The developer's guide to building Microsoft Teams apps
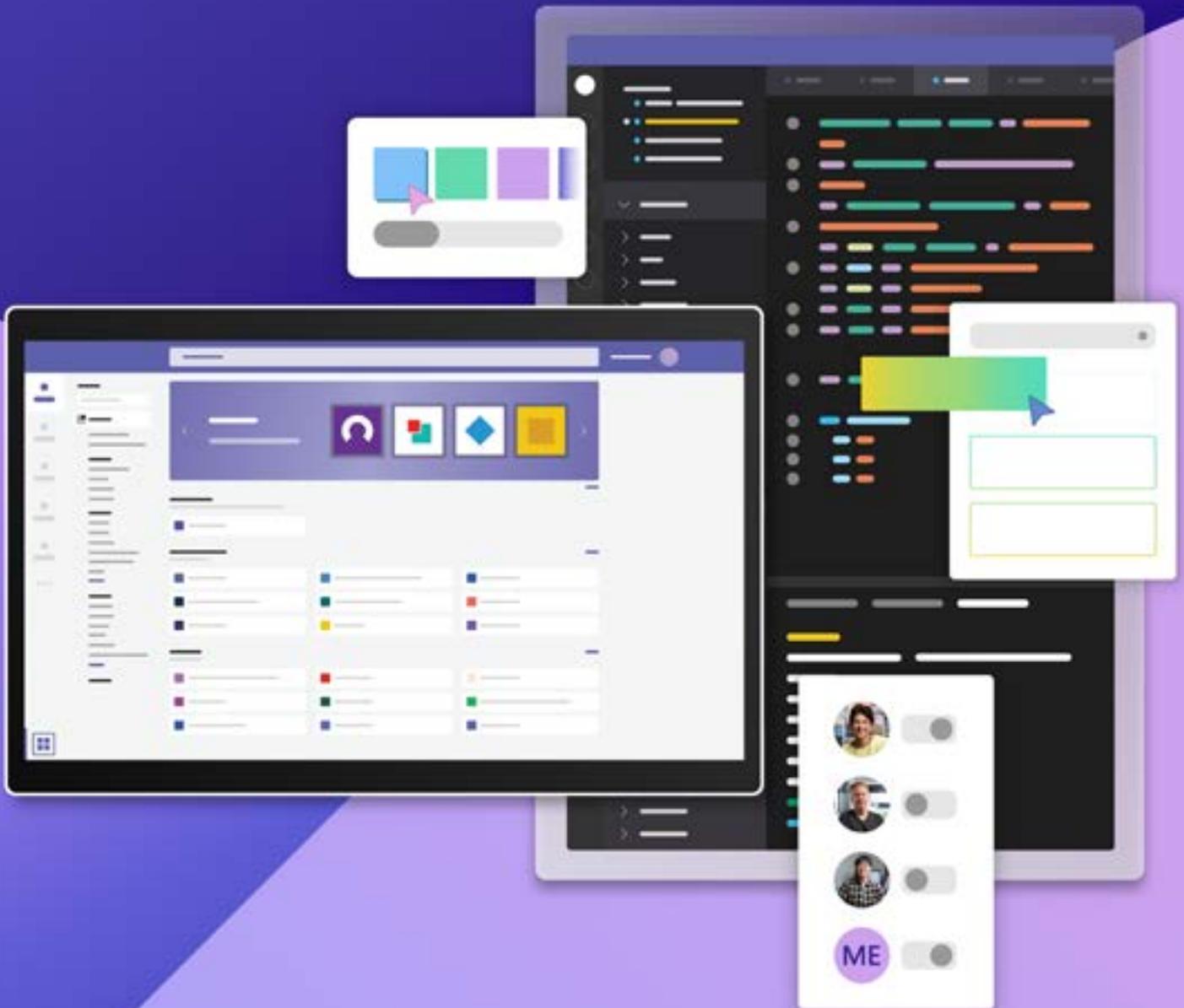
CONTENTS

01

The Microsoft Teams developer journey

Teams capabilities

Collaborative apps represent a paradigm shift away from an application-centric work model to one centered on the team. People don't go to apps—apps come to people. Collaborative apps meet users how, where, and when they need and want to work, and bring insights, appropriate business workflows, and accurate data into the right user and team interactions. This facilitates seamless access to and navigation between workflows and business data without interruption.

With engaging, collaborative apps in Teams, your organization can modernize communications and delight users at the same time. Everyone—from decision makers to developers—can experience unique and compelling benefits using Teams, including improved productivity, greater user satisfaction, and powerful development features. Teams offers an unmatched opportunity for developers to effectively plan, design, build, and distribute collaborative apps using a flexible development environment and a rich set of features, frameworks, tools, and SDKs. Offering diverse capabilities and entry points, Teams enables developers to create functionality that encourages deeper engagement, such as allowing users to explore an app from a specific tab.

