

© Copyright Microsoft Corporation. All rights reserved.

**FOR USE ONLY AS PART OF VIRTUAL TRAINING DAYS PROGRAM. THESE MATERIALS ARE NOT AUTHORIZED
FOR DISTRIBUTION, REPRODUCTION OR OTHER USE BY NON-MICROSOFT PARTIES.**

Collaborative Business Processes

Save time

gain efficiency in task completion

Streamline processes

remove errors from processes

Gain skills

continue to grow their skills and organizational impact

Be innovative

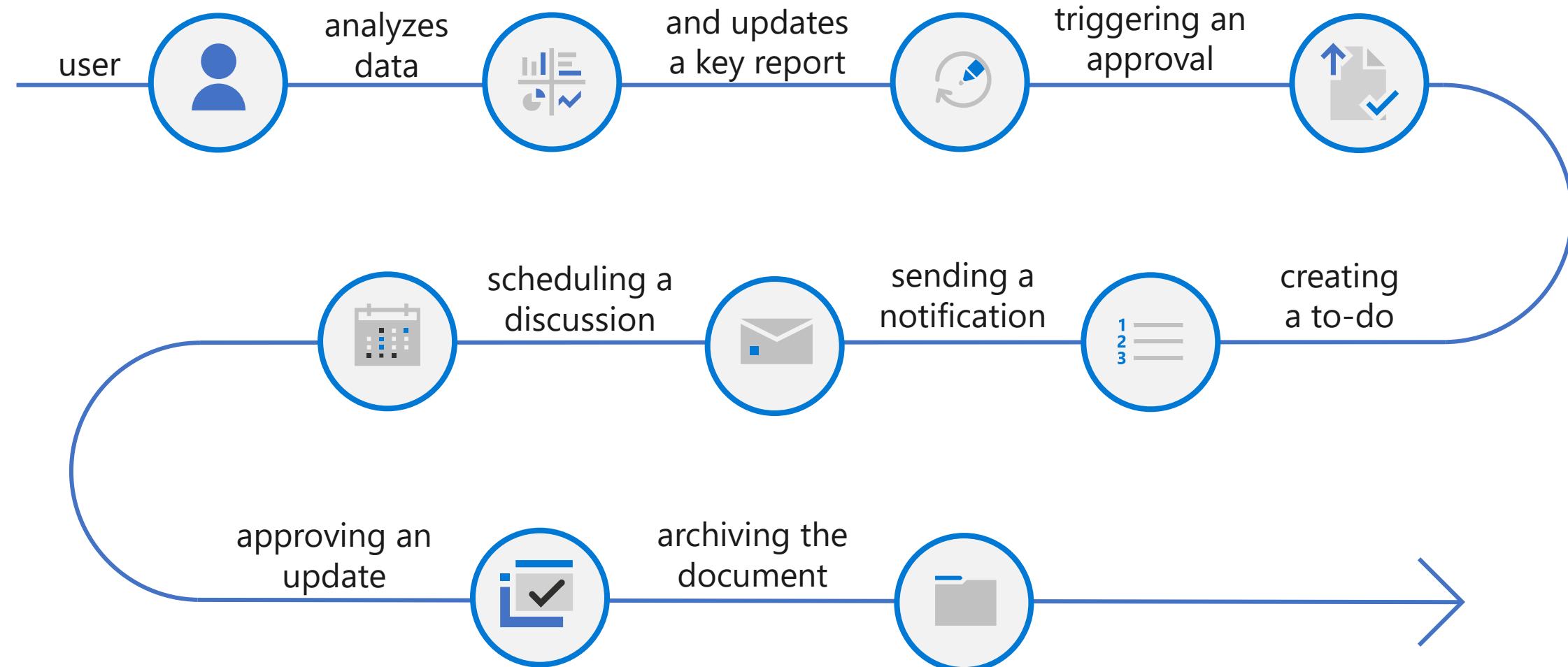
spend their time at work utilizing their creativity

Embed logic

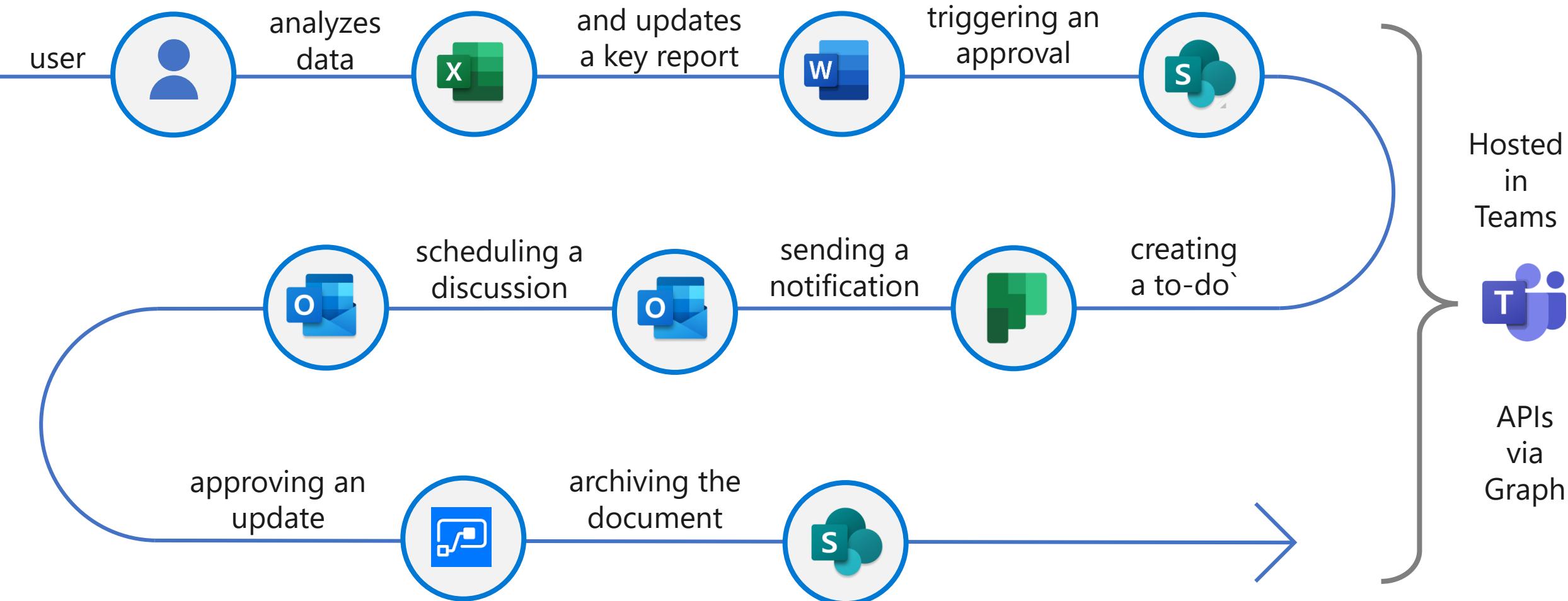
include business rules with content



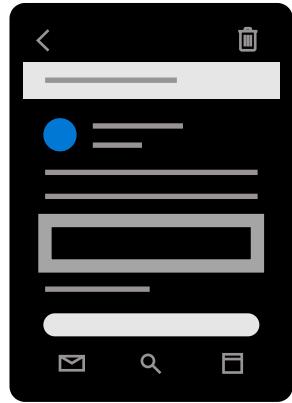
Microsoft 365 manages a wide range of everyday business processes



Microsoft 365 manages a wide range of everyday business processes



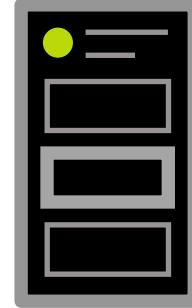
Extend every experience where work is done



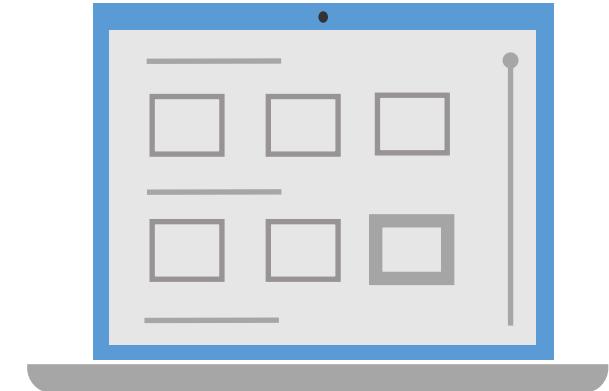
Outlook



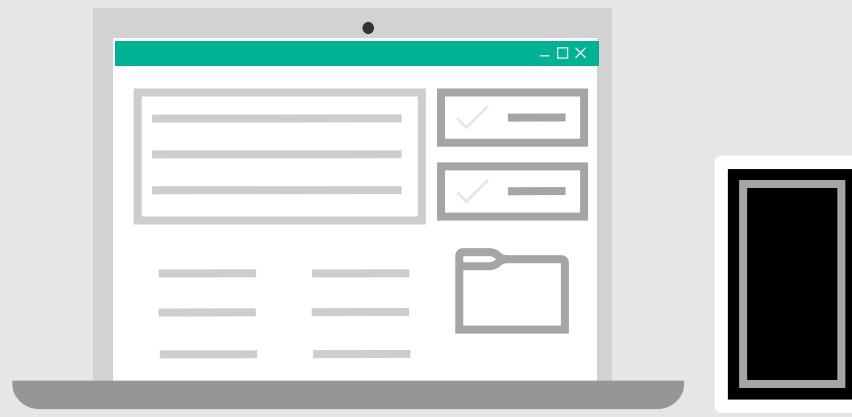
Teams



SharePoint



Windows apps



— Your applications —

Low code



Microsoft
Teams



Excel



SharePoint
& Power
Automate



Office
add-ins



Actionable
messages

Pro code

Implement Microsoft identity (Part 1)

- Getting started with Microsoft identity platform
- Register an Application



Overview of Microsoft identity

Introduction to Microsoft identity

Microsoft identity platform allows developers to build applications that sign in all Microsoft identities and get tokens to call Microsoft APIs such as Microsoft Graph, or APIs that developers have built.

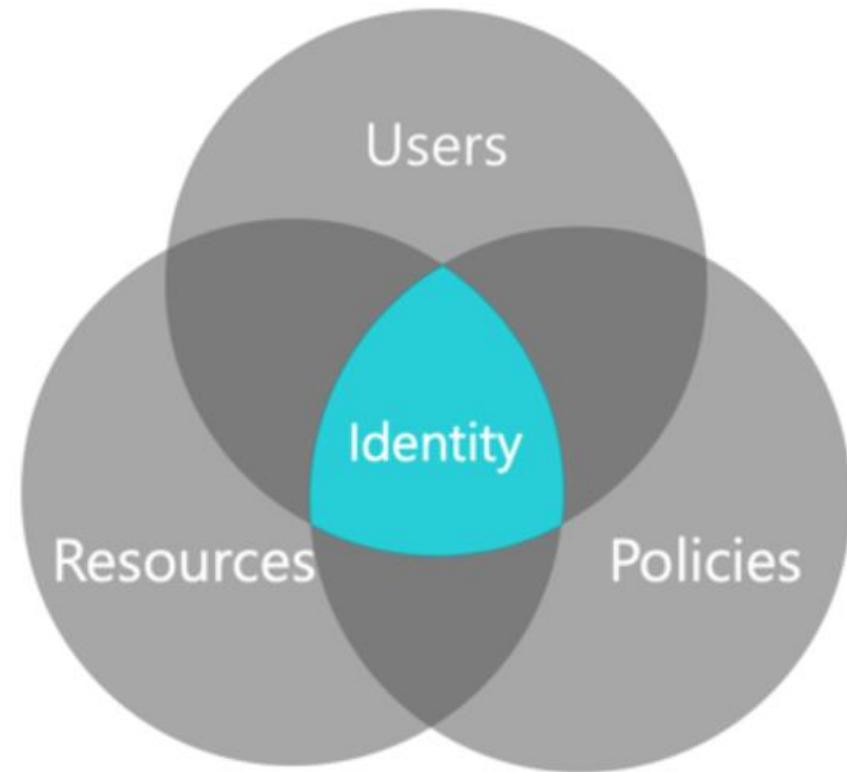
Overview of Microsoft identity

Users

- Azure AD – Employees and Enterprise Resources
- Azure AD B2B – Partners
- Azure AD B2C – Customers/citizens

Resources

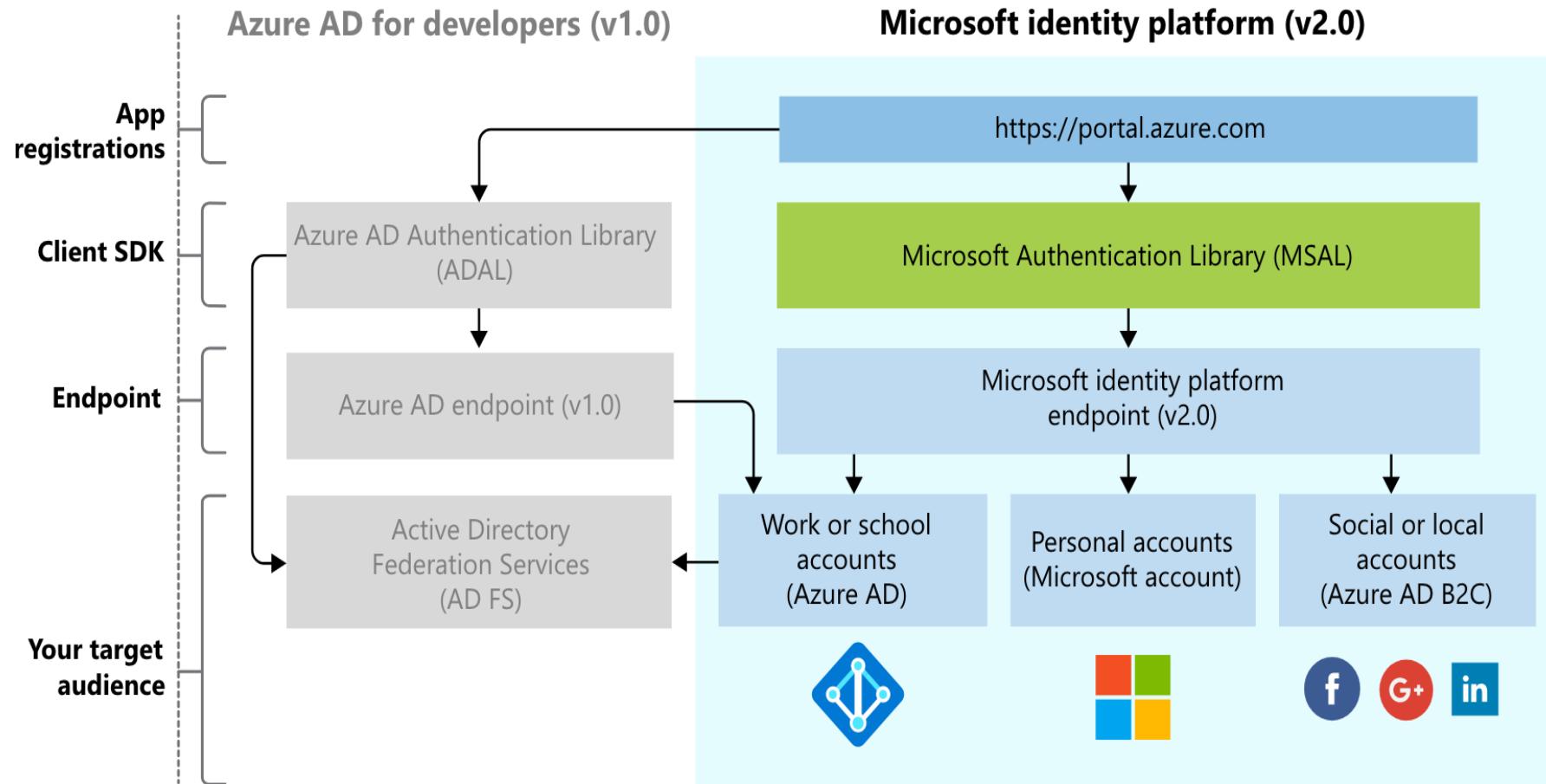
Policies



Overview of Microsoft identity platform

- Microsoft identity platform (v. 2.0) is an evolution of the Azure Active Directory v1.0 (Azure AD) developer platform.
- The Microsoft identity platform consists of:
 - OAuth 2.0 and OpenID Connect standard-compliant authentication service.
 - Open-source libraries.
 - Application management portal.
 - Application configuration API and PowerShell.
 - Developer content.

Overview of Microsoft identity platform (cont.)



