



Extending Microsoft 365: Exploring the Art of the Possible

Julie Turner

Setting Expectations

- ✓ This is a 100 level INTRODUCTION to extending the Microsoft 365 SaaS platform
- ✓ We are NOT going to show/review code
- ✓ We are going to talk about all the ways to extend the platform, what tools you might need, and what scenarios you can accomplish
- ✓ *This is a firehose of information!*

Julie Turner

Partner/CTO

Working with SharePoint since 2007

Microsoft MVP, Office Apps and Services since 2017

Microsoft 365 PnP Team since 2019

Open-source project co-maintainer: PnPjs & hTWOo



<https://julieturner.net/me>



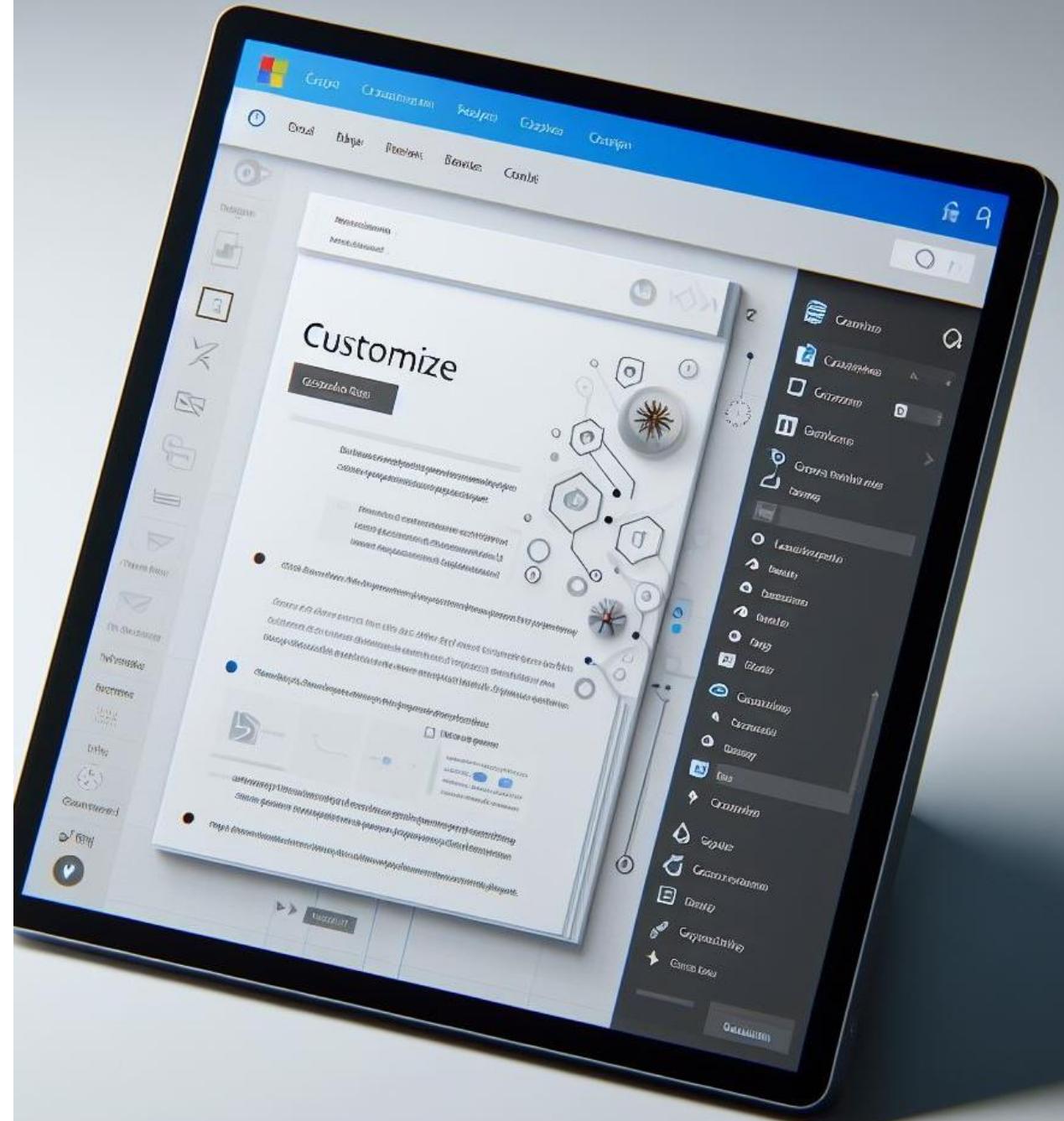
Agenda

- ❖ What is extensibility?
- ❖ Out of the Box Extensibility
- ❖ Low Code Extensibility
- ❖ Administration Tasks – Scripting
- ❖ Client-Side Extensions
- ❖ Server/Cloud Resources
- ❖ Authentication
- ❖ Design Frameworks
- ❖ Community Resources

What is extensibility?

Adjective
capable of being extended.

Extend - Verb
to increase in length, area, scope, etc.

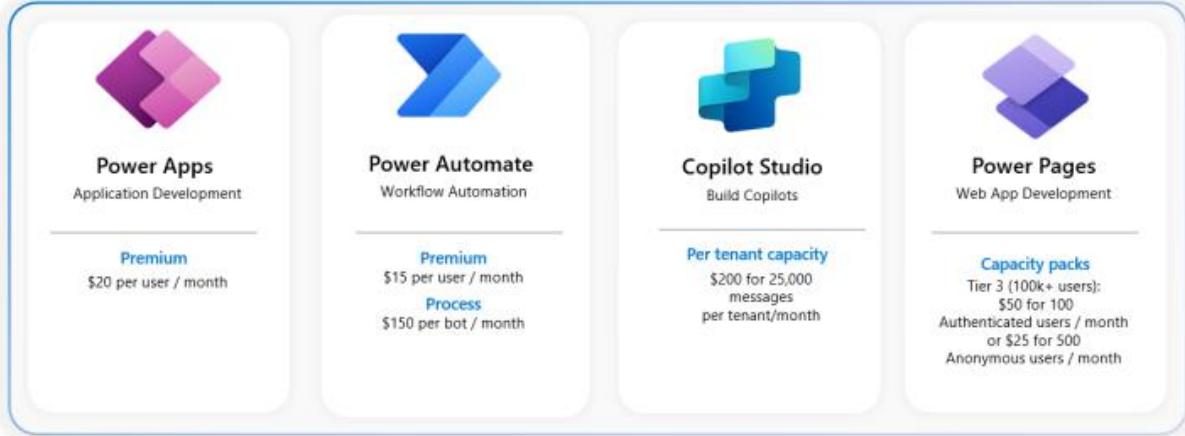


Out of the Box Extensibility

Templates

- ◆ Content Types and Document Templates
- ◆ Page Templates
- ◆ Site and List Templates
- ◆ Teams Templates
- ◆ List Formatting

Low Code Extensibility



PowerPlatform Connectors

Pair with CoPilot studio to allow others to create their own CoPilot experiences

LogicApps

- ◆ PowerAutomate base engine
- ◆ Standard/Consumption pricing
- ◆ Premium connectors billed differently

Pros

- ◆ Good for single entity business processes

Cons

- ◆ Cost to build/maintain + licensing (depending on connectors/platform)
- ◆ Complex logic can be hard to manage

Administration Tasks - Scripting

One-time or infrequently performed tasks

- ◆ PowerShell – Various Microsoft 365 Modules
- ◆ Command Line Interface
 - ◆ Azure CLI
- ◆ Community Tools
 - ◆ PnP PowerShell
 - ◆ CLI for Microsoft 365

Client-Side Extensions

- ◆ Browser Based
- ◆ Platform Hosted
- ◆ SPFx
- ◆ Self-Hosted
- ◆ Teams Apps
- ◆ Office Add-Ins (don't confuse with SharePoint Add-Ins)
- ◆ Native Mobile – iOS/Android
- ◆ Desktop Clients – iOS/Windows/Android

Tools – Client-Side Surfaces

- ◆ SharePoint
 - ◆ SPFx - SPPKG (Zip) files include code and manifest
 - ◆ Platform hosted (by default)
 - ◆ Supports SharePoint, Teams Tab/Personal App, Adaptive Card Extensions (ACE)
 - ◆ Teams
 - ◆ Teams Toolkit
 - ◆ Teams AI Library – add natural language capability
 - ◆ Self-hosted
 - ◆ Unified manifest for Microsoft 365 (Meta OS)
 - ◆ Office Add-Ins
 - ◆ Office Add-ins XML manifest
 - ◆ Unified manifest for Microsoft 365 (Meta OS)
 - ◆ Other
 - ◆ Microsoft365.com -> Unified Manifest (Meta OS)
 - ◆ Fluid Framework -> Azure Fluid Relay
 - ◆ LiveShare SDK – Leverages Fluid Framework and configured version of Azure Fluid Relay (Teams Meetings Only)
 - ◆ SharePoint Embedded – headless SharePoint storage with all the benefits of M365 functionality
- Delivery
- ◆ Microsoft App Source (App and Services listings)
 - ◆ Microsoft 365 – Integrated Apps (Teams, Microsoft365.com, Office Add-Ins Only)
 - ◆ SharePoint Store
 - ◆ SharePoint App Catalog (SPFx/Teams, Outlook Add-Ins)
 - ◆ Teams App Catalog (Teams)
 - ◆ Teams Side Loading (for development)

