

collab365.

SUMMIT EBOOK



collab365.

The Power of Power Apps Portals

Contents

Introduction	3
Definitions and Origins	3
When to use.....	4
What makes a Power Apps Portal?	5
Portal Admin Center	5
Azure Web App (the portal itself).....	5
ADX Solutions Deployed with Portal	6
Model Driven App (Portal Management).....	6
Portal Studio (WYSIWYG Editor)	6
Your Data	6
Demo	6
Environments.....	7
Apps	7
Solutions.....	10
Portal Management.....	11
Portal Overview	12

Authentication	18
More on Security	19
Web Roles	20
Website Access Permissions	20
Entity Permissions	20
Web Page Access Control Rules.....	20
How Security Works.....	21
Demo	22
Can I Customize It?	27
Liquid.....	28
Liquid Constructs	28
Web Templates	30
Demo	30
Learn More	33

Introduction

TARGET AUDIENCE

- New to Dynamics 365 world (WELCOME TO THE DARK SIDE)
- You have experience with the Power Platform, but not the portals
- If you want to learn something new!

The target audience for this eBook is people familiar with SharePoint and Power Platform, but who have some experience with Dynamics 365. If you've used Power Apps and Power Automate, you'll find this useful because Power Apps Portals have their roots in Dynamics 365 Portals. If you have experience with Power Apps but not the portals, you're welcome. And if you just want to learn something new, you're also welcome!

Definitions and Origins


DEFINITION & ORIGINS

- Adxstudio!
- It's all coming from the Dynamics world! (Renamed in Oct 2019 to Power Apps Portals)
- It uses CDS (DataFlex Pro!!) that's why it was named Power Apps Portals
- It's a website..
- ONE FACT.. Knowledge with Dynamics CRM is more beneficial than Power Apps

In 2015, Microsoft acquired a Canadian company called Adxstudio and built its portals product into Dynamics 365. In 2019, the product was renamed, 'Power Apps Portals' because it uses CDS (Dataflex Pro) and Microsoft wanted to take the Dynamics 365 world and push it into the Power Platform. The management of the portals is done through model-driven apps, but the whole idea is that it's based on CDS and that's why Microsoft took it from Dynamics and pushed it into the Power Platform.

Whenever you create a Power Apps portal it's actually a website in Azure. If you want to be good at Power Apps Portals you need to have some experience of Dynamics CRM.

When to use



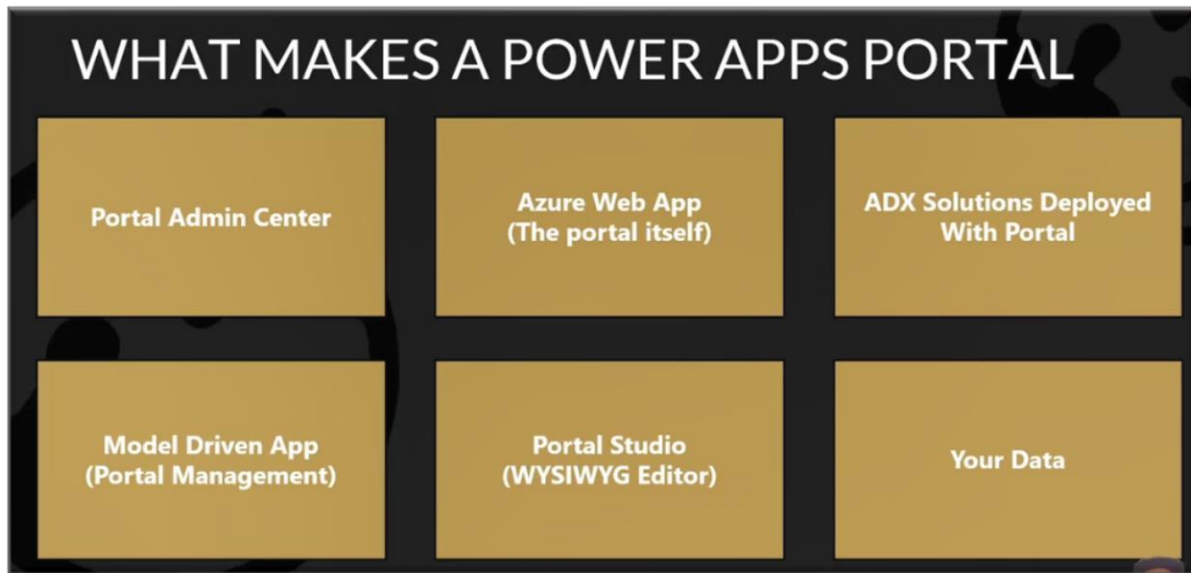
WHEN TO USE

- Other ways to develop applications.
Just ask a question... Is your data in CDS "DataFlex Pro"?
- Other factors can affect your decision

You might say, "I can create my website on top of Microsoft Azure, create it with WordPress, or have a custom ASP.NET application hosted somewhere – why would I need to use Power Apps Portals?" Well, in order to decide, you have to ask yourself where your data is right now. Is your data in CDS? If so, you would need to think about going with Power Apps Portals. However, that's not strictly true because it can actually be expensive. Whenever anonymous users go into your website, you would have to pay for that, so you would have to check the

licensing. If you need to have a document management system, you're probably not going to be doing that in Power Apps Portals as well.

What makes a Power Apps Portal?



Let's look at the components of a Power Apps Portal.

Portal Admin Center

Just like in SharePoint, each portal has an admin center, where you can manage SSL, manage the domain, and manage SharePoint integration so that you can upload and download documents to and from SharePoint.

Azure Web App (the portal itself)

For each portal, there's going to be an Azure website in the backend. You don't need to be worried about this, but just be aware.

ADX Solutions Deployed with Portal

Everything in the Azure portal is based on entities. The portal itself is based on CDS entities. Where do these entities come from? Whenever you create a new website and Azure and Power Apps Portals, you'll have ADX solutions in your Power Platform environment. The solutions will be deployed, and they will provide you with web resources and entities. If you click on any solution, you'll see what has been deployed.

Model Driven App (Portal Management)

How do we manage the entities? Whenever you create a new Power Apps portal you get a model-driven application to help you manage the portal.

Portal Studio (WYSIWYG Editor)

You manage the site itself using Portal Studio. You can go in and see what the page looks like, and you can create pages, write some text.

Your Data

The most important thing is going to be your domain data. The data for the portal itself is going to be managed by the model-driven application. However, there's also the data that you own, and this will be managed by you. You'll need some sort of model-driven application where you can manage your own data.

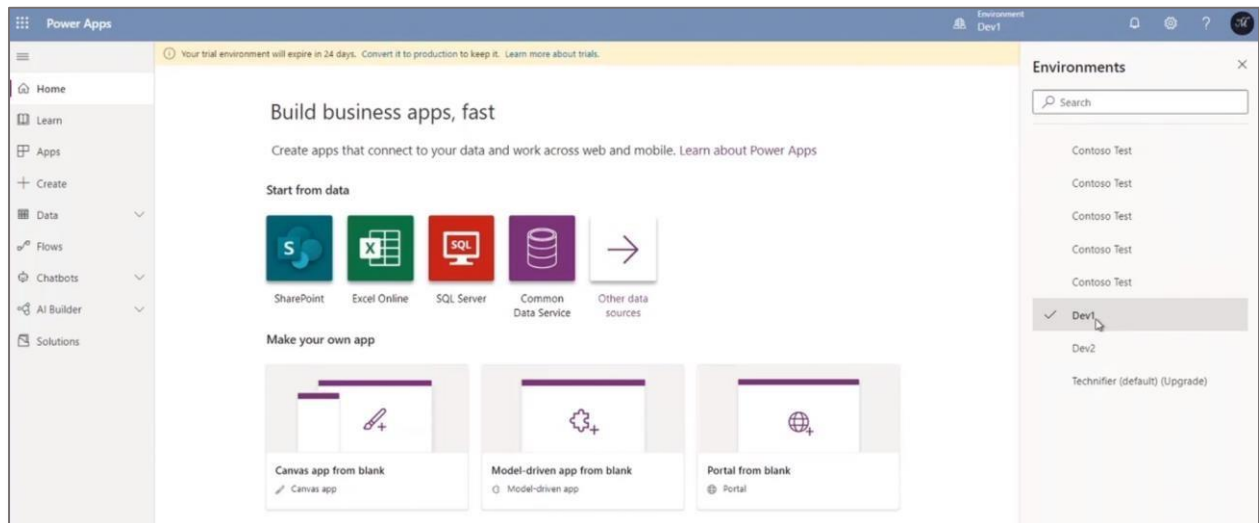
In summary, when you create a Power Apps portal you have the admin center, the Azure site in the backend (which users are going to be routed to), the ADX solutions (whenever you create a website they are going to deploy stuff to you), a model-driven app to manage the entities, Portal Studio for managing web pages, and a way of managing your own data.

Demo

Let's look at an environment.

