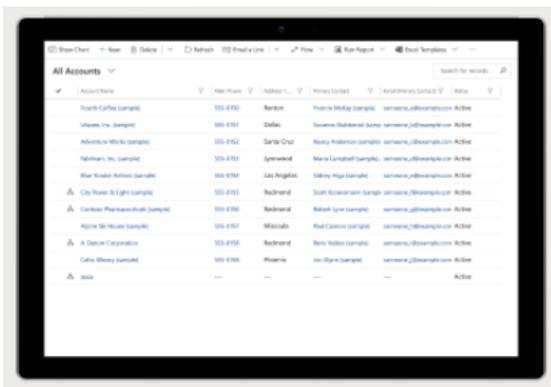
Cancel



1. Select Dataverse table.
2. Click on Next.

## ← New page

X



This option gives you two pages for each table. You get a view, which displays the rows and columns of the table. You also get a form, which lets people interact with the table by editing existing rows, or by creating new ones.

Choose data table for these pages

**1** Select existing table

Create new table

Select one or more tables

speak **2** X

Speaker **3**

Show in navigation

4

Add

Cancel

1. Select existing table.
2. Search for the right table.
3. Choose the right table or tables. Speaker in our case.
4. Click on Add.

The page will be added to the app.

The screenshot shows the Power Apps Studio interface. On the left, the 'Pages' blade is open, displaying a navigation menu with sections like 'Events view', 'Attendees view', 'Sessions view', etc., under 'New Group'. A red circle with a question mark is visible next to the 'New Group' section. In the center, a 'New Group' page titled 'Active Events' is being edited. The page has a table with columns 'Name' and 'Created On'. The right side of the screen shows the 'Display options' pane, where the title is set to 'New Group' and the ID is 'group\_2b27dfb6'. The status bar at the bottom indicates a responsive width of 1251 x 751 pixels.

1. You can see it in the Navigation on the right.
2. If you're satisfied with your Pages, click on Save button.

Once the app is saved:

This screenshot is identical to the previous one, showing the Power Apps Studio interface with the 'Pages' blade open and the 'New Group' page being edited. The navigation menu on the left remains the same. The central area shows the 'Active Events' page with no data available. The right-hand 'Display options' pane is also identical, showing the title 'New Group' and ID 'group\_2b27dfb6'. The status bar at the bottom still shows a responsive width of 1251 x 751 pixels.

1. Publish the app.

So far so good. What we've created till now:

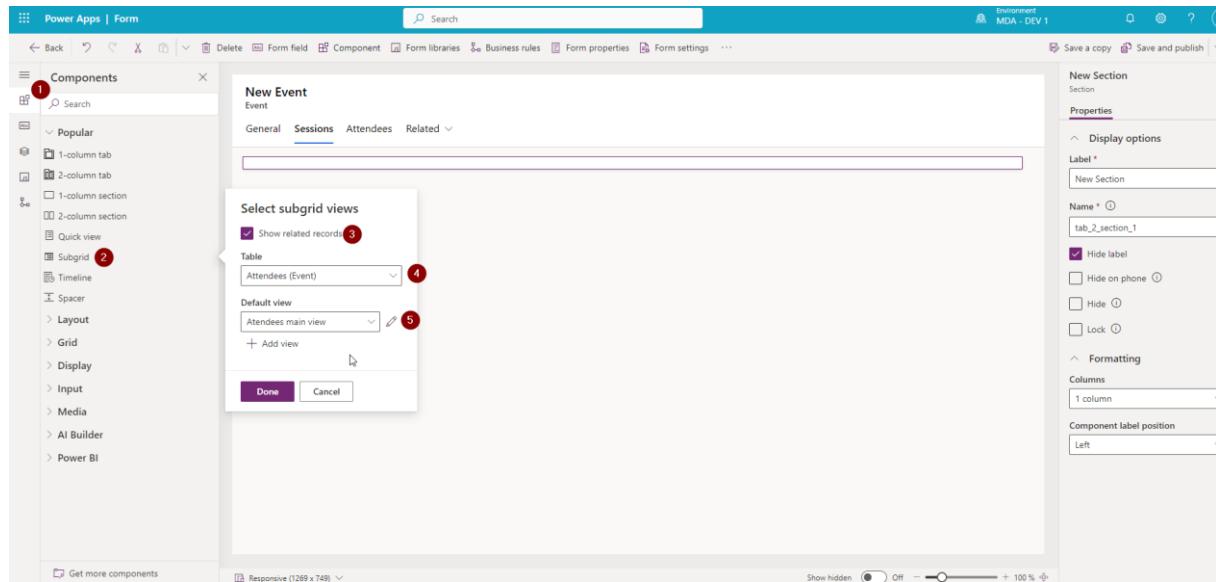
Display name	Name	Type	Managed	Last Modified	Owner	Status
Attendee	bal_attendee	Table	No	5 minutes ago		
Event	bal_event	Table	No	2 hours ago		
Events manager	bal_Eventsmanager	Site Map	No	1 minute ago		
Events manager	bal_Eventsmanager	Model-Driven App	No	1 minute ago		On
Session	bal_session	Table	No	2 hours ago		
Speaker	bal_speaker	Table	No	2 hours ago		
Venue	bal_venue	Table	No	2 hours ago		

1. Tables. With all the columns, and all the forms, and all the views... We're amazing.
2. App. Just. One. App.
3. Site map. Wait a minute! I didn't create it! You are right. It was created for you once you created the app. Read more about it: [Create a model-driven app site map using the site map designer.](#)

We have something to build on. Let's go back to our form and customize it a bit.