Identity topology

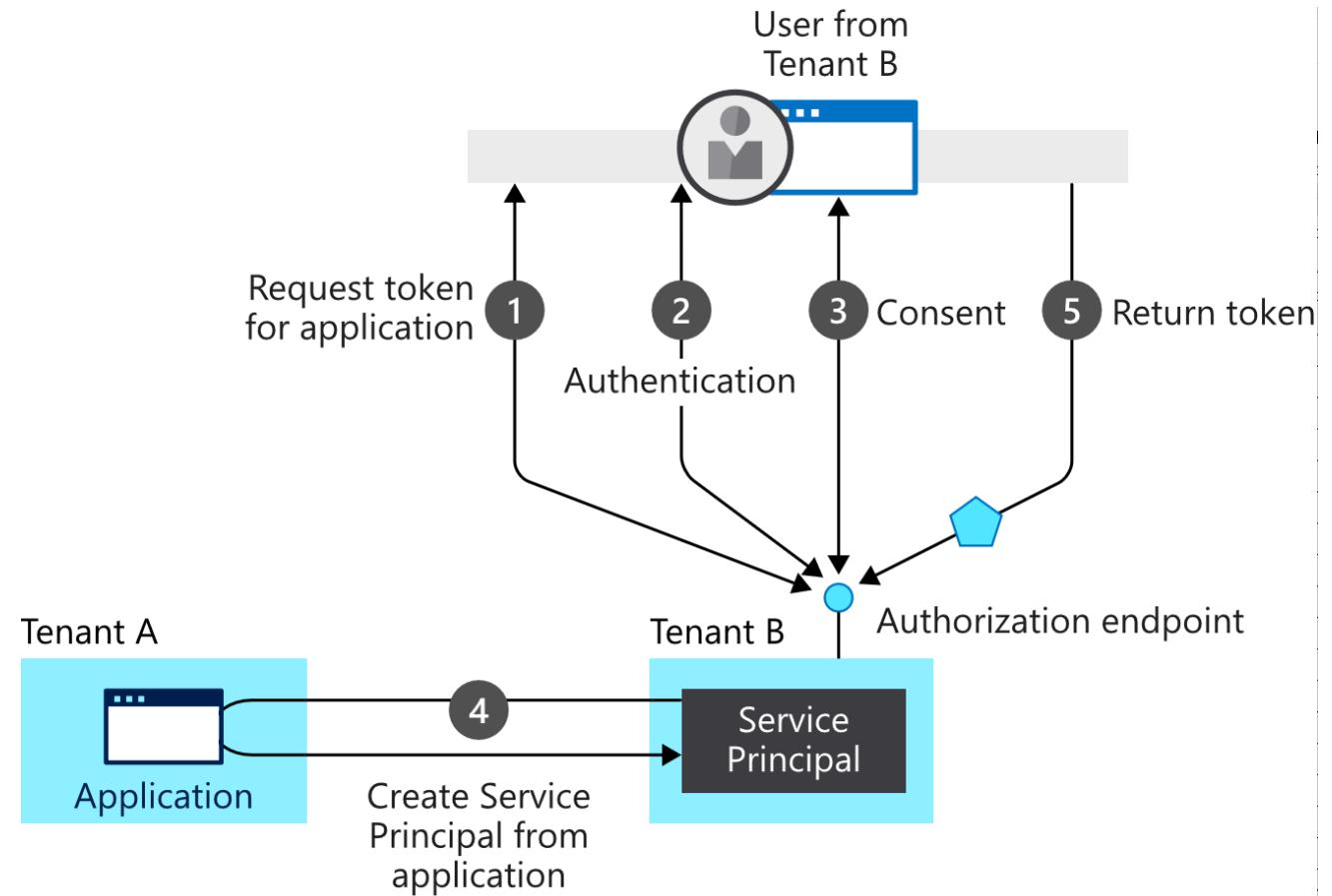
- Microsoft identity platform supports the following identity topologies:
 - Consumer
 - Enterprises
 - Business to Business (B2B)
 - Business to Customer (B2C)



Register an application

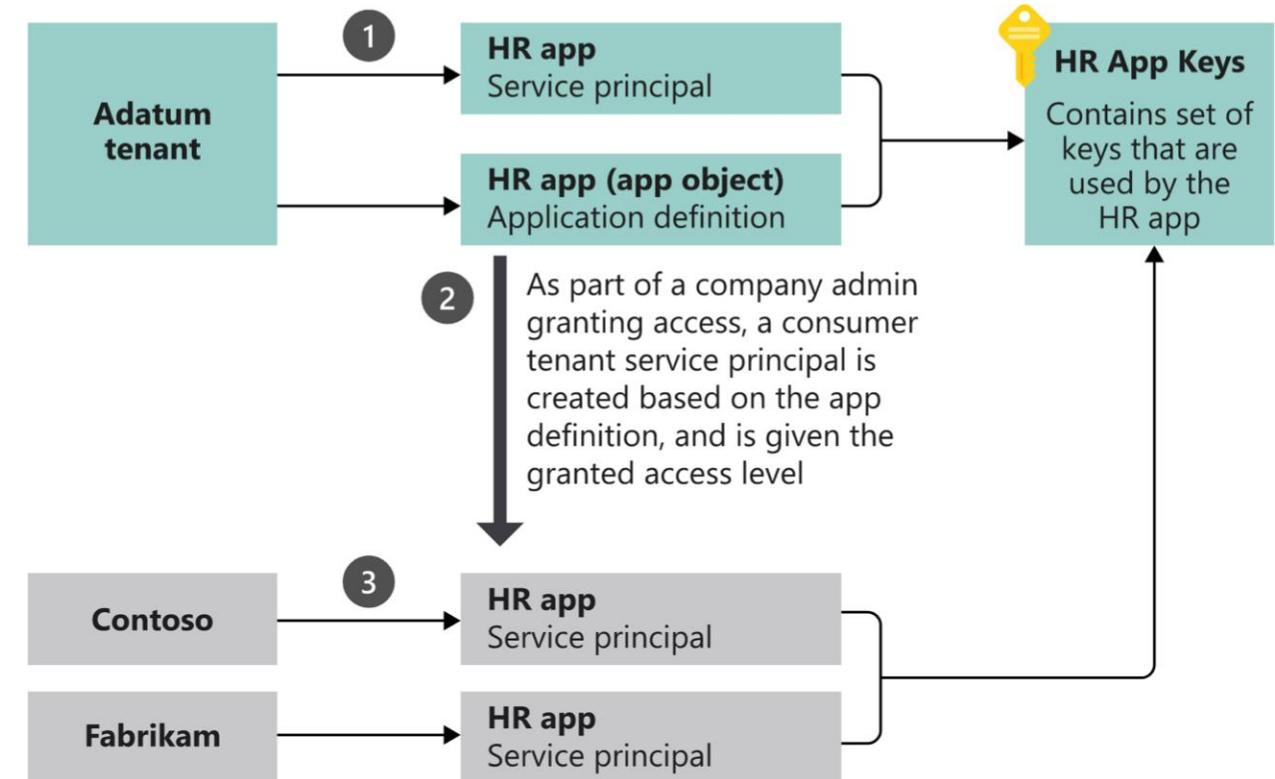
Overview of application model

- Authentication
- Authorization
- Application object
- Service principal
- Consent



Application and service principal objects

- Application object
 - The application properties stored in the manifest.
 - Defines the required application permissions.
- Service principal object
 - A copy of the application object used in non-home tenants to define application permissions.



Determine the account type

- Single organization
- Multiple organizations
- Personal Microsoft accounts

Home > App registrations > Register an application

Register an application

* Name
The user-facing display name for this application (this can be changed later).

ContosoApp_1 ✓

Supported account types
Who can use this application or access this API?

Accounts in this organizational directory only (Contoso Enterprises)
 Accounts in any organizational directory
 Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)

[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URL after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web https://contosoapp1/auth ✓

[Register](#)

Register an application

- Application name
 - Supported account types
 - Redirect URI (optional)

Register an application

*** Name**
The user-facing display name for this application (this can be changed later).

Supported account types
Who can use this application or access this API?

- Accounts in this organizational directory only (Contoso only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

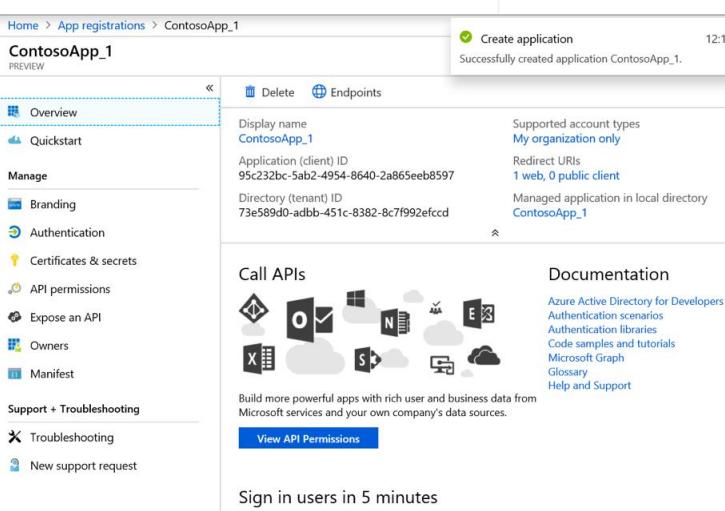
[Help me choose...](#)

Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. This can be changed later, but a value is required for most authentication scenarios.

Web e.g. <https://myapp.com/auth>

By proceeding, you agree to the Microsoft Platform Policies [\[?\]](#)

[Register](#)



Define app roles

Application roles are used to assign permissions to users.

- Azure AD App roles
- Azure AD security groups
- Application manager role

Configure Azure AD app manifest

Microsoft Azure | Search resources, services, and docs (G+/)

Home > Contoso - App registrations > Contoso App

Contoso App

Search (Cmd+/) Delete Endpoints

Overview Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Display name Contoso App
Application (client) ID ec78d011-e49e-4733-b69f-bd0763004190
Redirect URIs Add a Redirect URI
Directory (tenant) ID 56462a55-28d8-4735-aa26-33f8040d71a9
Object ID 6336904f-f188-4e43-a53c-3a8b1e9ee4b4
Supported account types My organization only
Application ID URI api://ec78d011-e49e-4733-b69f-bd0763004190
Managed application in local directory Contoso App

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

Call APIs Documentation

Microsoft identity platform
Authentication scenarios
Authentication libraries
Code samples
Microsoft Graph
Glossary
Help and Support

View API permissions

Sign in users in 5 minutes

JS .NET .NET Core Python Java Windows Android iOS macOS

Search (Cmd+/)

Overview Quickstart Manage Branding Authentication Certificates & secrets Token configuration (preview) API permissions Expose an API Owners Roles and administrators (Previous) Manifest

Troubleshooting New support request

Microsoft Azure | Search resources, services, and docs (G+/)

Home > Contoso - App registrations > Contoso App - Manifest

Contoso App - Manifest

Search (Cmd+/) Save Discard Upload Download

The editor below allows you to update this application by directly modifying its JSON representation. For more details, see: Understanding the Azure Active Directory application manifest.

```
1 "id": "6336904f-f188-4e43-a53c-3a8b1e9ee4b4",
2 "acceptMappedClaims": null,
3 "accessTokenAcceptedVersion": null,
4 "addIns": [],
5 "appId": "ec78d011-e49e-4733-b69f-bd0763004190",
6 "appRoles": [],
7 "oauth2AllowUrlPathMatching": false,
8 "createdDateTime": "2019-12-16T06:20:46Z",
9 "groupMembershipClaims": null,
10 "identifierUris": [
11   "api://ec78d011-e49e-4733-b69f-bd0763004190"
12 ],
13 "informationalUrls": {
14   "termsOfService": null,
15   "support": null,
16   "privacy": null,
17   "marketing": null
18 },
19 "keyCredentials": [],
20 "knownClientApplications": [],
21 "logoutUrl": null,
22 "name": "Contoso App",
23 "oauth2AllowIdTokenImplicitFlow": false,
24 "oauth2AllowImplicitFlow": false,
25 "oauth2Permissions": [],
26 "oauth2RequirePostResponse": false,
27 "optionalClaims": null,
28 "orgRestrictions": {},
29 "parentalControlSettings": {
30   "countriesBlockedForMinors": [],
31   "legalAgeGroupRule": "Allow"
32 },
33 "parentalControlSettings": {
34   "countriesBlockedForMinors": [],
35   "legalAgeGroupRule": "Allow"
36 }
```

Demo

Registering an application in Azure Active Directory

© Copyright Microsoft Corporation. All rights reserved.

**FOR USE ONLY AS PART OF VIRTUAL TRAINING DAYS PROGRAM. THESE MATERIALS ARE NOT AUTHORIZED
FOR DISTRIBUTION, REPRODUCTION OR OTHER USE BY NON-MICROSOFT PARTIES.**

Implement Microsoft identity (Part 2)

- Implement Authentication
- Implement Authorization



Implement Authentication

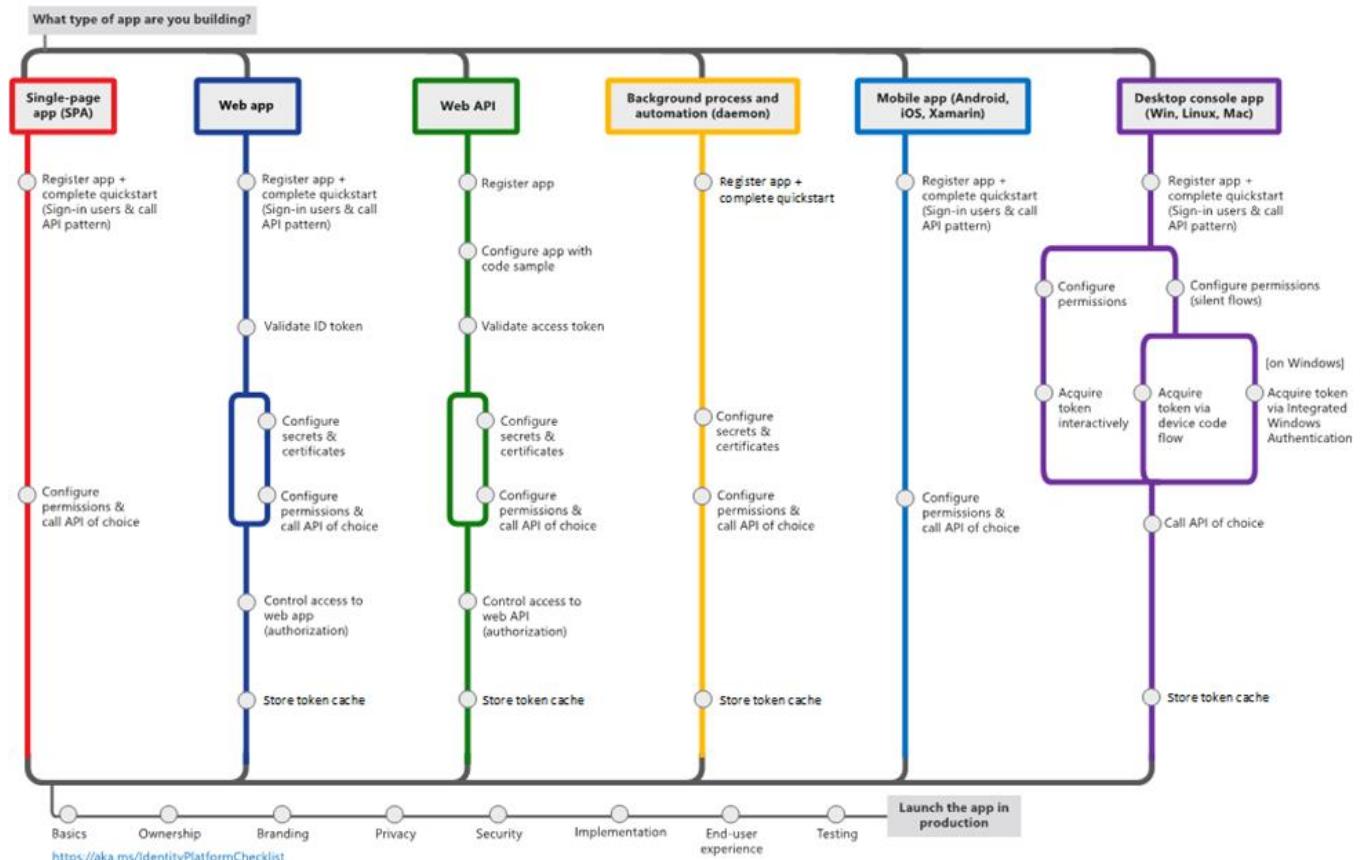
Authentication flows and application scenarios

Authentication categories

- Protected resources vs. client applications
- With users or without users
- Single-page, public client, and confidential client applications
- Sign-in audience
- Supported OAuth 2.0 flows
- Supported platforms

Microsoft identity platform

<http://aka.ms/identityPlatform>



Microsoft Authentication Library (MSAL)

Microsoft Authentication Library (MSAL) enables developers to acquire tokens from the Microsoft identity platform endpoint in order to access secured Web APIs.

- Application types and scenarios.
- Languages and frameworks.

Security tokens for Microsoft identities

- ID token

An ID token is a security token that allows the client to verify the identity of the user.

- Access tokens

Access tokens enable clients to securely call APIs protected by Azure AD. Access tokens are sometimes referred to as "User+App" or "App-Only"

Demo

Implementing Authentication



Implement Authorization

Overview of authorization model

Applications that integrate with Microsoft identity platform follow an authorization model that gives users and administrators control over how data can be accessed.

- OAuth 2.0 authorization code flow
- Scopes and permissions
- Delegated vs. application permissions
 - Effective permissions
- Requesting consent for an entire tenant
- Admin-restricted permissions

Admin consent

- Some permissions require consent from an administrator before they can be granted within a tenant. You can also use the admin consent endpoint to grant permissions to an entire tenant.
- The app registration UI displays the permissions and admin consent granted to your app. It has the following sections:
 - Configured permissions.
 - Other permissions granted.
 - Admin consent button
- Request the permissions from a directory admin.
- Understanding application authorization consent.

