# Exam MS-600: Building Applications and Solutions with Microsoft 365 Core Services – Skills Measured

**This exam was updated on March 23, 2021. Following the current exam guide, we have included a version of the exam guide with Track Changes set to "On," showing the changes that were made to the exam on that date.**

## Audience Profile

Candidates for this exam are Microsoft 365 Developers who design, build, test, and maintain applications and solutions that are optimized for the productivity and collaboration needs of organizations using the Microsoft 365 platform.

Candidates for this exam are proficient in Microsoft identity and Microsoft Graph. They have general knowledge on UI elements (including Adaptive Cards and Office UI Fabric (Fluent UI)), integration points (including Microsoft Teams, Office Add-ins, SharePoint Framework, Actionable Messages), and determining workload platform targets.

Candidates should have experience developing solutions on Microsoft Teams, Office Add-ins, or SharePoint Framework through all phases of software development. They should have a basic understanding of REST APIs, JSON, OAuth2, OData, OpenID Connect, Microsoft identities (including Azure AD and Microsoft accounts), Azure AD B2C, and permission/consent concepts.

## Skills Measured

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is NOT definitive or exhaustive.

NOTE: Most questions cover features that are General Availability (GA). The exam may contain questions on Preview features if those features are commonly used.

## Implement Microsoft Identity (20-25%)

### Register an Application

- determine the supported account type
- select authentication and client credentials for app type and authentication flow
- define app roles

### Implement Authentication

- configure the JavaScript implementation of Microsoft Authentication Library (MSAL) for endpoint and token cache

- plan and configure scopes for dynamic or static permission
- use the MSAL (JavaScript) login method

**Configure Permissions to Consume an API**

- configure Delegated permissions for the app
- configure Application permissions for the app
- identify admin consent requirements

**Implement Authorization to Consume an API**

- configure incremental consent scopes
- call MSAL (JavaScript) using AquireTokenSilent/AquireToken pattern

**Implement Authorization in an API**

- validate Access Token
- configure effective permissions for delegated scopes
- implement app permissions using roles
- use a delegated access token to call a Microsoft API

**Create a Service to Access Microsoft Graph**

- configure client credentials using a certificate
- configure Application permissions for the app
- acquire an access token for Microsoft Graph using an application permission and client credential certificate
- acquire an access token using the client secret

## Build Apps with Microsoft Graph (15-20%)

**Optimize Data Usage with query parameters**

- use $filter query parameter
- use $select query parameter
- order results using $orderby query parameter
- set page size of results using $skip and $top query parameters
- expand and retrieve resources using $expand query parameter
- retrieve the total count of matching resources using $count query parameter
- search for resources using $search query parameter
- determine the appropriate Microsoft Graph SDK to leverage

**Optimize network traffic**

- monitor for changes using change notifications
- combine multiple requests using $batch
- get changes using a delta query
- detect and handle throttling

**Access User data from Microsoft Graph**

- get the signed in user's profile
- get a list of users in the organization
- get the user's profile photo
- get the user object based on the user's unique identifier
- get the user's manager profile

**Access Files with Microsoft Graph**

- get the list of files in the signed in user's OneDrive for Business
- download a file from the signed in user's OneDrive for Business using file unique id
- download a file from a SharePoint Online Site using the relative path to the file
- get the list of files trending around the signed in user
- upload a large file to OneDrive for Business
- get a user object from an owner list in a group and retrieve that user’s files

**Manage a group lifecycle on Microsoft Graph**

- get the information on a group by id
- get the list of members in a group
- get the list of owners of a group
- get the list of groups where the signed in user is a member
- get the list of groups where the signed in user is an owner
- provision a group
- provision a Team with a group
- delete a group

## Extend and Customize SharePoint (15-20%)

**Describe the components of a SharePoint Framework (SPFx) web part**

- identify the appropriate tool to create an SPFx Web Part project
- describe properties of client-side web parts
- describe Office UI Fabric (Fluent UI) in client-side web parts
- describe when to use an app page
- differentiate between app page and web part

- describe rendering framework options
- describe branding and theming in SharePoint Online

**Describe SPFx extensions**

- identify the appropriate tool to create an SPFx Extension project
- describe page placeholders from Application Customizer
- describe the ListView Command Set extension
- describe the Field Customizer extension

**Describe the process to package and deploy an SPFx solution**

- describe the options for preparing a package for deployment
- describe the options for packaging a solution
- describe the requirements of tenant-scoped solution deployment
- describe the requirements of domain isolated web parts
- describe the options to deploy a solution
- describe how to build a Microsoft Teams tab by using SPFx

**Describe the consumption of Microsoft Graph**

- describe the purpose of the MSGraphClient object
- describe the methods for granting permissions to Microsoft Graph