## FORM CUSTOMIZATIONS

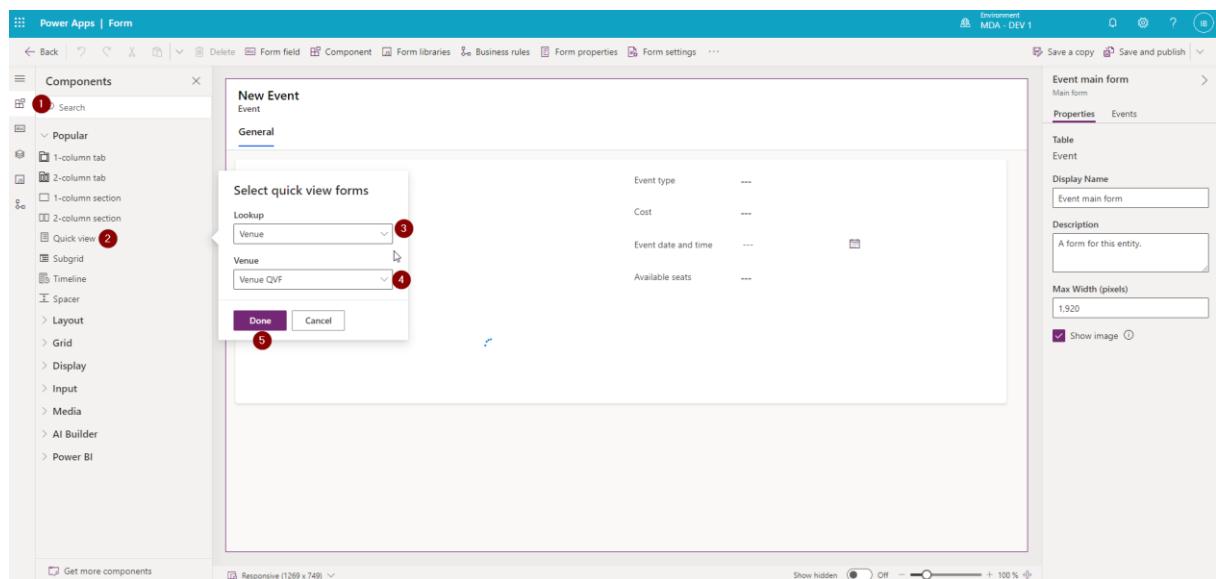
Let's open the event form and add a subgrid. A subgrid is what we need if we want to show e.g. Attendees who signed up for the event.



1. Go to Components.
2. Select Subgrid.
3. Tick Show related records.
4. Select Attendees table.
5. Select the view and click on Done.

## ADD QUICK VIEW FORM TO MAIN FORM

Quick view form is a form which allows you to show some additional information from the lookup column which is already on the form. You can edit it like we did with the Main form above, defining which columns should be visible. We're gonna add some Venue details.



1. Go to Components

2. Select Quick view.
3. Select Lookup column.
4. Select QVF.
5. Click on Done.

You can see the added QVF:

New Event

Event

General

Event name	*	---	Event type	---
Event status	Draft		Cost	---
Venue	---		Event date and time	---
<input type="button" value="Name"/> <span style="color: red;">!</span>			Available seats	---
<input type="button" value="Address"/>				
Sponsor	---			
Event description				

1. The QVF will be visible if the Lookup column is set.

## REMOVE THE VIEW FROM THE APP

If you don't want particular view to be visible in the app:

The screenshot shows the 'Events main view' table with columns: Name, Event type, Event status, Event date and time, Sponsor, Available seats, and Cost. A blue circle with a question mark is over the 'Edit filters' button. In the 'Views' pane on the right, under 'In this app', 'Events main view' is selected. A red circle with a number 1 is over the 'Events view' item in the navigation pane. A red circle with a number 2 is over the 'Remove' button for 'Event A'. A red circle with a number 3 is over the 'Not in this app' section.

1. Select the table views.
2. Click on the view you don't want to see and select Remove.
3. See which views were removed from the app.

Now you can open the app and:

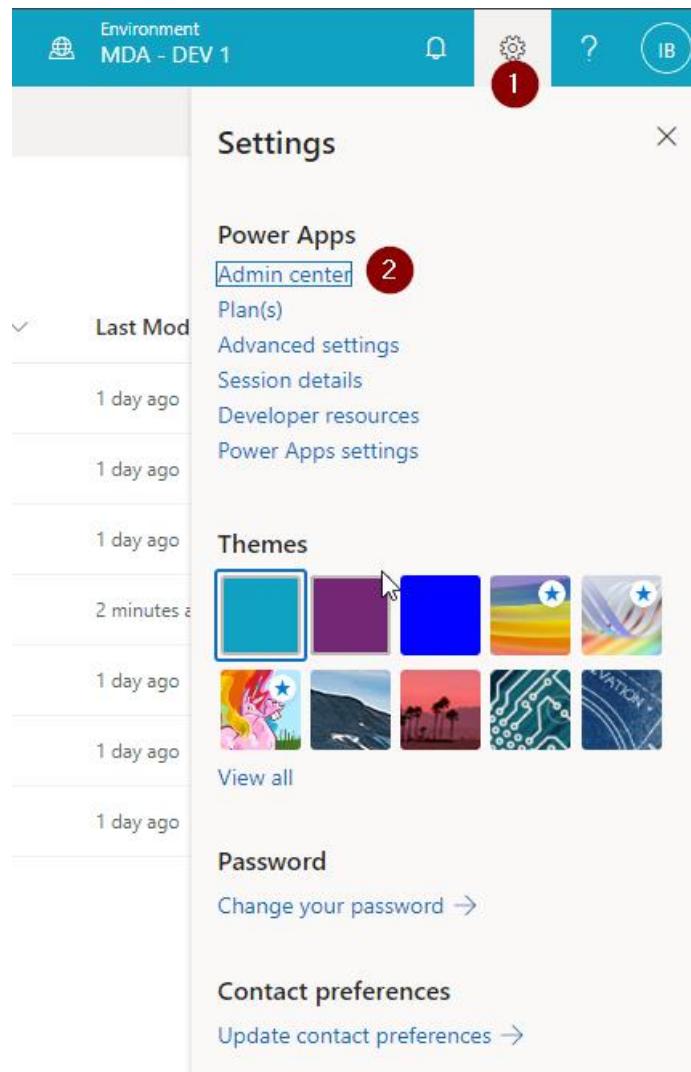
The screenshot shows the 'Events main view' table with the same columns as before. A red circle with a number 1 is over the 'Events' item in the navigation pane. A red circle with a number 2 is over the 'Default' button in the 'Events main view' dropdown menu. The table displays a message: 'We didn't find anything to show here'.