With app development in Teams, you own the technology stack and the entire user experience. The full stack of Microsoft technologies means you can build and distribute collaborative apps in ways that make sense to you—just choose the right combination of your preferred tools, programming languages, and SDKs to target specific business needs:

Extend app functionality by adding more Teams capabilities and using Microsoft Graph to connect the data and insights from Microsoft 365.

Take advantage of an integrated developer environment and the DevOps benefits of GitHub, Microsoft Visual Studio, and Visual Studio Code with Microsoft 365 and Azure.

Use the robust Developer Portal to manage, configure, deploy, and distribute your apps.

Monetize your apps with options like SaaS offers, in-app purchases, and more.

This e-book takes a closer look at how you can extend Teams to help users collaborate and employ the tooling that Microsoft offers to make building apps easier. It covers the key capabilities developers need to build rich collaborative experiences that help users thrive in the flow of work.

02

Why Teams?

Dedicated to supporting where, when, and how people want to work, Teams is the collaboration platform of choice across industries, borders, and organizations of every size and shape. With Teams apps, you can bring business workflows and insights to users where they work, helping them collaborate, drive better and faster decisions, and accelerate business outcomes.

Build collaborative apps

Collaborative apps—an app pattern designed to bring people, processes, and data together to help users thrive in the hybrid workplace—are transforming how people in every organization work together. With more than 270 million active users, Teams offers developers an unmatched opportunity to build collaborative apps. You can create collaborative experiences with low-code tools like Power Platform and professional code options like Visual Studio and Visual Studio Code. You can build your app once and deploy it anywhere, helping to ensure maximum adoption and user reach. Apps built for Teams work across all desktops and mobile, and app permission policies can be created and assigned through the Teams admin center.

Teams promotes stronger app discoverability and product awareness by allowing you to push, list, and distribute your app on the Teams store. The Teams store helps people find your apps through updated app categories, curated app collections, and intelligent recommendations based on what

colleagues and peers are using. Likewise, with Teams, you can certify your app through the Microsoft 365 App Compliance program, which focuses on security, privacy, and data-handling standards. To monetize your app, include it as a SaaS offering on the Microsoft Commercial Marketplace, or add an in-app purchasing option that includes a paywall experience directly from within the app.

A collaborative app example:
MURAL for Teams

MURAL for Teams extends the Teams app's personal tabs and search-based message extensions to other Microsoft-hosted apps. For example, it enables you to deliver a connected experience across Teams, Outlook, and Office. With a search-based message extension, you can insert a MURAL directly into an Outlook message as an interactive Adaptive Card.

MURAL for Teams is the first app in the Visual Collaboration category to achieve Microsoft 365 certification, which validates that the app offers:

> **Increased confidence that member data is protected**

> **Trusted, strong security that reduces customer risk**

> **Reduced time spent investigating security and compliance**

Bring app solutions to life

Teams apps serve as a bridge between the Teams client (web, mobile, or desktop) and APIs, folding these directly into the context of Teams. You can bring your solutions to life through Teams capabilities like tabs, bots, Adaptive Cards, messaging extensions, webhooks, and task modules. These capabilities, in turn, can be powered by Microsoft Azure, Microsoft Graph, SharePoint Framework, or Power Platform. The Teams SDK also enables you to build apps for Teams, Outlook, and Office using a single application and deployment model. Your collaborative apps build on top of the world-class security, governance, and compliance of the Microsoft 365 platform, making it easier for you to build apps that are both secure to use and compliant.

03

Getting started with Teams apps

Teams apps consist of cloud services that integrate with Teams extension points (also known as features) and a manifest that registers these services with the Teams platform. Teams core features and functionalities that you can build in your app are called Teams extension points because they enable integration and interaction. Creating apps with Teams involves deciding what to build (based on Teams extension points and user engagement), building the APIs, creating a Teams app package, and distributing the package to target users.

Note that the Teams client provides the extension point that people use to interact with your app, while you host the APIs and logic that power your app.[1] A Teams app package containing icons and a manifest JSON file is installed in Teams. The manifest file includes details about:

• **App metadata (like the app name)**

• **Teams extensibility points that the app uses (like tabs, bots, or messaging extensions)**

• **Pointers to the APIs that power the app (like the ID or tab URL)**

[1]*The Teams platform is not a hosting service; the APIs powering your app must be hosted by you and accessible by HTTPS over the internet.*

Before building your Teams app, it's important to understand the different extensibility points available across the Teams UI, including tabs, bots, and messaging extensions. By extending Teams capabilities, you tap into the power of contextual relevance, making the user experience clearer, fuller, and more effective. For example, you can build your services with context-specific information like messages your app was triggered from or information related to the channel your app is installed in.