SPFx – Web Part

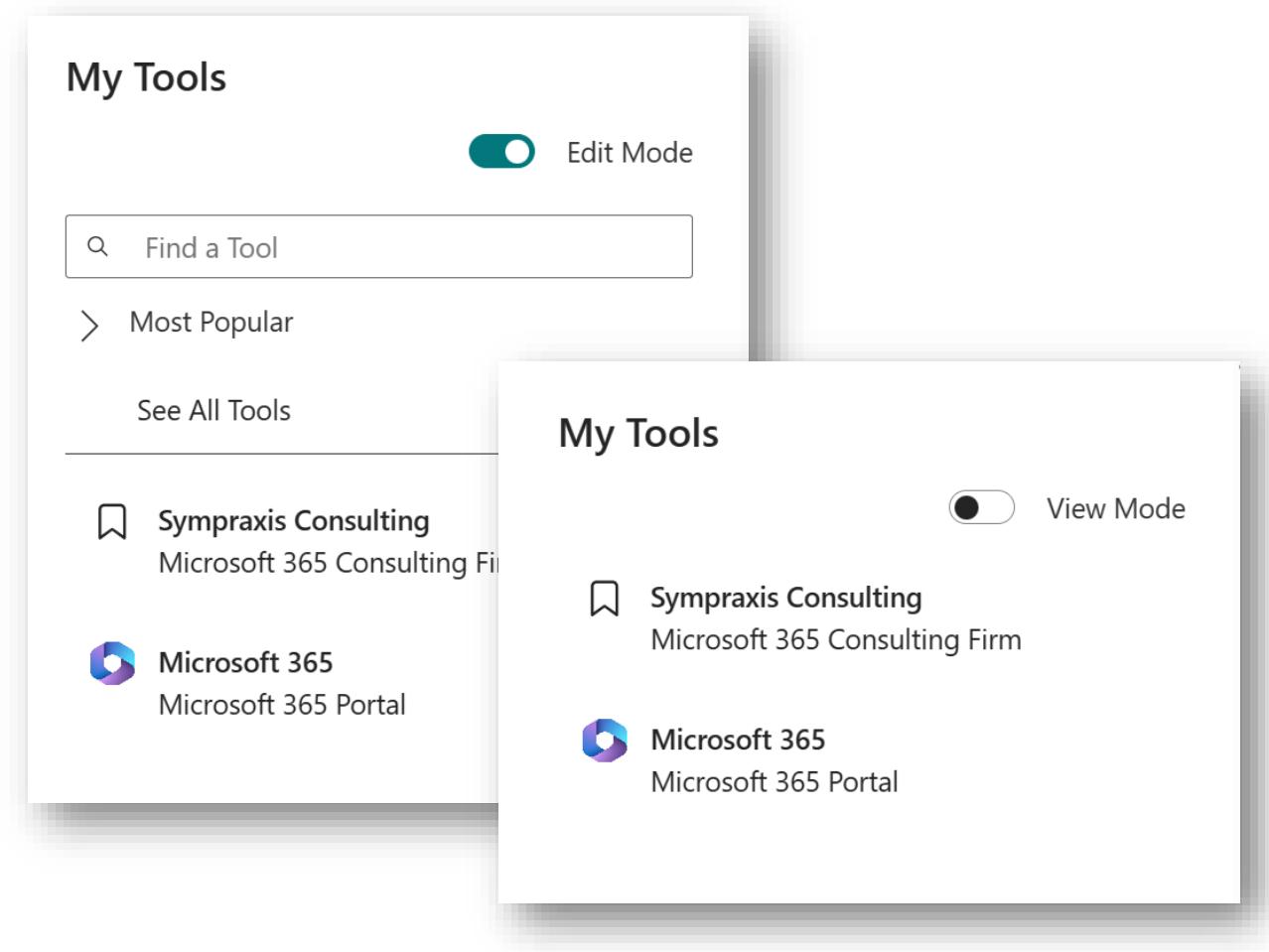
Create customization snippet that can be added (1->many times)

◆ DIV Object

- ◆ Surfaced anywhere in page that 1st party web parts can be added

◆ UX Scenarios

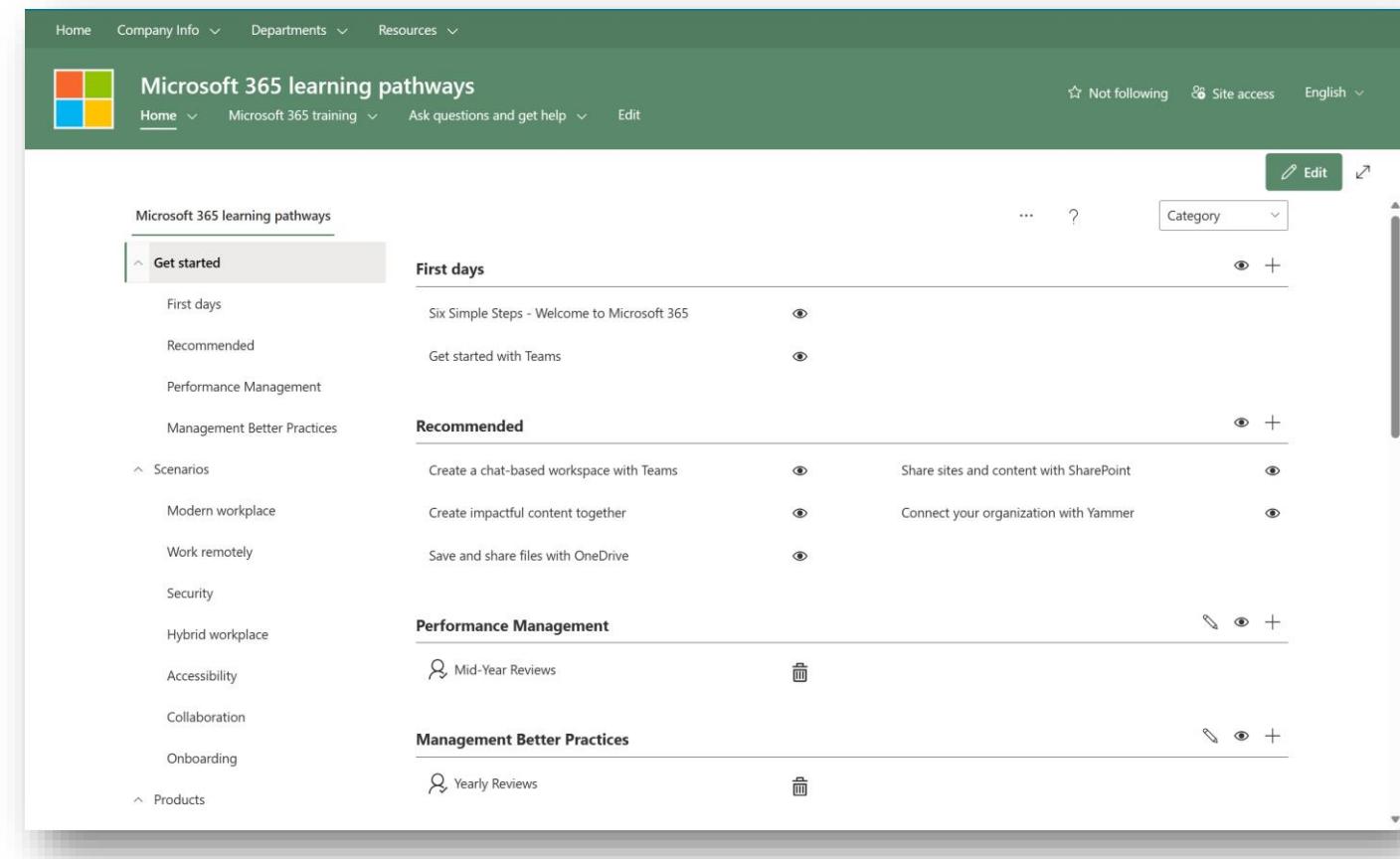
- ◆ In page context user tools
- ◆ Surface information from other sites
- ◆ Surface information from external sources
- ◆ Create mini-apps
- ◆ Can also surface as Teams Tab or Personal App



SPFx – Single App Part Pages (special web part)

Ability to create page with embedded full-page webpart in site.

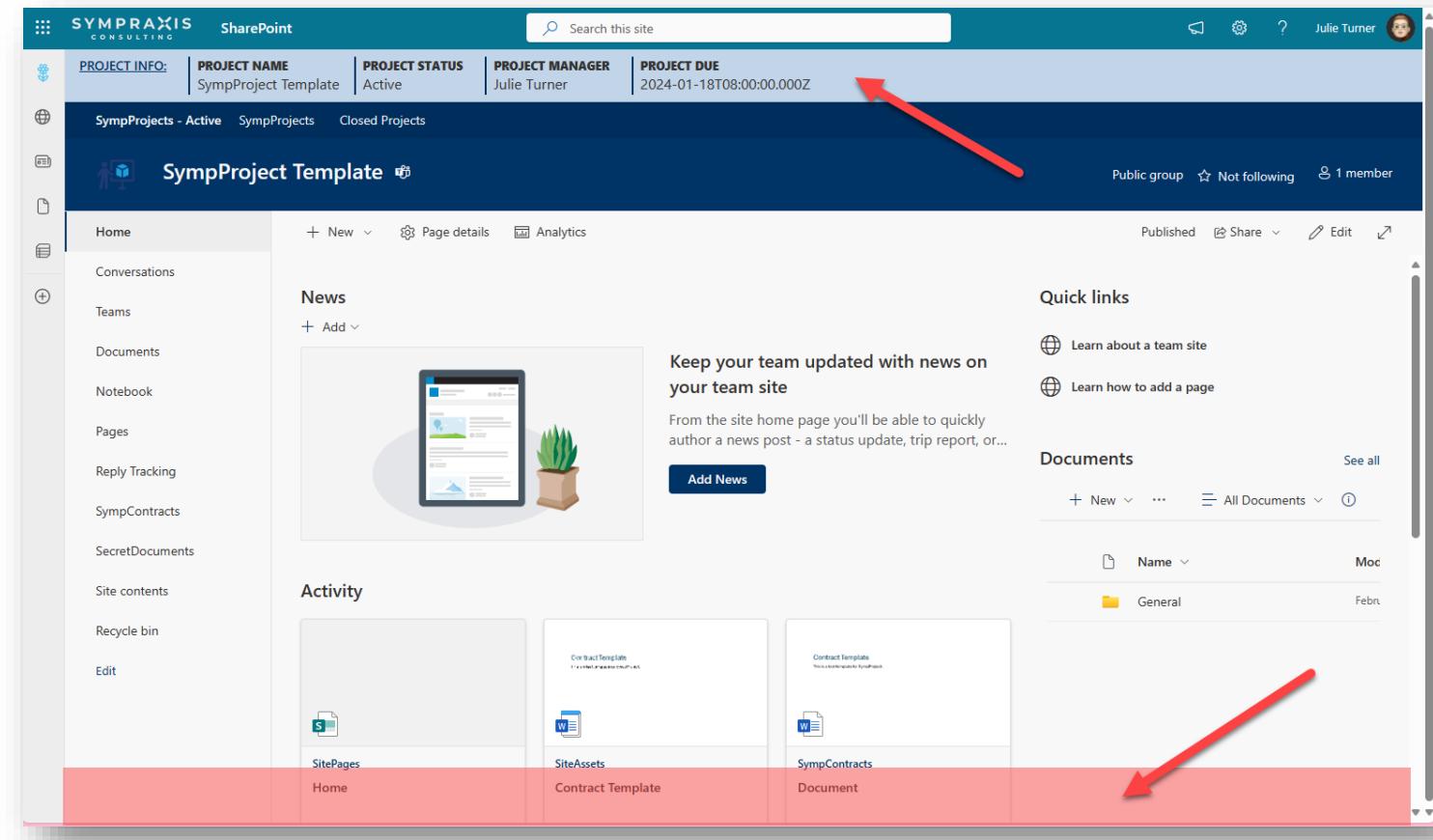
- ◆ DIV Object
 - ◆ Single column
 - ◆ No page toolbar
 - ◆ One web part
- ◆ UX Scenarios
 - ◆ Single Page Application
 - ◆ Full solution scenarios
 - ◆ Requires full page experience
- ◆ More UX: use an HTML modal dialog element



SPFx – Application Customizers

Runs on every page in the site/tenant unless you code it otherwise

- ◆ DIV object – in page
 - ◆ Header (between suite bar and hub navigation)
 - ◆ Footer (pinned to viewport)
- ◆ UX Scenarios
 - ◆ Site/Tenant specific information
 - ◆ Interactive Site-specific functionality
- ◆ Scenarios – No UX
 - ◆ Analytics
- ◆ More UX: use an HTML modal dialog element