**Describe the consumption of third-party APIs secured with Azure AD from within SPFx**

- describe the purpose of the AadHttpClient object
- describe the methods for granting permissions to consume a third-party API

**Describe Web Parts as Teams Tabs**

- describe the considerations for creating a SPFx Web Part to be a Teams Tab
- describe the options for deploying a SPFx Web Part as a Teams Tab

## Extend Teams (20-25%)

**Create a Microsoft Teams app manifest**

- configure an app manifest using App Studio
- manually create an app manifest to deploy a SPFx Web Part to Teams
- create an app package for a Microsoft Teams app

**Deploy a Teams app**

- describe the options for deploying a Teams app
- sideload an app in Microsoft Teams
- publish a Teams app to an organization app catalog

**Create and use task modules**

- create a card-based task module
- create an iframe-based task module
- invoke a task module from a tab
- invoke a task module from a bot
- chain task module invocations

**Create a webhook**

- create an outgoing webhook
- create an incoming webhook

**Implement custom Teams tabs**

- create a personal tab
- create a channel/group tab
- create a tab with a deep link
- add authentication to a tab

**Create a messaging extension**

- create a search command extension
- create an action command extension using an adaptive card
- create an action command extension using parameters

**Create a conversational Bot**

- create a personal bot
- create a group/channel bot
- use proactive messaging with a bot
- send actionable messages from a bot
- add authentication to a bot

## Extend Office (15-20%)

**Describe fundamental components and types of Office Add-ins**

- describe task pane and content Office Add-ins

- describe dialog boxes
- describe custom functions
- describe Add-in commands
- describe the purpose of Office Add-ins manifest

**Describe Office JS APIs**

- describe the Office Add-in programming model
- describe Office Add-in developer tools
- describe the capabilities of the Excel JavaScript API
- describe the capabilities of the Outlook JavaScript API
- describe the capabilities of the Word JavaScript API
- describe the capabilities of the PowerPoint JavaScript API
- describe the capabilities of custom functions

**Describe customization of Add-ins**

- describe the options for persisting state and settings
- describe Office UI Fabric (Fluent UI) in Office Add-ins
- describe when to use Microsoft Graph in Office Add-ins
- describe authorization when using Microsoft Graph in Office Add-ins

**Describe testing, debugging, and deployment options**

- select deployment options based on requirements
- describe testing and debugging concepts for Office Add-ins

**Describe actionable messages**

- describe the features of actionable messages with an adaptive card
- describe the scenarios for refreshing an actionable message

**The exam guide below shows the changes that were implemented on March 23, 2021.**

## Audience Profile

Candidates for this exam are Microsoft 365 Developers who design, build, test, and maintain applications and solutions that are optimized for the productivity and collaboration needs of organizations using the Microsoft 365 platform.

Candidates for this exam are proficient in Microsoft identity and Microsoft Graph. They have general knowledge on UI elements (including Adaptive Cards and Office UI Fabric (Fluent UI)), integration points (including Microsoft Teams, Office Add-ins, SharePoint Framework, Actionable Messages), and determining workload platform targets.

Candidates should have experience developing solutions on Microsoft Teams, Office Add-ins, or SharePoint Framework through all phases of software development. They should have a basic understanding of REST APIs, JSON, OAuth2, OData, OpenID Connect, Microsoft identities (including Azure AD and Microsoft accounts), Azure AD B2C, and permission/consent concepts.

## Skills Measured

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is NOT definitive or exhaustive.

NOTE: Most questions cover features that are General Availability (GA). The exam may contain questions on Preview features if those features are commonly used.

## Implement Microsoft Identity (20-25%)

**Register an Application**

- determine the supported account type
- select authentication and client credentials for app type and authentication flow
- define app roles

**Implement Authentication**

- configure the JavaScript implementation of Microsoft Authentication Library (MSAL) for endpoint and token cache
- plan and configure scopes for dynamic or static permission
- use the MSAL (JavaScript) login method

**Configure Permissions to Consume an API**

- configure Delegated permissions for the app
- configure Application permissions for the app
- identify admin consent requirements

**Implement Authorization to Consume an API**

- configure incremental consent scopes
- call MSAL (JavaScript) using AquireTokenSilent/AquireToken pattern

### Implement Authorization in an API

- validate Access Token
- configure effective permissions for delegated scopes
- implement app permissions using roles
- use a delegated access token to call a Microsoft API

### Create a Service to Access Microsoft Graph

- configure client credentials using a certificate
- configure Application permissions for the app
- acquire an access token for Microsoft Graph using an application permission and client credential certificate
- acquire an access token using the client secret