1. Go to the Events page.
2. See the views that weren't removed from the app.

## OH ADMIN, MY ADMIN

Admin Center is a center for Admins. Power Platform Admins, to be precise. Read more about it here: [Manage Microsoft Dataverse settings](#).

To acces it you need to open the environment and:



1. Click on Setting icon.
2. Click on Admin center.

## SECURITY ROLE

The Admin center will open. At first, we're gonna take a look at Security roles.

An environment is a space to store, manage, and share your organization's business data, apps, chatbots, and flows. When an environment is managed, it allows you greater visibility and control.

Environment	Type	State	Dataverse	Managed	Region	Created on	Created by	Last activity
MDA - DEV 2	...	Developer	Ready	Yes	No	United States	01/13/2024 5:46 PM	Piotr Rusak
MDA - DEV 1	...	Developer	Ready	Yes	No	United States	01/13/2024 5:45 PM	Piotr Rusak

1. Click on Environments
2. Select the right Environment (not the one on the right – the right for you).

You will see an overview of Environment options.

MDA - DEV 1

Environment URL	State	Region
bpp-mda-dev1.crm.dynamics.com	Ready	United States

Details

Organization ID	Environment ID
3203fb33-fcb1-ee11-a564-000d3a106325	7181830c-3ff8-e772-a9c0-adc1bbe33012

Auditing

Auditing enabled	Retain audit logs for	Free up capacity
No	Forever	Delete audit logs

Version

Dataaverse version
9.2.23113.00214

Updates

2023 release wave 2
On See what's new in the release Deployment Schedule

Recent operations

Full history

Access

- Security roles See all **1**
- Teams See all
- Users See all
- S2S apps See all
- Business Units (Preview) See all

Resources

- Power Pages sites
- Power Apps
- Flows

1. Click on See all below Security roles.

You will see all the security roles available in the environment.

Power Platform admin center

Environments > MDA - DEV 1 > Settings > Security roles

Role ↑	Business unit	Managed	Modified on
AIB Roles	... bpp-mda-dev1	Yes	01/13/2024 9:38 AM
AIB SML Roles	... bpp-mda-dev1	Yes	01/13/2024 9:38 AM
App Deployment Orchestration Role	... bpp-mda-dev1	Yes	01/13/2024 7:15 AM
App Opener	... bpp-mda-dev1	Yes	01/13/2024 7:03 AM
Async ingestion	... bpp-mda-dev1	Yes	01/13/2024 9:51 AM
Basic User <span style="color: red;">1</span>	... bpp-mda-dev1	Yes	01/13/2024 9:38 AM
BizQAApp	... bpp-mda-dev1	Yes	01/13/2024 9:53 AM
Bot Author	... bpp-mda-dev1	Yes	01/13/2024 7:35 AM
Bot Contributor	... bpp-mda-dev1	Yes	01/13/2024 7:35 AM
Bot Transcript Viewer	... bpp-mda-dev1	Yes	01/13/2024 7:35 AM

1. Click on three dots next to Basic User.
2. Click on the Copy button.

It's good practice to create a separate Basic User role for the specific solution, which you can further customize.

Power Platform admin center

Environments > MDA - DEV 1 > Settings > Security roles

Copy role

Create another security role by copying this one. Only its privileges will be copied over

Name \* Events manager Basic User 1

2 Copy Cancel

1. Write the name (I usually use <Solution name> Basic User convention).
2. Click on Copy.

Now having the copy, we want to add it to our solution. Go to solution:

Power Apps

Objects 1

+ New 2 Add existing 3 Publish all customizations ...

All (7)

Events manager 1

Overview

Objects

History

Pipelines

Events manager 2

Objects 3

Events manager

Analytics

App

Automation

Card

Chatbot

Dashboard

Report

Security 2

Column security profile

Site

Table 3

More

Name	Type	Managed	Last Modif...	Owner	Status
bal_attendee	Table	No	1 day ago	-	
bal_event	Table	No	1 day ago	-	
bal_Eventsmanager	Site Map	No	1 day ago	-	
bal_Eventsmanager	Model-Driven App	No	2 minutes ago	-	On
bal_session	Table	No	1 day ago	-	
bal Speaker	Table	No	1 day ago	-	
bal_venue	Table	No	1 day ago	-	

1. Select Add Existing.
2. Security.
3. Security role.

The screenshot shows the 'Add existing roles' dialog box. At the top, there's a search bar with 'ev' typed into it. Below the search bar, a message says 'Select roles from other solutions or roles that aren't in solutions yet. Adding roles that aren't already in solutions will also add them to Dataverse.' A table lists a single role: 'Events manager Basic User'. The 'Add' button at the bottom is highlighted with a red circle containing the number '3'.

1. Search for the Events manager Basic User role.
2. Select it.
3. Click Add.

The security role was added to the solution. Now we can go to:

The screenshot shows the 'Events manager > Security roles' list. A context menu is open over the single row ('Events manager Basic User'). The 'Edit' option in the menu is highlighted with a red circle containing the number '2'.

1. Security roles.
2. Click on three dots next to the role and select Edit.

Now you can define to which tables it has permissions.