SPFx – Field Customizers

Runs on every page in the site/tenant unless you code it otherwise

- ◆ DIV Object
 - ◆ Replace the rendering of the field, with or without the fields value
- ◆ UX Scenarios
 - ◆ Transform the value
 - ◆ Provide extra visualizations
- ◆ More UX: use an HTML modal dialog element

Organization Tools			
Tool Favorite Count	Tool Pinned Co...	Title	
1	2	Sympraxis Consulting	
4	5	Microsoft	
2	2	Microsoft 365	

Modified	Modified By	Actions	Process Stage
December 6, 2022	Julie Turner	<button>Take Action</button>	Test
February 3, 2023	Julie Turner		Initial

SPFx – Form Customizers

Customize the new, edit, view form for a SharePoint list

- ◆ Full Page Experience
- ◆ Build ANY logic you want
- ◆ Go extra mile and build custom security
- ◆ Scenarios UX
 - ◆ Complex data entry
 - ◆ Security scenarios
- ◆ More UX: use an HTML modal dialog element

The screenshot shows a SharePoint SympTools 'Edit' form for a 'Tool' item. The form fields include:

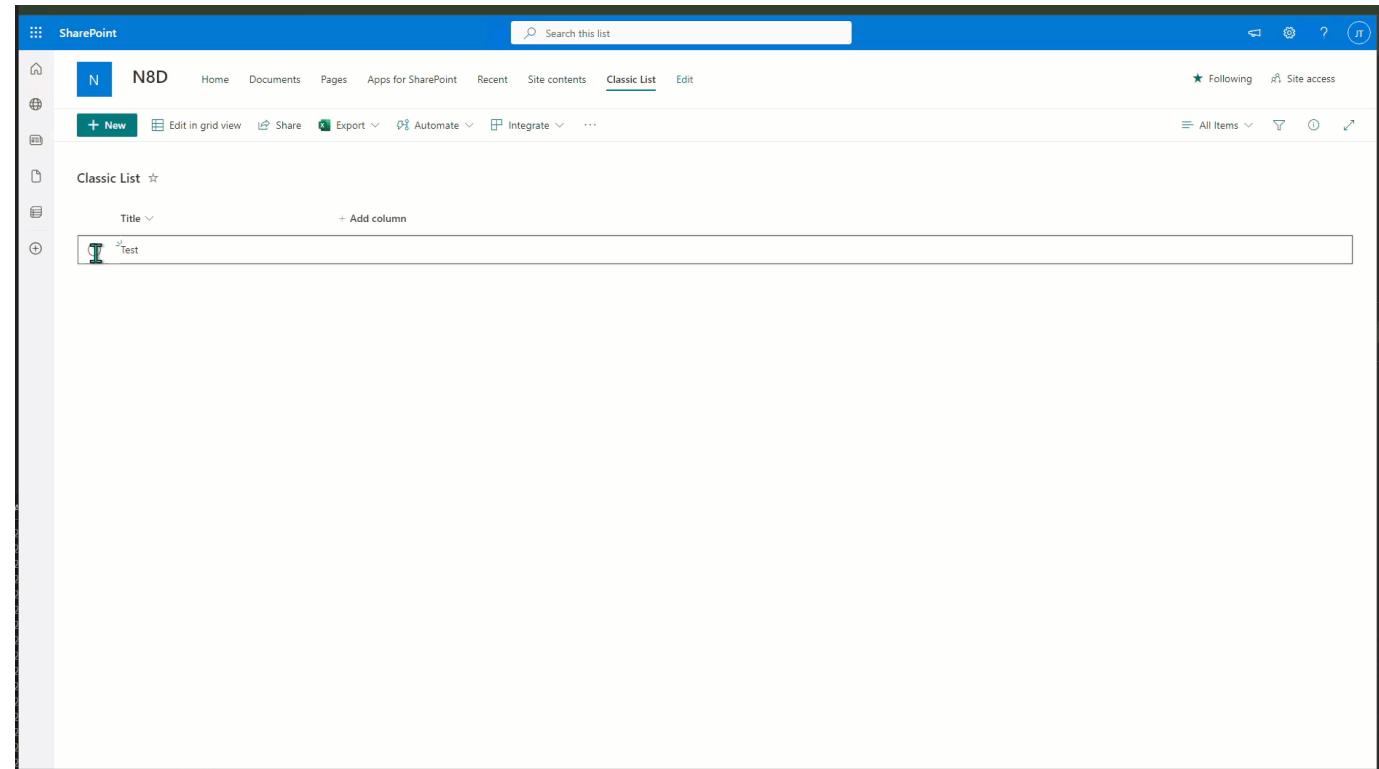
- Name: Sympraxis Consulting
- Description: Microsoft 365 Consulting Firm
- Tool Url: URL: https://sympraxisconsulting.com/
https://sympraxisconsulting.com/
- About Url: URL: Description
- Tool Keywords: consulting
- Tool Organization:
 - Sympraxis Global HQ
 - Sympraxis Mid-West
 - Sympraxis Iceland
 - Sympraxis North
 - Sympraxis South
- Tool Image or Icon:
 - Icon selected
 - Search for Icon
 - Current Selected Icon: magnifying glass
- Tool Owner: Julie Turner
- Backup Tool Owner: Emily Mancini
- Request Archive
- Tool Admins only
- Tool Status: Approved
- Favorite Count: 1
- Pinned Count: 2

At the bottom are Save and Cancel buttons.

SPFx – Command Set Customizers

Runs on every list/library in a site/tenant unless you code it otherwise

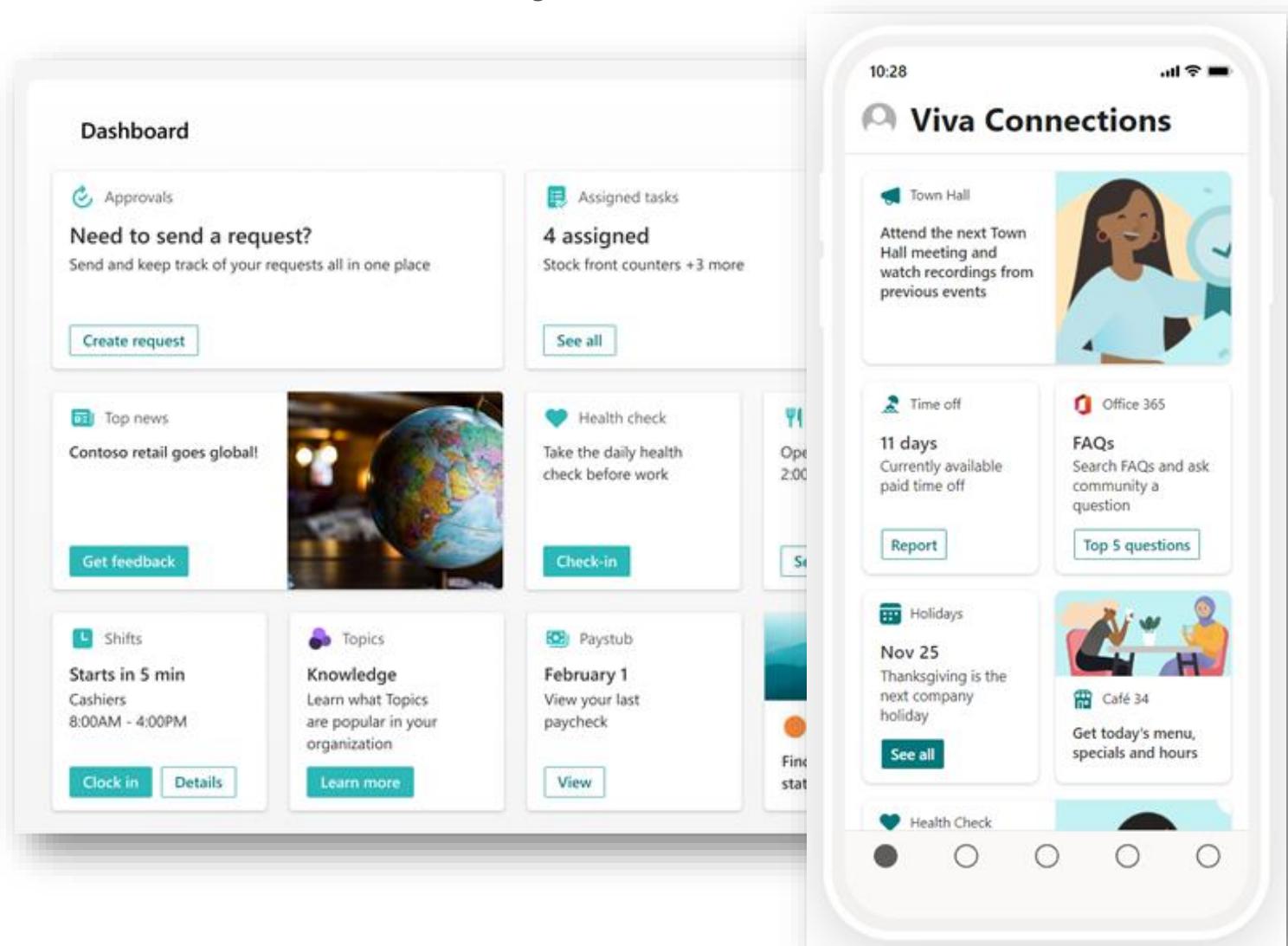
- ◆ DIV Object
 - ◆ In toolbar or quick launch menu
 - ◆ Provides “button” based on list item context
- ◆ UX Scenarios
 - ◆ Trigger actions for entire list
 - ◆ Trigger actions for selected item
 - ◆ Trigger actions for multiple items
- ◆ More UX: use an HTML modal dialog element