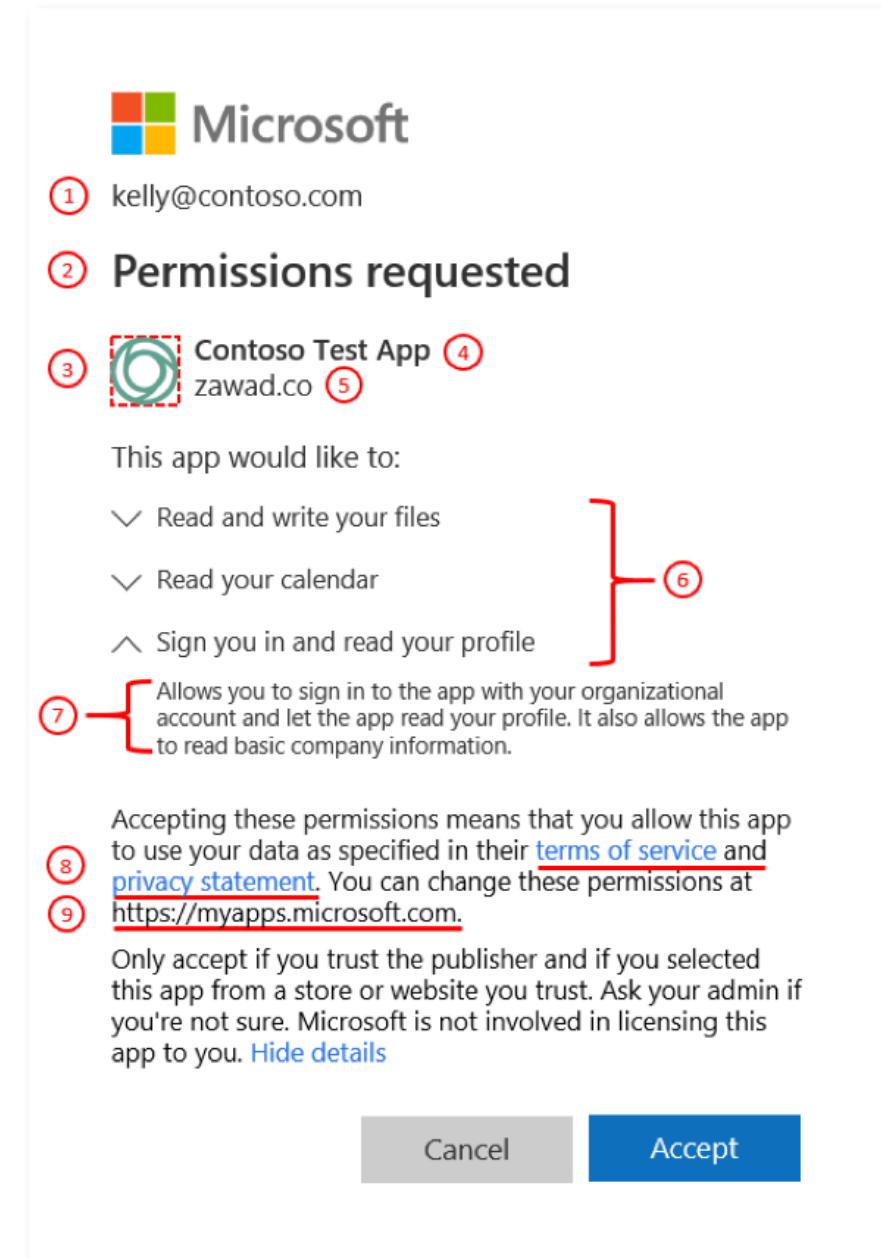
Application consent experiences

- **User consent flow**

when an application developer directs users to the authorization endpoint with the intent to record consent for only the current user.

- **Admin consent flow**

when an application developer directs users to the admin consent endpoint with the intent to record consent for the entire tenant. To ensure the admin consent flow works properly, application developers must list all permissions in the RequiredResourceAccess property in the application manifest.



Add permissions to access Web APIs

- Microsoft APIs
- APIs my organization uses
- My APIs

The screenshot shows the Microsoft Azure portal interface for managing API permissions. On the left, a sidebar menu includes options like Overview, Quickstart, Manage, Authentication, Certificates & secrets, Token configuration (preview), API permissions (selected), Expose an API, Owners, Roles and administrators (Preview), Manifest, Support + Troubleshooting, Troubleshooting, and New support request. The main content area is titled 'Contoso App - API permissions' and shows 'Configured permissions' with two entries: 'Microsoft Graph (1)' and 'WebApp-OpenIDConnect-DotNet (1)'. A red arrow points from the 'API permissions' link in the sidebar to the '+ Add a permission' button in the main content area. To the right, a 'Request API permissions' panel is open, featuring tabs for 'Microsoft APIs' (selected), 'APIs my organization uses', and 'My APIs'. Below the tabs, a section titled 'Commonly used Microsoft APIs' lists various services with their icons and brief descriptions.

API / Permissions name	Type	Description
User.Read	Delegated	Sign in and read
Employees.Read.All	Delegated	Read-only access

Request API permissions

Select an API

Microsoft APIs **APIs my organization uses** **My APIs**

Commonly used Microsoft APIs

Microsoft Graph Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.	Azure Rights Management Services Allow validated users to read and write protected content	Dynamics 365 Business Central Programmatic access to data and functionality in Dynamics 365 Business Central
Flow Service Embed flow templates and manage flows	Intune Programmatic access to Intune data	Office 365 Management APIs Retrieve information about user, admin, system, and policy actions and events from Office 365 and Azure AD activity
OneNote Create and manage notes, lists, pictures, files, and more in OneNote notebooks	Power BI Service Programmatic access to Dashboard resources such as Datasets, Tables, and Rows in Power BI	SharePoint Interact remotely with SharePoint data
Skype for Business Integrate real-time presence, secure	Yammer Access resources in the Yammer web	

Secure custom APIs with Microsoft identity

Securing a web API with Microsoft identity that can be called by other applications involves two main tasks:

- Register & configure an Azure AD application
 - Define scopes
- Code the web API project, configured to support Microsoft identity
 - Add support to validate the current request has the necessary scopes

Call the Microsoft Graph API

The Microsoft Graph API provides programmatic access to Azure AD through REST API endpoints. Applications can use the Microsoft Graph API to perform create, read, update, and delete (CRUD) operations on directory data and objects.

- Provides the following features:
 - REST API endpoints.
 - Authentication with Azure AD.
 - Role-based authorization (RBAC).
 - Differential query.
 - Directory extensions.
 - Secured by permission scopes
- Enables application scenarios, such as:
 - Line-of-business (single tenant) application.
 - Software-as-a-service (multi-tenant) application.

Demo

Implementing authorization to
consume an API

© Copyright Microsoft Corporation. All rights reserved.

**FOR USE ONLY AS PART OF VIRTUAL TRAINING DAYS PROGRAM. THESE MATERIALS ARE NOT AUTHORIZED
FOR DISTRIBUTION, REPRODUCTION OR OTHER USE BY NON-MICROSOFT PARTIES.**

Work with Microsoft Graph

- Overview of Microsoft Graph.
- Optimize data usage with query parameters.
- Optimize network traffic.
- Access user data with Microsoft Graph.
- Access files with Microsoft Graph.
- Manage a group lifecycle on Microsoft Graph.



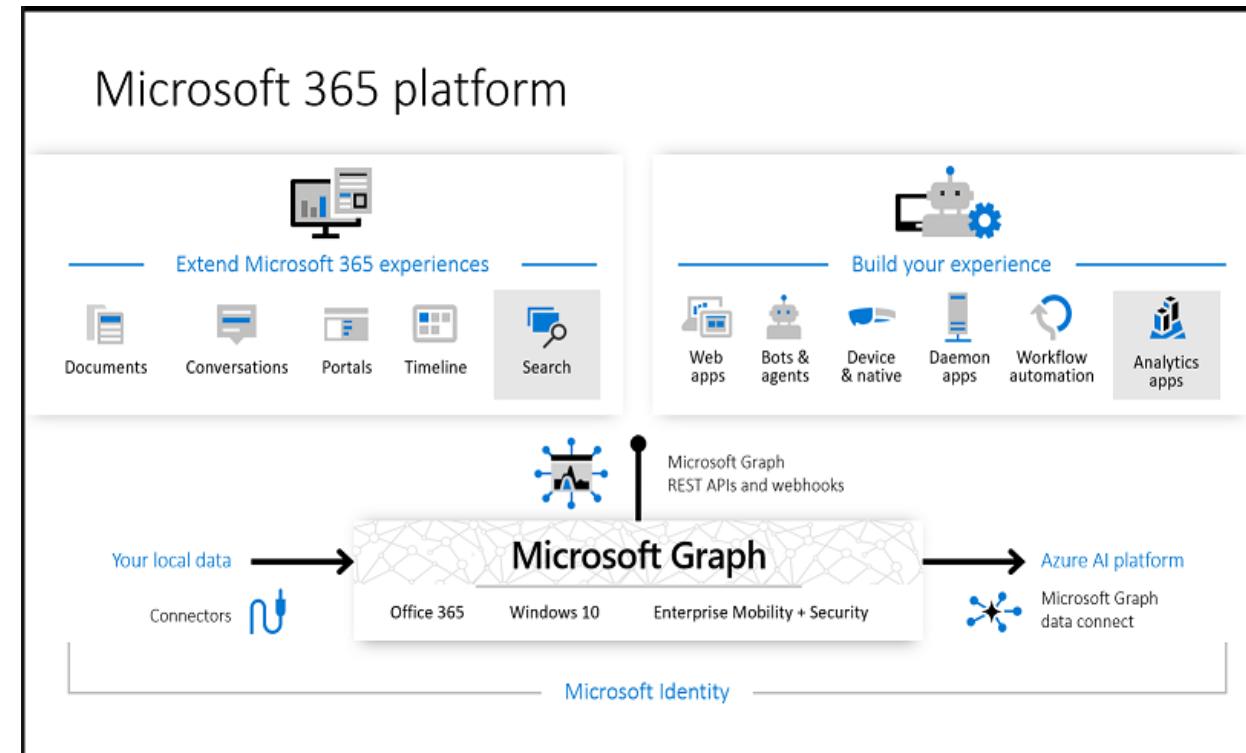
Overview of Microsoft Graph

What is Microsoft Graph?

Microsoft Graph is a RESTful web API that enables you to access Microsoft cloud service resources.

Three main components:

- The Microsoft Graph API.
- Microsoft Graph data connect.
- Microsoft Graph connectors.



Access the Microsoft Graph

- **Use the Microsoft Graph REST API:** use the platform, framework, and programming language you're most comfortable with to issue common HTTP requests and process HTTP responses.
- **Use a Microsoft Graph native SDK:** Abstract the details of constructing and processing HTTP requests
- Authentication options:
 - **Azure AD accounts:** M365 and O365 resources
 - **Microsoft accounts:** consumer resources

Work with the Microsoft Graph API

- Use **Microsoft Graph Explorer** to test
- Request:
{HTTP method}https://graph.microsoft.com/{version}/{resource}?{query-parameters}
- Response:
 - Status code
 - Response message
 - nextLink



Optimize data usage with query parameters

Introduction to query parameters

- Microsoft Graph APIs implement many of the OData protocol's query parameters.
- Use query parameters to:
 - Order results
 - Limit number of results
 - Search resources
 - Select specific data properties
 - Filter data

The \$select query parameter

Use \$select to:

- Only fetch required data
- Get properties that are not returned by default

Request the *from* and *subject* properties of the signed-in user:

`https://graph.microsoft.com/v1.0/me/messages?$select=from,subject`

The \$orderby query parameter

Use \$orderby to specify sort-order of results

More efficient than fetching all results and ordering on client

Fetch users in the organization ordered by their display name:

`https://graph.microsoft.com/v1.0/users?$orderby=displayName`

The \$skip and \$top query parameters

- Use the \$top query parameter to specify the page size of the result set.

https://graph.microsoft.com/v1.0/users?\$top=5

- Send URL value of the @odata:nextLink property to Microsoft Graph to retrieve next page of results.

*"@odata.nextLink": "https://graph.microsoft.com/v1.0/users?\$top=5&\$skiptoken=X%274453707 ...
6633B900000000000000000000000000%27"*

- Use the \$skip query parameter to set the number of items to skip at the start of a collection.

https://graph.microsoft.com/v1.0/me/events?\$orderby=createdDateTime&\$skip=20

The \$expand query parameter

- Use \$expand to fetch additional resources related to the requested resource.
- Provide a comma-separated list of properties to be expanded and included in the results.

Fetch root drive information along with the top-level child items in a drive:

```
GET https://graph.microsoft.com/v1.0/me/drive/root?$expand=children
```

The \$count query parameter

Add \$count=true as a query string parameter to include a count of the total number of items in a collection with the results

To fetch the contact collection of the current user, along with the number of items in the contact collection:

`https://graph.microsoft.com/v1.0/me/contacts?$count=true`

The \$search query parameter

- Use \$search to restrict the results of a request to match a search criterion.
- Supports Keyword Query Language (KQL) structured queries that allow you to target searches to specific fields.
- You can search message and person collections.
 - Searches on messages search the *from*, *subject*, and *body* properties by default
 - Searches on people search the *displayName* and *emailAddress* properties

Fetch messages in the signed-in user's inbox that contain "pizza" in the *from*, *subject*, or *body* properties

GET https://graph.microsoft.com/v1.0/me/messages?\$search="pizza"

The \$filter query parameter

- Use \$filter to find resources that match a specified query.
- Use to retrieve just a subset of a collection.
- Support for \$filter operators varies across Microsoft Graph APIs

Fetch all unread mail in the signed-in user's inbox using the *equals* logical operator:

`https://graph.microsoft.com/v1.0/me/mailFolders/inbox/messages?$filter=isRead eq false`

Demo

Using query parameters when querying
Microsoft Graph via HTTP



Optimize network traffic

Receive change notifications

Build applications that respond to changes in data within Microsoft Graph:

- Subscribe to changes on supported resources
- Client receives notifications to the URL provided when creating the subscription
- Send an HTTP 202 response to acknowledge receipt of notification
- Validate the clientState property
- Update your application based on business logic

Perform batch requests

Use \$batch to combine multiple Graph requests into a single HTTP request, reducing the number of HTTP calls.

- Create a JSON array of requests that you then POST to \$batch.
- Use **dependsOn** to specify dependencies between requests in a batch.

```
{  
  "requests": [  
    {  
      "id": "1",  
      "method": "GET",  
      "url": "..."  
    },  
    {  
      "id": "2",  
      "dependsOn": [ "1" ],  
      "method": "GET",  
      "url": "..."  
    },  
    {  
      "id": "3",  
      "method": "GET",  
      "url": "..."  
    },  
    {  
      "id": "4",  
      "dependsOn": [ "2" ],  
      "method": "GET",  
      "url": "..."  
    }  
  ]  
}
```

Get changes using a delta query

Efficiently poll Microsoft Graph for changes to sets of data.

- Make a GET request with the `delta` function on the desired resource
- Graph response will include a `nextLink` URL if there are additional pages or a `deltaLink` URL if there is no more data about the existing state of the resource.
- For future requests, use the `deltaLink` URL to retrieve changes to the resource since the previous request
- Optional query parameters must be specified in the initial request. They will then be encoded in the `nextLink` or `deltaLink` tokens.

Handle throttling

Throttling limits the number of concurrent calls to a service to prevent overuse of resources.

When throttling occurs, Microsoft Graph returns HTTP status code 429 and the requests fail

- Wait the number of seconds specified in the **Retry-After** header
- Retry the request

Demo

Reducing traffic with batched requests



Access user data with Microsoft Graph

Work with users in Microsoft Graph

Access and manipulate user resources without having to perform additional calls, look up specific authentication information, and directly issue queries against other Microsoft Graph resources.

Accessing users through Microsoft Graph:

- /users/{id | userPrincipalName}
- /me

Get information about users

Task	HTTP Request
Get a list of users in the organization	<code>GET /users</code>
Get the user's profile photo	<code>GET /me/photo/\$value</code> <code>GET /users/{id userPrincipalName}/photo/\$value</code>
Get metadata about the profile photo	<code>GET /me/photo</code> <code>GET /users/{id userPrincipalName}/photo</code>
Get the user's manager profile	<code>GET /me/manager</code> <code>GET /users/{id userPrincipalName}/manager</code>

Demo

Querying user data from
the Microsoft Graph



Access files with Microsoft Graph

Work with files in Microsoft Graph

- Microsoft Graph exposes two resource types for working with files:
 - Drive resource
 - Drivelist resource
- Drive and Drivelist resources expose data in three different ways:
 - Properties expose simple values.
 - Facets expose complex values.
 - References point to collections of other resources.

Get information about a Drive

A Drive is the top-level container for a file system, such as OneDrive or SharePoint document libraries. Developers can use Microsoft Graph to retrieve the **properties** and **relationships** of a Drive resource.

Task	HTTP Request
Get information about a Drive	<code>GET /me/drive</code> <code>GET /users/{id userPrincipalName}/drive</code>
Get a list of items in a Drive	<code>GET /me/drive/root/children</code>
Get the list of trending items around the signed-in user	<code>GET /me/drive/recent</code>

Download files

Developers can use Microsoft Graph to download the contents of the primary stream (file) of a DriveItem.

Only driveItems with the **file** property can be downloaded.

HTTP request

```
GET /me/drive/items/{item-id}/content
```

Upload a large file

To upload a file using an upload session, there are two steps:

1. Create an upload session.
2. Upload bytes to the upload session.

HTTP request:

```
POST /me/drive/root:/{{item-path}}:/createUploadSession
```

Get a user object from an owner list in a group and retrieve that user's files

The following order is needed to traverse through Microsoft Graph based on this topic's scenario:

- Enumerate owners of the groups to get a user ID.
- Grant the appropriate permissions for the list owner's API call if necessary.
- Get the user's drive ID.
- List the items in the user's drive using the user ID and the drive ID returned from previous HTTP responses.



Manage a group lifecycle on Microsoft Graph

Work with groups in Microsoft Graph

The group resource within Microsoft Graph represents multiple things as there are different types of groups. The types of groups accessible from Microsoft Graph include:

- Office 365 groups
- Security groups

Get group information

Developers can get a list of groups, or specific groups, with the Microsoft Graph API or one of the multiple Microsoft Graph SDKs.

HTTP request

```
GET https://graph.microsoft.com/v1.0/groups  
GET https://graph.microsoft.com/v1.0/groups/{ID}
```

Get group Information

Task	HTTP Request
Get group owners	GET /groups/{id}/owners
Get group members	GET /groups/{id}/members
Get the list of groups where a user is an owner	GET https://graph.microsoft.com/v1.0/me/ownedobjects
Get the list of groups where a user is a member	GET https://graph.microsoft.com/v1.0/me/memberof GET https://graph.microsoft.com/v1.0/users/{ID userPrincipalName}/memberOf

Manage groups

Task	HTTP Request
Create a group	POST /groups
Create a Team with a group	PUT /groups/{id}/team
Delete a group	DELETE https://graph.microsoft.com/v1.0/groups/{ID}

© Copyright Microsoft Corporation. All rights reserved.

