Study guide for Exam MS-600: Building Applications and Solutions with Microsoft 365 Core Services

Purpose of this document

This study guide should help you understand what to expect on the exam and includes a summary of the topics the exam might cover and links to additional resources. The information and materials in this document should help you focus your studies as you prepare for the exam.

Useful links	Description
Review the skills measured as of October 31, 2022	This list represents the skills measured AFTER the date provided. Study this list if you plan to take the exam AFTER that date.
Review the skills measured prior to October 31, 2022	Study this list of skills if you take your exam PRIOR to the date provided.
Change log	You can go directly to the change log if you want to see the changes that will be made on the date provided.
How to earn the certification	Some certifications only require passing one exam, while others require passing multiple exams.
Certification renewal	Microsoft associate, expert, and specialty certifications expire annually. You can renew by passing a **free** online assessment on Microsoft Learn.
Your Microsoft Learn profile	Connecting your certification profile to Learn allows you to schedule and renew exams and share and print certificates.
Passing score	A score of 700 or greater is required to pass.
Exam sandbox	You can explore the exam environment by visiting our exam sandbox.

Useful links	Description
Request accommodations	If you use assistive devices, require extra time, or need modification to any part of the exam experience, you can request an accommodation.
Take a practice test	Are you ready to take the exam or do you need to study a bit more?

Updates to the exam

Our exams are updated periodically to reflect skills that are required to perform a role. We have included two versions of the Skills Measured objectives depending on when you are taking the exam.

We always update the English language version of the exam first. Some exams are localized into other languages, and those are updated approximately eight weeks after the English version is updated. Other available languages are listed in the **Schedule Exam** section of the **Exam Details** webpage. If the exam isn't available in your preferred language, you can request an additional 30 minutes to complete the exam.

Note

The bullets that follow each of the skills measured are intended to illustrate how we are assessing that skill. Related topics may be covered in the exam.

Note

Most questions cover features that are general availability (GA). The exam may contain questions on Preview features if those features are commonly used.

Skills measured as of October 31, 2022

Audience Profile

Microsoft Teams application developers design, build, test, and maintain modern enterprise-grade applications and solutions with Microsoft Teams that are optimized for the productivity and collaboration needs of organizations using the Microsoft 365 platform.

Candidates for this exam are proficient in designing collaboration solutions using the Microsoft identity platform, Microsoft Graph, SharePoint Framework, and Microsoft Power Platform to build custom apps that extend Microsoft Teams, SharePoint, and Viva Connections.

Candidates should have a working knowledge of developing solutions with Microsoft Teams, Microsoft Power Platform, Dataverse for Teams, and/or SharePoint Framework through all phases of software development. They should have a basic understanding of modern web development technologies, source control and CI/CD strategies, Azure compute, data, and messaging workloads, and the Microsoft identity platform, including OAuth2, OData, OpenID Connect, and permission/consent concepts.

- Design collaborative app solutions and existing app integration (5–10%)
- Implement Microsoft identity (15–20%)
- Build apps with Microsoft Graph (10–15%)
- Extend and customize Microsoft 365 with SharePoint Framework (15–20%)
- Extend Microsoft Teams (30–35%)
- Test, deploy, monitor, and maintain modern enterprise-grade app solutions (5–10%)

Design collaborative app solutions and existing app integration (5–10%)

Assess Microsoft Teams and existing app integration opportunities

- Assess current Teams adoption level using M365 Maturity Model
- Identify opportunities for no-code or low-code development (Microsoft Power Platform, Azure Logic Apps, Dataverse for Teams)
- Assess Teams integration opportunities with existing applications

Design development and production environments

- Design test, acceptance, production, and deployment strategy to meet solution requirements
- Design development environment by using Microsoft 365 developer tenant
- Recommend Azure services for Teams app use cases (serverless compute workflows)
- Specify app requirements for CI/CD integration
- Develop an appropriate tooling strategy for a Teams solution, including the Teams Toolkit vs Yeoman Teams Generator
- Select an appropriate development framework and runtime platform for a Teams solution (including Teams Toolkit for React or Teams Toolkit Blazor)
- Select an appropriate runtime platform for a Teams solution (including Node.js, .NET, and/or SPFx)

Design solution based on technical and business requirements

- Design authentication scenarios, including for guest users and multitenant authentication
- Design UI by using Microsoft Teams UI Library, Fluent UI React Northstar, Adaptive Cards, and Microsoft Teams UI Kit for Figma

Implement Microsoft identity (15–20%)

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
- Implement single sign-on authentication for a bot
- Implement single sign-on authentication for a tab