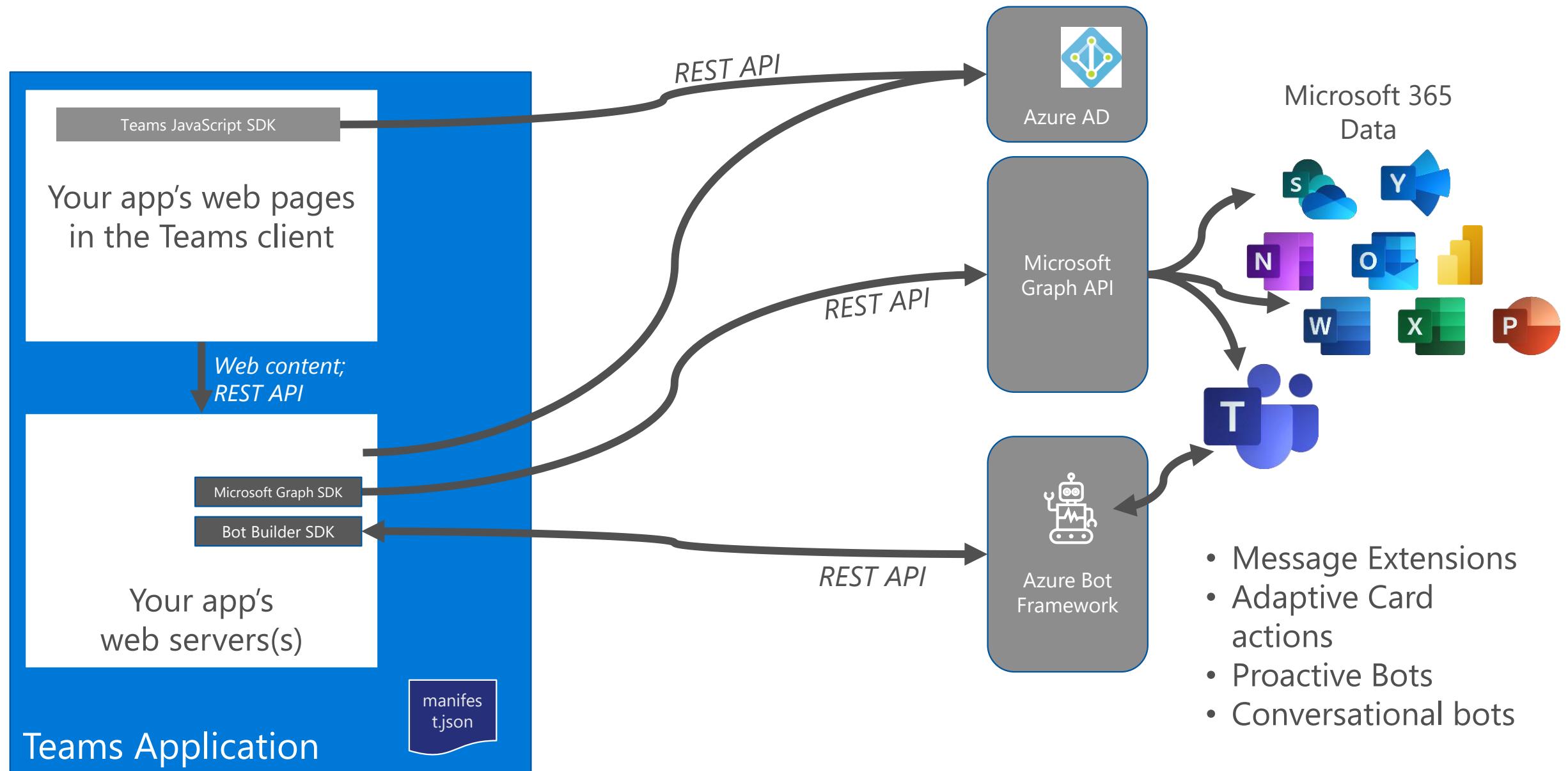
Viva Connections Adaptive Card Extension (ACE)

Optimized Adaptive Card based UX scenarios for job-related tasks

- ◆ Adaptive Card(s)
 - ◆ Med/Large Card View
 - ◆ Quick View “action card(s)”
- ◆ Optimized for mobile
 - ◆ “Home Site” Dashboard
 - ◆ Surfaced in Teams mobile
- ◆ UX Scenarios
 - ◆ User based notifications
 - ◆ Business Process tasks
 - ◆ Announcements

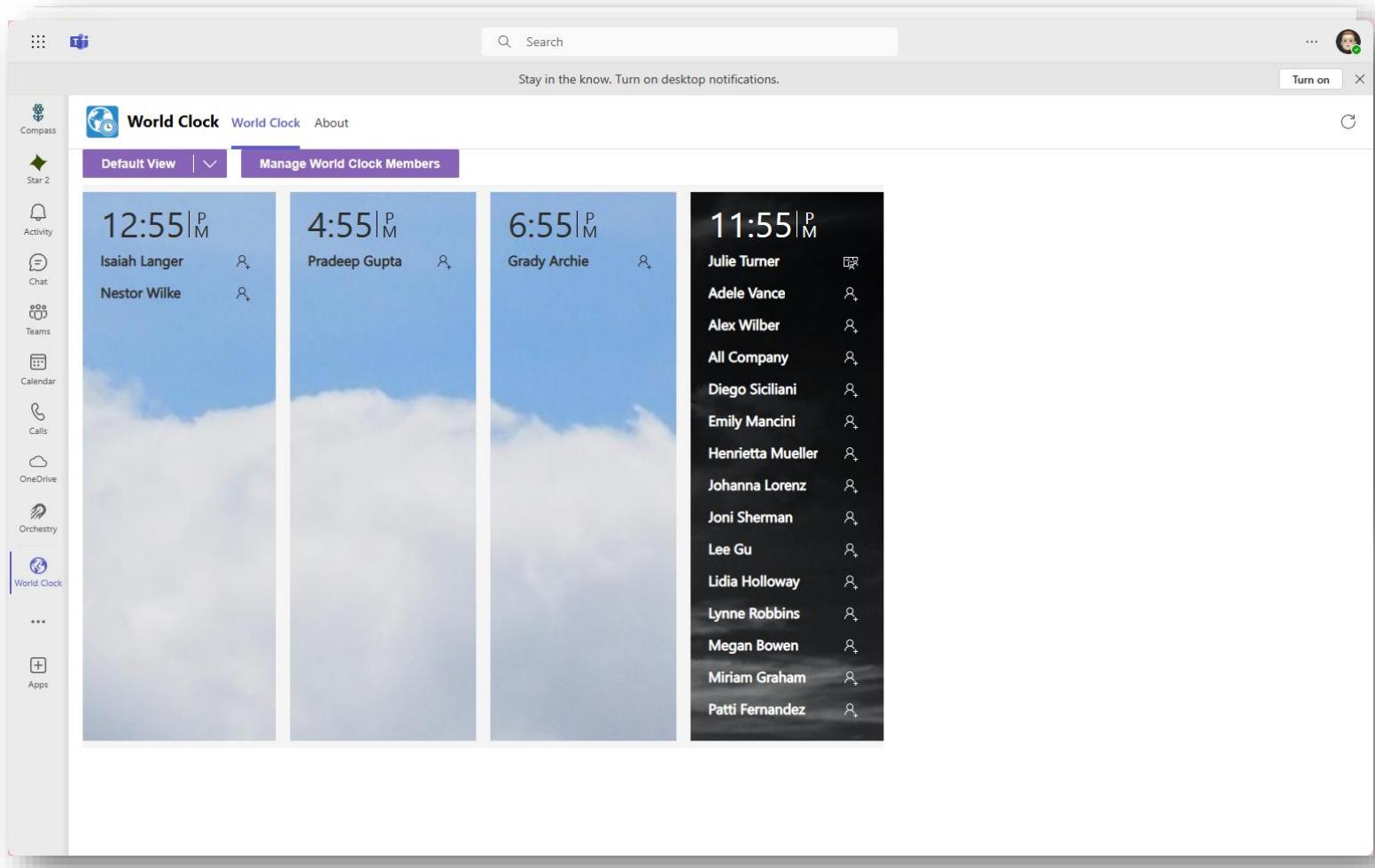


Teams Development



Teams - Tabs

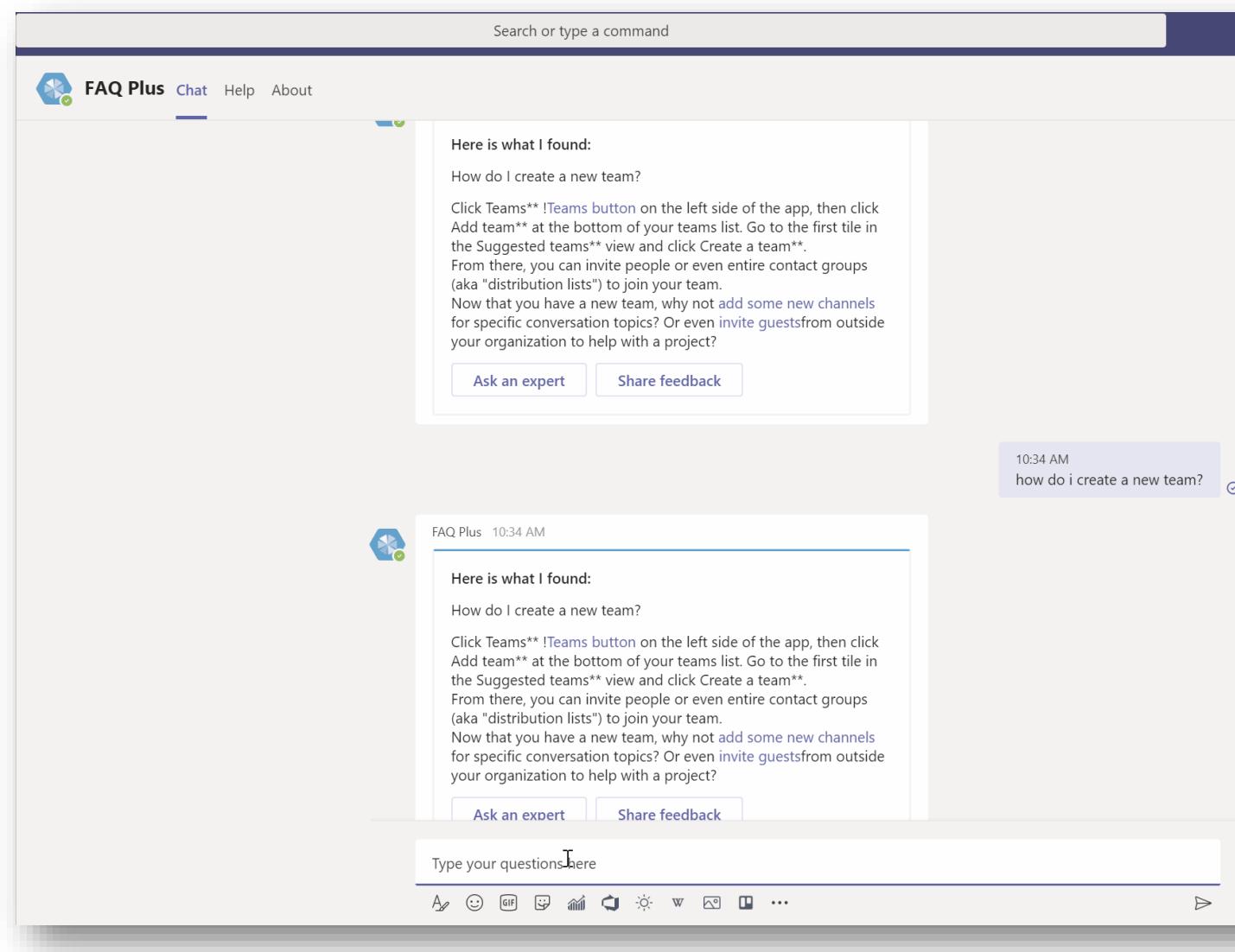
- ◆ iFrame your externally hosted web application in Teams
- ◆ Channel/Group Chat/Meetings Tabs are configurable
- ◆ Personal Tabs are not
- ◆ Can also be deployed to SharePoint



Teams - Bots

Also known as chatbot or conversational bot.