**FOR USE ONLY AS PART OF VIRTUAL TRAINING DAYS PROGRAM. THESE MATERIALS ARE NOT AUTHORIZED
FOR DISTRIBUTION, REPRODUCTION OR OTHER USE BY NON-MICROSOFT PARTIES.**

Extend and Customize SharePoint

- SharePoint Framework web parts.
- SharePoint Framework extensions.
- Package and deploy a SPFx solution.
- Consumption of Microsoft Graph.
- Consumption of third-party APIs secured with Azure AD from within SPFx.
- Web Parts as Teams Tabs.
- Branding and theming in SharePoint



SharePoint Framework web parts

Introduction to SharePoint Framework

SharePoint Framework is the development model used for the creation of custom experiences for SharePoint and Office 365.

- Use SharePoint Framework (SPFx) to create the following:
 - SharePoint Framework (SPFx) web parts.
 - SPFx extensions.
 - Single Part App Pages.
 - Microsoft Teams customizations.
- Key features of the SharePoint Framework.

SharePoint Framework development tools & libraries

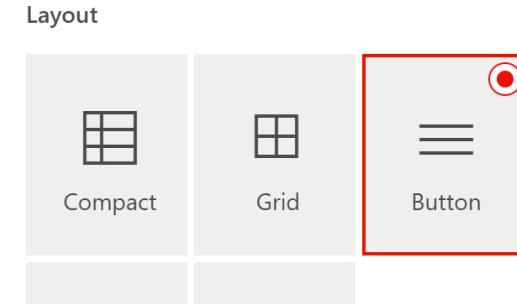
SharePoint Framework includes several client-side JavaScript libraries you can use to build your solutions.

- The tools and libraries used to develop client-side web parts include:
 - TypeScript.
 - JavaScript frameworks (e.g. React, Angular, Vue.js, Handlebars).
 - Node Package Manager (npm).
 - Node.js.
 - Gulp task runner (use to build packages and solutions).
 - Webpack.
 - Source code editors (e.g. Visual Studio Code, Atom, WebStorm).
 - SharePoint REST APIs.
 - Patterns and Practices (PnP).
 - Yeoman generators.
- Generate SharePoint Framework Projects using Yeoman.

SharePoint client-side web parts

SharePoint client-side web parts are controls that appear inside a SharePoint page but run locally in the browser. Integrate web part properties with SharePoint.

- Client-side web part properties.
- Property pane field types supported:
 - Button
 - Checkbox
 - Choice group
 - Dropdown
 - Horizontal rule
 - Label
 - Link
 - Slider
 - Textbox
 - Multi-line Textbox
 - Toggle
 - Custom
- Configure the property pane.



Alignment

Center

Title text

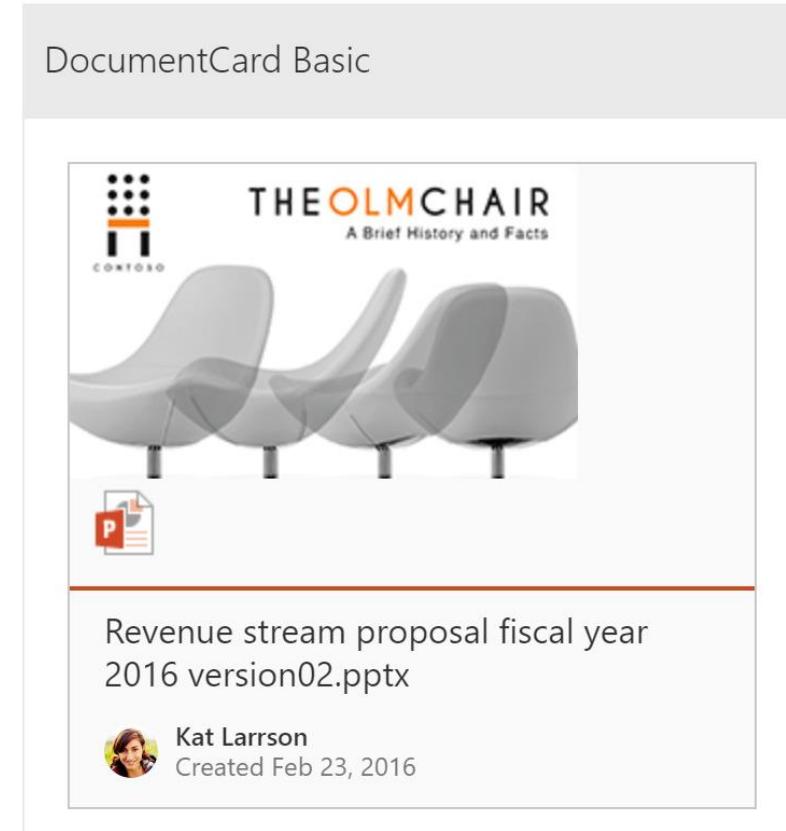
One line

Two lines

Office UI Fabric React components in SharePoint client-side web part

Two areas of UI Fabric that developers can use:

- UI Fabric Core
 - Colors
 - Icons
 - Fonts
 - Grids
- Fabric React Components: A series of reusable components
 - Buttons
 - Navigation
 - Dropdowns
 - Labels



Office UI Fabric React components in SharePoint client-side web part

- Using Office UI Fabric Core and Fabric React in SharePoint Framework.
- Office UI Fabric Core package.
- Office UI Fabric React components.
- The CSS challenge with Office UI Fabric.

Single part app pages

Single part app pages provide the ability to host SharePoint Framework web parts or Teams applications in SharePoint Online with a locked layout. End users cannot modify or configure a page that uses the Single Part App Page layout.

- App pages have following characteristics:
 - Single Part App Pages cannot be edited by end users using a browser.
 - Currently support hosting only single web part or Microsoft Teams application.
 - Page layout can only be changed programmatically from normal page layout to a Single Page App Page.
 - End-users cannot parameterize exposed web part or Teams application.
- Use the Single Part App page in your tenant.
- Change Page layout using JavaScript in browser console.
- Change page layout using PnP PowerShell.

Demo

Introduction to SharePoint Framework (SPFx)



SharePoint Framework extensions

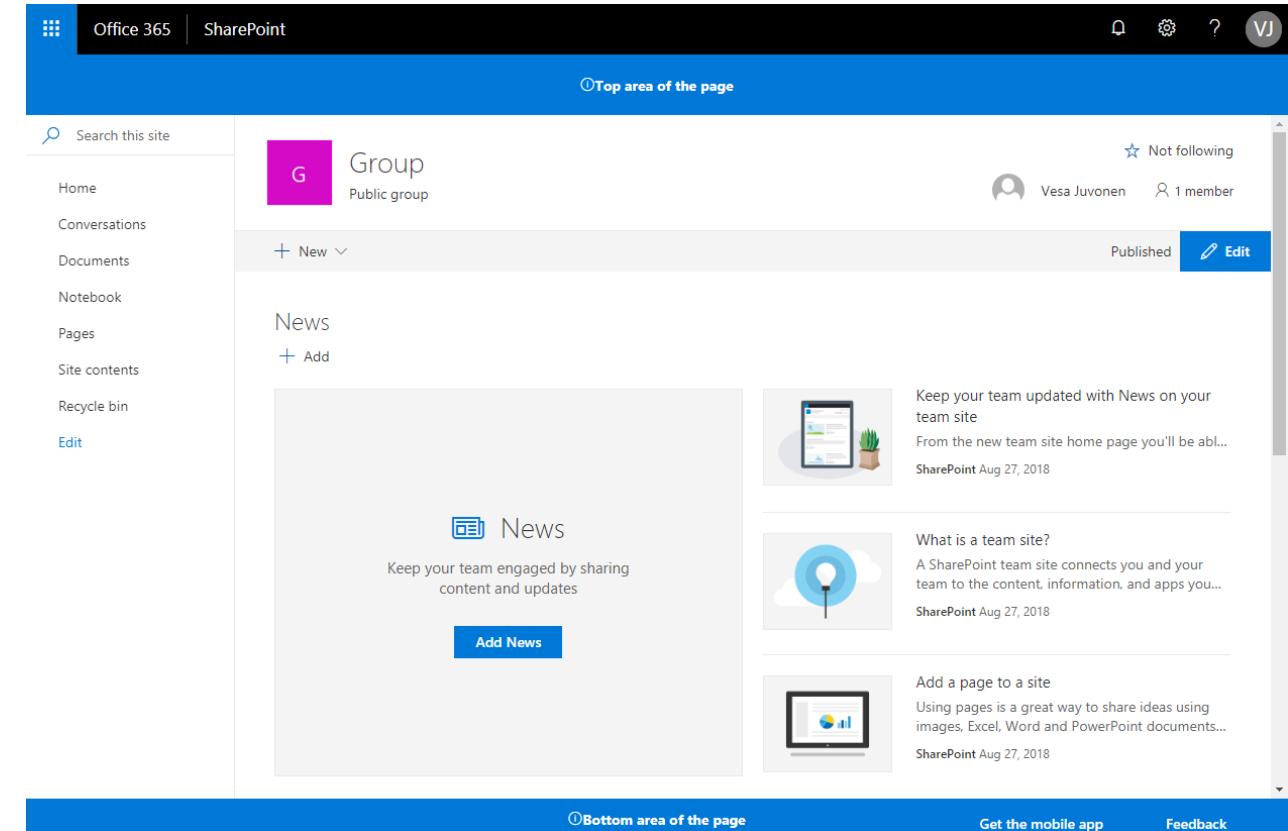
SharePoint Framework extensions

SharePoint Framework extensions allow you to extend the development capabilities of SharePoint and Teams.

- Three specific types of extensions.
 - Application Customizers: Adds scripts to the page and accesses well-known HTML element placeholders and extends them with custom renderings.
 - Command Sets: Extends the SharePoint command surfaces to add new actions and provides client-side code that you can use to implement behaviors.
 - Field Customizers: Provides modified views to data for fields within a list.

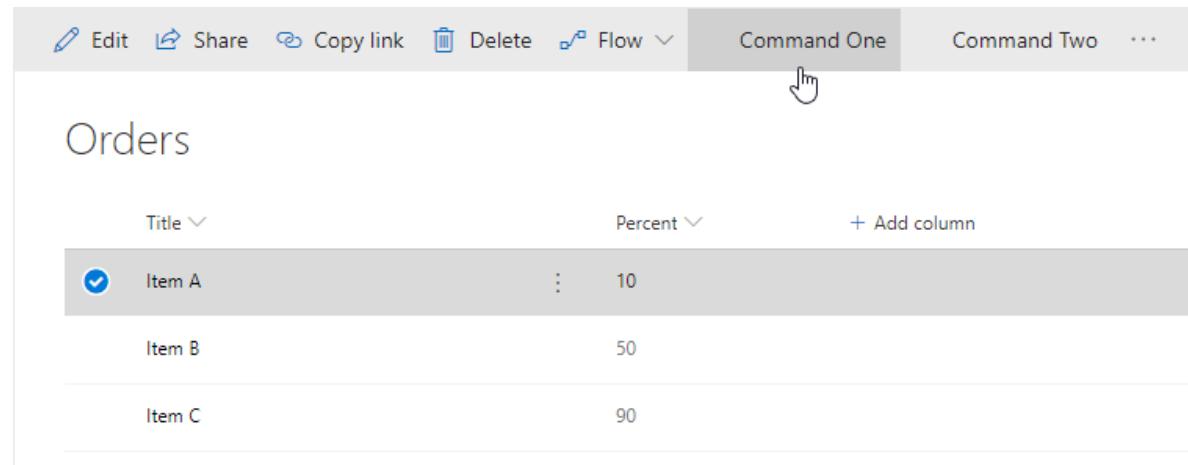
The Application Customizer extension

Application Customizers provide access to well-known locations on SharePoint pages that you can modify based on your business and functional requirements.



The ListView Command Set extension

ListView Command Set extension allows you to add custom controls and functionality within the Command Bar at the top of the page.



The Field Customizer extension

Field Customizer extensions allow you to customize how specific fields render within a SharePoint List.

- Examples:
 - Status
 - Percentage

Title	Percentage	Add column
Item A	<div style="width: 35%;">35</div>	+ Add column
Item B	<div style="width: 60%;">60</div>	+ Add column
Item C	<div style="width: 100%;">100</div>	+ Add column

Demo

Creating SharePoint Framework
Extensions



Package and deploy a SPFx solution

Prepare a package for deployment

At times, developers may need to provision a SharePoint list or a document library along with your client-side solution package so that the list or library is available for your client-side components, such as web parts.

- Provision SharePoint assets with your solution package.
- Create SharePoint items in your solution.
- Upgrade SharePoint items.
- Update SharePoint Framework packages.

Package a solution

Each package configuration file includes some optional settings to override the places where the task looks for various source files and manifests, as well as defining the location to write the package. Additionally, it includes a required solution definition, which instructs the packager on the relationships between various components.

- Solution definition (ISolution).
- Feature definition (IFeature).
- File paths.

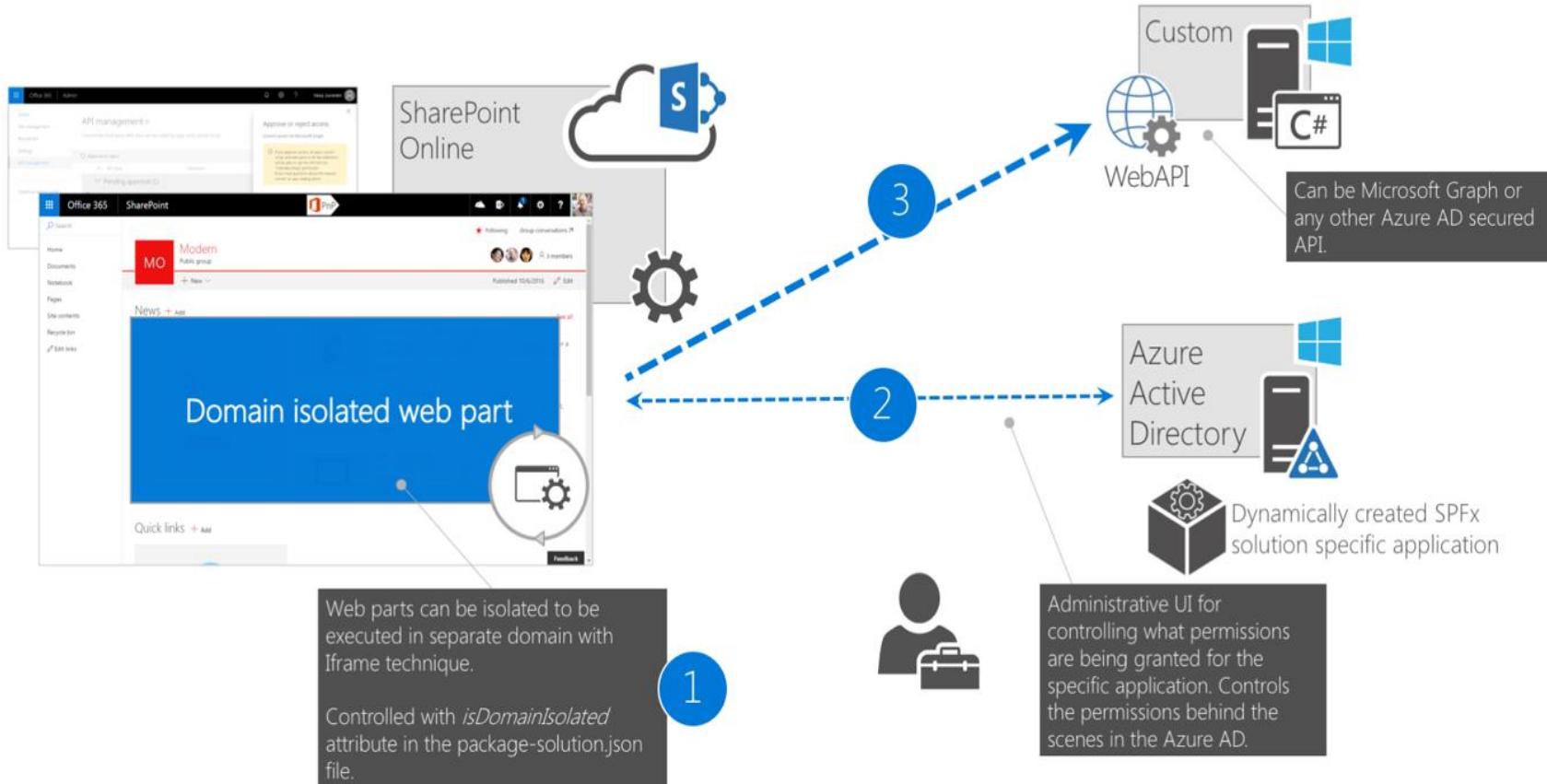
Tenant-scoped solution deployment

The Tenant-Wide Deployment option for SharePoint Framework extensions is supported for application customizers and for ListView Command Sets. It provides tenant app catalog managers with an easy option for managing which extensions are activated by default across the tenant or based on web/list templates used in the sites.

- Tenant-scoped solution deployment option.
- Solution-specific requirements.
- Configure solution to be available across the tenant.
- Approve tenant-wide deployment in app catalog.