Tabs display web-based content in a central location where people can discuss and work on it together. You can add tabs as part of a channel inside a team, group chat, or personal app for an individual user. An example use case would be exposing your existing web application inside Teams.

Bots in Teams can range from simple notifications to conversational bots with AI-powered language understanding. Examples include a customer service bot or a weather forecasting bot.

Messaging extensions enable users to search and share external information in conversation, such as sending a card or creating a help ticket based on the content of a channel post. For example, users can look up order or customer information in an external system and share it in a conversation without leaving Teams.

Webhooks and connectors allow users to subscribe to a channel for notifications and messages from an external system, such as your APIs.

Meeting extensions deliver a specific user experience for each stage of the meeting life cycle, including pre-meeting, in-meeting, and post-meeting. Example use cases include preparing the meeting agenda before a meeting, using the agenda to support the conversation and take notes and actions during a meeting, and sharing the outcomes with participants after a meeting.
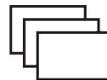
Adaptive Cards are actionable snippets of content that can be added to a conversation through a bot or messaging extension. They simplify the sharing of structured data in conversations in a consistent way across all devices and applications. An example use case is gathering information through input forms.

Microsoft Graph API provides access to data and insights stored in Microsoft 365 as well as information about teams, channels, users, and messages. Example use cases include showing upcoming meetings or retrieving information about people in your organization or the files you've been working on.

Depending on business needs, you might choose a single extension point (like a personal conversation bot) or combine multiple extension points. For more information, see Microsoft Docs | Teams app features.

04
App development experience

Because Teams is a collaborative social framework, there are many options available to solve your users' needs. Building an app in Teams sets you on the path for successful development, distribution, and future-state goals. Based on your business requirements and preferred programming language, you can choose from multiple options for developer tools and SDKs in Teams. Collectively, these choices enable you to design, build, debug, distribute, and manage Teams apps. For environments, consider the Teams Toolkit, the open-source community tool Yeoman Generator for Microsoft Teams (Yo Teams), or the Developer Portal. Supported frameworks include Blazor, SharePoint Framework, and Bot Framework. For building solutions, web frameworks like React, Angular, and VueJS are available. For programming languages, choose from familiar options like JavaScript, TypeScript, .NET, Java, or Python.
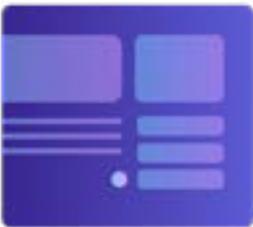
Design

Use the Teams UI Kit and UI Library to create the right experience and design a UI that fits diverse user requirements.
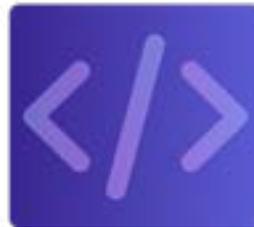
Development

Use popular IDEs like Visual Studio or Visual Studio Code, command-line tools, APIs, and SDKs to integrate apps into Teams conversations and channels.

Integration with Microsoft 365

Access data and insights stored in Microsoft 365 using Microsoft Graph, expose Teams apps across Microsoft 365 (including Outlook and Office.com), and publish to the marketplace.

Testing, deployment, distribution, and management

Test and debug your app locally or in the cloud. Then, deploy the app to SharePoint or Azure using Teams Toolkit or Yo Teams. Finally, configure, manage, and distribute the app directly to the store with the Developer Portal.

Teams development tools and SDKs

The tools and SDKs available for app development in Teams are not alternatives of each other; rather, a combination of these different tools and SDKs can be used in your solution. Table 1 provides an overview of the tools and SDKs available, their intended users or purpose, and how they can help with app development.

Tool or SDK	Intended users or purpose	Benefits for Teams app development
Teams Toolkit for Visual Studio Code	JavaScript and TypeScript developers	Create new projects in Teams, set up apps in the Teams Developer Portal, run and debug in Teams, and configure cloud hosting. Add other Teams capabilities (such as tabs or bots) and more Azure resources to your Teams app. Collaborate on Teams projects and add SSO, CI/CD pipelines, and API connections.
Teams Toolkit for Visual Studio	.NET developers	Create new projects in Teams, set up apps in the Teams Developer Portal, run and debug in Teams, and configure cloud hosting.
Yo Teams	CLI tool for developers who are familiar with Yeoman and Gulp	Create and scaffold projects that include one or more Teams features such as bots, messaging extensions, tabs, and more.
TeamsFx	CLI tool for JavaScript, TypeScript, and .NET developers; includes SDK	CLI offers the keyboard-centric equivalent of the Teams Toolkit functionality, including deployment from CI/CD pipelines. SDK offers simplified authentication to APIs secured with the Microsoft identity platform, such as Microsoft Graph and custom APIs.
Bot Framework	Build conversational Teams bots with C#, Java, Python, and JavaScript	Bot SDK offers a rich bot SDK to create Teams bots and message extensions.
Teams JS SDK	Integrate apps with Teams client capabilities	When you expose your web application in Teams as a tab, make it Teams-aware and build task modules that integrate with your application.
Microsoft Graph SDK	Connect data and insights from Microsoft 365	Simplify building high-quality, efficient, and resilient apps that access Microsoft Graph.
SharePoint Framework	Build Microsoft 365–hosted solutions with single sign-on (SSO) using TypeScript or JavaScript	Host tabs and task modules on Microsoft 365 without additional services on Azure. Easily access resources secured with the Microsoft identity platform, like Microsoft Graph or custom APIs.
Adaptive Cards SDK	Build declarative UI elements with JSON, which transforms into a native Teams UI on web, desktop, and mobile versions	Extend bots and messaging extensions with a rich and interactive UI that works on any platform.
Fluent UI SDK	Implement web UI that is consistent with Teams; supports React	Use web components that implement Microsoft Design Language and seamlessly integrate with Teams UI, such as buttons, dropdowns, grids, people and date pickers, and more.