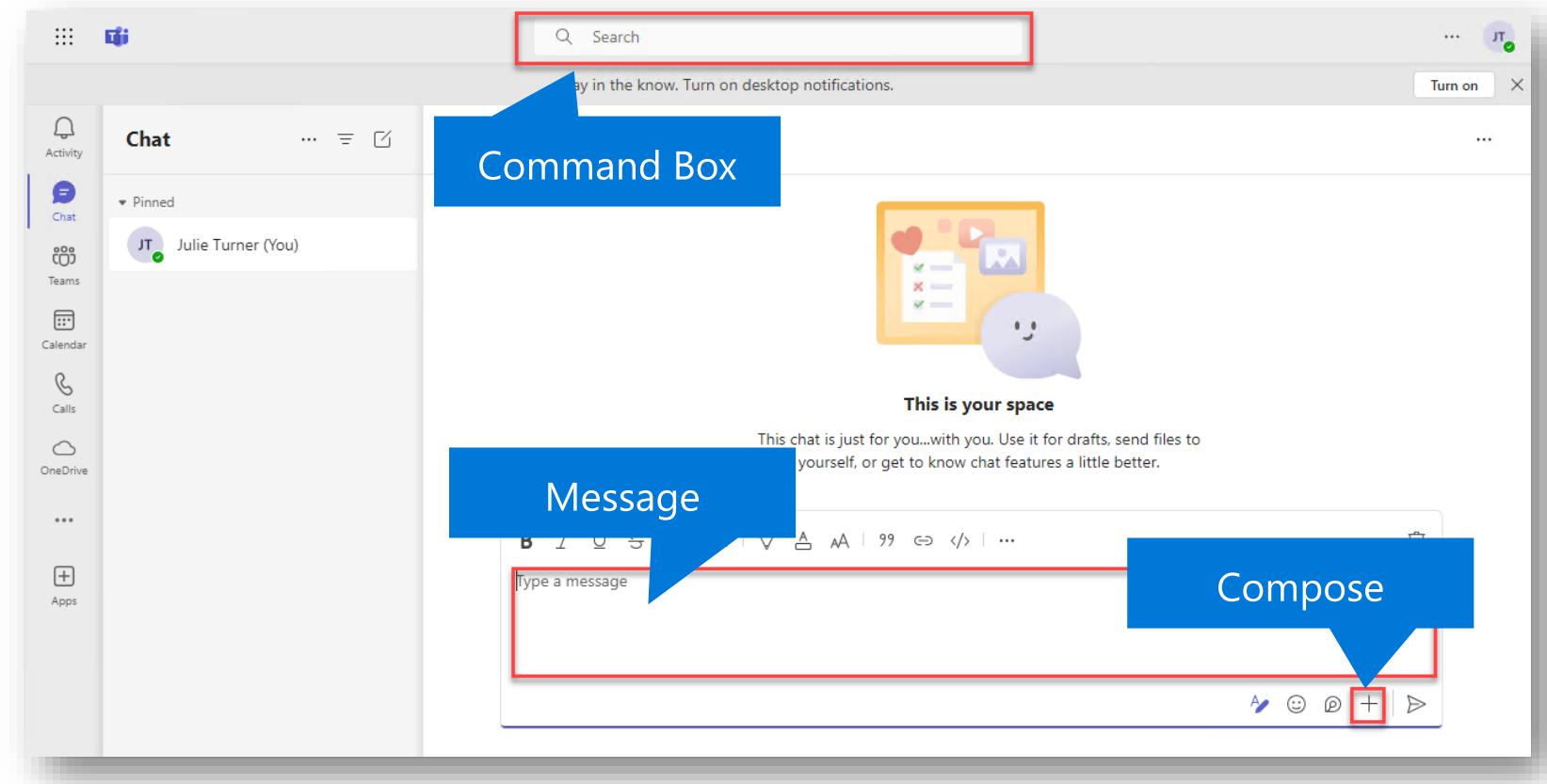
- ◆ Uses Bot Framework
 - ◆ Conversational bots allow users to interact with your web service using:
 - ◆ Text
 - ◆ Interactive cards
 - ◆ Dialogs
- (referred as task modules in TeamsJS v1.x)



Teams – Messaging Extensions

Interact with web service through buttons and forms within the Microsoft Teams client

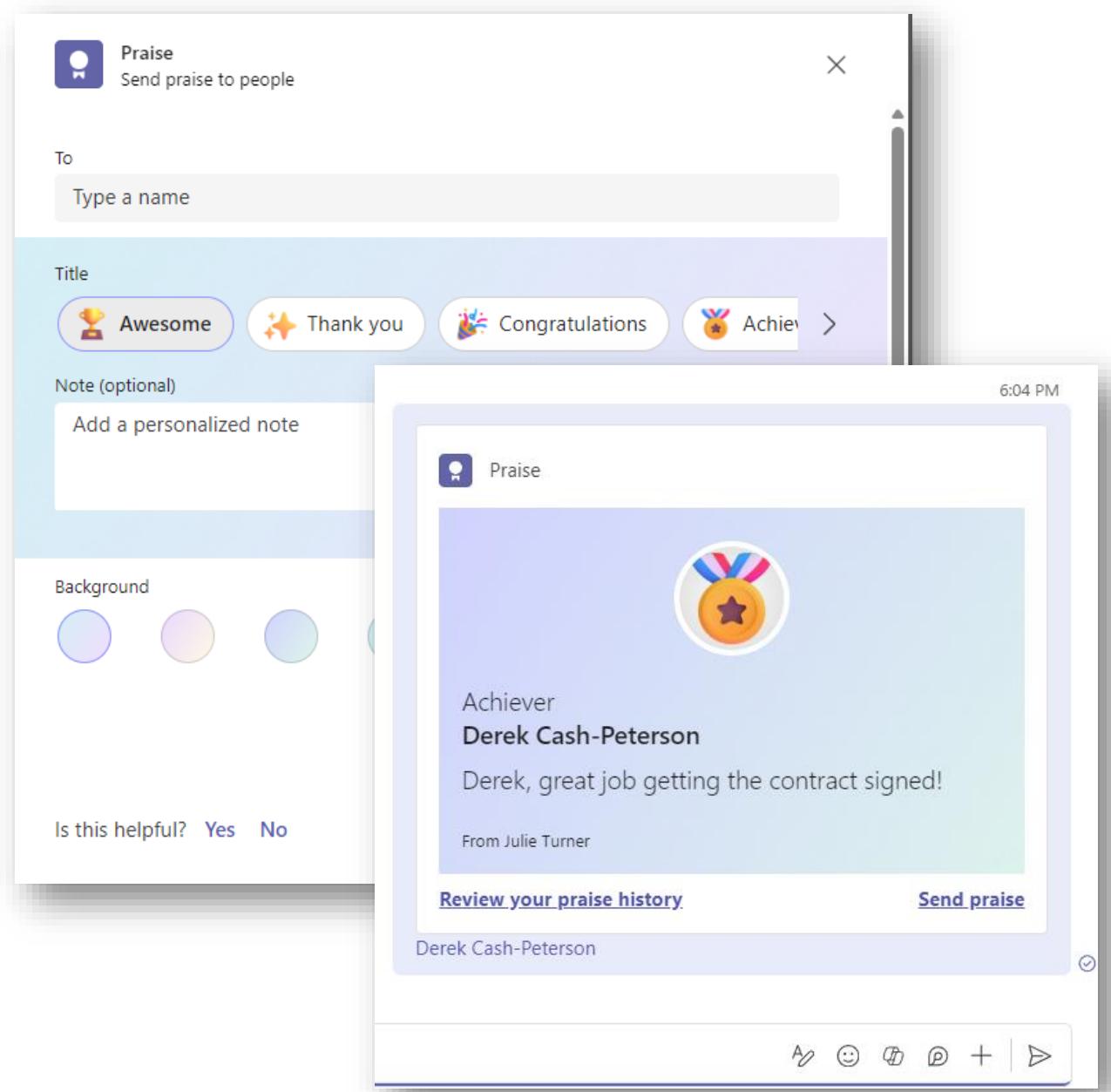
- ◆ Bot Framework
- ◆ API Message Ext
 - ◆ call your REST API directly.
 - ◆ limited functionality
- ◆ Actions
- ◆ Search
- ◆ Link Unfurling



Teams – Actions

Interact with user in chat/post UX

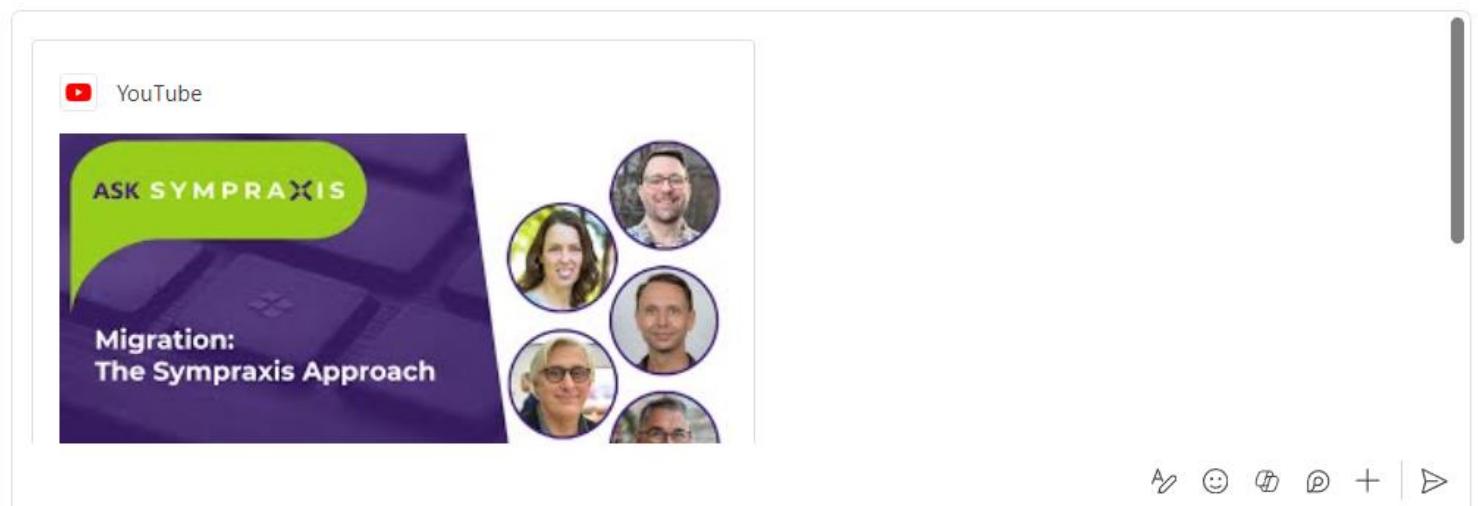
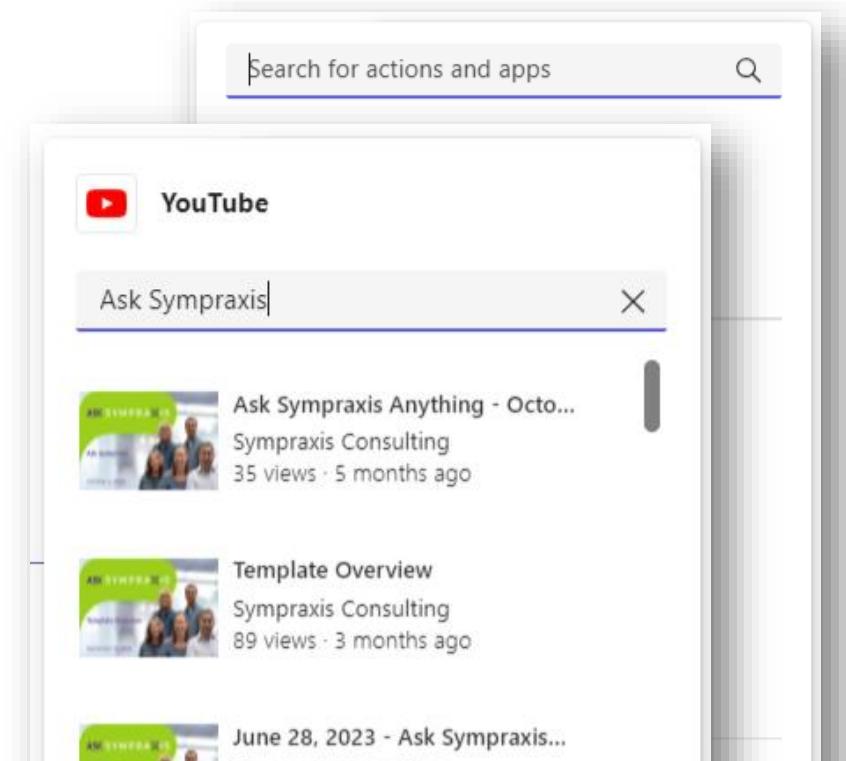
- ◆ Present the users with a modal pop-up to collect or display information
- ◆ Service responds by inserting a message into the conversation directly
- ◆ Multiple forms can be stitched together to behave like a “wizard”
- ◆ Invoke actions from compose message area, the command box, or inside the message (existing message sent to the service)