Domain isolated web parts

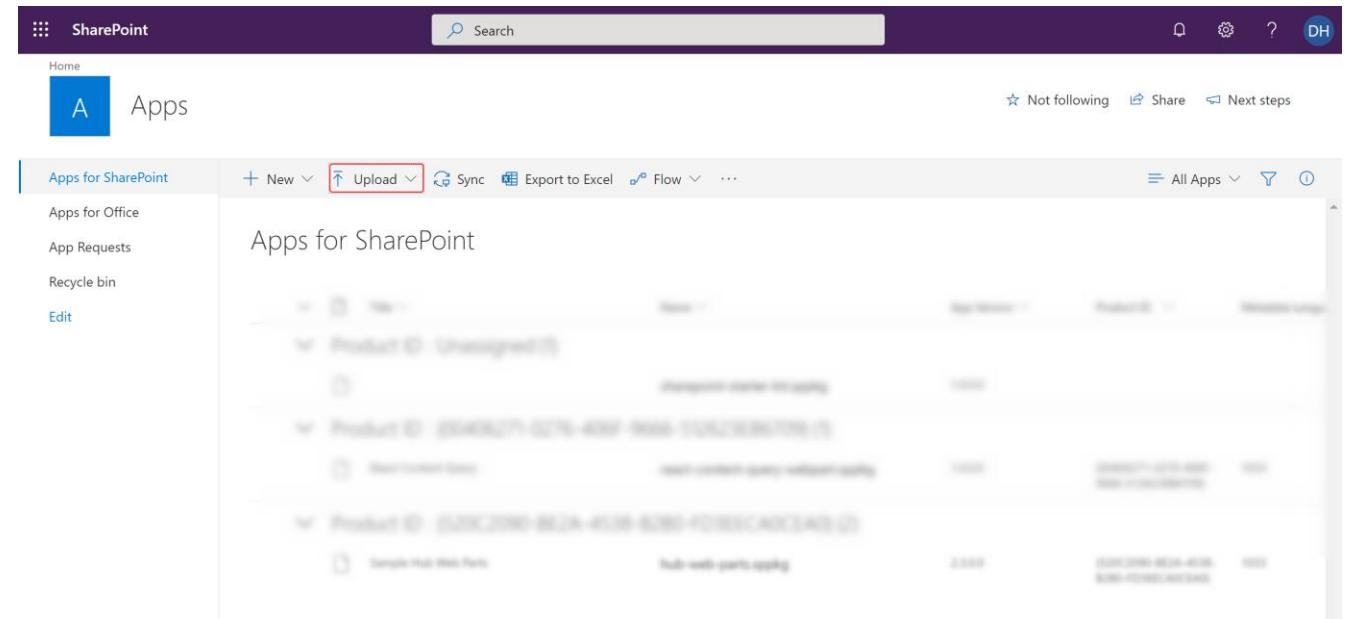
Domain isolated web parts – supporting also isolated permissions



Optimize SharePoint Framework builds for production

Package the solution using the following commands, then upload into the app catalog for SharePoint.

- gulp build
- gulp bundle –ship
- gulp package-solution –ship



Demo

Deploying a SharePoint Framework Solution



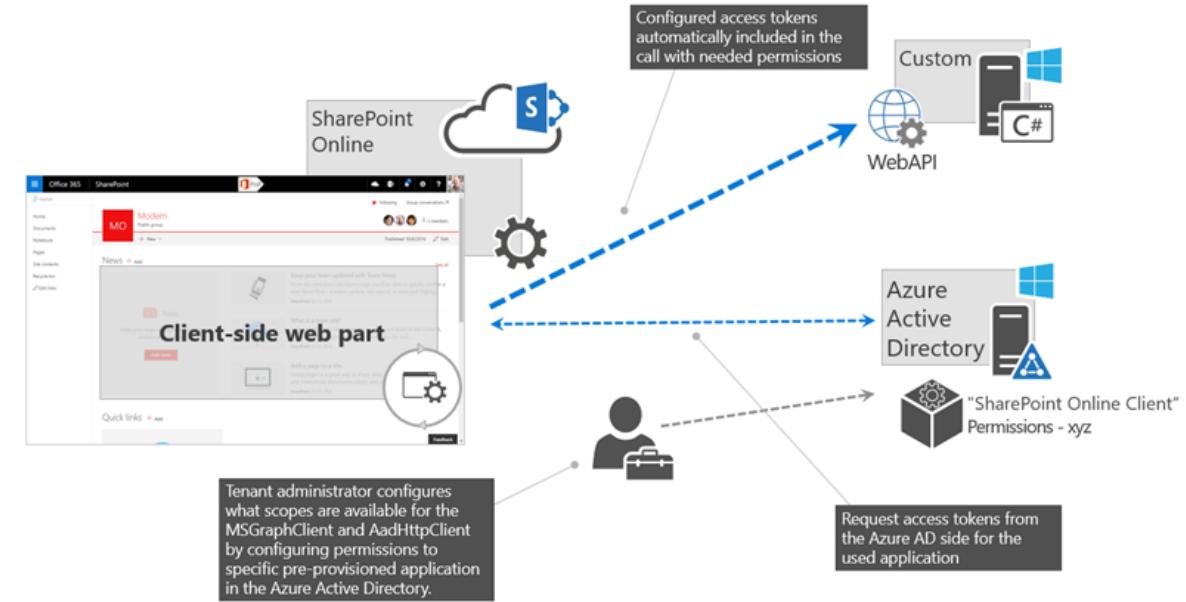
Consumption of Microsoft Graph and third-party APIs

Use the MSGraphClient to connect to Microsoft Graph

- When building SharePoint Framework solutions, developers can easily connect to Microsoft Graph by using the MSGraphClient.
- MSGraphClient is a new HTTP client introduced in SharePoint Framework v1.6.0 that simplifies connecting to the Microsoft Graph inside SharePoint Framework solutions.
- MSGraphClient wraps the existing [Microsoft Graph JavaScript Client Library](#), offering developers the same capabilities as when using the client library in other client-side solutions.

Understand AadHttpClient

- SharePoint Framework allows you to specify which Azure AD applications and permissions your solution requires, and a tenant administrator can grant the necessary permissions if they haven't yet been granted.
- By using the AadHttpClient, you can easily connect to APIs secured by using Azure AD without having to implement the OAuth flow yourself.



Grant permissions to consume a third-party API

When building SharePoint Framework solutions, you might want to consume public APIs, such as stock or weather information.

```
ts

this.context.httpClient
  .get('https://jsonplaceholder.typicode.com/todos/1', HttpClient.configurations.v1)
  .then((res: HttpClientResponse): Promise<any> => {
    return res.json();
})
  .then((response: any): void => {
    console.log(response);
});
```



Branding and theming in SharePoint

SharePoint site theming

Like the Microsoft brand palette, the SharePoint themes are designed to build on the Microsoft brand, while at the same time allowing for flexibility to enliven your partnerships without dominating them. They reveal shared goals and personality, and they reflect diversity and ability to optimize the SharePoint experience.

SharePoint site theming

Theming experience components:

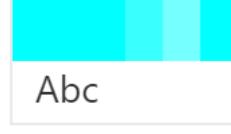
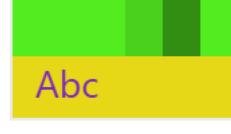
- Color palette.
- Font scheme.
- Background image.
- Master page.
- Master page preview.
- Composed look.
- Change the look.
- Design gallery.

Site design and site scripts

Use site designs and site scripts to automate provisioning new or existing modern SharePoint sites that use your own custom configurations.

- Site designs.
- Site script.
- Use PowerShell or REST to work with site designs an site scripts.
- Scoping.

Site design and site scripts

- Company themes
-  Dee Purple
 -  Custom Cyan
 -  Whitefox theme
 -  Thrive Theme

© Copyright Microsoft Corporation. All rights reserved.

**FOR USE ONLY AS PART OF VIRTUAL TRAINING DAYS PROGRAM. THESE MATERIALS ARE NOT AUTHORIZED
FOR DISTRIBUTION, REPRODUCTION OR OTHER USE BY NON-MICROSOFT PARTIES.**

Extend Microsoft Teams

- Microsoft Teams Apps.
- Webhooks in Microsoft Teams.
- Tabs in Microsoft Teams.
- Messaging extensions in Microsoft Teams.
- Conversational bots in Microsoft Teams.



Microsoft Teams apps

Components of a Microsoft Teams app

A Microsoft Teams app consists of three primary components:

- **The Microsoft Teams client** (web, desktop, or mobile) provides the extensions points and UI elements users will interact with.
- **Teams app package** creates the app installed by users.
 - Manifest file
 - Icons
- **Web services** hosted by developers provide the APIs and logic that power the app.

Microsoft Teams platform UX elements

The Microsoft Teams Platform provides flexible UI elements for apps to take advantage of.

Microsoft Teams platform elements:

- Cards and card actions.
- Task modules.
- Deep links.
- Web content pages.

Extensible points in the Teams client

There are multiple places where the Microsoft Teams client can be extended to allow users to interact with the app.

Options for extending the Teams client:

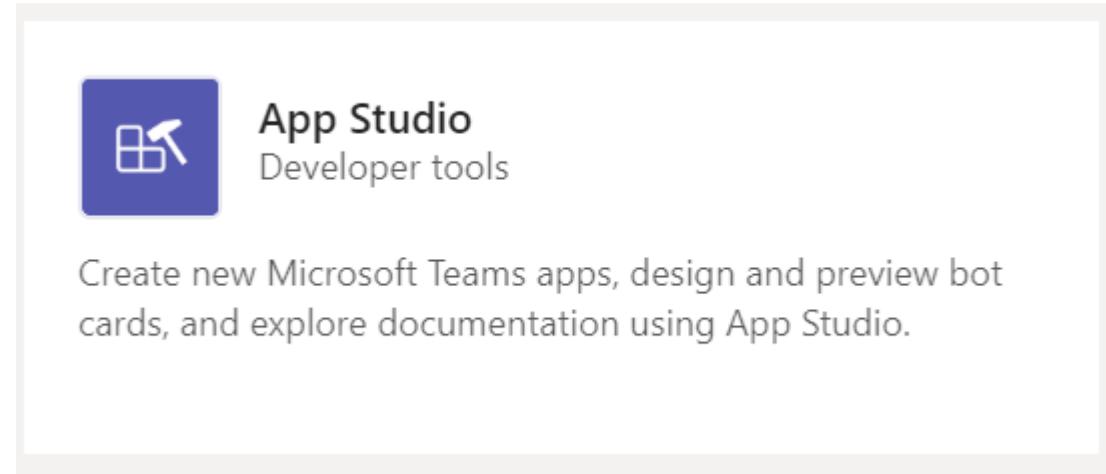
- Teams, channels and group chats
- Personal apps
- Messages

App Studio for Microsoft Teams app development

App Studio streamlines the process of creating manifests and packages for Teams apps.

Use App Studio to:

- Create and edit app manifests
- Design and preview cards
- Find documentation
- Access UI controls in the React control library



Options for distributing a Teams app

Three options for distributing a Microsoft Teams app.

- Share app package directly.
- Publish app to organizational app catalog.
- Publish app to public App Store.

All app installations in Microsoft Teams are context-specific

Demo

Understanding the components of a
Microsoft Teams App



Webhooks in Microsoft Teams

Webhooks and Connectors in Microsoft Teams

Outgoing and incoming webhooks both allow developers to connect web services to channels and teams inside Microsoft Teams.

Outgoing webhooks allow users to send text messages from a channel to the developer's web services without having to use the Microsoft Bot Framework.

Incoming webhooks work as a type of *connector*, allowing users to subscribe to receive notifications and messages from a developer's web services.

Outgoing webhooks in Microsoft Teams

Outgoing webhooks post data from Teams to any chosen service capable of accepting a JSON payload. Once an outgoing webhook is added to a team, it acts like bot, listening in channels for messages using @mention, sending notifications to external web services, and responding with rich messages that can include cards and images.

- Key features:
 - Scoped Configuration.
 - Reactive Messaging.
 - Standard HTTP messaging exchange.
 - Teams API support method.
- Limitations.

Incoming webhooks in Microsoft Teams

- Incoming webhooks function as connectors, providing a simple way for an external app to share content in team channels. They are useful for tracking and notification.
- Teams provides a unique URL to which you send a JSON payload with the message that you want to POST, typically in a card format.
- Key features:
 - Scoped configuration.
 - Secure resource definitions.
 - Actionable messaging support.
 - Independent HTTPS messaging support.
 - Markdown support.

Demo

Working with webhooks in Microsoft Teams



Tabs in Microsoft Teams

What is a tab in Teams?

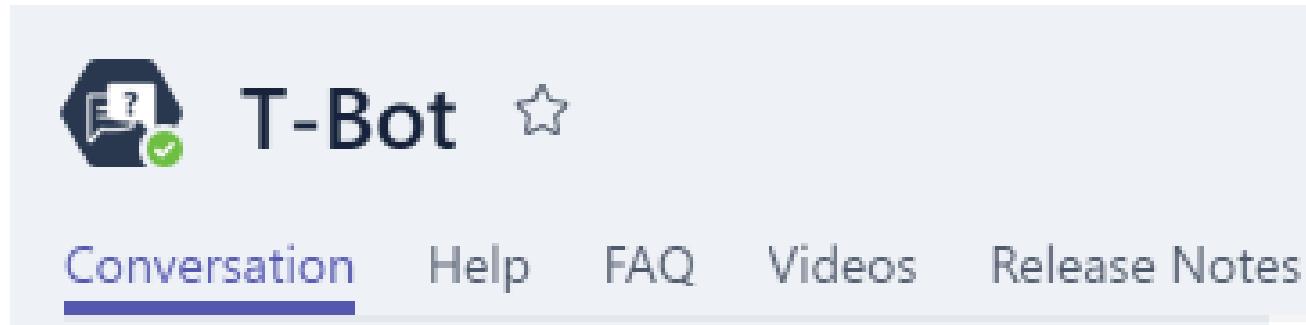
In Microsoft Teams, tabs display rich interactive web content. Developers can create two types of tabs—static and configurable.

- Static tabs provide content for individual users.
- Configurable tabs provide a single type of content for an entire team.
- Tab scope:
 - Teams (team scope)
 - Group chat (groupchat scope)
 - Personal (personal scope)
- Differences between tabs and a browser.

Static tabs

A **static tab** is a content page that is declared directly in your manifest and—unlike a configurable **tab**—does not require a configuration page.

- Create tab content.
- Static tabs on mobile clients.



Configurable tabs

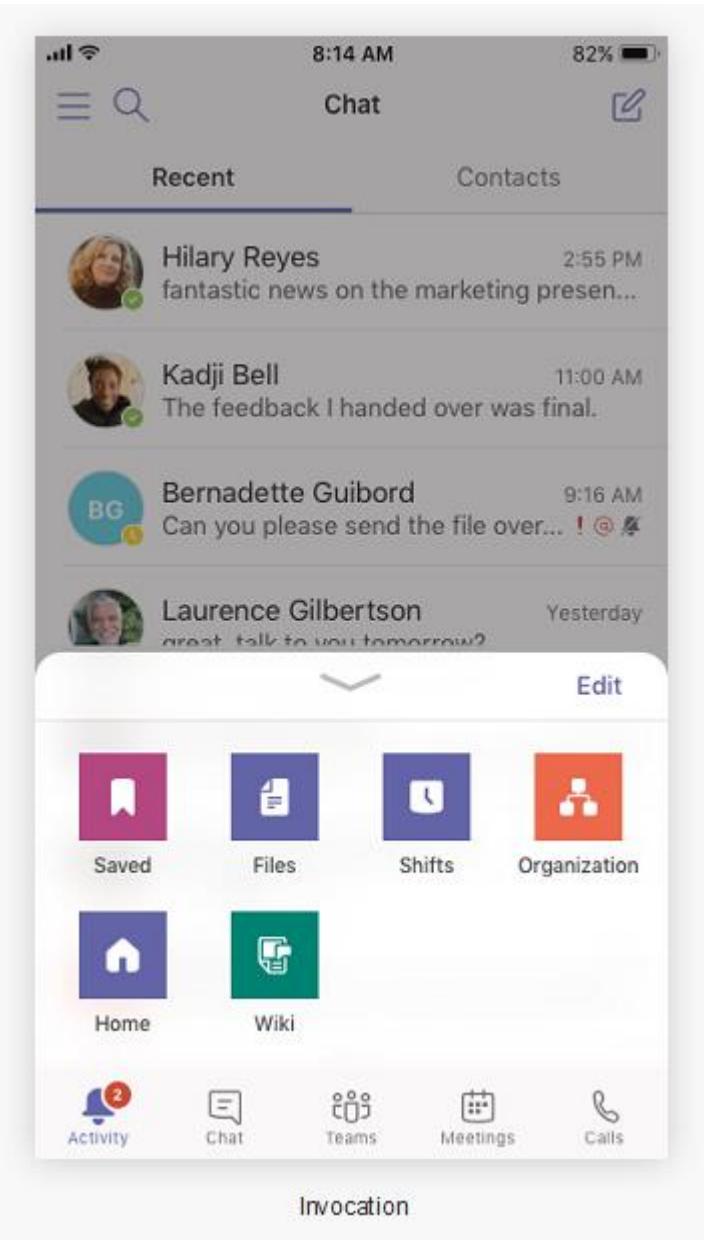
A configurable tab becomes part of a channel and provides one type of information to a team.

- Configurable tab scope.
- Creating a configurable tab involves the following:
 - Create the configuration page: For configurable tabs, you must provide a configuration page to present options and gather information so users can customize the content in, and experience with, your tab. You can also enable users to update or remove a tab after they add it.
 - Create the content page: A content page is an HTML page that you host. You can also provide a page for users to specify what happens to content when they remove a tab.

Tabs on mobile

Group and channel tabs are available on mobile clients.

- Developer considerations for mobile support:
 - Testing on mobile clients.
 - Responsive design.
 - Authentication.
 - Low bandwidth and intermittent connections.
- Design considerations for mobile.
- Tabs with bots on mobile.
- UI components.