## Build Apps with Microsoft Graph (15-20%)

### Optimize Data Usage with query parameters

- use $filter query parameter
- use $select query parameter
- order results using $orderby query parameter
- set page size of results using $skip and $top query parameters
- expand and retrieve resources using $expand query parameter
- retrieve the total count of matching resources using $count query parameter
- search for resources using $search query parameter
- determine the appropriate Microsoft Graph SDK to leverage

### Optimize network traffic

- monitor for changes using change notifications
- combine multiple requests using $batch
- get changes using a delta query
- detect and handle throttling

### Access User data from Microsoft Graph

- get the signed in user's profile
- get a list of users in the organization
- get the user's profile photo
- get the user object based on the user's unique identifier
- get the user's manager profile

### Access Files with Microsoft Graph

- get the list of files in the signed in user's OneDrive for Business
- download a file from the signed in user's OneDrive for Business using file unique id
- download a file from a SharePoint Online Site using the relative path to the file
- get the list of files trending around the signed in user
- upload a large file to OneDrive for Business
- get a user object from an owner list in a group and retrieve that user's files

### Manage a group lifecycle on Microsoft Graph

- get the information on a group by id
- get the list of members in a group
- get the list of owners of a group
- get the list of groups where the signed in user is a member
- get the list of groups where the signed in user is an owner
- provision a group
- provision a Team with a group
- delete a group

## Extend and Customize SharePoint (15-20%)

### Describe the components of a SharePoint Framework (SPFx) web part

- identify the appropriate tool to create an SPFx Web Part project
- describe properties of client-side web parts
- describe Office UI Fabric (Fluent UI) in client-side web parts
- describe when to use an app page
- differentiate between app page and web part
- describe rendering framework options
- describe branding and theming in SharePoint Online

### Describe SPFx extensions

- identify the appropriate tool to create an SPFx Extension project
- describe page placeholders from Application Customizer
- describe the ListView Command Set extension
- describe the Field Customizer extension

### Describe the process to package and deploy an SPFx solution

- describe the options for preparing a package for deployment
- describe the options for packaging a solution

- describe the requirements of tenant-scoped solution deployment
- describe the requirements of domain isolated web parts
- describe the options to deploy a solution
- describe how to build a Microsoft Teams tab by using SPFx

**Describe the consumption of Microsoft Graph**

- describe the purpose of the MSGraphClient object
- describe the methods for granting permissions to Microsoft Graph

**Describe the consumption of third-party APIs secured with Azure AD from within SPFx**

- describe the purpose of the AadHttpClient object
- describe the methods for granting permissions to consume a third-party API

**Describe Web Parts as Teams Tabs**

- describe the considerations for creating a SPFx Web Part to be a Teams Tab
- describe the options for deploying a SPFx Web Part as a Teams Tab

## Extend Teams (20-25%)

**Create a Microsoft Teams app manifest**

- configure an app manifest using App Studio
- manually create an app manifest to deploy a SPFx Web Part to Teams
- create an app package for a Microsoft Teams app

**Deploy a Teams app**

- describe the options for deploying a Teams app
- sideload an app in Microsoft Teams
- publish a Teams app to an organization app catalog

**Create and use task modules**

- create a card-based task module
- create an iframe-based task module
- invoke a task module from a tab
- invoke a task module from a bot
- chain task module invocations

**Create a webhook**

- create an outgoing webhook
- create an incoming webhook

**Implement custom Teams tabs**

- create a personal tab
- create a channel/group tab
- create a tab with a deep link
- add authentication to a tab

**Create a messaging extension**

- create a search command extension
- create an action command extension using an adaptive card
- create an action command extension using parameters

**Create a conversational Bot**

- create a personal bot
- create a group/channel bot
- use proactive messaging with a bot
- send actionable messages from a bot
- add authentication to a bot

## Extend Office (15-20%)

**Describe fundamental components and types of Office Add-ins**

- describe task pane and content Office Add-ins
- describe dialog boxes
- describe custom functions
- describe Add-in commands
- describe the purpose of Office Add-ins manifest

**Describe Office JS APIs**

- describe the Office Add-in programming model
- describe Office Add-in developer tools
- describe the capabilities of the Excel JavaScript API
- describe the capabilities of the Outlook JavaScript API
- describe the capabilities of the Word JavaScript API
- describe the capabilities of the PowerPoint JavaScript API

- describe the capabilities of custom functions

**Describe customization of Add-ins**

- describe the options ~~of~~ for persisting state and settings
- describe Office UI Fabric (Fluent UI) in Office Add-ins
- describe when to use Microsoft Graph in Office Add-ins
- describe authorization when using Microsoft Graph in Office Add-ins

**Describe testing, debugging, and deployment options**

- select deployment options based on requirements
- describe testing and debugging concepts for Office Add-ins

**Describe actionable messages**

- describe the features of actionable messages with an adaptive card
- describe the scenarios for refreshing an actionable message