Configure permissions to consume an API

- Configure delegated permissions for the app
- Configure application permissions for the app
- Specify and implement consent requirements
- Grant consent to an application

Implement authorization to expose or consume an API

- Configure incremental consent scopes
- Grant scoped permissions to API
- Call MSAL (JavaScript) using AcquireTokenSilent/AcquireToken pattern

Implement authorization in an API

- Validate access token
- Configure effective permissions for delegated scopes
- Implement app permissions by 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
- Consume certificates from Key Vault, including using managed identities
- Configure application permissions for the app
- Acquire an access token for Microsoft Graph using an application permission and client credential certificate

Build apps with Microsoft Graph (10–15%)

Access data from Microsoft Graph

- Determine whether to retrieve data using a Microsoft Graph SDK or the Graph REST API
- Get the signed-in user's profile
- Get a list of users in the organization
- Get the user object based on the user’s unique identifier
- Access user data from Microsoft Graph
- Create, read, update, and delete data

- Traverse Microsoft Graph to retrieve data
- Upload files larger than 10 mb to Teams
- Implement Microsoft Graph Toolkit

Customize responses by using query parameters

- Use query parameters, including $filter, $select, $orderby, $count, $skip, and $stop
- Expand and retrieve resources using $expand query parameter
- Search for resources using $search query parameter
- Access paged date using $odata.nextLink, $skip, and $skiptoken parameters
- Test queries and responses by using Microsoft Graph Explorer

Optimize usage of Microsoft Graph resources

- Monitor for changes using change notifications
- Combine multiple requests using $batch
- Get changes using a delta query
- Detect and handle throttling

Manage a team lifecycle by using Microsoft Graph

- Get the information on a team by ID
- Get the list of members in a team
- Get the list of owners of a team
- Get the list of teams where the signed in user is a member
- Provision a team using an existing Teams template
- Delete or archive a team
- Update an existing team
- Enable resource-specific consent

Extend and customize Microsoft 365 with SharePoint Framework (15–20%)

Describe the components of a SharePoint Framework (SPFx) solution

- 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
- Explain 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 Adaptive Card Extensions
- Describe Viva Connections and Viva dashboard

- Describe the types of SPFx extensions (Application Customizer, Field Customizer, Command Set 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 SPFx solution
- Describe how to build a Microsoft Teams tab by using SPFx
- Manually create an app manifest to deploy a SPFx Web Part to Teams

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 Microsoft Azure Active Directory (Azure AD), part of Microsoft Entra from within SPFx

- Describe the purpose of the AadHttpClient object
- Identify 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 Microsoft Teams (30–35%)

Create a Microsoft Teams app manifest

- Configure an app manifest using the Developer Portal
- Create an app package for a Microsoft Teams app

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
- Implement device permissions in a personal tab
- Create a channel/group tab
- Create a tab with a deep link
- Create an adaptive card 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
- Add link unfurling

Create conversational, call, and online media bots

- Create a personal bot
- Create a group/channel bot
- Send actionable messages from a bot
- Create a call bot
- Create an online media bot
- Handle incoming calls
- Transfer incoming calls

Create a meeting extension

- Describe the capabilities of Microsoft Teams meeting apps
- Enable and configure apps for Teams meetings
- Create or integrate apps within the meeting lifecycle
- Implement shared meeting stage

Test, deploy, monitor, and maintain modern enterprise-grade app solutions (5–10%)

Test apps and solutions

- Design test plans, including Bots, Connectors, Messaging extensions, Meeting extensions, SSO, and Adaptive cards
- Specify required sample data for testing
- Deploy sample data by using M365 CLI or PowerShell scripts
- Execute test plans, including Bots, Connectors, Messaging extensions, Meeting extensions, SSO, and Adaptive cards

Deploy apps and solutions

- Create or modify app definitions in the Developer Portal for Teams

- Manage Teams apps packages by using the Developer Portal for Teams
- Upload a custom app to a single user or team
- Upload an app to the App Catalog
- Upload an app to the Developer Portal for Teams
- Automate deployment of Teams apps

Monitor and maintain published apps

- Recommend audit settings using Microsoft 365 Audit logging
- Review and evaluate performance using App Insights
- Monitor Microsoft Graph API permissions by using the Microsoft 365 compliance center
- Assess new SDK versions for impact to published apps
- Analyze logs, usage data, and audit data

Study Resources

We recommend that you train and get hands-on experience before you take the exam. We offer self-study options and classroom training as well as links to documentation, community sites, and videos.