Sales and Close Actions

Security Role: Events manager Basic User

Working on solution: Events manager

Details Core Records Sales Service Business Management Service Management Customization Missing Entities Business Process Flows Custom Entities

Delegated Authorization  
Delete Record  
Delete Row Module  
Dynamic Search  
DynamicSearchAttribute  
DynamicSearchIdentity  
DynamicSearch  
DynamicSearchAttribute  
DynamicSearchIdentity  
EM Ammee  
EM Event  
EM Session  
EM Speaker  
EM Vehicle  
EventAttachmentList  
Empty Image Configuration  
Empty Link Chat Configuration  
EmptyRelationshipHistory  
Environment Variable Definition  
Expiremed Email  
Exported Document  
External clients  
Favorita Knowledge Article  
NaturalLanguageSetting  
New Contact  
New Event  
New Machine  
New Machine Group  
New Machine Image  
New Machine Network  
New Row  
Row Section  
Form Step

Key ②

None Selected User Business UNIT Parent Child Business Units Organization

1. Find the tables for which you want to set a permission.
2. Define proper permission level.
3. Save security role.

## ADD YES/NO COLUMN AND SHOW IT AS TOGGLE ON FORM

Add new column of Yes/No type.

### New column

X

Previously called fields. [Learn more](#)

Display name \*

Online event

Description ⓘ

Data type \* ⓘ

Yes/no

1

Behavior ⓘ

Simple

Required ⓘ

Optional

Searchable ⓘ

Choices

Label *	Value *
⋮ <input checked="" type="checkbox"/> No	0
⋮ <input checked="" type="checkbox"/> Yes	1

Default choice \*

No

Advanced options ▾

**Save**

**Cancel**

Edit Form and add it to it.

1. Click on the column on form.
2. Select Component.
3. Add Toggle component.

## ← Add Toggle

Field [\(i\)](#)

Table column \*

Online event

Show component on 1

- Web
- Mobile
- Tablet

2

**Done**

**Cancel**

1. Define on which devic the toggle component should be visible.
2. Click on Done.

Now you've got yourself a toggle. It ain't much, but it's an honest component.

## POPULATE FEW ATTENDEES USING MS FORMS INTEGRATION

Go to <https://forms.office.com/Pages/DesignPageV2.aspx?origin=shell> using same tenant where you are building whole solution.

Create simple MS Form survey with 6 Text Fields:

First Name (Mandatory)

Last Name (Mandatory)

Email address (Mandatory)

Company

Job Title

Phone Number

## Baltic Power Platform Meetup - May 2024

1. First Name \*

Enter your answer

2. Last Name \*

Enter your answer

3. Email address \*

Enter your answer

4. Company

Enter your answer

5. Job Title

Enter your answer

6. Phone Number

Enter your answer

Set unique Theme color for the MS Form

Publish MS Form and preserve Title of the Event + Short URL + MS Form Id

## Short URL

Send and collect responses X

Anyone can respond  
Anonymous response, doesn't require sign-in

Only people in cleverdevs can respond

Specific people in cleverdevs can respond

[Copy link](https://forms.office.com/e/vD2Krq7e7r)

Shorten URL

Respondees will see the form like this.

[Feedback](#)

## MS Form Id

The screenshot shows a Microsoft Edge browser window with the following details:

- Address bar: https://forms.office.com/pages/responsepage.aspx?id=DGx5P2SsS0-2gYhB1ZtQGFaUK4S0dJGq9Au7\_5oxr5UMTVPRUhHQzZHMVJMRDl3TfETlhPWjVRNC4u
- Navigation bar: Back, Forward, Stop, Refresh, Address bar, Favorites, etc.
- Toolbar: M365 Admin Center, SPO Admin Center, PP Admin Center, Private, BLOGS, MS Blogs, Roadmaps & Ideas, Licensing, Power Platform, XrmToolbox, MDA, Co
- Content area:
  - Form title: Baltic Power Platform Meetup - May 2024
  - Form fields:
    - \* Required
    - 1. First Name \*
    - Enter your answer
    - 2. Last Name \*
    - Enter your answer

Duplicate Form few times (at least 2 times)

For duplicated MS Forms specify different Title and Theme Color (just to differentiate forms)

For all duplicated forms preserve Title of the Event + Short URL + MS Form Id

## CREATING MANUALLY EVENTS IN DATAVERSE

Go to MDA App and open Events section

Create Events corresponding to created MS Forms (at least 3)

Name	Event type	Event status	Event date and time	Sponsor	Available seats	Cost
Baltic Power Platform Meetup - February 2024	Workshop	Draft	2/23/2024 12:00 AM	Uncle Raspberry from S...	50	0.00
Baltic Power Platform Meetup - March 2024	Workshop	Draft	3/22/2024 12:00 AM	Three Musketeers	40	0.00
Baltic Power Platform Meetup - April 2024	Conference	Draft	4/25/2024 12:00 AM	Pinky Winky	120	100.00

## PLAYING WITH POWER AUTOMATE

Go to Events manager solution

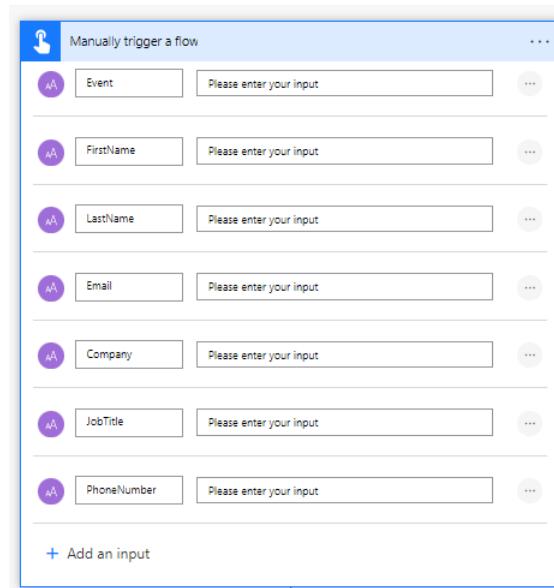
Go and create 4 Connection References to following Services:

- Microsoft Dataverse
- Microsoft Forms
- Office 365 Outlook
- Power Automate Management

Inside Solution create first Flow – Instant with Title:

[DEMO-1] Process Response (Child Flow) [Instant]

and Trigger



With 6 Input Parameters (single line of text)

Event

FirstName

LastName

Email

Company

JobTitle

PhoneNumber

Using **List rows** action from Dataverse find Event that equals to Event (input property)

Use FetchXML below:

```
<fetch top='1'>

    <entity name='bal_event'>

        <attribute name='bal_availableseats' />
        <attribute name='bal_cost' />
        <attribute name='bal_eventdateandtime' />
        <attribute name='bal_eventdescription' />
        <attribute name='bal_eventid' />
        <attribute name='bal_eventstatus' />
        <attribute name='bal_eventtype' />
        <attribute name='bal_name' />
        <attribute name='bal_sponsor' />
        <attribute name='bal_venue' />
        <filter>
            <condition attribute='bal_name' operator='eq'
value='@{triggerBody()['text']}' />
        </filter>
    </entity>
</fetch>
```

Rename action to **FindEvent**

Drop Compose Action and paste code below to this action (save flow and make sure Save works)

```
@{if(greaterOrEquals(length(outputs('FindEvent'))?['body/value']),1),
concat('bal_events(',outputs('FindEvent')?['body/value'][0]['bal_eve
ntid'], ')'), null)}
```

Add new row to Attendees table (use Dataverse Add new row action)

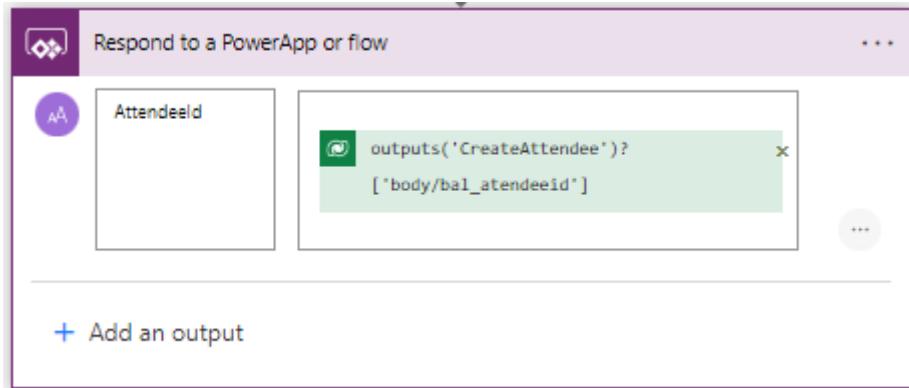
Fill in columns like on screen below (ask host in case of any issues)

Rename this action to CreateAttendee

CreateAttendee

*Table name	EM Attendees
*Full name	<code>fx concat(triggerBody()['text_1'], ' ', triggerBody()['text_2'])</code>
Atendee	Unique identifier for entity instances
Company	<code>triggerBody()['text_4']</code>
Email address	<code>triggerBody()['text_3']</code>
Event (EM Events)	<code>outputs('EventData')</code>
First name	<code>triggerBody()['text_1']</code>
Import Sequence Number	Sequence number of the import that created this record.
Job title	<code>triggerBody()['text_5']</code>
Last name	<code>triggerBody()['text_2']</code>
Owner (Owners)	Owner Id
Phone number	<code>triggerBody()['text_6']</code>
Record Created On	Date and time that the record was migrated.
Registration date	<code>fx utcNow()</code>
Status Reason	Active
Time Zone Rule Version Number	For internal use only.
Trigger Flow	
UTC Conversion Time Zone Code	Time zone code that was in use when the record was created.
Hide advanced options ^	

Place Respond to Power Apps action with following response parameter

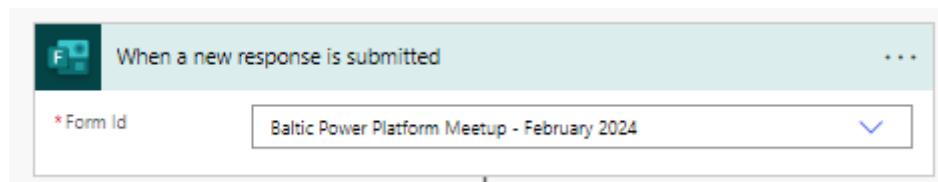


Save flow and get back to this flow Details page

Setup Run only users for this flow (this action is required for all Child Flows)

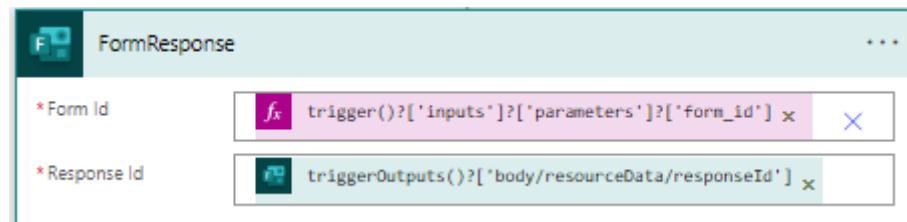
Create solution aware flow that listens to Responded to one of your MS Forms (pick one)

Trigger – When new response is submitted:



Then read details of response submitted (DO NOT hardcode MS Form Name – you can reference this from trigger input)

```
@trigger()?'inputs']?'parameters']?['form_id']
```



Place Compose action and prepare JSON object with following values:

```
{
  "Event": "Baltic Power Platform Meetup - February 2024",
  "FirstName": "{SELECT-FROM-FormResponse}",
  "LastName": "{SELECT-FROM-FormResponse}",
  "Email": "{SELECT-FROM-FormResponse}",
  "Company": "{SELECT-FROM-FormResponse}",
  "JobTitle": "{SELECT-FROM-FormResponse}",
  "PhoneNumber": "{SELECT-FROM-FormResponse}"
}
```

The screenshot shows the 'AttendeeData' JSON object in the Microsoft Power Automate interface. The object has the following structure:

```
{
  "Event": "Baltic Power Platform Meetup - February 2024",
  "FirstName": "",
  "LastName": "",
  "Email": "",
  "Company": "",
  "JobTitle": "",
  "PhoneNumber": ""
}
```

Each field value is a placeholder expression: `outputs("FormResponse")?` followed by a URL like `['body/r1acb5addd6434a3d82e8698a21d91653']`. This indicates that the values are being populated from a previous step in the flow.