Teams – Search

Provide user with custom search results

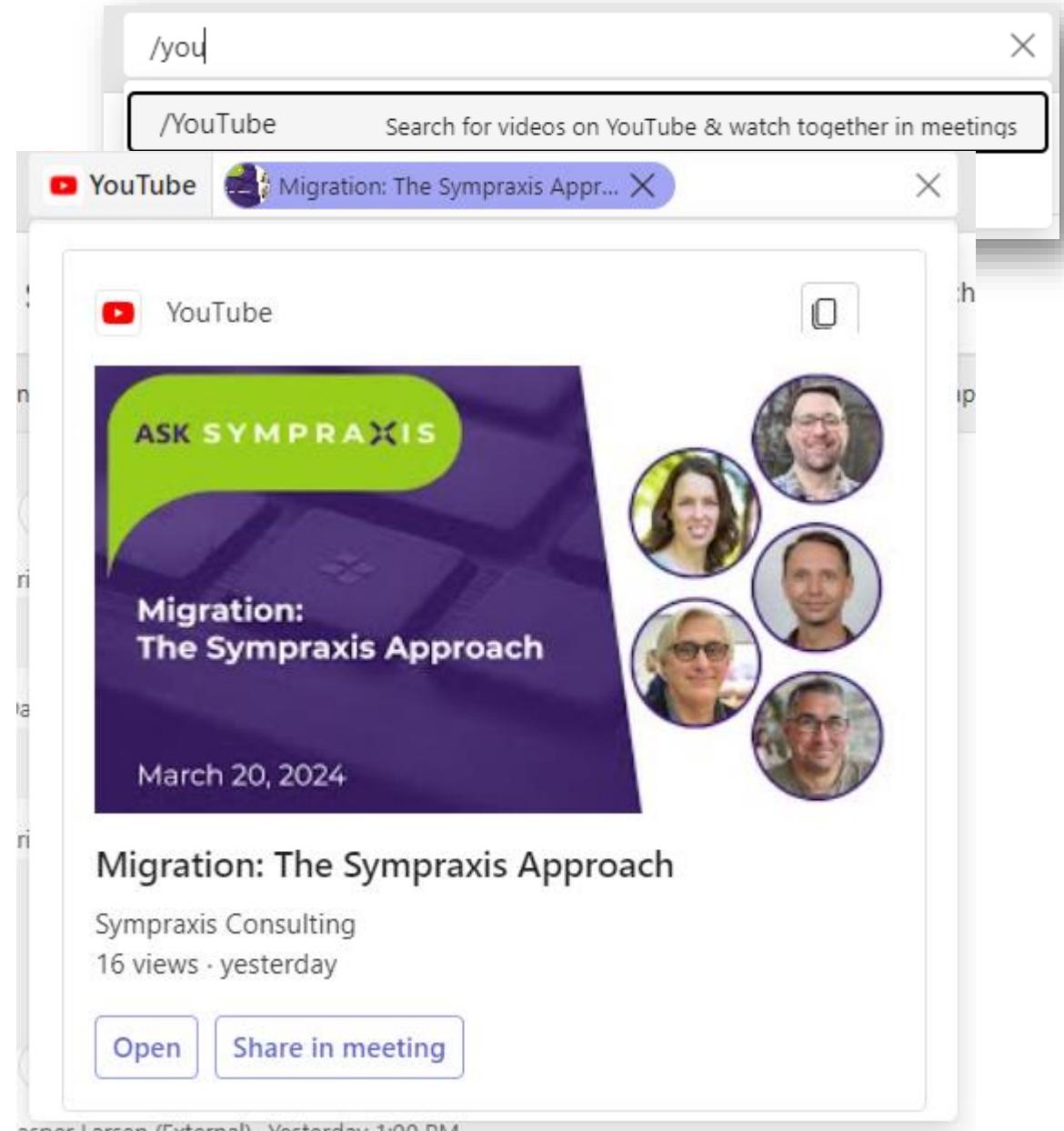
- ◆ Invoke message includes the search string submitted by the user
- ◆ Respond with a list of cards and card previews
- ◆ Trigger from command k or compose message bu not the message itself.



Teams – Search

Provide user with custom search results

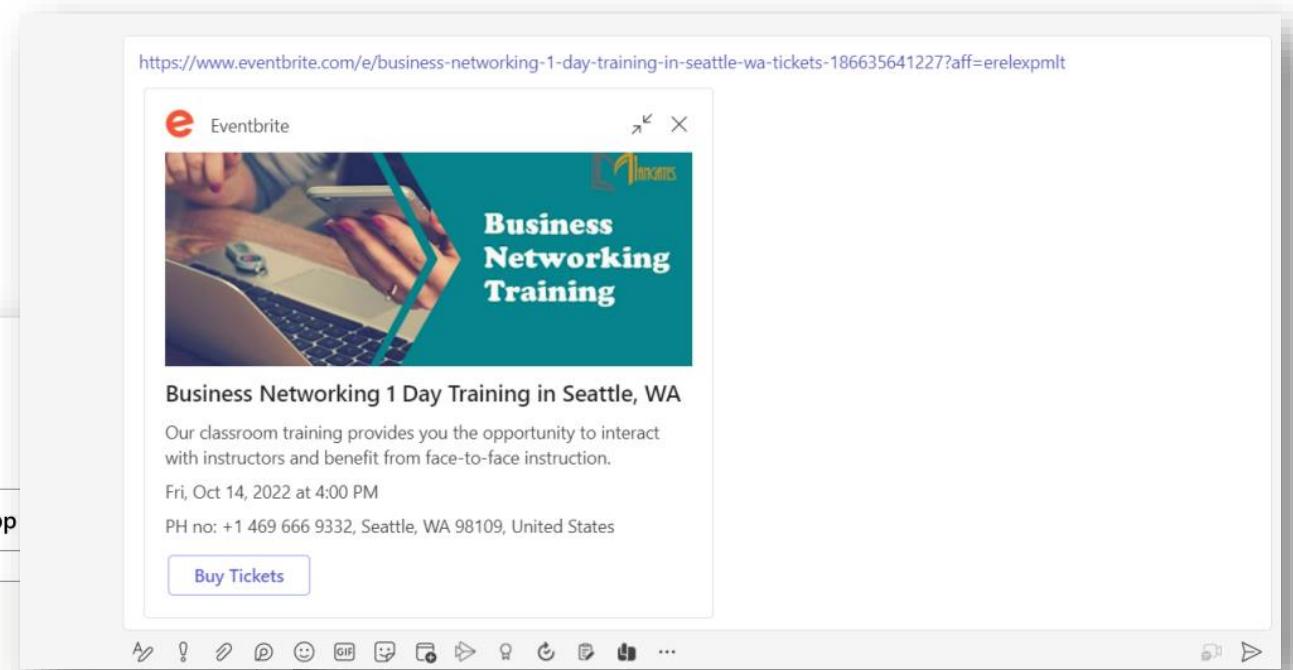
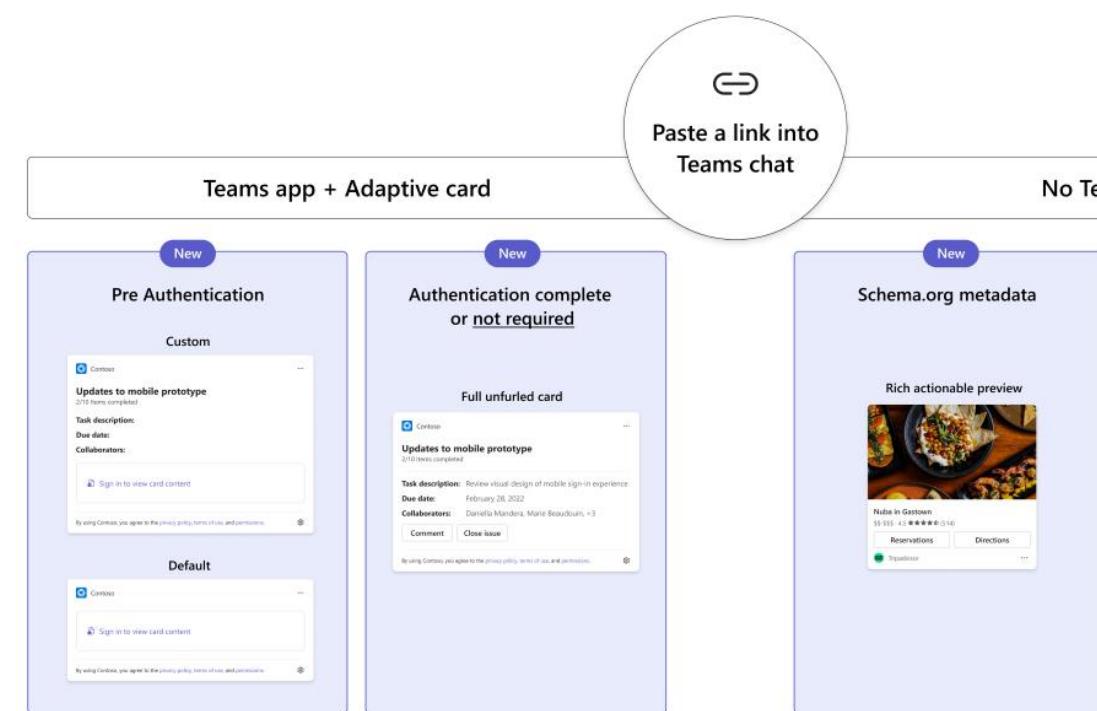
- ◆ invoke message includes the search string submitted by the user
- ◆ respond with a list of cards and card previews
- ◆ Trigger from command box or compose message but not the message itself.