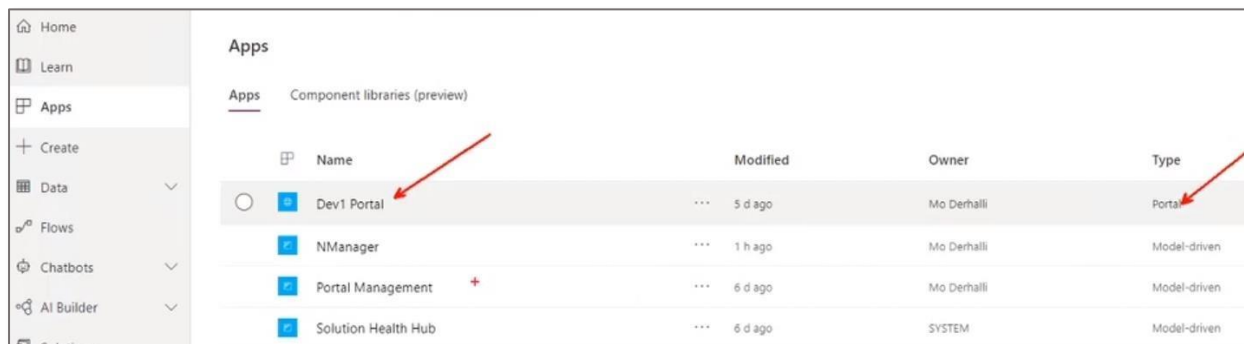
Environments

This is a trial environment called “Dev1”.

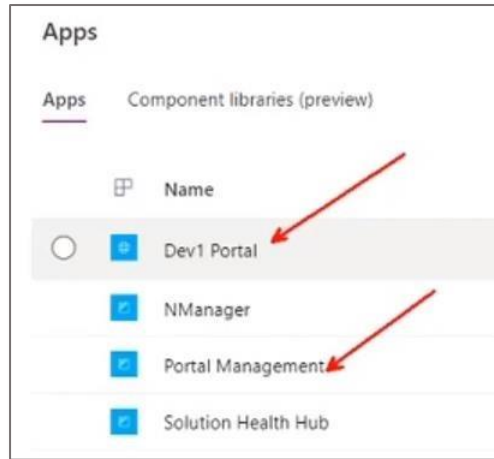


Apps

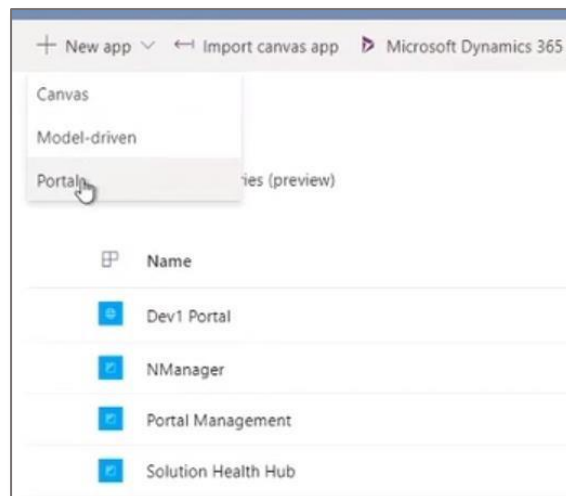
If you go to 'Apps', you'll notice a portal. You can tell it's a portal because it says 'Portal' in the Type column.



You'll notice this 'little brother', which is Portal Management.



The portal was created by clicking 'New App' and selecting 'Portal'. It was named 'Dev1 Portal' and the model-driven app was named 'Portal Management'.

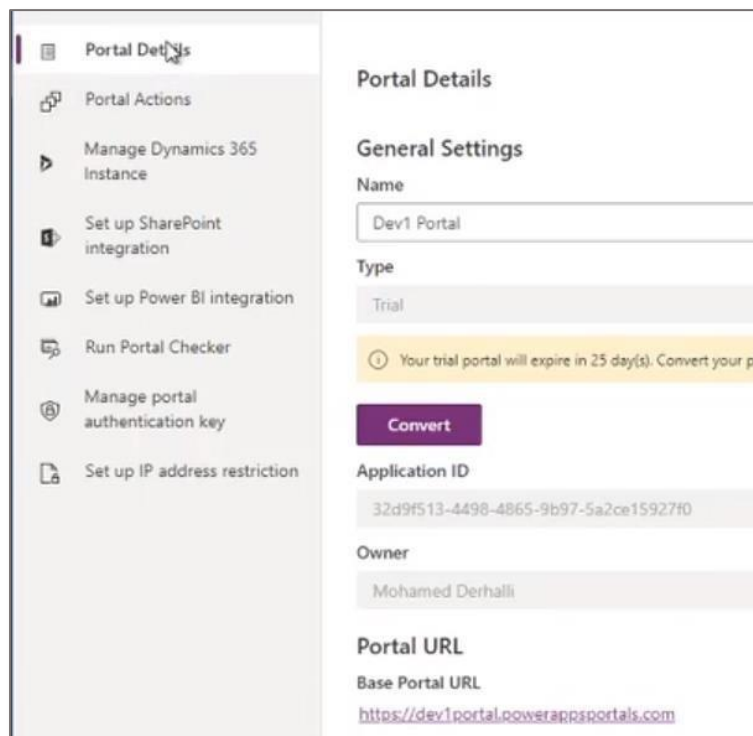


Admin Center

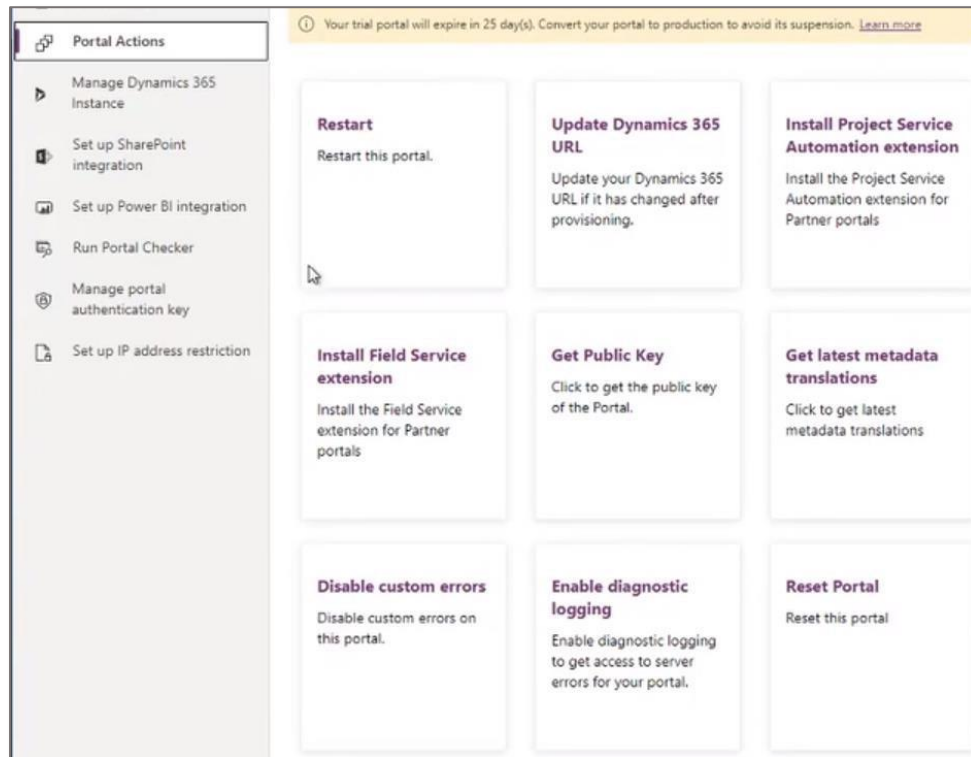
To access the admin center, click on the ellipsis next to the portal name and select 'Settings'.



In Portal settings, you can see the portal name, the URL for the portal, which anyone can just access, and you can click on 'Administration'. This opens a new tab where you can see the Application ID. If you go to Microsoft Azure and open the available app registrations, you'll find this one. This is a trial version of Power Apps Portals, but if you have a live version, you'll notice options on the left-hand side for SSL and registering your domain.

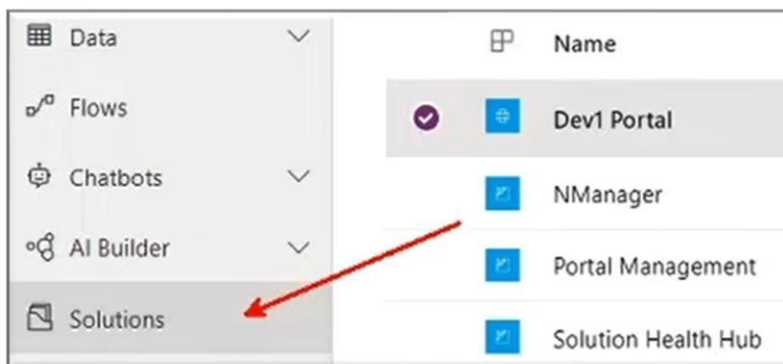


In Portal Actions, there are actions like Restart and Disable custom errors.



Solutions

Back in the Environment window is a 'Solutions' option.



If you click on this, you'll see the Dynamics 365 Portals. Microsoft hasn't renamed them Power Apps Portals yet. All these solutions have been deployed:

Display name	Name
Technifier	Technifier
Common Data Service Starter Portal	CDSStarterPortal
Dynamics 365 Portals - Base HTML Editor	BaseHtmlEditor_portal
Dynamics 365 Portals - Timeline	MicrosoftPortalTime...
Dynamics 365 Portals - Azure Storage	MicrosoftAzureStora...
Dynamics 365 Portals - Feedback	Feedback
<input type="radio"/> Dynamics 365 Portals - Web Forms	MicrosoftPortalWebF...
Dynamics 365 Portals - Identity Workflows	MicrosoftIdentityWo...
Dynamics 365 Portals - Identity System Workfl...	MicrosoftIdentitySys...