Study resources	Links to learning and documentation
Get trained	Choose from self-paced learning paths and modules or take an instructor led course
Find documentation	Microsoft 365 developer documentation Microsoft 365 documentation Microsoft Graph documentation
Ask a question	Microsoft Q&A | Microsoft Docs
Get community support	Microsoft 365 - Microsoft Tech Community SharePoint - Microsoft Tech Community Teams Developer - Microsoft Tech Community
Follow Microsoft Learn	Microsoft Learn - Microsoft Tech Community
Find a video	Exam Readiness Zone

Change log

Skill area prior to October 31, 2022	Skill area as of October 31, 2022	Changes
Audience Profile		Minor
Design collaborative app solutions and existing app integration	**Design collaborative app solutions and existing app integration**	No change
Assess Microsoft Teams and existing app integration opportunities	Assess Microsoft Teams and existing app integration opportunities	No change
Design development and production environments	Design development and production environments	No change
Design solution based on technical and business requirements	Design solution based on technical and business requirements	No change
Implement Microsoft identity	**Implement Microsoft identity**	No change
Register an application	Register an application	No change
Implement authentication	Implement authentication	No change
Configure permissions to consume an API	Configure permissions to consume an API	No change
Implement authorization to expose or consume an API	Implement authorization to expose or consume an API	No change
Implement authorization in an API	Implement authorization in an API	No change
Create a service to access Microsoft Graph	Create a service to access Microsoft Graph	No change
Build apps with Microsoft Graph	**Build apps with Microsoft Graph**	No change
Access data from Microsoft Graph	Access data from Microsoft Graph	No change
Customize responses by using query parameters	Customize responses by using query parameters	No change
Optimize usage of Microsoft Graph resources	Optimize usage of Microsoft Graph resources	No change
Manage a team lifecycle by using Microsoft Graph	Manage a team lifecycle by using Microsoft Graph	No change

Skill area prior to October 31, 2022	Skill area as of October 31, 2022	Changes
Extend and customize Microsoft 365 with SharePoint Framework	**Extend and customize Microsoft 365 with SharePoint Framework**	No change
Describe the components of a SharePoint Framework (SPFx) solution	Describe the components of a SharePoint Framework (SPFx) solution	No change
Describe the process to package and deploy an SPFx solution	Describe the process to package and deploy an SPFx solution	No change
Describe the consumption of Microsoft Graph	Describe the consumption of Microsoft Graph	No change
Describe the consumption of third-party APIs secured with Azure AD from within SPFx	Describe the consumption of third-party APIs secured with Microsoft Azure Active Directory (Azure AD), part of Microsoft Entra from within SPFx	Minor
Describe Web Parts as Teams Tabs	Describe Web Parts as Teams Tabs	No change
Extend Microsoft Teams	**Extend Microsoft Teams**	No change
Create a Microsoft Teams app manifest	Create a Microsoft Teams app manifest	No change
Create and use task modules	Create and use task modules	No change
Create a webhook	Create a webhook	No change
Implement custom Teams tabs	Implement custom Teams tabs	No change
Create a messaging extension	Create a messaging extension	No change
Create conversational, call, and online media bots	Create conversational, call, and online media bots	No change
Create a meeting extension	Create a meeting extension	No change
Test, deploy, monitor, and maintain modern enterprise-grade app solutions	**Test, deploy, monitor, and maintain modern enterprise-grade app solutions**	No change
Test apps and solutions	Test apps and solutions	No change
Deploy apps and solutions	Deploy apps and solutions	No change
Monitor and maintain published apps	Monitor and maintain published apps	No change

Skills measured prior to October 31, 2022

- Design collaborative app solutions and existing app integration (5–10%)
- Implement Microsoft identity (15–20%)
- Build apps with Microsoft Graph (10–15%)
- Extend and customize Microsoft 365 with SharePoint Framework (15–20%)
- Extend Microsoft Teams (30–35%)
- Test, deploy, monitor, and maintain modern enterprise-grade app solutions (5–10%)

Design collaborative app solutions and existing app integration (5–10%)

Assess Microsoft Teams and existing app integration opportunities

- Assess current Teams adoption level using M365 Maturity Model
- Identify opportunities for no-code or low-code development (Microsoft Power Platform, Azure Logic Apps, Dataverse for Teams)
- Assess Teams integration opportunities with existing applications

Design development and production environments

- Design test, acceptance, production, and deployment strategy to meet solution requirements
- Design development environment by using Microsoft 365 developer tenant
- Recommend Azure services for Teams app use cases (serverless compute workflows)
- Specify app requirements for CI/CD integration
- Develop an appropriate tooling strategy for a Teams solution, including the Teams Toolkit vs Yeoman Teams Generator
- Select an appropriate development framework and runtime platform for a Teams solution (including Teams Toolkit for React or Teams Toolkit Blazor)
- Select an appropriate runtime platform for a Teams solution (including Node.js, .NET, and/or SPFx)