Teams – Link Unfurling

Provide user with customized adaptive card in response to links with specific host.

Can be enabled without having to have the user install the customization

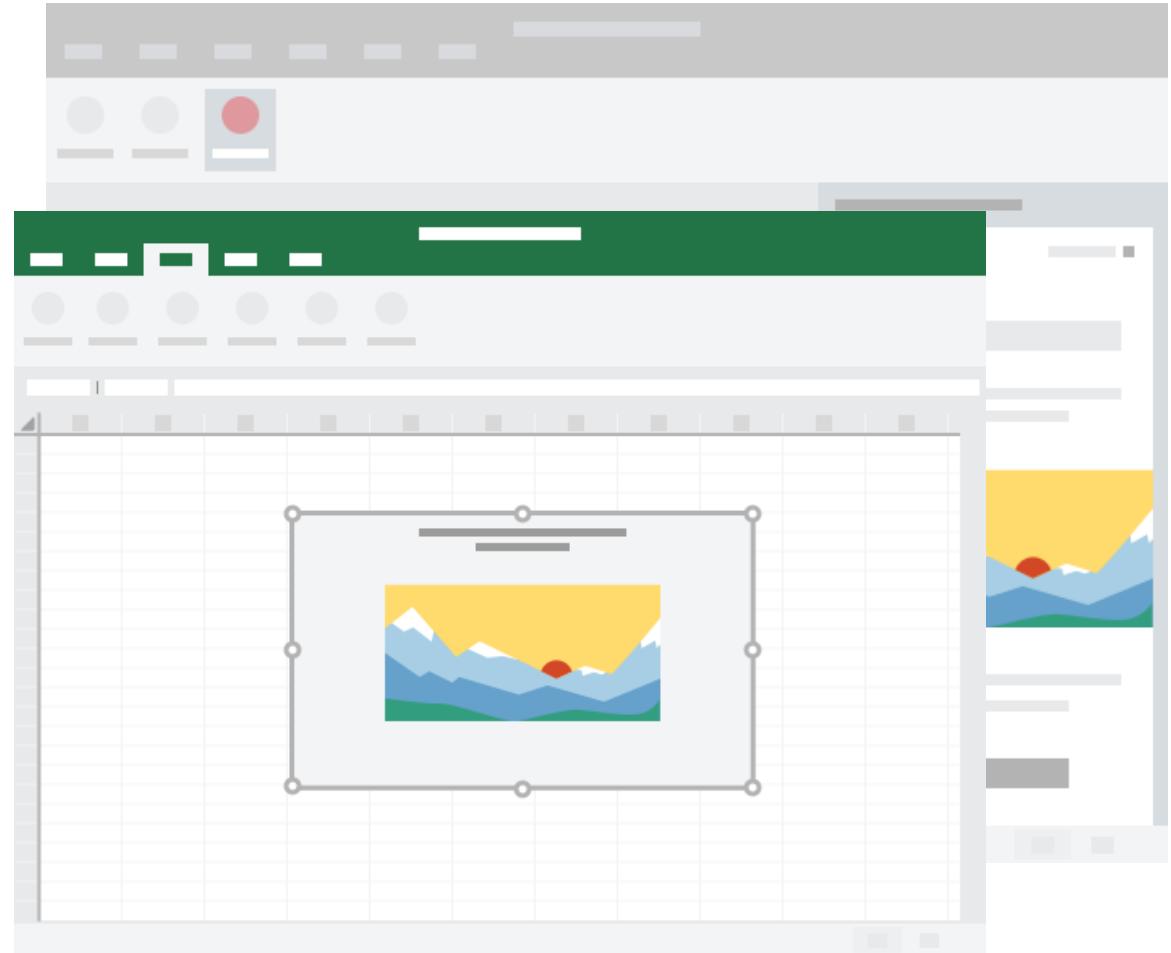


Works in desktop and mobile

Office Add-Ins

Extend Office Client functionality (Outlook, Word, Excel, PowerPoint, OneNote, Project, Visio)

- ◆ Custom ribbon buttons and menu commands
 - ◆ collectively called "add-in commands"
- ◆ Insertable task panes
- ◆ Embed Web-Based objects called "content add-ins" which can surface external content
 - ◆ web pages, videos, etc
 - ◆ Excel/PowerPoint only



Server/Cloud Extensions

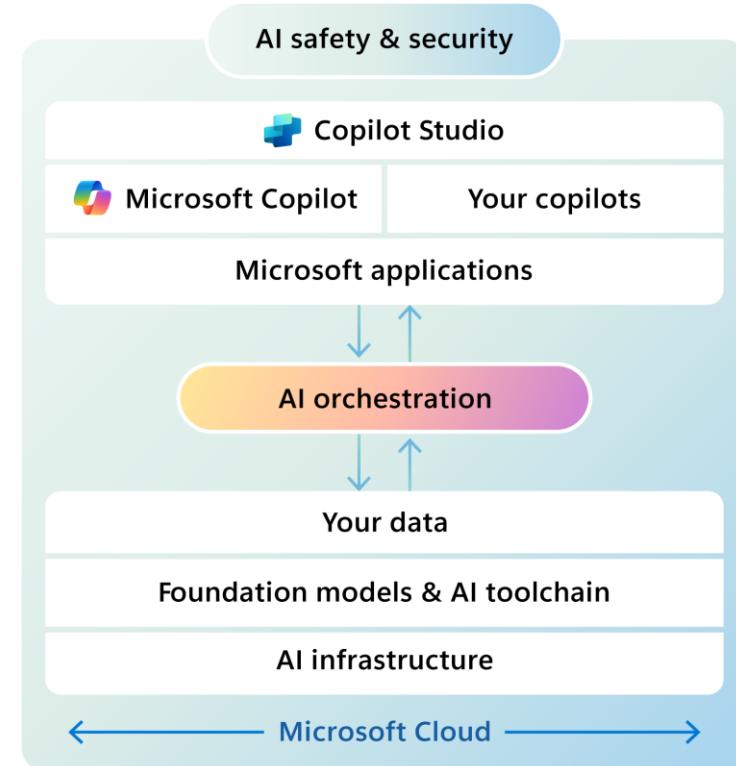
- ◆ Self-Hosted
 - ◆ Traditional On-Prem solutions
- ◆ Cloud-Hosted
 - ◆ Timer
 - ◆ Queues
 - ◆ HTTP Triggered

Tools - Cloud

Cloud based services often used when building extensions to Microsoft 365

- ◆ App Services
- ◆ App Insights
- ◆ Azure Storage
- ◆ Key Vault*
- ◆ Azure OpenAI Assistants
- ◆ Azure AI Studio

Copilot stack



Tools – Cloud – Data Sources

- ◆ SQL Server
- ◆ On-Premise Data Gateway
- ◆ Cosmos DB
- ◆ SharePoint Embedded
- ◆ Microsoft Graph APIs (SharePoint APIs)
- ◆ Microsoft Graph Connectors – extends Microsoft 365 search and CoPilot
- ◆ Triggers (Webhooks)
- ◆ 3rd Party APIs
- ◆ “Other Clouds”

Authentication

- ◆ Supported Flows
- ◆ OAUTH APIs
- ◆ MSAL
- ◆ Azure Identity
- ◆ Managed Identity

Authentication protocol	Authentication	Authorization	Multifactor Authentication	Conditional Access
Header-based authentication	✓	✓	✓	✓
LDAP authentication	✓			
Open Authorization (OAuth) 2.0 authentication	✓	✓	✓	✓
OIDC authentication	✓	✓	✓	✓
Password-based single sign-on (SSO) authentication	✓	✓	✓	✓
RADIUS authentication	✓		✓	✓
Remote Desktop Gateway services	✓	✓	✓	✓
Secure Shell (SSH)	✓		✓	✓
Security Assertion Markup Language (SAML) authentication	✓	✓	✓	✓
Windows Authentication - Kerberos Constrained Delegation	✓	✓	✓	✓

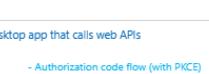
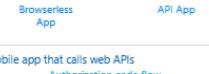
Supported Flows

- ◆ Authorization code
- ◆ Implicit
- ◆ Client credentials
- ◆ Device Code
- ◆ On-Behalf-Of
- ◆ Resource Owners
 Password Credentials

Scenario	Detailed scenario walk-through	OAuth 2.0 flow and grant	Audience
  	Authorization code flow (with PKCE)	Single-page app	Authorization code with PKCE
  	Implicit flow	Single-page app	Implicit
 	Web app that signs in users	Authorization code	Work or school accounts, personal accounts, and Azure AD B2C
  	Authorization code flow	Web app that calls web APIs	Authorization code
  	+ Authorization code flow (with PKCE) + Integrated Windows Authentication + Username/Password	Desktop app that calls web APIs	Interactive by using authorization code with PKCE
		Integrated Windows authentication	Work or school accounts
		Resource owner password	Work or school accounts and Azure AD B2C
  	Device Code Flow	Device code	Work or school accounts, personal accounts, but not Azure AD B2C
   iOS   	Authorization code flow (with PKCE)	Mobile app that calls web APIs	Interactive by using authorization code with PKCE
		Resource owner password	Work or school accounts, personal accounts, and Azure AD B2C
   	Client Credentials flow	Daemon app that calls web APIs	Client credentials
  	On behalf of flow	Web API that calls web APIs	On-behalf-of
			Work or school accounts and personal accounts

Supported Platforms/Languages

- ◆ .NET
- ◆ .NET Framework
- ◆ Java
- ◆ JavaScript
- ◆ macOS
- ◆ Native Android
- ◆ Native iOS
- ◆ Node.js
- ◆ Python
- ◆ Windows 10/UWP
- ◆ Xamarin.iOS
- ◆ Xamarin.Android

Scenario	Windows	Linux	Mac	iOS	Android
Single-page app Authorization code flow (with PKCE) 	JS MSAL.js	JS MSAL.js	JS MSAL.js	JS MSAL.js	JS MSAL.js
Single-page app Implicit flow 	JS MSAL.js	JS MSAL.js	JS MSAL.js	JS MSAL.js	JS MSAL.js
Web app that signs in users 	.NET Core Web app	node	node	node	node
Web app that calls web APIs Authorization code flow 	.NET Core Web app	ASP.NET Core + MSAL.NET MSAL Java			
Desktop app that calls web APIs - Authorization code flow (with PKCE) - Integrated Windows Authentication - Username/Password 	.NET Core Desktop App	MSAL.NET MSAL Java	MSAL.NET MSAL Java	MSAL.NET MSAL Java	MSAL.NET MSAL Java
Mobile app that calls web APIs Authorization code flow (with PKCE) 	iOS Mobile App	MSAL.NET	iOS MSAL.NET	iOS MSAL.NET	iOS MSAL.NET
Daemon app Client Credentials flow 	.NET Core Daemon Web app	MSAL.NET MSAL Java	MSAL.NET MSAL Java	MSAL.NET MSAL Java	MSAL.NET MSAL Java
Web API that calls web APIs On behalf of flow 	.NET Core API App	ASP.NET Core + MSAL.NET MSAL Java			