If we click on 'Dynamics 365 Portals – Feedback' we can see that these two entities have been deployed: Web Page and Feedback.

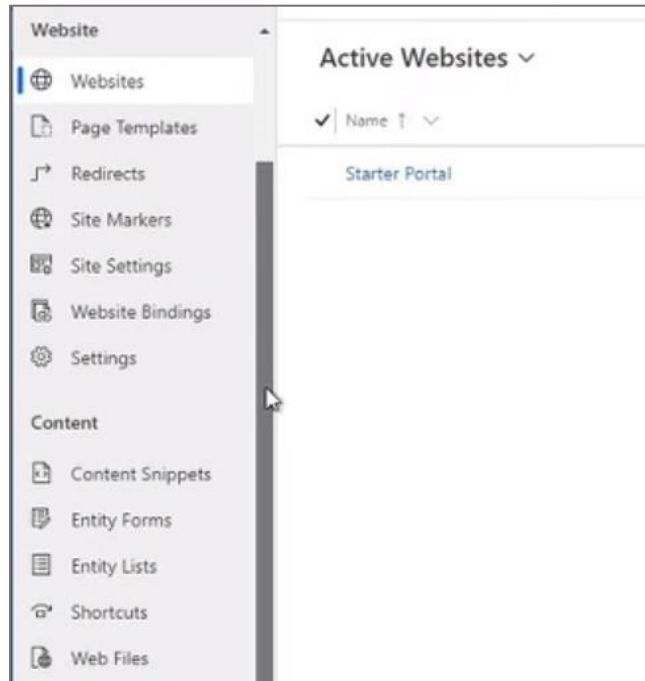
Display name	Name	Type
Web Page	adx_webpage	Entity
<input type="radio"/> Feedback	feedback	Entity

Other options for a solution include Site map, Web resource, Entity, Option set, and so on.

Portal Management

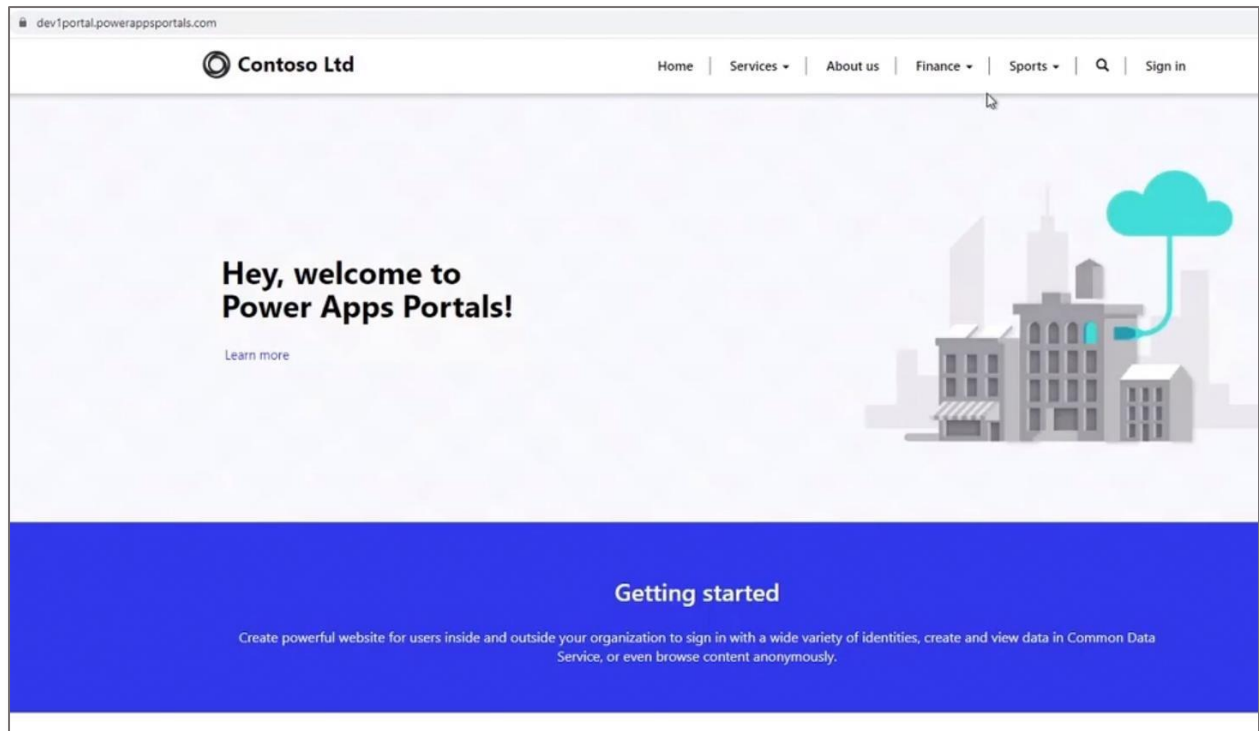
So, we have the solution, the portal and the portal's little brother, Portal Management.

You can manage all your entities from within this model-driven application. In Site Settings, you can allow authentication by Twitter, LinkedIn, and so on. In Web Files, you can upload images. There are also Web Forms, Web Pages, Web Templates, a Security section.

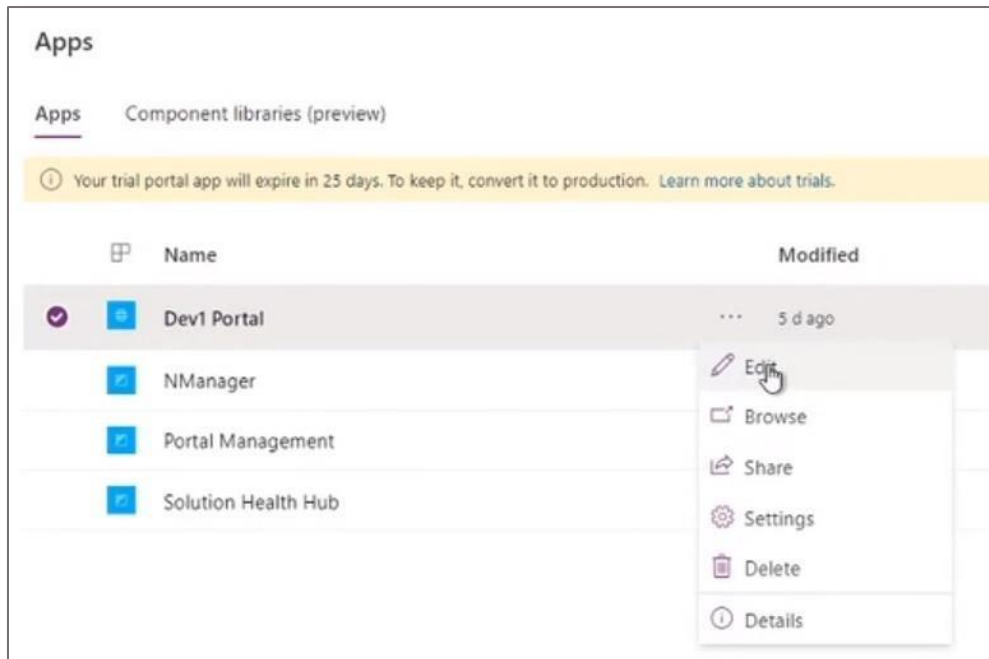


Portal Overview

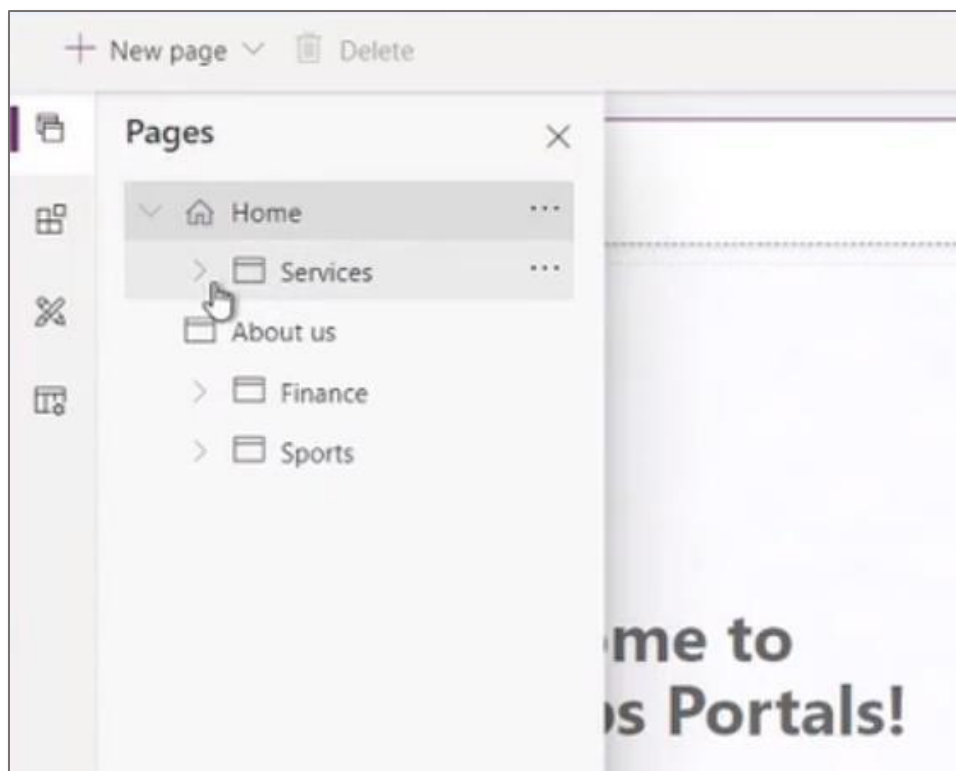
Let's take a look at the portal itself, by clicking on the portal name (Dev1 Portal) in the Apps window.