Demo

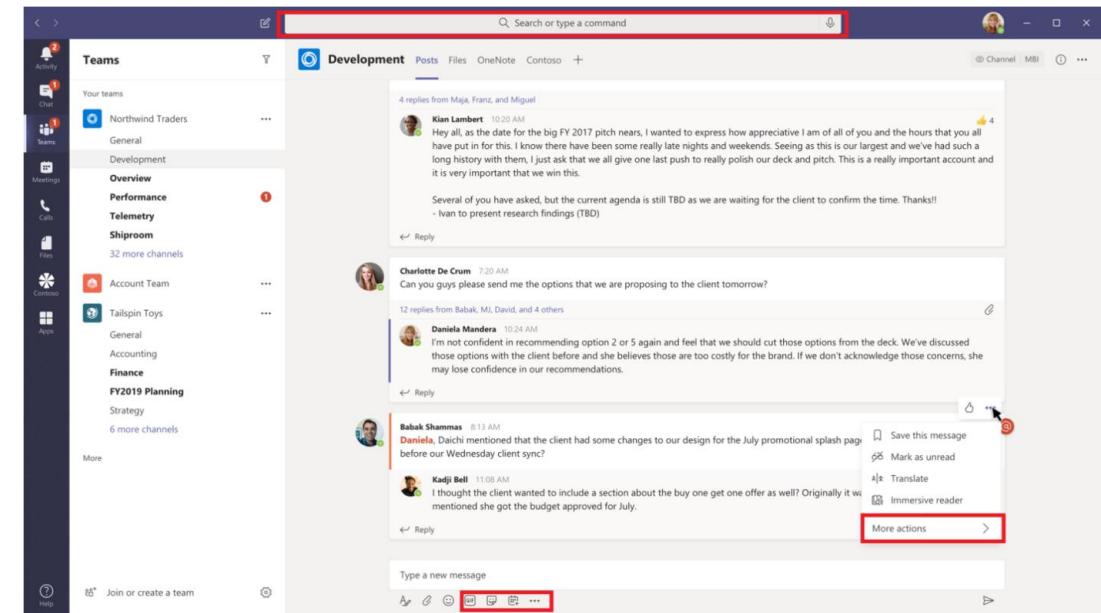
Creating tabs in Microsoft Teams



Messaging extensions in Microsoft Teams

What are messaging extensions for Teams?

- Messaging extensions allow users to query your service for information, and post information to your service.
 - Users can search, or initiate actions, in an external system from the compose message area, the command box, or directly from a message.
 - Developers can then send the results of that interaction back to the Microsoft Teams client, typically in the form of a richly formatted card.
- Search command invoke locations.
- Action command invoke locations.



Overview of creating messaging extensions

- Steps to create a messaging extension:
 1. Prepare your development environment.
 2. Create and deploy your web service (use a tunneling service like ngrok to run it locally.)
 3. Register your web service with the Bot Framework.
 4. Create your app package.
 5. Upload your package to Microsoft Teams.
- Define a messaging extension in the app manifest.
- Register with the Bot Framework.

Search-based messaging extensions

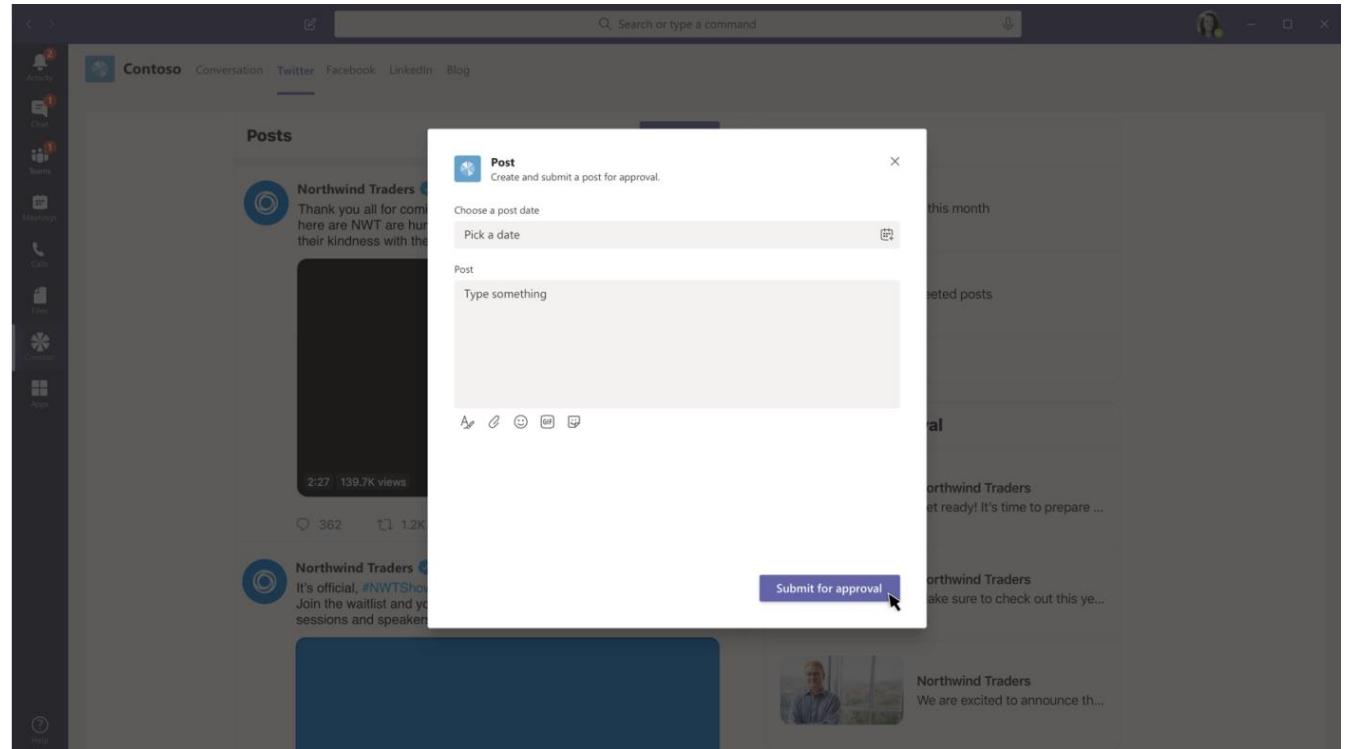
Creating a search-based messaging extension involves defining the commands for the messaging extension in the app manifest and setting up your service to receive and respond to queries.

- Define search-based messaging extension commands.
- Receive queries.
- Respond to queries.

Action-based messaging extensions

Action commands allow developers to present users with a modal popup to collect or display information.

- Define action-based messaging extensions.
- Create and send the task module
- Respond to the submit action.



Demo

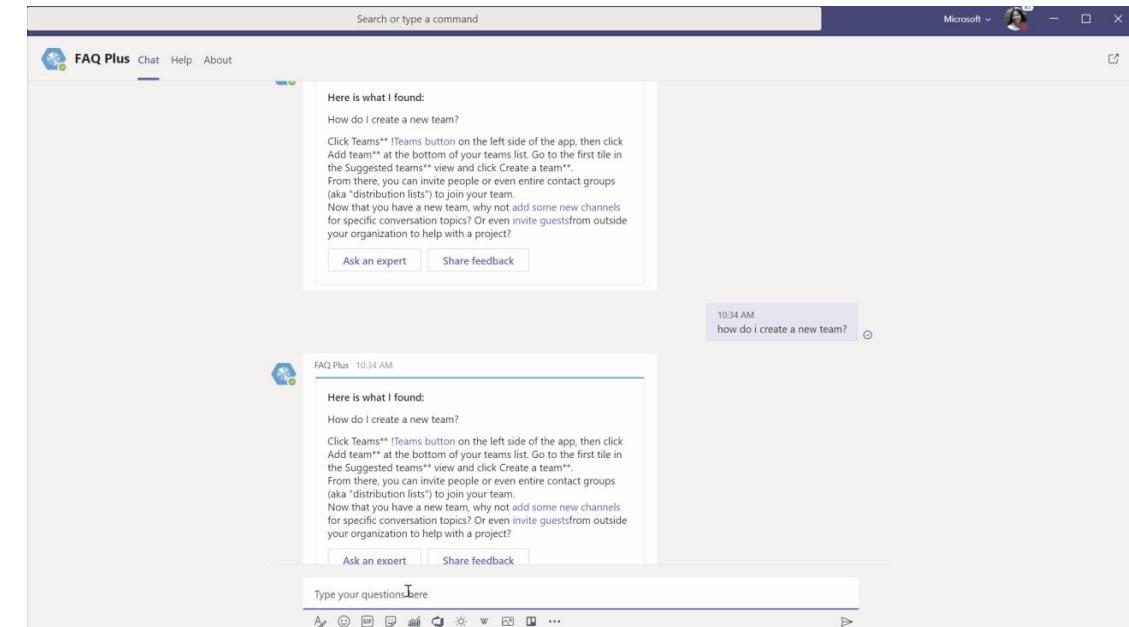
Understanding messaging
extensions in Microsoft Teams



Conversational bots in Microsoft Teams

Overview of Bots in Teams

- Bots are apps that users interact with in a conversational way.
- Users can enter text or graphics or use speech.
- Every interaction between a user and a bot generates an activity object.
- The Bot Framework Service sends the activity objects between the user's channel and the bot.



Overview of Bots in Teams

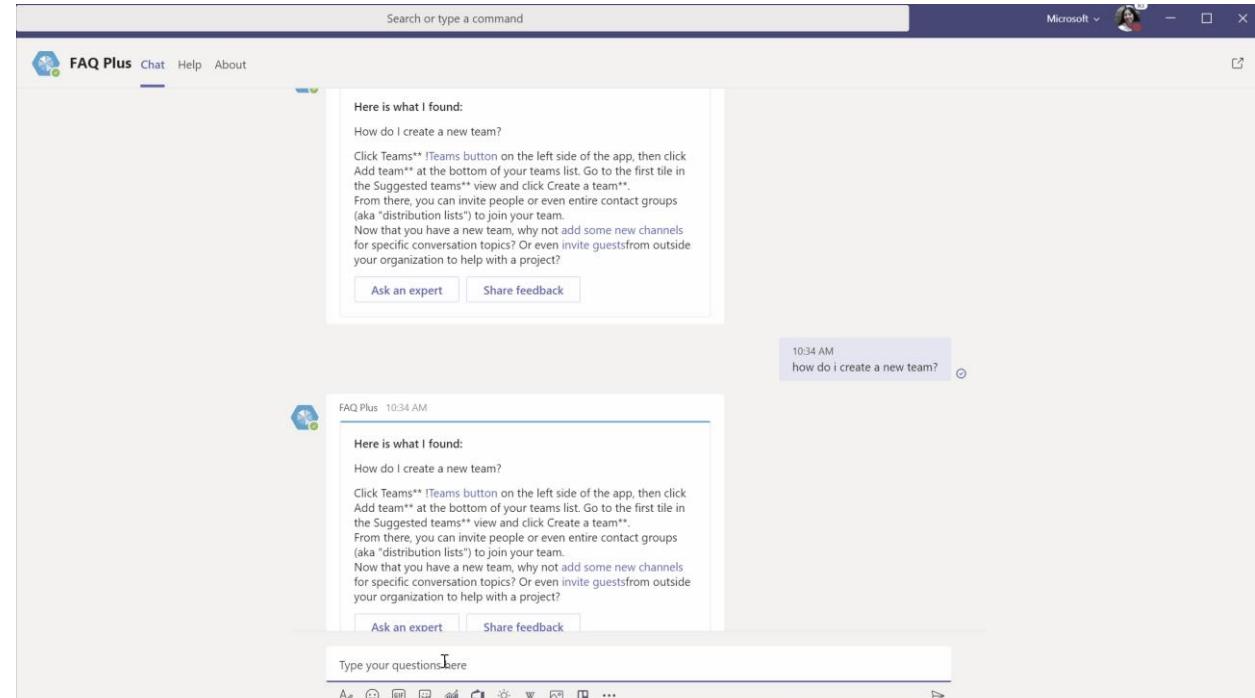
Conversational bots in Teams apps allow users to interact with the web service through text, interactive cards, and task modules.

- Types of conversations:
 - In a channel.
 - In a group chat.
 - In a one-on-one chat.
- Teams App Bot Scenarios.

Bots Conversation Basics

Basic conversation is handled through the Bot Framework Connector, a single REST API to enable the bot to communicate with Teams and other channels.

- Message content.
- Receiving messages.
- Sending messages.





Teams plus Power Platform

Power Platform can accelerate development of custom tools for Teams

 Focus of today's discussion

Microsoft Teams as a Platform

Technical effort required

Ready-made apps



Low-code apps & workflows
(Power Platform)



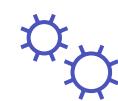
High-code custom solutions



Departmental tools



Employee resources



Processes and workflow

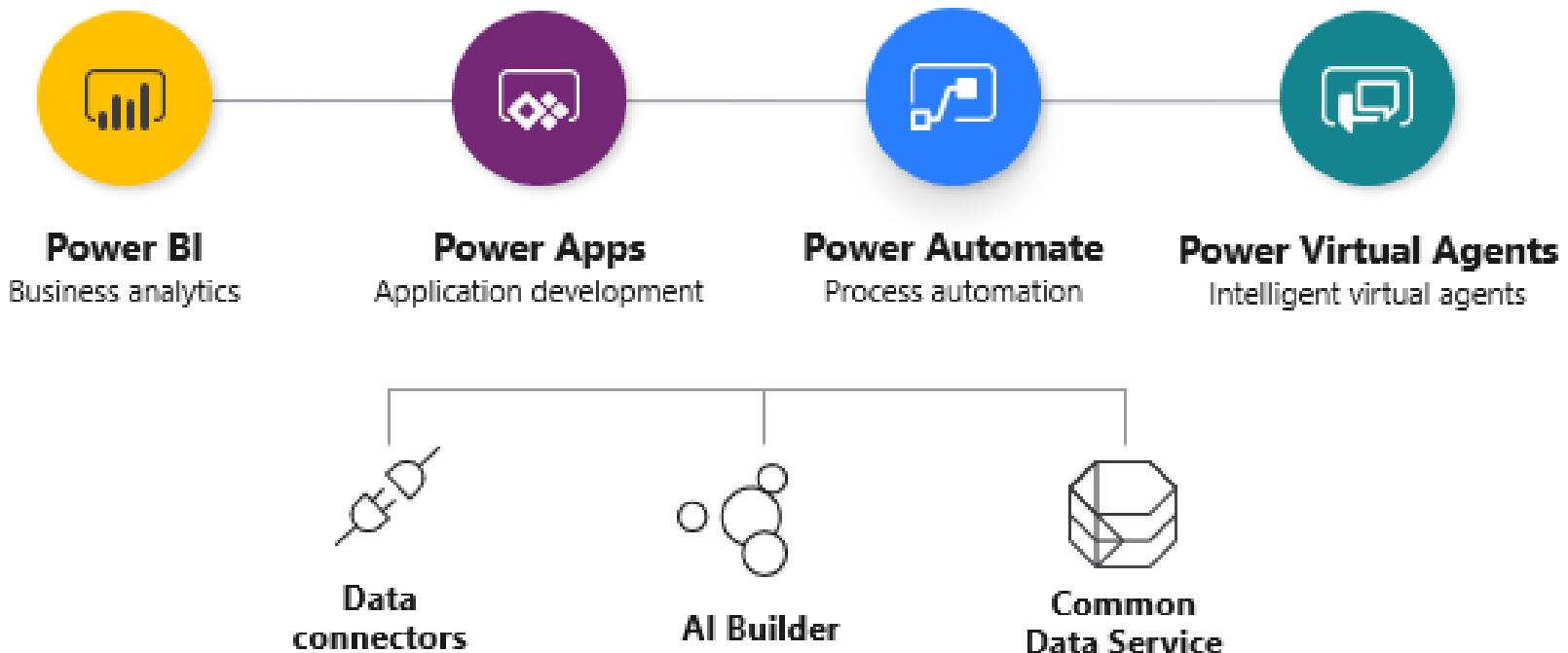


Support and info

Introduction to Power Platform

Microsoft Power Platform

Innovation anywhere. Unlocks value everywhere.





Power Apps overview

Introduction to Power Apps



Microsoft PowerApps is a service that lets you build business apps that run in a browser or on a phone or tablet, and no coding experience is required.

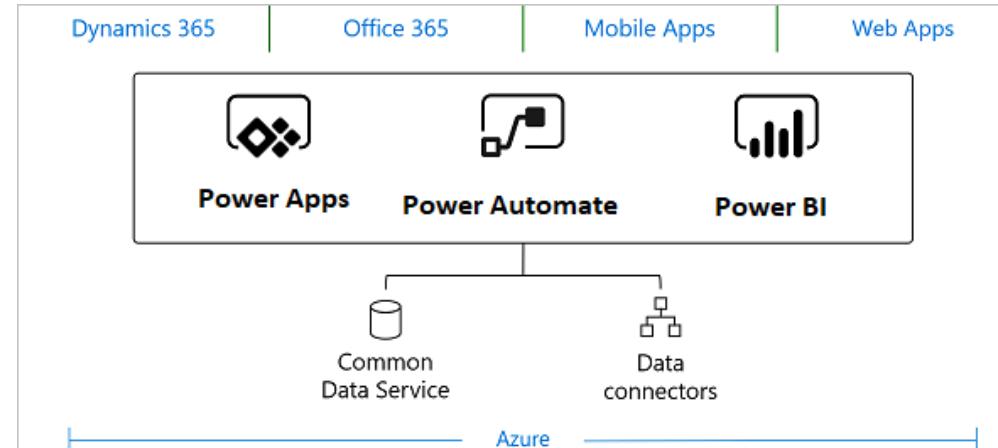
There are three types of PowerApps:

- Canvas apps
- Model-driven apps
- Portals

In this lesson, we will explore creating Canvas Apps

Connecting Power Apps to Data

- Connectors act as a bridge from the data source to the Power App (or workflow in Power Automate), thereby allowing information to be transported back and forth.
- Power Apps supports over 275 connectors for popular services and on-premises data sources, including SharePoint, SQL Server, Office 365, Salesforce, and Twitter.
- Connections to on-premises data sources, such as an on-premises SharePoint service, require the use of an on-premises data gateway.
- When an app or workflow creates a connection to a data source, the connector typically requires some form of authentication.
- The Common Data Service is a type of data source that lets you securely store and manage data that is used by business applications.