Microsoft Authentication Library (MSAL)

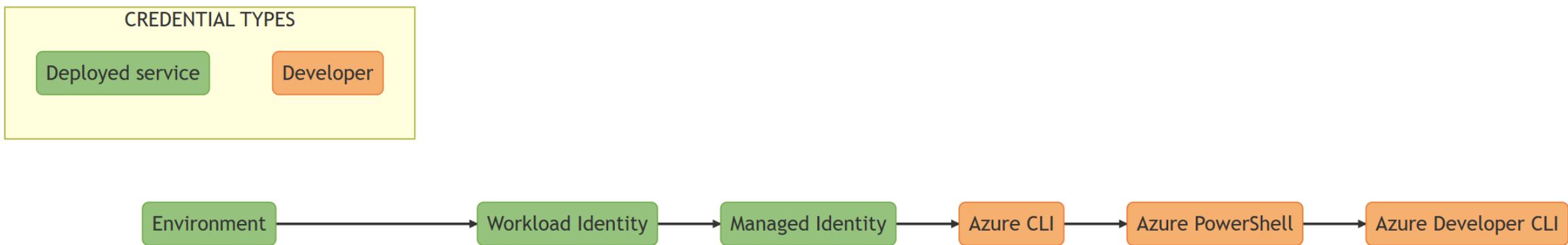
- ◆ Can acquire tokens on behalf of a user or application
 - ◆ when applicable to the platform
- ◆ Maintains a token cache for you
 - ◆ handles token refreshes when they're close to expiring
- ◆ Helps you specify which audience you want your application to sign in.
 - ◆ The sign in audience can include personal Microsoft accounts, social identities with Azure AD B2C organizations, work, school, or users in sovereign and national clouds.

MSAL Library	Supported platforms and frameworks
MSAL.NET	.NET Framework, .NET, Xamarin Android, Xamarin iOS, Universal Windows Platform
MSAL for Android	Android
MSAL Angular	Single-page apps with Angular and Angular.js frameworks
MSAL for iOS and macOS	iOS and macOS
MSAL Java	Windows, macOS, Linux
MSAL.js	JavaScript/TypeScript frameworks such as Vue.js, Ember.js, or Durandal.js
MSAL Node	Web apps with Express, desktop apps with Electron, Cross-platform console apps
MSAL Python	Windows, macOS, Linux
MSAL React	Single-page apps with React and React-based libraries (Next.js, Gatsby.js)
MSAL Go (Preview)	Windows, macOS, Linux

@Azure/Identity

DefaultAzureCredential

- ◆ Environment - account information specified via environment variables and use it to authenticate.
- ◆ Workload Identity - deployed to Azure Kubernetes Service with Managed Identity enabled
- ◆ Managed Identity - deployed to an Azure host with Managed Identity enabled
- ◆ Azure CLI - developer has authenticated an account via the Azure CLI `az login` command
- ◆ Azure PowerShell - developer has authenticated using the Azure PowerShell module `Connect-AzAccount` command
- ◆ Azure Developer CLI - developer has authenticated an account via the Azure Developer CLI `azd auth login` command



Managed Identity

Managed identities in Azure are a service that allows Azure resources to authenticate cloud services without the need for storing credentials in code or configuration files.

- ◆ You don't need to manage credentials.
 - ◆ Credentials aren't even accessible to you.
- ◆ You can use managed identities to authenticate to any resource that supports Microsoft Entra authentication, including your own applications.
- ◆ Managed identities can be used at no extra cost.
- ◆ System-assigned
 - ◆ 1:1 relationship with the azure resource and its lifecycle is tied to the resource
- ◆ User-assigned
 - ◆ 1:many relationship to azure resources
- ◆ You authorize a managed identity to have access to one or more services

Authorizing a managed identity

The screenshot shows the Azure portal interface for managing access control (IAM) on a resource named "Key vault".

On the left sidebar, the "Access control (IAM)" option is selected, indicated by a red arrow.

In the main content area, the "Add" button is highlighted with a red arrow, and the "Add role assignment" dialog is open.

The "Members" tab is selected in the "Add role assignment" dialog. The "Selected role" is set to "Reader".

The "Assign access to" section shows the "Managed identity" option selected, indicated by a red arrow.

The "Members" table shows a single row: "No members selected".

A modal window titled "Select managed identities" is displayed, listing two managed identities under "Subscription": "Function App (2)" and "func-...". A red arrow points to the "Subscription" dropdown.

```
m365 login --authType browser

m365 aad approleassignment add --appObjectId "1022615c-4433-4731-a933-53a9d2770e76" --resource "Microsoft Graph" --scopes "Files.ReadWrite.All,Group.Read.All,Mail.Send,User.Read.All"

m365 aad approleassignment add --appObjectId "1022615c-4433-4731-a933-53a9d2770e76" --resource "SharePoint" --scopes "Sites.FullControl.All"

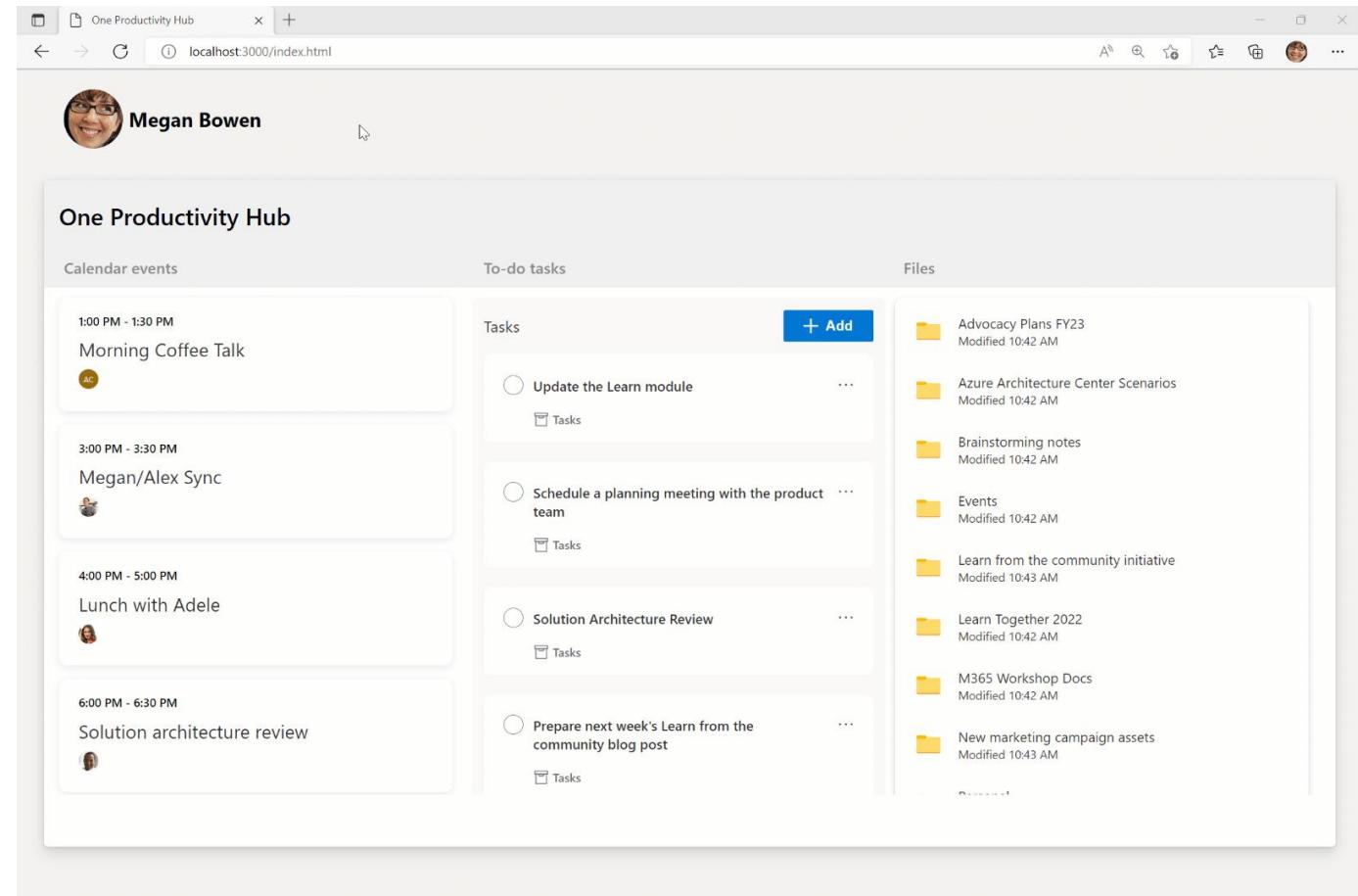
m365 aad approleassignment add --appObjectId "1022615c-4433-4731-a933-53a9d2770e76" --resource "SharePoint" --scopes "TermStore.ReadWrite.All"
```

Microsoft 365 Design Framework Options

- ◆ Fluent UI – FKA Office UI Fabric – FKA Northstar
 - ◆ CSS Classes and SASS mixins for colors, fonts, icons (font based), animations, grid
 - ◆ No HTML/CSS component library
 - ◆ Has REACT library – Uses CSS-in-JS
 - ◆ Technical debt upgrading between major versions-> [Fluent UI GitHub](#)
 - ◆ Figma Design Guides
- ◆ hTWOo – Open-Source Community Project
 - ◆ Pure HTML/CSS solutions – Fully A11y compliant
 - ◆ REACT component library – provides extendability to rootElementAttributes (and more)
 - ◆ Extendible Style-Guide built on Pattern Labs
 - ◆ SVG based Icon Library

Data Bound Component Options

- ◆ Microsoft Graph Toolkit Web Components
 - ◆ Uses FluentUI Web Components built with FAST
- ◆ PnP Reusable React Controls
 - ◆ Uses FluentUI v9
 - ◆ Version 3.x is bound to PnPjs v2.x which is no longer supported



Community Support

- ◆ Community Calls
- ◆ Videos
- ◆ Guidance
- ◆ Samples/Solutions
- ◆ SDKs
- ◆ Tools
- ◆ Extensions
- ◆ Forums

Microsoft 365 & Power Platform Community

Learn from others how to build apps on Microsoft 365 & Power Platform.

Don't reinvent the wheel. Focus on what truly matters for your organization.

Changing the world one contribution at a time!

[SEE INITIATIVES →](#)



Resources

<https://symp.info/m365-ext-links>