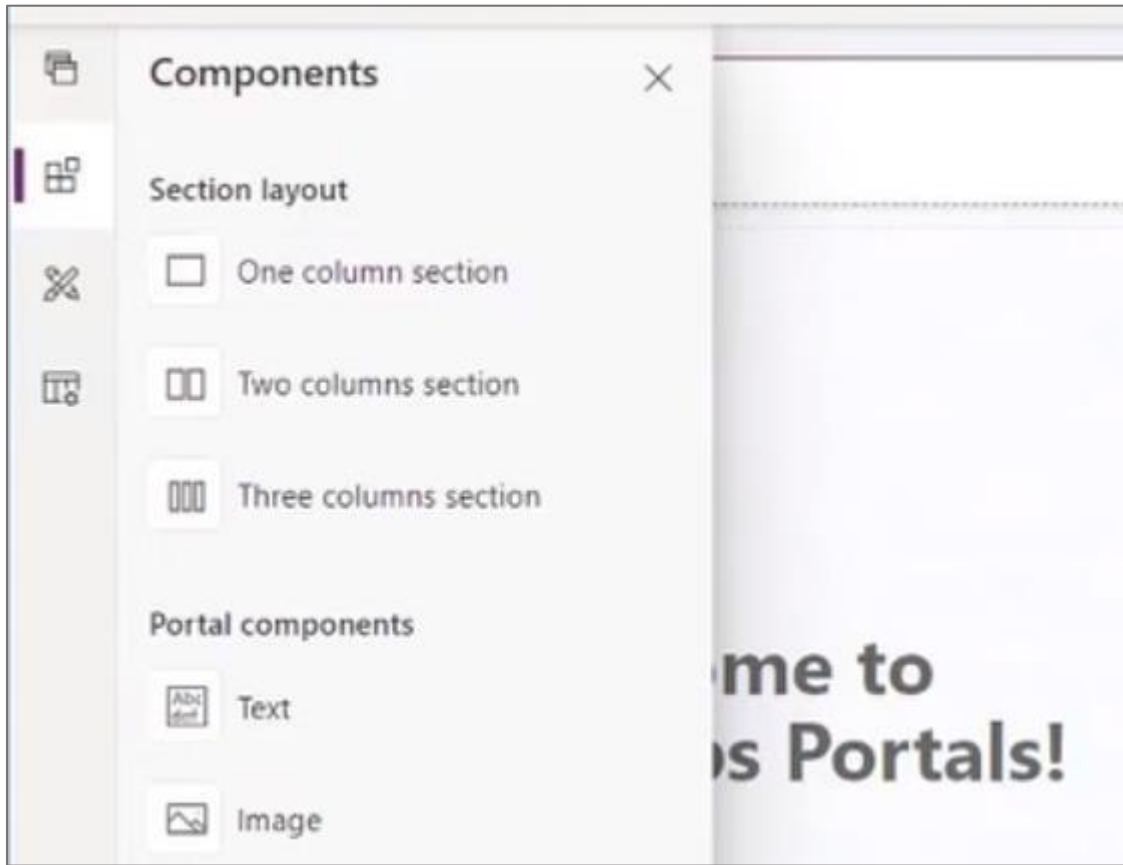
This is the portal's default home page. A few links have been added already (About us, Finance, Sports), but, otherwise, this is out of the box. To edit the page, click on the ellipsis next to the portal name and select 'Edit'.



This takes you to the portal studio. You can see the pages on the left. Some pages have 'children' in a dropdown. Here, you can create new pages.

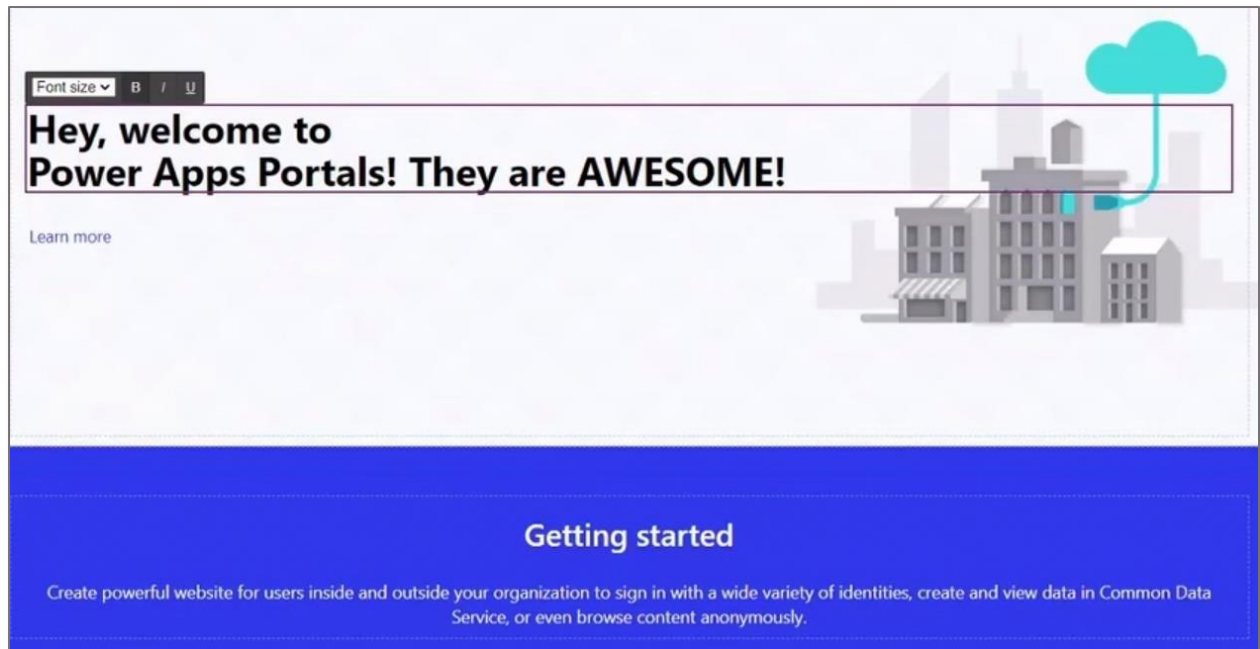


In Components, you can add new sections just like in SharePoint with one column, two columns, and so on. Within those sections, you can add text, images, lists, forms, breadcrumbs. There's also integration with Power BI.

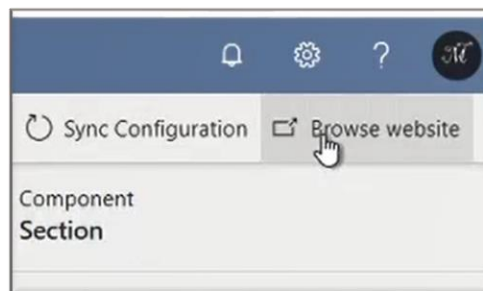


Beneath the Components option on the left are options to upload your own CSS styles and create templates.

The heading below was created using the text component. Whenever you modify something it's going to be saved immediately – see bottom right.



Click on 'Browse website' to see how your updates affect the live website.

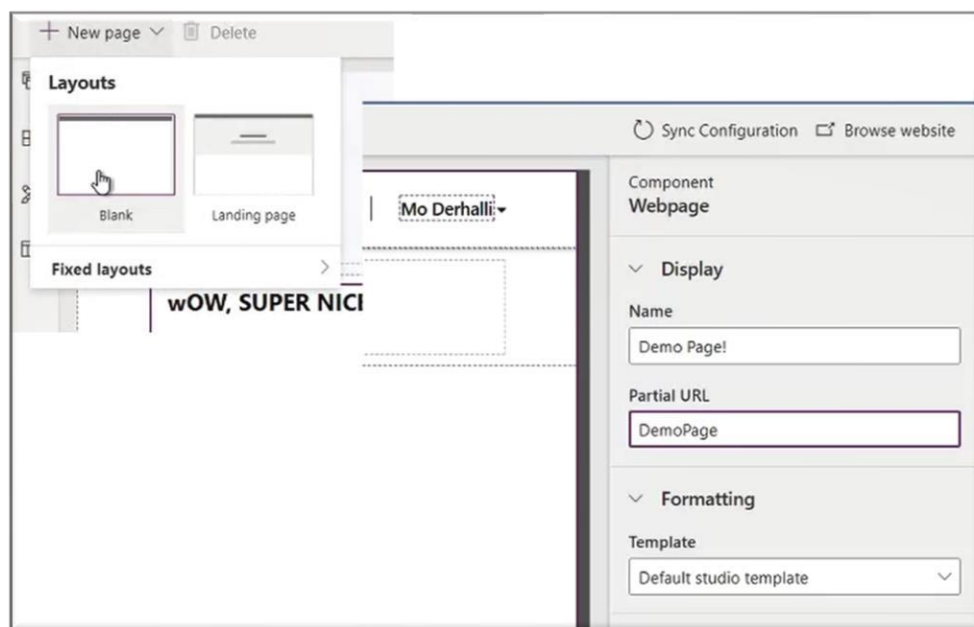


Here, we've added a two columns section from the Components option. In the left column, we added some text and formatted it and in the right column we added one of the existing images, but you can also upload your own images.