Creating a Canvas Power App

- There are three ways to start creating a canvas Power App:
 - Start from blank
 - Start from a template
 - Start from data
- When starting with a blank canvas, the app creator can add data sources to the app by connecting to data sources using connectors (for example, using the SharePoint connector) or by selecting the desired CDS entities.
- Controls are the user interface elements of the application, whose properties can be configured for appearance and behavior. The app creator builds the canvas app by adding controls. Controls may be associated with data sources.
- Saving changes to a canvas app automatically publishes them only for the app creator and anyone else who has permissions to edit the app.
- When an app creator finishes making changes, he or she must explicitly publish the changes to make them available to everyone with whom the app is shared.
- Once the app is published, the app creator can specify which users in the organization can run the app and which users can modify and even reshare the app.

Running a Power App

- The process of running a Power App varies depending on the app type.
- Canvas apps can be run on mobile devices, SharePoint Online, and Microsoft Teams.
- App creators that are part of Microsoft Partner Network can share a canvas app with customers to generate leads for their business by using [AppSource.com](#) and Power Apps Test Drive solutions.
- Model-driven apps run on a mobile device using the Dynamics 365 mobile app (the Power Apps mobile app cannot be used to run a model-driven app).
 - If a user does not have the Dynamics 365 app for phones and tablets installed, the app can be run on a web browser on a tablet, as long as the device has a sufficiently high screen resolution.

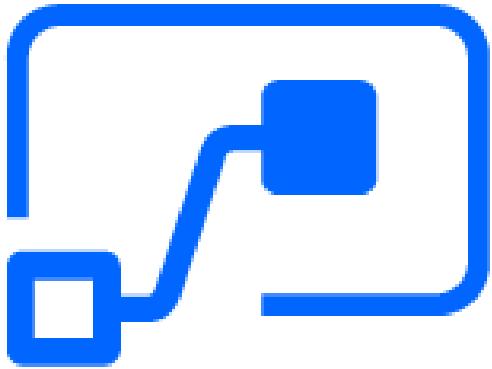
Demo

Create a Power App from a SharePoint
data source



Power Automate overview

Introduction to Power Automate



Microsoft Power Automate is an online workflow service that enables an organization to work smarter and more efficiently by automating workflows across its most common apps and services. Workflow examples include:

- Instantly respond to high-priority notifications or emails
- Capture, track, and follow up with new sales leads
- Copy files from one service to another
- Collect data about your business and share that information with your team
- Automate approval workflows

You can create the following types of flows in Power Automate:

- Automated flows
- Instant flows
- Scheduled flows
- Business process flows
- UI flows

Connecting Flows to Data

- Power Automate is similar to Power Apps in that it uses connectors to act as a bridge from the data source to the workflow.
 - Connectors can either be public or custom.
 - Connectors make triggers and actions available that can be used by flows.
 - Triggers are used by flows to start the execution of the workflow.
 - Actions are used by flows to perform a defined set of actions during execution.
- Some of the triggers that are more commonly used in flows include:
 - When an item is created
 - When an item is created or modified
 - When an item is deleted
- Some of the actions that are more commonly used in flows include:
 - Add an attachment
 - Copy a file
 - Check in a file
 - Check out a file

Creating a Flow

- There are three ways to start creating a flow:
 - Start from blank
 - Start from a template
 - Start from a connector
- The Power Automate Designer is a feature-rich tool that enables users to configure every detail of flow logic.
- The following elements are used when creating a flow:
 - Triggers
 - Actions
 - Dynamic content
 - Peek code
 - Functions
 - Advanced functionality
- The Flow Checker checks for any errors or warnings in your flow and calls out the actions where the errors or warnings occur.

Running a Flow

- Once a flow is saved without errors, it is active and available to run when the trigger condition is satisfied.
- Button (instant) flows run when manually triggered.
- Flows can be scheduled to run when a trigger condition occurs, or they can be run immediately by pressing the Run button.
- Flows can be shared with other users as co-owners or run-only users.
 - When a user adds another user or group as an owner of a flow, that flow becomes a team flow (which only show up under Team flows). Co-owners of the flow can also modify the flow using the connections that already exist.
 - Run-only sharing is an option when the flow is manually triggered (button flows). The user with whom the flow is shared can only run it; the user cannot edit the flow.
- Flows can be run from the following sources:
 - A flow can be executed from within SharePoint when a flow containing a SharePoint List is shared.
 - Users can create, manage, and run flows from within Microsoft Teams.
 - Power Automate flows can be created, managed, and run from the Power Automate mobile phone app.

Demo

Create a flow using Power Automate

Create a Power Automate Flow

Assign an additional owner to the Power Automate Flow
(optional)

© Copyright Microsoft Corporation. All rights reserved.

**FOR USE ONLY AS PART OF VIRTUAL TRAINING DAYS PROGRAM. THESE MATERIALS ARE NOT AUTHORIZED
FOR DISTRIBUTION, REPRODUCTION OR OTHER USE BY NON-MICROSOFT PARTIES.**

Extend Office

- Office Add-ins.
- Office JS APIs.
- Customization of Add-ins.
- Testing, debugging, and deployment options for Office Add-ins.
- Actionable messages.



Office Add-ins

Overview of the Office Add-in platform

All Office Add-ins built upon the Office Add-ins platform share a common framework. Developers use familiar web technologies such as HTML, CSS, and JavaScript to build a solution that can run in Office across multiple platforms.

- Add-ins allow developers to provide:
 - Additional functionality in Office.
 - Embedded rich, interactive objects.
- Benefits of Office Add-ins:
 - Cross-platform support.
 - Centralized deployment and distribution.
 - Public availability via AppSource.
 - Built using standard web technology

Overview of the Office Add-in platform

- Components of an Office Add-in:
 - Manifest.
 - Web application .
- Supported Office Add-in types, hosts, and platforms:
 - Extend functionality
 - Create new objects

Overview of the Office Add-ins platform

Office Add-ins run in various Office applications on certain platforms.

The following table lists Office applications, platforms, and add-in types.

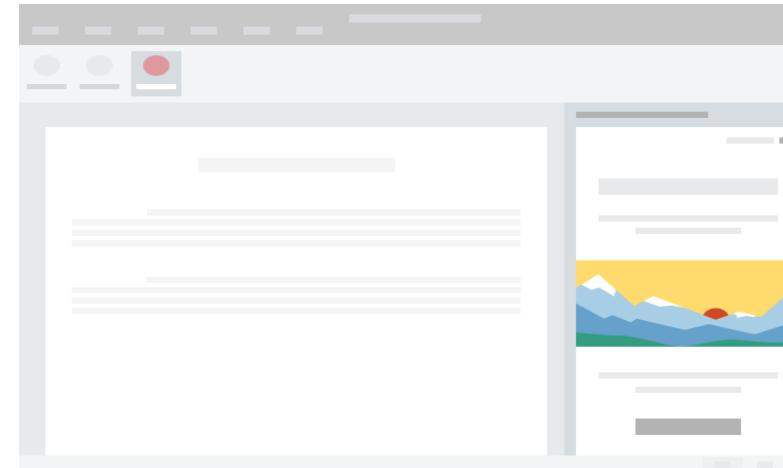
Office application	Supported platforms	Available add-in types
Excel	Windows, Mac, iPad, web browser	task pane, content, custom functions (excluding iPad)
OneNote	web browser	task pane
Outlook	Windows, Mac, iOS, Android, web browser	task pane, contextual, UI-less functions, module (Windows only)
PowerPoint	Windows, Mac, iPad, web browser	task pane, content
Project	Windows	task pane
Word	Windows, Mac, iPad, web browser	task pane

Task panes and content Office Add-ins

Office Add-ins provide several options for how your solution can interact with an Office application.

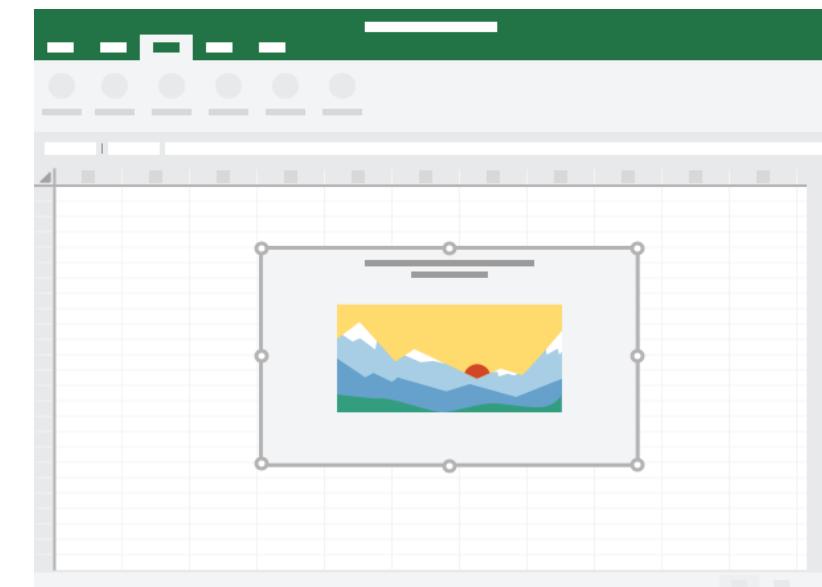
- Task pane Add-ins

Allow interaction through panel displayed within Office application.



- Content Add-Ins

Used to insert object into Excel or PowerPoint presentation.

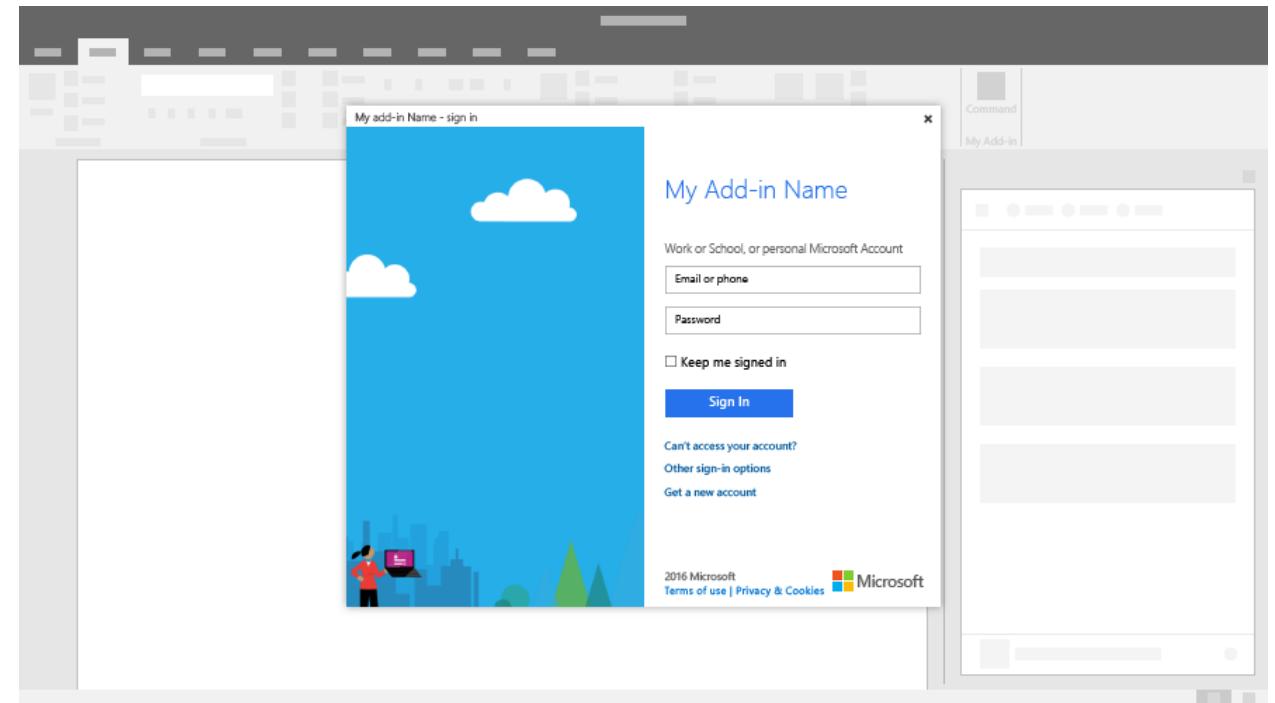


Dialog boxes in Office Add-ins

Dialog boxes are surfaces that float above the active Office application window.

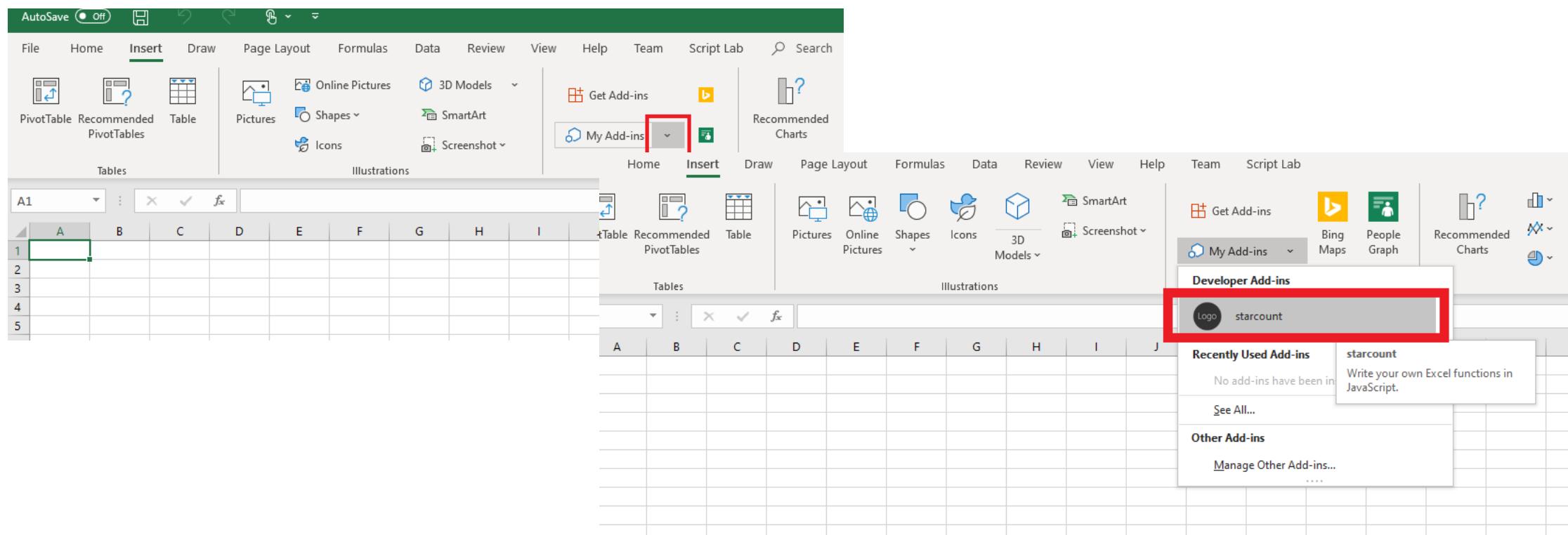
- Office Add-ins platform enables to display a dialog for users to:
 - Sign into an integrated service.
 - Confirm the user's action.
 - Run a task that might be confined in a task pane.
- Dialog window is not modal

User can continue to interact with Office application and the add-in while dialog window is displayed.



Custom functions

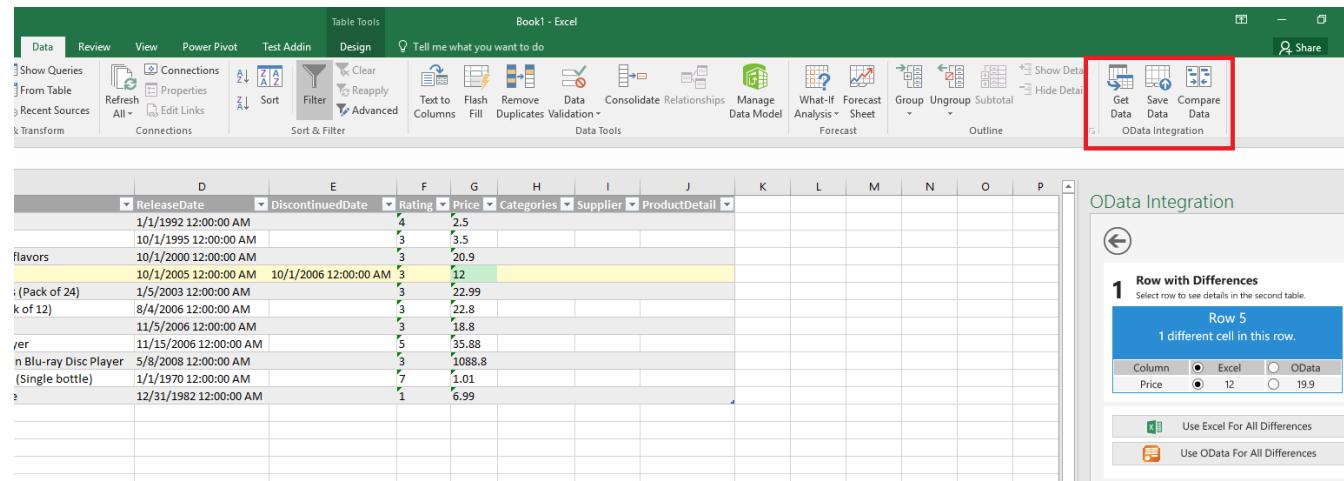
Custom functions enable developers to add new functions to Excel by defining those functions in JavaScript as part of an Add-in.