Design solution based on technical and business requirements

- Design authentication scenarios, including for guest users and multitenant authentication
- Design UI by using Microsoft Teams UI Library, Fluent UI React Northstar, Adaptive Cards, and Microsoft Teams UI Kit for Figma

Implement Microsoft identity (10–20%)

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
- Implement single sign-on authentication for a bot
- Implement single sign-on authentication for a tab

Configure permissions to consume an API

- Configure delegated permissions for the app
- Configure application permissions for the app
- Specify and implement consent requirements
- Grant consent to an application

Implement authorization to expose or consume an API

- Configure incremental consent scopes
- Grant scoped permissions to API
- Call MSAL (JavaScript) using AcquireTokenSilent/AcquireToken pattern

Implement authorization in an API

- Validate access token
- Configure effective permissions for delegated scopes
- Implement app permissions by 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
- Consume certificates from Key Vault, including using managed identities
- Configure application permissions for the app
- Acquire an access token for Microsoft Graph using an application permission and client credential certificate

Build Apps with Microsoft Graph (10–15%)

Access data from Microsoft Graph

- Determine whether to retrieve data using a Microsoft Graph SDK or the Graph REST API
- Get the signed-in user's profile
- Get a list of users in the organization
- Get the user object based on the user’s unique identifier
- Access user data from Microsoft Graph
- Create, read, update, and delete data

- Traverse Microsoft Graph to retrieve data
- Upload files larger than 10 mb to Teams
- Implement Microsoft Graph Toolkit

Customize responses by using query parameters

- Use query parameters, including $filter, $select, $orderby, $count, $skip, and $stop
- Expand and retrieve resources using $expand query parameter
- Search for resources using $search query parameter
- Access paged date using $odata.nextLink, $skip, and $skiptoken parameters
- Test queries and responses by using Microsoft Graph Explorer

Optimize usage of Microsoft Graph resources

- Monitor for changes using change notifications
- Combine multiple requests using $batch
- Get changes using a delta query
- Detect and handle throttling

Manage a team lifecycle by using Microsoft Graph

- Get the information on a team by ID
- Get the list of members in a team
- Get the list of owners of a team
- Get the list of teams where the signed in user is a member
- Provision a team using an existing Teams template
- Delete or archive a team
- Update an existing team
- Enable resource-specific consent

Extend and Customize Microsoft 365 with SharePoint Framework (15–20%)

Describe the components of a SharePoint Framework (SPFx) solution

- 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
- Explain 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 Adaptive Card Extensions
- Describe Viva Connections and Viva dashboard

- Describe the types of SPFx extensions (Application Customizer, Field Customizer, Command Set 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 SPFx solution
- Describe how to build a Microsoft Teams tab by using SPFx
- Manually create an app manifest to deploy a SPFx Web Part to Teams

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
- Identify 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 Microsoft Teams (30–35%)

Create a Microsoft Teams app manifest

- Configure an app manifest using the Developer Portal
- Create an app package for a Microsoft Teams app

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
- Implement device permissions in a personal tab
- Create a channel/group tab
- Create a tab with a deep link
- Create an adaptive card 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
- Add link unfurling

Create conversational, call, and online media bots

- Create a personal bot
- Create a group/channel bot
- Send actionable messages from a bot
- Create a call bot
- Create an online media bot
- Handle incoming calls
- Transfer incoming calls

Create a meeting extension

- Describe the capabilities of Microsoft Teams meeting apps
- Enable and configure apps for Teams meetings
- Create or integrate apps within the meeting lifecycle
- Implement shared meeting stage

Test, deploy, monitor, and maintain modern enterprise-grade app solutions (5–10%)

Test apps and solutions

- Design test plans, including Bots, Connectors, Messaging extensions, Meeting extensions, SSO, and Adaptive cards
- Specify required sample data for testing
- Deploy sample data by using M365 CLI or PowerShell scripts
- Execute test plans, including Bots, Connectors, Messaging extensions, Meeting extensions, SSO, and Adaptive cards

Deploy apps and solutions

- Create or modify app definitions in the Developer Portal for Teams

- Manage Teams apps packages by using the Developer Portal for Teams
- Upload a custom app to a single user or team
- Upload an app to the App Catalog
- Upload an app to the Developer Portal for Teams
- Automate deployment of Teams apps

Monitor and maintain published apps

- Recommend audit settings using Microsoft 365 Audit logging
- Review and evaluate performance using App Insights
- Monitor Microsoft Graph API permissions by using the Microsoft 365 compliance center
- Assess new SDK versions for impact to published apps
- Analyze logs, usage data, and audit data