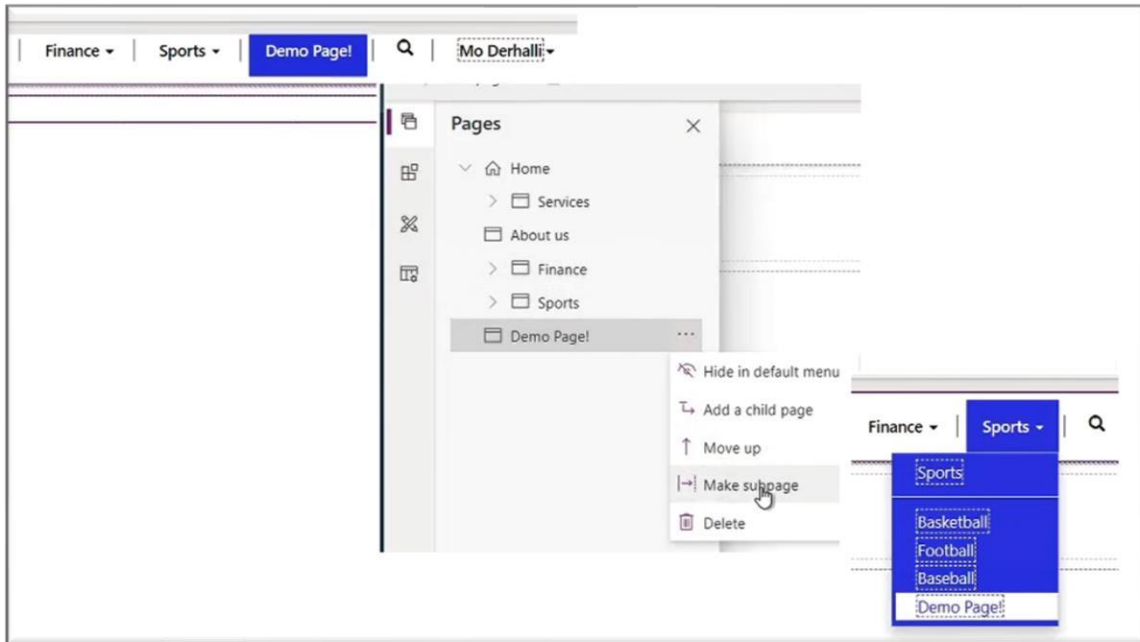


Creating a new page

We clicked on 'New page', selected 'Blank', and entered a Name and Partial URL, which is appended to the end of the website address. You can also add templates here that you have created in Portal Management.



The name of the page appears as a link in the web page's navigation bar. If you want it to appear elsewhere in the navigation, click on Pages, then the ellipsis by the page name, and select from the options. In the example, the page has been made a subpage of Sports.



Authentication

How can users access your website?

AUTHENTICATION

- Anonymous access
- Local Authentication (Users with their existing emails, credentials stored in Contact entity)
- Azure AD
- Facebook, Twitter, etc..

Important Note

Every user needs a contact record, even Azure AD users will have contact records.

- Anonymously, just by opening the website.
- Register to the website with local authorisation, by entering their email addresses and choosing a password. Their credentials will be stored in the system, so next time they type their email address and password they will log into the website.
- Azure AD – logging in with an existing Azure account
- Facebook, Twitter, etc.

You can use any method to log in – they will all be connected to the same account. So, later on, if you remove Twitter you can log in with Facebook, and the other way round.

Many ways are connected to the same user.

Power Apps Portals in the backend uses the Contact entity. Every single user, even Azure AD users, will be stored in the Contact entity.

More on Security



Web Roles

Web roles are the centre of authentication for Power Apps Portals because whenever you create a user account you assign them to a web role. If we create a website for a university, we'll have teachers, students, the admissions team, and so on, and we will create a web role for every one of these groups. This is like assigning permission levels to the users in SharePoint.

Website Access Permissions

Website access permissions let you assign permission for your users to access content snippets or reusable text, such as footer text or a disclaimer. Do you want users to be able to change these snippets? You have the notion of site markers and weblinks, which also fall under website access permissions. You add website access permissions to a web role and assign a web role to a user.

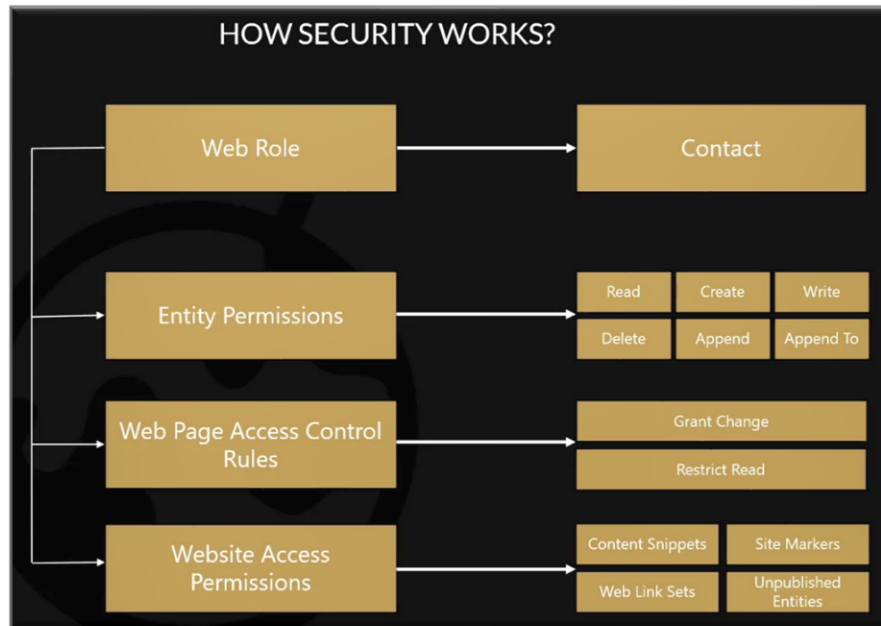
Entity Permissions

If you assign teachers or students to a web role then you would need the web role to have permissions for a specific entity. You may want teachers to be able to edit the list of schools but students only to be able to read it. The students will have READ permissions to the entity, but teachers will have READ and WRITE permissions.

Web Page Access Control Rules

This allows me to control permissions to the page itself rather than individual entities.

How Security Works



The web role is connected to a contact, or rather it's assigned to a contact, and within the web role will be entity permissions, web page access control rules and website access permissions.

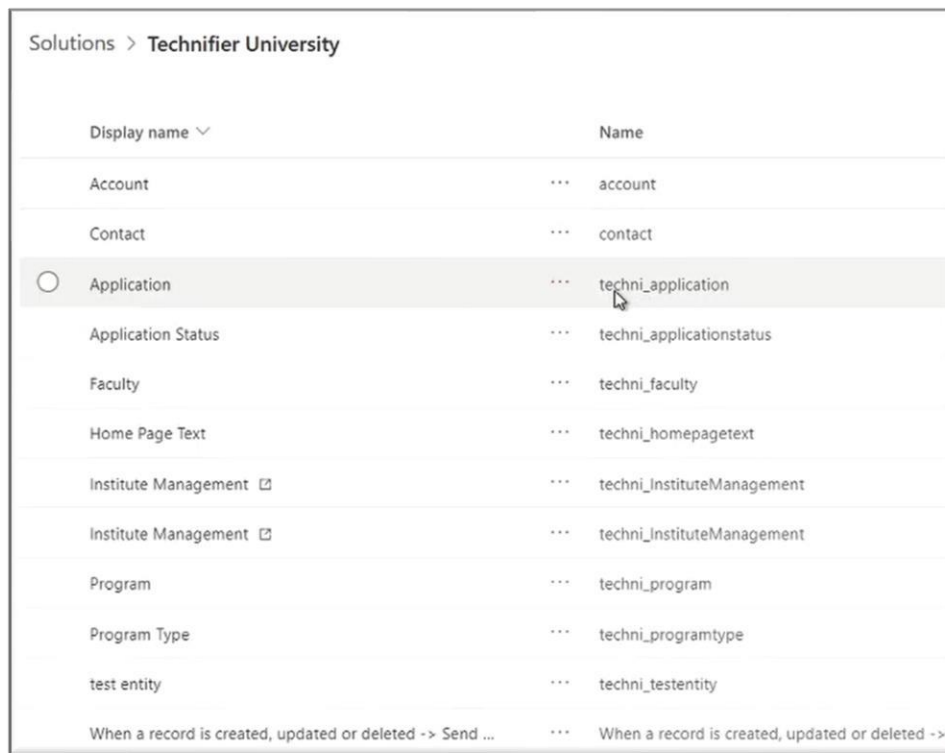
Entity permissions have settings such as Read, Create, Write, Delete, Append and Append To, which apply on an entity level.