Add-in commands

Add-in commands are UI elements that extend the Office UI and start actions in the developer's Add-in.

- Add-in commands in Excel, Word, PowerPoint, and OneNote.
- Add-in commands in Outlook.



Demo

Understanding fundamental components
and types of Office Add-ins



Office JavaScript APIs

Office Add-in programming model

The Office Add-in programming model relies on two JavaScript object models:

- Host-specific JavaScript API
 - Host-specific APIs for Excel and Word providing strongly-typed objects that you can use to access specific elements in host application.
- Common API
 - Enables you to access features such as:
 - UI.
 - Dialogs.
 - Client settings that are common across multiple types of Office applications.

Office Add-in developer tools

- Office JavaScript API requirement sets.
- Use Office JavaScript APIs.
- Make asynchronous calls using proxy objects.

```
$ yo office

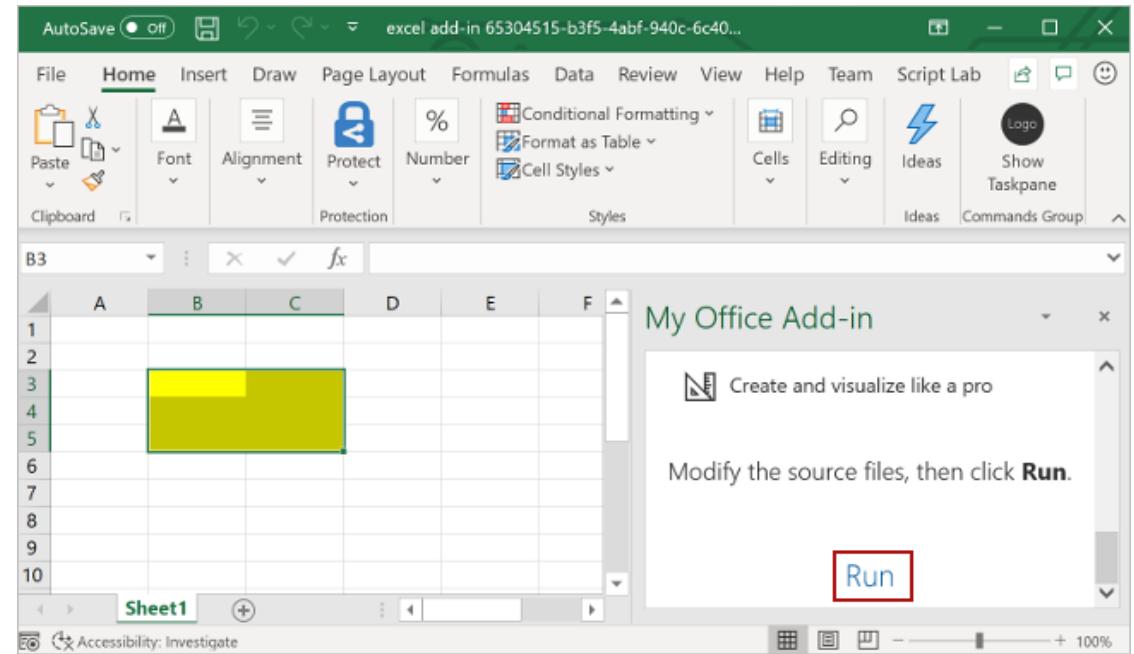
  _----_
  |--(o)--|
  \-----/
    ( _ ^U^ _ )
     /__A__\ \
      |   |
      '-'-'-
      | o | Y |
      '-'-'-

Welcome to the Office
Add-in generator, by
@OfficeDev! Let's create
a project together!

? Choose a project type: Office Add-in Task Pane project
? Choose a script type: Javascript
? What do you want to name your add-in? My Office Add-in
? Which Office client application would you like to support? Excel
```

Excel JavaScript API

- Object model:
 - A Workbook contains one or more Worksheets.
 - A Worksheet gives access to cells through Range objects.
 - A Range represents a group of contiguous cells.
 - Ranges are used to create and place Tables, Charts, Shapes, and other data visualization or organization objects.
 - A Worksheet contains collections of those data objects that are present in the individual sheet.
 - Workbooks contain collections of some of those data objects (such as Tables) for the entire Workbook.
- Ranges.
- Charts, tables and other data objects.



Outlook JavaScript API

Outlook JavaScript APIs give Add-ins access to the user's mailbox, messages, and appointments in Outlook.

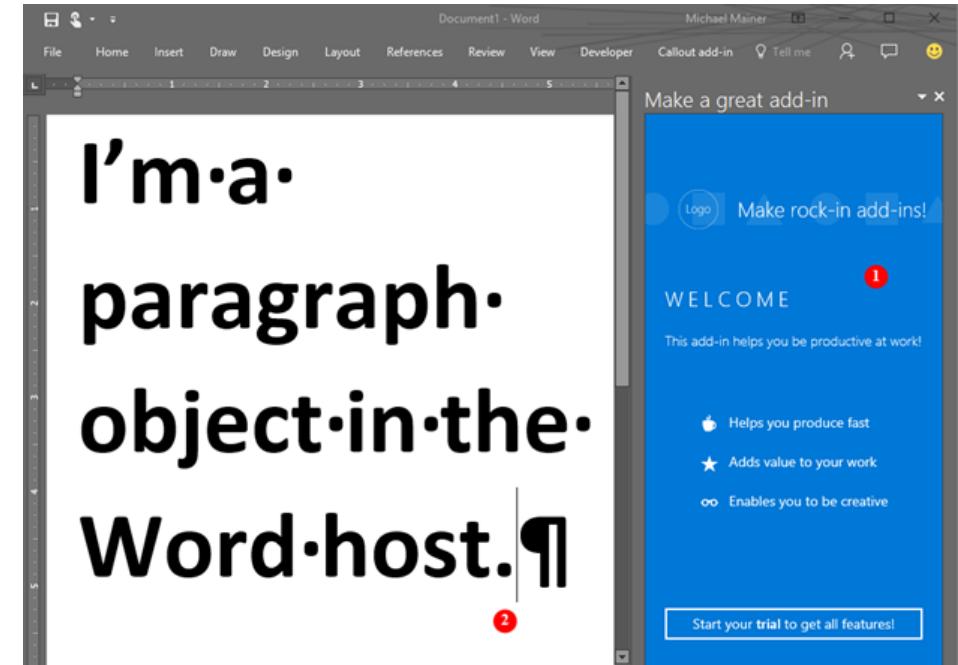
- Object model:
 - Mailbox object.
 - Item object.
- Key features.
- Module Add-ins.

Item type	Modes
Message	<ul style="list-style-type: none">• Read• Compose
Appointment/meeting	<ul style="list-style-type: none">• Attendee (read)• Organizer (compose)

Word JavaScript API

The Word JavaScript APIs give Add-ins access to Word documents.

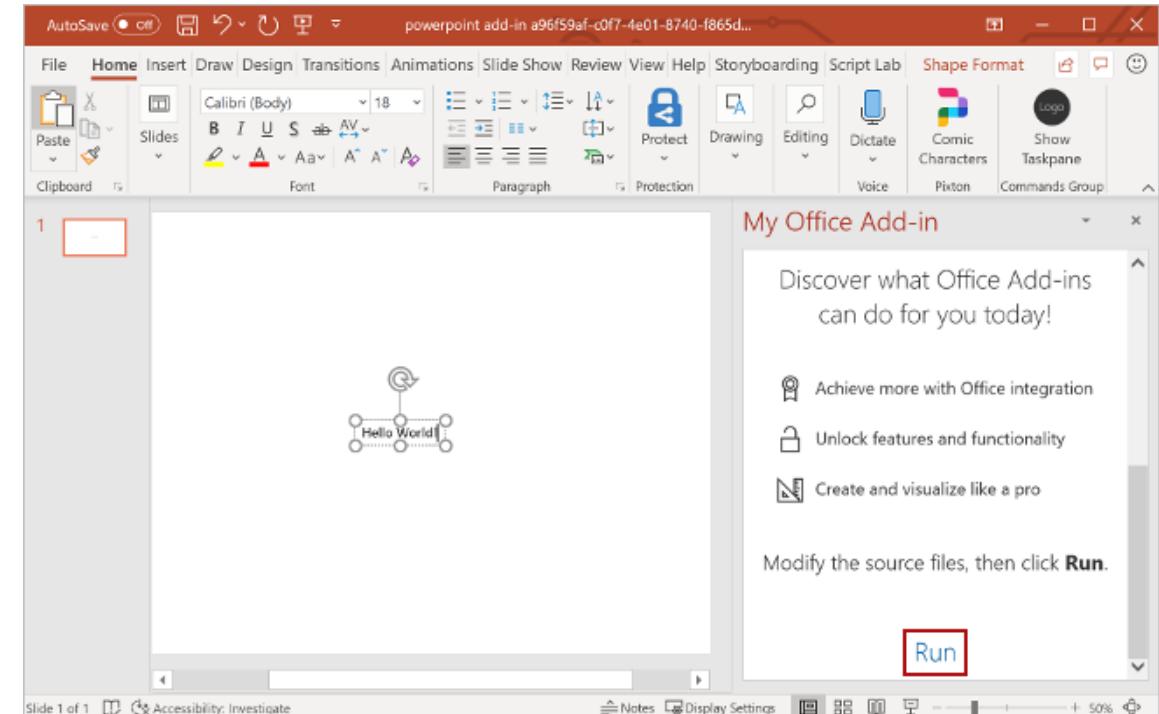
- Object model:
 - A document contains a body and at least one section.
 - A body can contain:
 - Paragraph
 - Table
 - List
 - Text
 - Objects
- Work with document text.
- Search.



PowerPoint JavaScript API

With PowerPoint add-ins, you can use familiar web technologies such as HTML, CSS, and JavaScript to build a solution that can run in PowerPoint across multiple platforms.

- Object model:
 - PowerPoint JavaScript API.
 - Common APIs.
- PowerPoint Add-Ins.
- PowerPoint add-in scenarios.
- ActiveViewChanged event.
- Navigation.
- Get the URL.
- Create a presentation.



Custom functions

Custom functions have several unique capabilities and restrictions because they run in a separate runtime from other add-in interactions, like task panes.

Users within Excel can access custom functions just as any native function in Excel, such as SUM().

- You can create custom functions that:
 - Perform simple tasks calculations.
 - Or more complex tasks such as streaming real-time data from the web into a worksheet.

Demo

Understanding Office JavaScript APIs



Customization of Add-ins

Overview of customizing Office Add-ins

Office Add-ins extend the Office experience by providing contextual functionality that users can access within Office clients.

- Developers consider the following design factors:
 - Design explicitly for Office.
 - Focus on a few key tasks; do them well.
 - Favor content over chrome.
 - Make it enjoyable and keep users in control.
 - Design for all platforms and input methods.

Persisting add-in state and settings

- Office Add-ins platform provides several ways for your add-in to persist state and settings. Your options depend on the Office applications you plan to support and the type of add-in you plan to develop.
- Options to persist state and settings.

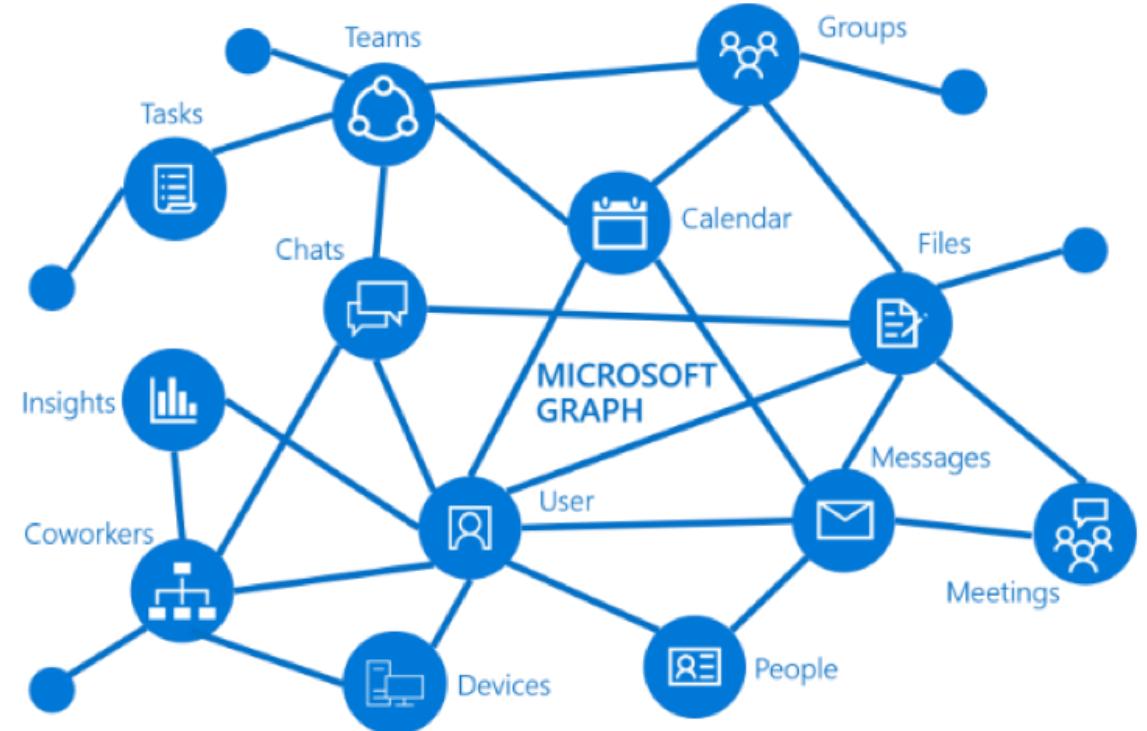
Office UI Fabric in Office Add-ins

As you build your add-in, you have UI design factors to consider.

- Office UI Fabric has two main areas:
 - Fabric Core provides basic elements like font, icons, and color.
 - Fabric React components includes Fabric Core elements and adds input, navigation, and notification components, among others.
- Fabric components:
 - Breadcrumb
 - Button
 - Checkbox
 - ChoiceGroup
 - Dropdown
 - Label
 - List
 - Pivot
 - TextField
 - Toggle

Microsoft Graph in Office Add-ins

- Integrating data from online service providers increases the value and adoption of add-ins.
- Microsoft Graph REST APIs provide a way for your add-in to access user's data in services like:
 - Azure Active Directory.
 - Office 365 services
 - Enterprise Mobility and Security services
 - Windows 10 services.
 - Dynamics 365.



Demo

Understanding customization of Add-ins



Testing, debugging, and deployment options for Office Add-ins

Deployment options for Office Add-in

Factors to consider:

Consider...	Examples
Add-in lifecycle stage	local developer testing, ready for public use
Add-in interaction or feature support	task pane add-in, content add-in, add-in commands
Target Office applications	Excel, Outlook
Target platforms	Windows, Mac
Scope of user base	your organization, general public

Deployment options for Office Add-in

Developers have several options for deploying an Add-in.

- Deployment options.
 - Centralized deployment via the Office 365 admin center.
 - SharePoint app catalog deployment.
 - Outlook add-in deployment.
- Deployment options by Office host.
 - Word, Excel, and PowerPoint.
 - Outlook.

Testing and debugging for Office Add-ins

At various points during your add-in's life cycle, developers need to verify functionality and fix bugs.

Option	Description	Best when...
Sideload	Install your add-in locally.	Developer building and testing add-in
Centralized deployment	Distribute your add-in to users via the Microsoft 365 admin center.	Add-in ready for use in your organization on Office 365 or in a hybrid environment
SharePoint catalog	Distribute add-in to users via SharePoint.	Task pane or content add-in ready for use in your organization that's using an on-premises environment; Excel, Word, or PowerPoint is targeted but Mac isn't a target platform
AppSource	Make add-in available to the public.	Add-in ready for public use
Exchange server	Distribute add-in to users via Exchange.	Outlook add-in ready for use in an organization whose environment doesn't use Azure Active Directory identity service
Network share	Make add-in available to network users via a shared folder.	Add-in development and users are on Windows

Testing and debugging for Office Add-ins

- Sideload an Office Add-in.
- Debug an Office Add-in.
 - A web browser with the browser's built-in developer tools.
 - Visual Studio, provided you prepared your add-in using this IDE.
 - Visual Studio Code for custom functions projects only.
 - Runtime logging on Windows and Mac.
- Validate the manifest file.
- Test required Office clients and platforms.



Actionable messages

Overview of actionable messages

Actionable messages enable users to complete the task on messages from services they use every day right from Outlook. Actionable messages can be posted via a group or inbox connector or can be sent directly over email.

Actionable messages via email:

- The recipient must be an individual, not a group.
- The recipient must be visible on the message. Do not put the recipient in the BCC field.
- The recipient must have a mailbox on Outlook.com or Exchange Online in Office 365.

Actionable message with an adaptive card

Adaptive cards communicate information immediately, and their user interface allows users to view and interact with the cards. Adaptive cards allow users to provide a quick response without having to open an app.

- Adaptive card layout:
 - Title.
 - Descriptive text.
 - One or more columns of detail.
- Refresh adaptive cards:
 - Approval scenario.
 - Task status.
 - Survey.

Demo

Understanding actionable messages

Thanks for joining us!

Resources for Microsoft 365 developers



You can find Microsoft 365 developer training modules on [Microsoft Learn!](#)



Become Microsoft 365 Certified! Earn a Developer Associate certification
[Exam MS-600](#): Building Applications and Solutions with Microsoft 365 Core Services (beta)



Check-out the Microsoft 365 Developer Center! Get the latest
Microsoft 365 developer documentation, trainings, blog posts and
more. <https://developer.Microsoft.com/Microsoft-365>