Use Parse JSON action to parse output of AttendeeData to give Hints in next part of the flow

Use JSON Object notation as Sample Payload to generate JSON Schema

The screenshot shows the configuration of a Microsoft Power Automate flow named 'AttendeeDataParsed'. The top bar has a purple header with the flow name and three dots on the right. The main area is divided into sections: 'Content' and 'Schema'. The 'Content' section contains a single step: 'outputs('AttendeeData')'. The 'Schema' section displays a JSON schema definition:

```
{
  "type": "object",
  "properties": {
    "Event": {
      "type": "string"
    },
    "FirstName": {
      "type": "string"
    },
    "LastName": {
      "type": "string"
    }
  }
}
```

Below the schema is a button labeled 'Generate from sample'.

Call Child Flow that was created before – match all parameters by name (should be easy)

The screenshot shows the configuration of a 'Run a Child Flow' step. The top bar has a blue header with the step name and three dots on the right. The configuration table lists eight parameters:

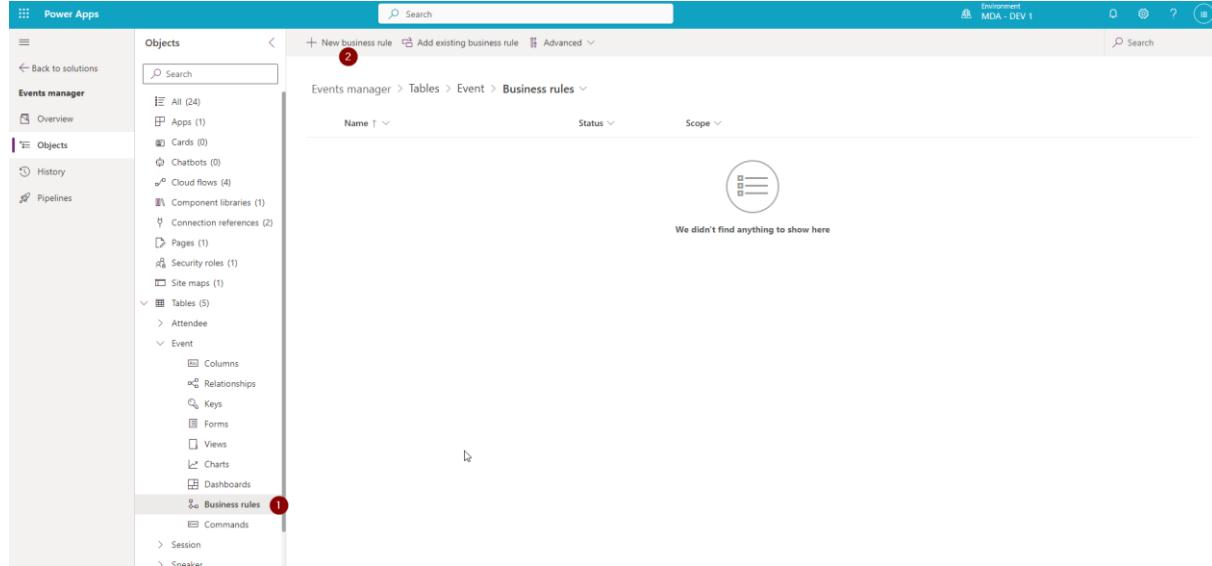
Parameter	Value
*Child Flow	[DEMO-1] Process Response (Child Flow) [Instant]
*Event	(.) body('AttendeeDataParsed')?['Event']
*FirstName	(.) body('AttendeeDataParsed')?['FirstName']
*LastName	(.) body('AttendeeDataParsed')?['LastName']
*Email	(.) body('AttendeeDataParsed')?['Email']
*Company	(.) body('AttendeeDataParsed')?['Company']
*JobTitle	(.) body('AttendeeDataParsed')?['JobTitle']
*PhoneNumber	(.) body('AttendeeDataParsed')?['PhoneNumber']

Bravo – you have just created automation that gives you ability to add Attendees to particular event.

But wait – we created MS Forms for few other events – should we manually create flows that will listen to those other MS Forms ?? Of course – not (we will present 2 nifty ways of saving your precious time)