Settings for web page access control rules include Grant Change and Restrict Read.

Website access permissions are assigned to content snippets, site markers, weblink sets and unpublished entities.

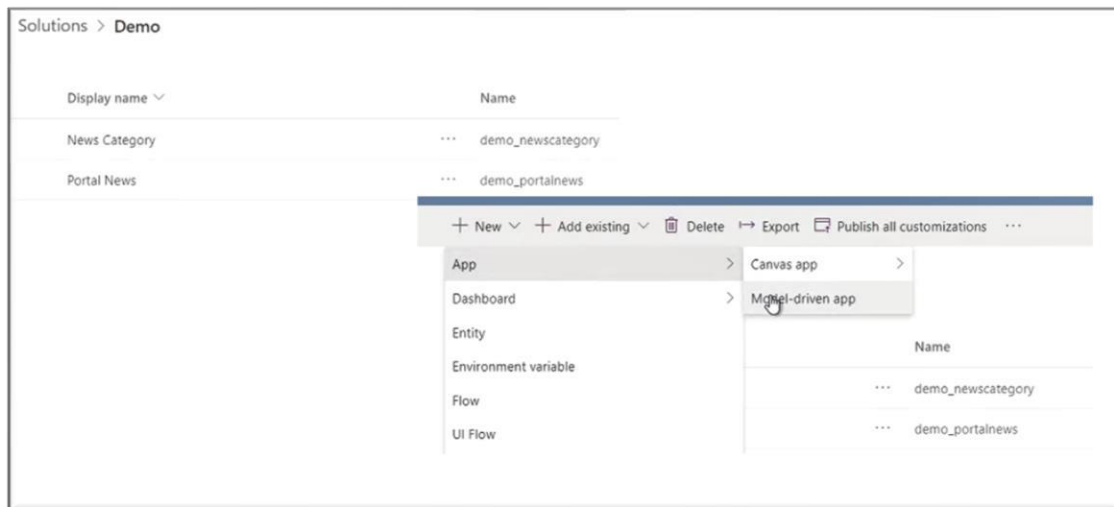
Demo

Let's look at a live environment called, 'Technifier', which has a solution called, 'Technifier University'. When you create a solution, you are given the option to add a prefix to custom entities and fields. In this example, the entities are all prefixed with 'techni_'.

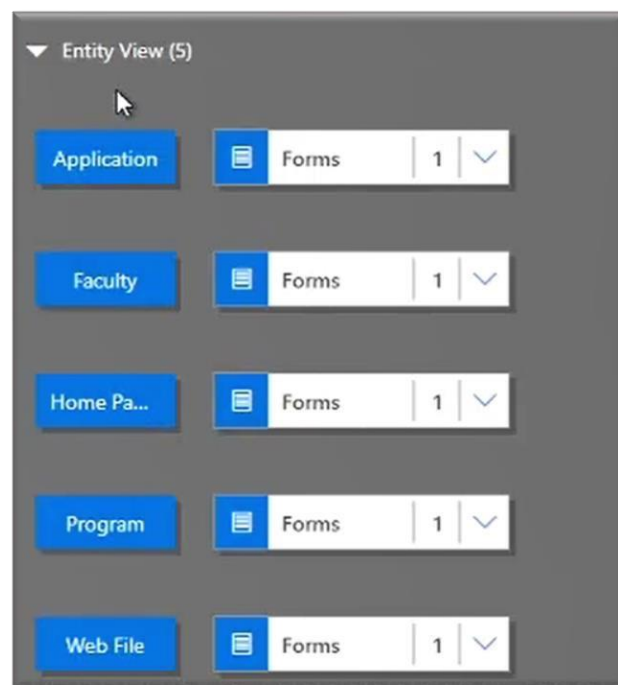


Display name ▾	Name
Account	*** account
Contact	*** contact
<input type="radio"/> Application	*** techni_application
Application Status	*** techni_applicationstatus
Faculty	*** techni_faculty
Home Page Text	*** techni_homepagetext
Institute Management 	*** techni_InstituteManagement
Institute Management 	*** techni_InstituteManagement
Program	*** techni_program
Program Type	*** techni_programtype
test entity	*** techni_testentity
When a record is created, updated or deleted -> Send ...	*** When a record is created, updated or deleted ->

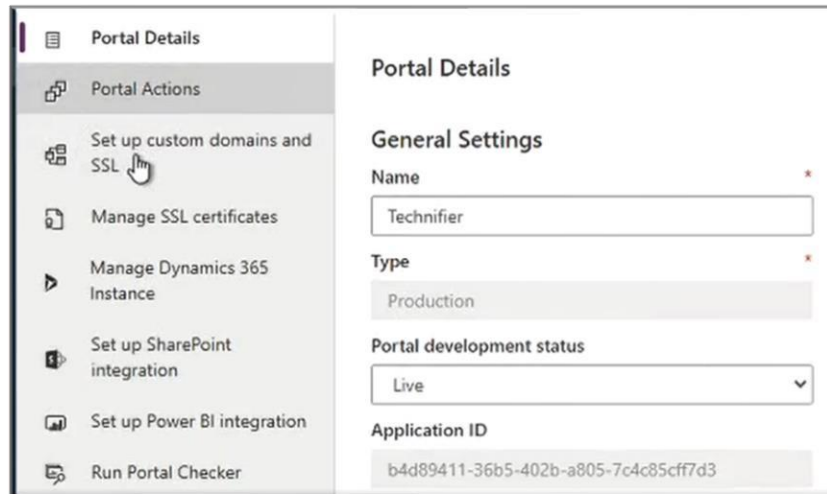
We added a new entity called, 'Portal News', and added a new field called, 'News Category'. To the category, we added Option set data types: 'Finance' and 'Sports'. Once you have created the entities you can add a new model-driven app.



In the example university solution, we've created a model-driven application with a set of entities to manage student admissions.



The website was bound to a custom domain in the Settings/Administration pane for the app portal and has the URL 'technifier.powerappsportals.com'.



Portal Details

General Settings

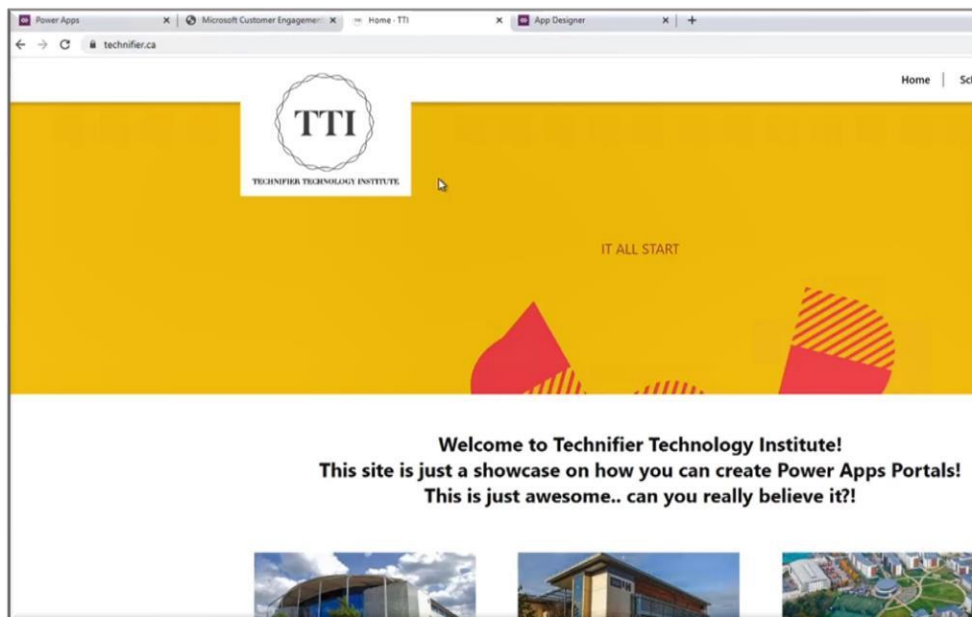
Name: Technifier

Type: Production

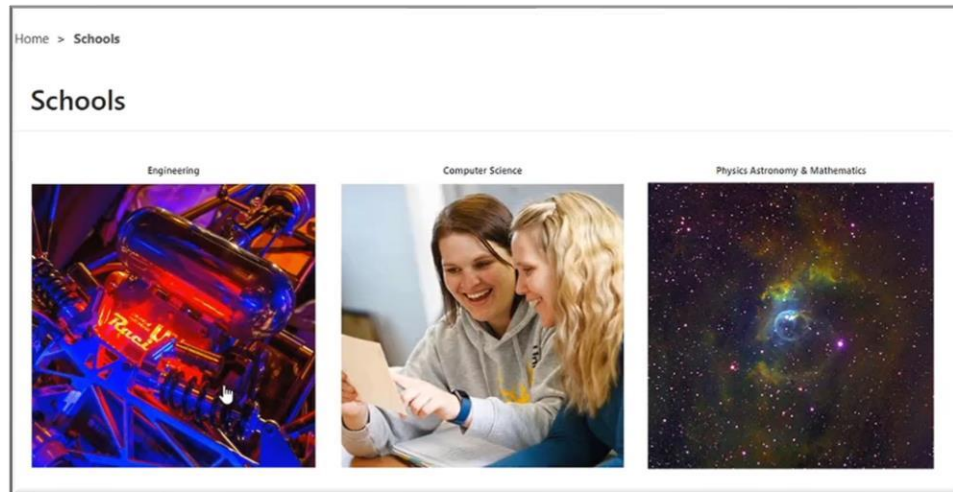
Portal development status: Live

Application ID: b4d89411-36b5-402b-a805-7c4c85cff7d3

Because of the domain binding, you just enter 'technifier.ca' to see the website.



The Schools page shows the different faculties, which have been created in a custom entity. The site has been designed with various web technologies such as CSS, HTML, and jQuery.



Courses are listed on the site and for each one there's an application form.

Apply

First Name *


Last Name *

Email *

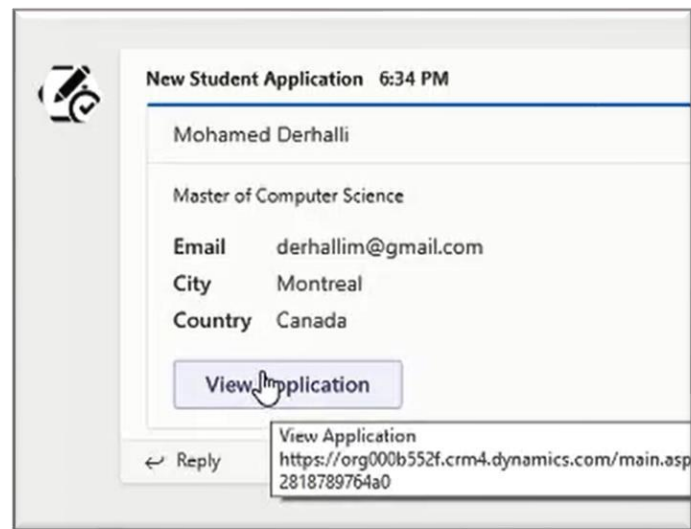
Phone Number *

Country *

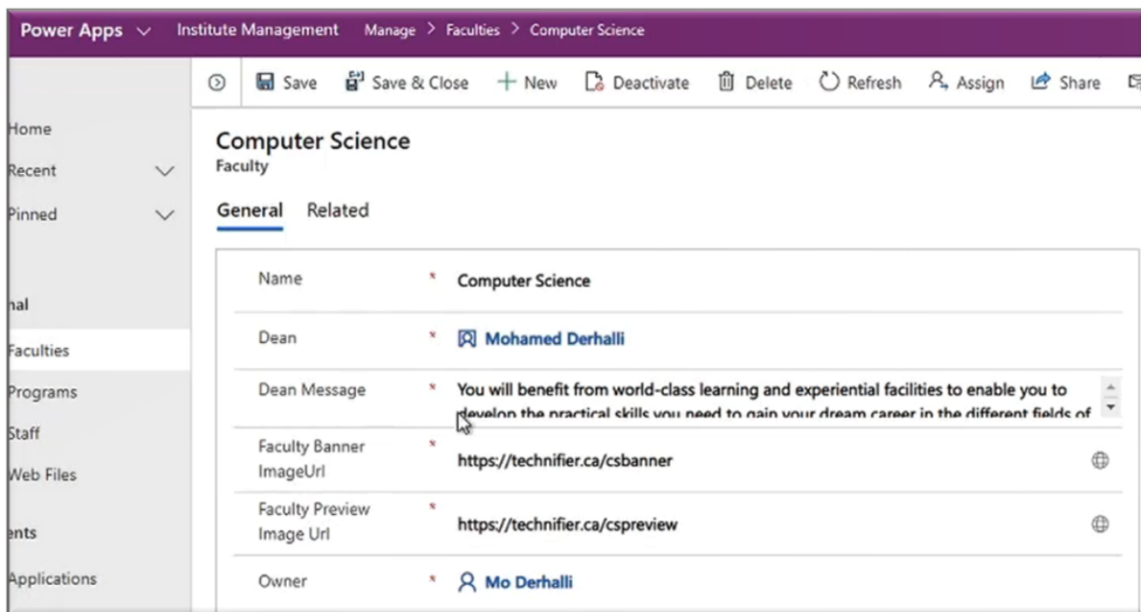
City *


Generate a new image
Play the audio code
 Enter the code from the image

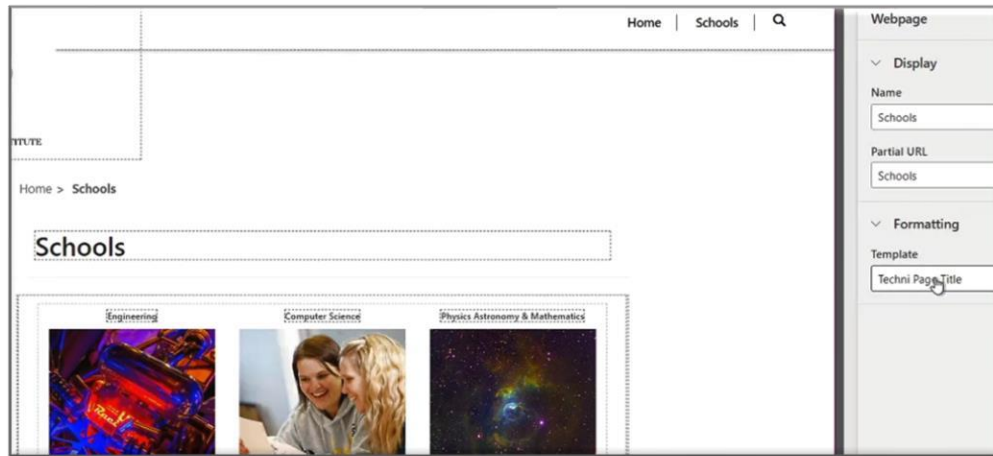
When an applicant completes a form, the admissions administrator receives a confirmation in Microsoft Teams and they can view the application straightaway.



Each faculty is created as an entity and can contain a banner message and preview image.

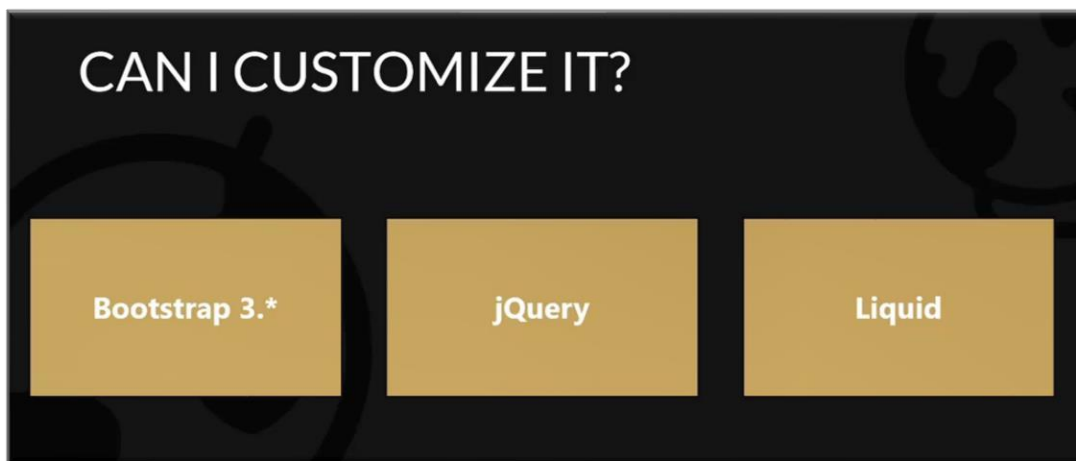


The Schools page was created in the Apps portal using a template.



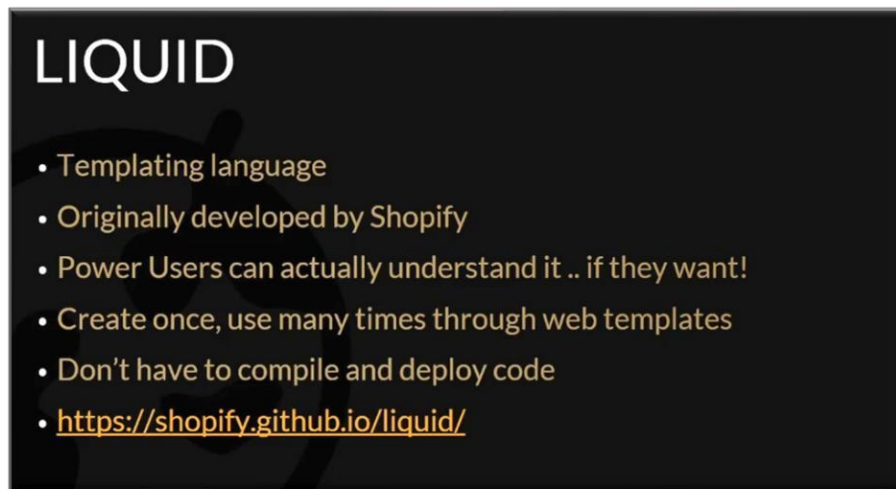
The code editor was used to fetch the faculties from the system, get the required columns and display HTML bootstrap classes.