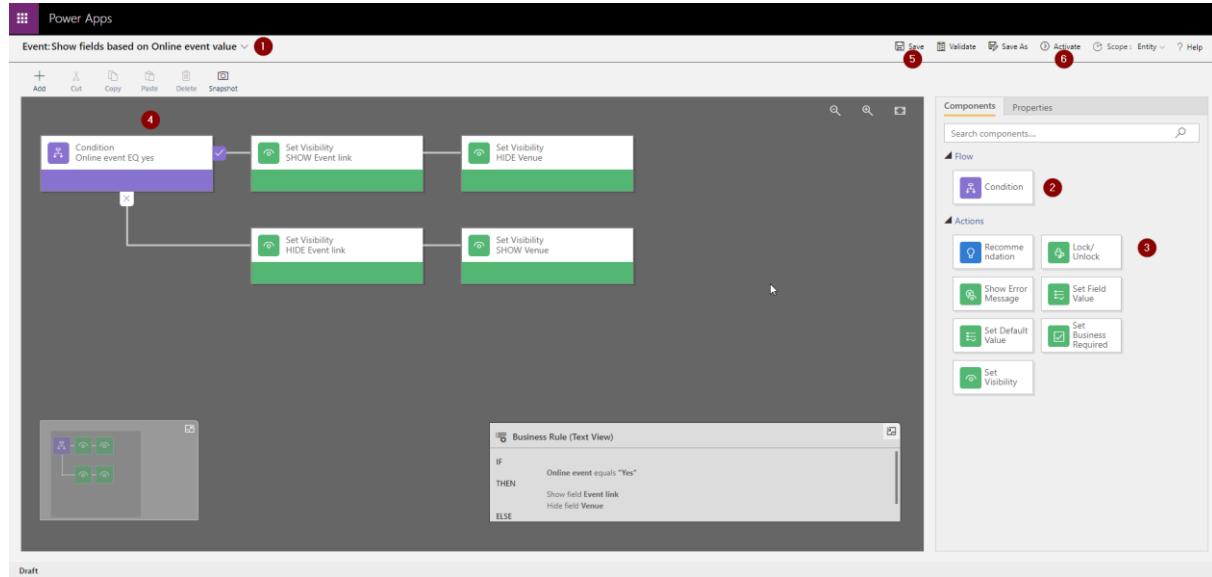
## CREATE BUSINESS RULE

Business rules are useful if you want to e.g. based on a specific condition define fields visibility or values on form. Go to solution.



1. Click on Business rules.
2. Click on New business rule.

In our example we will create a simple business rule which shows and hides fields based on the Online event field value.



1. Set business rule name.
2. Add Condition.
3. Add Actions.
4. Save business rule.
5. Activate it.

Now the new business rule will apply to the form.

Baltic Power Platform Meetup - April 2024 · Saved  
Event · Event main form ▾

General

Event name *	Baltic Power Platform Meetup - April 2024	Event type	Conference
Event status	Draft	Cost	100.00
Online event	<input checked="" type="radio"/> No <span style="color: red;">!</span>	Event date and time	4/24/2024 10:00 PM
Venue	...	Available seats	120
Sponsor	Pinky Winky		
Event description	April 2024 - conference		
			

1. Set Online event (the toggle field!) to No.
2. Venue lookup field is visible.

Baltic Power Platform Meetup - April 2024 · Unsaved  
Event · Event main form ▾

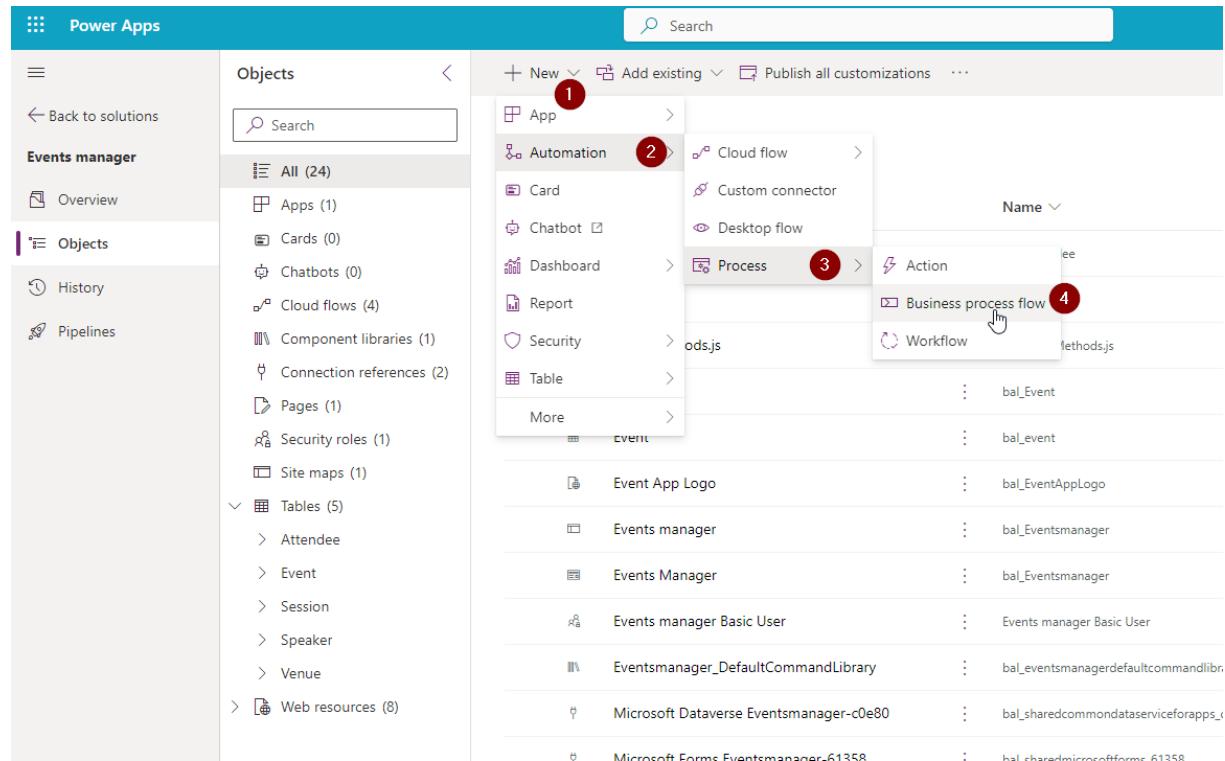
General

Event name *	Baltic Power Platform Meetup - April 2024	Event type	Conference
Event status	Draft	Cost	100.00
Online event	<input checked="" type="radio"/> Yes <span style="color: red;">!</span>	Event date and time	4/24/2024 10:00 PM
Event link	...	Available seats	120
Sponsor	Pinky Winky		
Event description	April 2024 - conference		
			

1. Set Online field to Yes.
2. Event link field is visible.

## BUSINESS PROCESS FLOW

Business process flow (BPF) allows us to create steps e.g. which should be followed when filling in the data. We will create a simple BPF for the Event table.



1. Go to solution and select New.
2. Automation.
3. Process.
4. Business process flow.

## New business process flow

Use business process flows to define a set of steps for people to follow to take them to a desired outcome.

Display name \*

 1

Name \*

 bal\_ eventbpf

Table \*

 Event 2

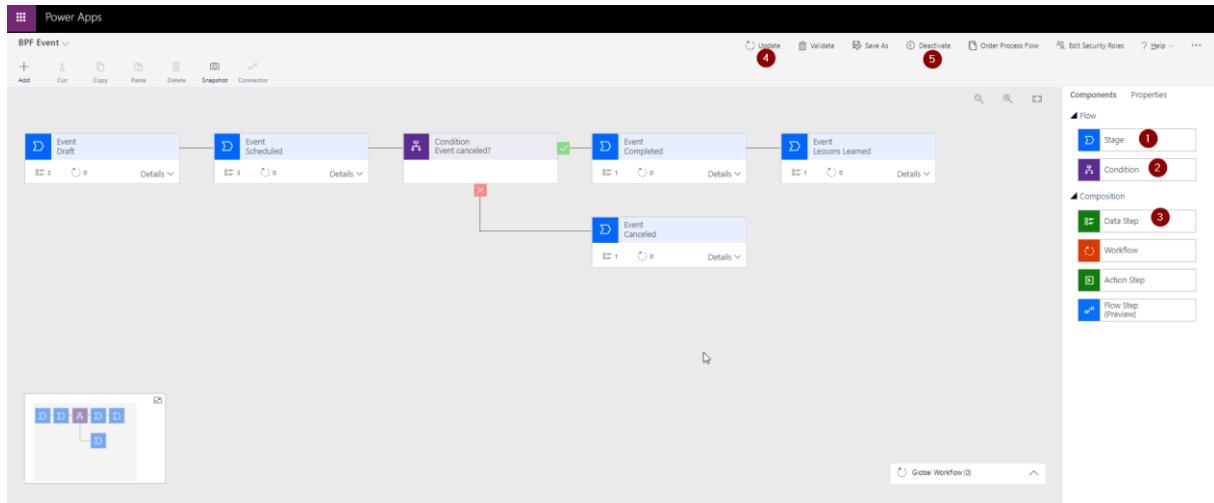
3

Create

Cancel

1. Give your process a name.
2. Connect it to a specific table.
3. Click on the Create button.

Once the BPF is created, you can edit it:



1. Add stages.
2. Add conditions.
3. Add data steps to stages (in our example).

We created a quite simple BPF with 5 stages: Draft, Scheduled, Completed, Lessons Learned and Canceled. There is one condition which influences the following stages.

Once we create our BPF logic we can:

4. Update it.
5. Activate or Deactivate it. Only the Activated BPF will be available on the form.

And that's how the BPF presents on Event form:

The screenshot shows the Power Apps Events Manager interface. On the left, there's a navigation sidebar with Home, Recent, Pinned, and Event Manager sections. Under Event Manager, there are sub-sections for Events, Attendees, Sessions, Speakers, Venues, and Send Email To Attendee. The main area is titled 'New Event' and shows the 'Event - Event main form' details. The 'General' tab is selected. The form includes fields for Event name, Event status (set to Draft), Online event (No), Venue, Sponsor, and Event description. Above the form, there are five circular icons representing stages: Draft (red dot, active), Scheduled (grey dot, inactive), Completed (grey dot), and Lessons Learned (grey dot). A tooltip 'Active for less than one minute' is displayed over the Draft icon. The 'Draft' stage also has a sub-tooltip 'Active for less than one minute' with a red dot icon. The 'Scheduled' stage has a tooltip 'Active for less than one minute' with a grey dot icon.

1. We can see the BPF with various stages.
2. For each stage we can see the predefined data step with column that need to be filled in for that stage.

Power Apps

Objects

Events manager > Security roles

Display name ↑	Name ↓	Type	Managed ↓	Last Modified... ↓	Owner ↓	Status
Events manager Basic User	Events manager Basic User	Security Role	No	6 days ago	-	

Search

New Add existing Edit Publish Advanced Remove

Events manager Overview History Pipelines

All (30) Apps (1) Cards (0) Chatbots (0) Cloud flows (6) Component libraries (1) Connection references (4) Pages (1) Processes (1) Security roles (1) Site maps (1) Tables (6) BPF Event EM Attendee EM Event EM Session EM Speaker EM Venue Web resources (0)

Events manager Basic User

Edit Publish Advanced Remove

## CUSTOMIZING MDA APPLICATION – MODERN COMMANDING / CUSTOM PAGES / POWER AUTOMATE INTEGRATION

In next 2 parts we will present how to extend MDA using:

- Custom Modern Commanding buttons
- Custom UI – Power Apps Custom Pages
- Custom Automations – multiple ways of executing Power Automate directly from Model Driven Apps

### HOW TO CALL POWER AUTOMATE FLOWS FROM MDA – METHODS COMPARISON

#	Method	Sync	Async	Notes
1	Using trigger from Dataverse (Legacy) – “When a row is selected”	No	Yes	This has OOTB method of display and integration
2	Calling Power Automate flow from Canvas App Custom Page (via {Flow-Name}.Run method)	Yes	Can be async too	Sync method can wait maximum 120 seconds for response.
3	Relying on column modification – manually updating rows from OOTB UI	No	Yes	Every row is connected exactly to 1 flow
4	Relying on column modification – patching selected records from Modern Commanding using PowerFx	No	Yes	Fast and easy to write
5	Relying on column modification – patching selected records from Modern Commanding using JS code	No	Yes	Harder to write – but overall UI might be much better (rich set of built in UI dialogs and operations in JS)
6	Business Events triggering – from Modern Commanding using PowerFx and Environment.{ActionName}	No	Yes	<b>DOES NOT WORK – REPORTED TO MICROSOFT</b> (seems to be a bug)
7	Business Events triggering – from Modern Commanding using JS and calling Custom API with params	No	Yes	Hard to write for Low Coders (but really powerful)
8	Business Events triggering – from Custom Page using PowerFx and Environment.{ActionName}	No	Yes	Easy to write and kick off
9	Run flow from BPF step	No	Yes	You can play with this to customize BPF stages and do some background operations.