Can I Customize It?



Yes, you can customize your site using Bootstrap, jQuery, and Liquid.

Liquid



- Templating language
- Originally developed by Shopify
- Power Users can actually understand it .. if they want!
- Create once, use many times through web templates
- Don't have to compile and deploy code
- <https://shopify.github.io/liquid/>

Liquid is the templating language for Power Apps Portals. It was created by Shopify, and then Adxstudio and Microsoft started using it for web pages. It's relatively easy for Power Apps users to understand it – if you can understand the formulas in Power Apps you should be able to understand Liquid. If you use Liquid with web templates, which are essentially the heart of customisation in Power Apps Portals, you can use the templates anywhere you want – for example, create a page and bind it to a web template. The nice thing about Liquid is you don't have to keep deploying code, you just do your edits, and everything is refreshed.

Find out more about Liquid: <https://shopify.github.io/liquid>

Liquid Constructs

In Liquid, placeholders, objects, properties/attributes and logic/tags are enclosed in curly brackets. It's just like handlebars in JavaScript.

LIQUID CONSTRUCTS

- **Placeholders** `{{ }}` : `{{SomeValue}}`
- **Objects and Properties/Attributes**: `{{Product.Name}}`
- **Logic/Tags**: `{% %}`:

```
{% if cart.item_count > 0 %}  
<p>You have {{ cart.item_count }} item(s) in your cart</p>  
{% else %}  
<p>There's nothing in your cart :(</p>  
{% endif %}
```

There are also operators, loops, and filters.

LIQUID CONSTRUCTS

- **Operators**: `==, !=, >, <, >=, <=, or, and, contains`
- **Loops**

```
{% for image in product.images %}  
  
{% endfor %}
```

- **Filters**: `{{ article.published_at | date: '%d %B %Y' }}`

Web Templates

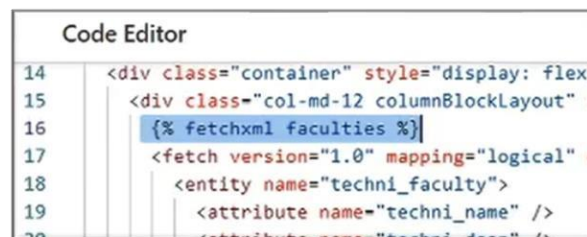
Web Templates

- Allows you to write reusable code
- Liquid code can be written anywhere you can have HTML
- Page templates can be based on web templates
- You can utilize many of the Liquid objects, such as site markers, web link sets, etc.
- You can do all sorts of crazy stuff with web templates in Power Apps Portals

Web templates are the core of customisation. They allow you to write reusable code – for example, create a template and bind it to a web page. Similar to a web service where you want to request some stuff from a system bound to your code, you can just use a web template. Liquid code can be written anywhere you want to write HTML– you don't have to create web templates; you can just write Liquid code in the HTML pages. Page templates can be based on web templates. You can utilize many Liquid objects, such as page objects, weblinks, site markers and snippets. You can do all sorts of crazy stuff with web templates in Power Apps Portals!

Demo

As we saw earlier, the Faculties section of the demo site shows the available faculties (schools): Engineering, Computer Science and Physics Astronomy & Mathematics. In Code Editor in the Faculties page, we use '`fetchxml faculties`' to fetch the faculty names.



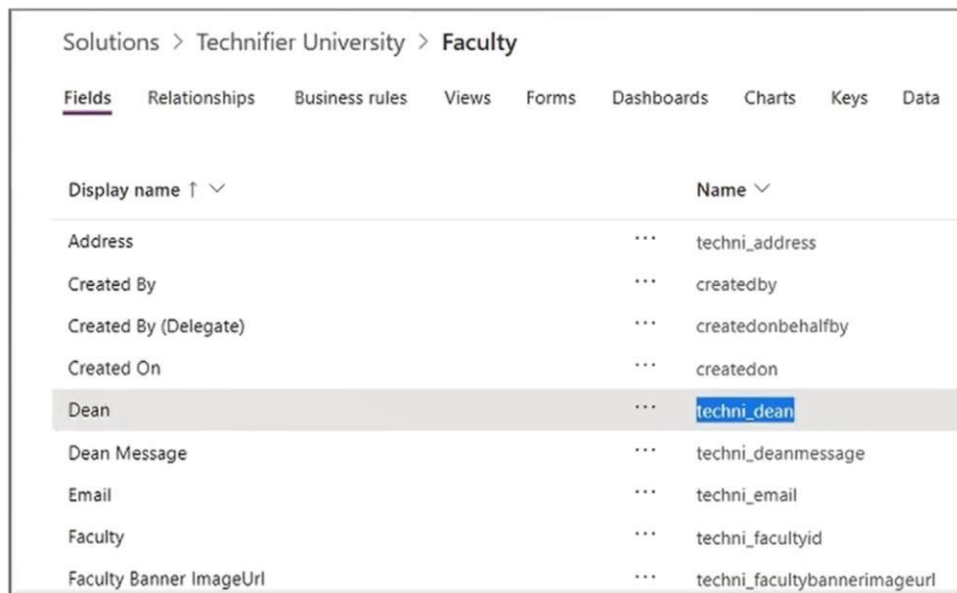
```
Code Editor
14 <div class="container" style="display: flex;
15 <div class="col-md-12 columnBlockLayout" s
16 {% fetchxml faculties %}
17 <fetch version="1.0" mapping="logical" c
18 <entity name="techni_faculty">
19 <attribute name="techni_name" />
20 <attribute name="techni_dept" />
```

We use this syntax to fetch the entity ('`techni_faculty`') and its columns (attributes) (it's a bit like CAML Query in SharePoint):



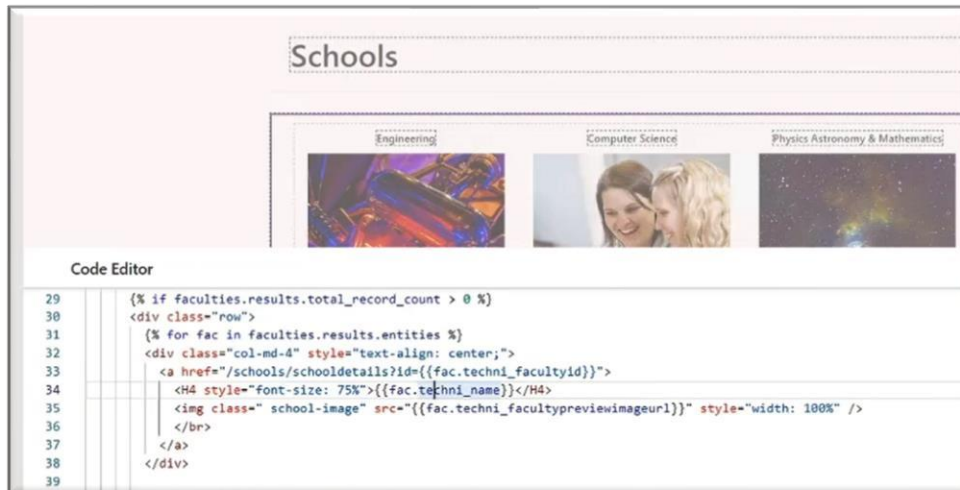
```
16 {  
17 <fetch version="1.0" mapping="logical" output-format="xml-platform" distinct="true" returntotalrecordcount="true">  
18 <entity name="techni_faculty">  
19 <attribute name="techni_name" />  
20 <attribute name="techni_dean" />  
21 <attribute name="techni_facultybannerimageurl" />  
22 <attribute name="techni_facultypreviewimageurl" />  
23 <attribute name="techni_facultyid" />  
24 </entity>  
25 </fetch>  
26 }
```

The internal names for the columns have been added in the Faculty entity of the solution:



Solutions > Technifier University > Faculty	
Fields	Relationships Business rules Views Forms Dashboards Charts Keys Data
Display name ↑ ▾	Name ▾
Address	*** techni_address
Created By	*** createdby
Created By (Delegate)	*** createdonbehalfby
Created On	*** createdon
Dean	*** techni_dean
Dean Message	*** techni_deanmessage
Email	*** techni_email
Faculty	*** techni_facultyid
Faculty Banner ImageUrl	*** techni_facultybannerimageurl

In the Code Editor, we define the faculty entities, text, links, images, and styling.



You can use site markers to point to pages so that later, if the page location changes, your links won't be broken.

Active Site Markers ▾	
✓ Name ↑ ▾	Page ▾
Access Denied	Access Denied
Event Feedback	---
Home	Home
Knowledge Base	---
Knowledge Base Search Results	---
Page Not Found	Page Not Found
Poll Archives	---
Profile	Profile
SchoolDetails	SchoolDetails
Search	Search

Learn More

- <https://www.xrmtoolbox.com> – Windows application that lets you do all sorts of customisation, administration, and configuration on top of Power Apps Portals.
- It's recommended that you find out more about ads, polls, portal languages, content snippets, etc.
- In Google, search 'site:*.powerappsportals.com' to see existing sites built with Power Apps Portals. Search 'site:*.microsoftcrmportals.com' for sites built with Dynamics 365 Portals.