Table 1: Teams tools and SDKs and their uses

Teams Toolkit

The Teams Toolkit guides you through the process of creating your app, providing everything you need to quickly build, debug, and launch. It allows you to integrate your application with all extensibility points in Microsoft Teams. It also offers templates for common scenarios that you can use as a starting point, such as bot notifications, command bots, or apps that are available in Teams, Outlook, and Office.com.

The Teams Toolkit enables you to create and deploy Teams apps with integrated identity, access to cloud storage, data from Microsoft Graph, and other services in Azure and Microsoft 365. It offers full-stack debugging, hot reload, secure tunneling, integrated support for hosting, support for serverless functions, and CI/CD actions for GitHub and Azure DevOps. It also provides integrations with the various tools, programming languages, and frameworks developers use most often, including first-class support for React, SharePoint Framework, and ASP.NET Core Blazor. The Teams Toolkit can also be used to debug and deploy front-end web apps from other frameworks such as Angular.

The Teams Toolkit allows you to build Teams apps faster. If you prefer to use JavaScript or TypeScript, you can use Teams Toolkit for Visual Studio Code; if you're a .NET developer, use Teams Toolkit for Visual Studio; and if you prefer command-line tooling, use the TeamsFx CLI. TeamsFx CLI is a feature of TeamsFx that runs tasks with a keyboard-centric experience while building Teams apps. TeamsFx includes a library that simplifies connecting to services secured with the Microsoft identity platform and lets you build apps with zero configuration.

Yeoman Generator

The Yeoman Generator for Microsoft Teams (also known as Yo Teams) scaffolds a project once the developer has answered a few questions about the Teams app they're building. It consists of the Yeoman Generator for Teams Apps (generator-teams) and the package of Gulp tasks (yoteams-build-core and yoteams-deploy). It offers additional features such as simple deployment, a unit testing framework, sample unit tests, and more. Visit the Microsoft Teams App Project Generator – #YoTeams GitHub page to learn more.

SDKs for building Teams apps

There are multiple SDKs available for you to power your Teams app.
You can combine them based on what you need to build.

Teams JavaScript (Teams JS) SDK

The Teams JS SDK allows you to make web apps Teams-aware. Whether you're bringing an existing web app into Teams or building task modules, using the Teams JS SDK gives you access to the context of Teams, Outlook, and Office.com.

Bot Framework SDK

The Bot Framework SDK helps you build messaging extensions and conversational bots. It supports the C#, JavaScript, Python, and Java programming languages. It also helps you build conversational experiences with language understanding, QnA Maker, and a sophisticated composition of bot replies.

Microsoft Graph SDK

Microsoft Graph SDK lets you simplify building high-quality, efficient, and resilient apps that access data and insights from Microsoft 365. It's available for the C#, PowerShell, TypeScript, JavaScript, Java, Go, PHP, and Python (preview) programming languages.

TeamsFx SDK

TeamsFx can help you build apps that use the Microsoft Graph API or any API secured with the Microsoft identity platform to fetch data like basic user information. Teams Fx handles the OAuth 2.0 authorization code grant flow, getting access tokens that can be used to make Microsoft Graph API calls.

SharePoint Framework

Build Microsoft 365–hosted apps with SharePoint Framework using code, Teams tabs, and task modules. Easily access resources secured with the Microsoft identity platform, such as Microsoft Graph and custom APIs.

Adaptive Cards SDK

Adaptive Cards are actionable snippets of content that can be added to conversations through bots or messaging extensions. These cards provide rich communication to your audience using text, graphics, and buttons.

Fluent UI SDK

Fluent UI implements Microsoft Design Language, which allows developers to build apps that seamlessly integrate with Teams. These components work across different use cases, Teams themes, and screen sizes.
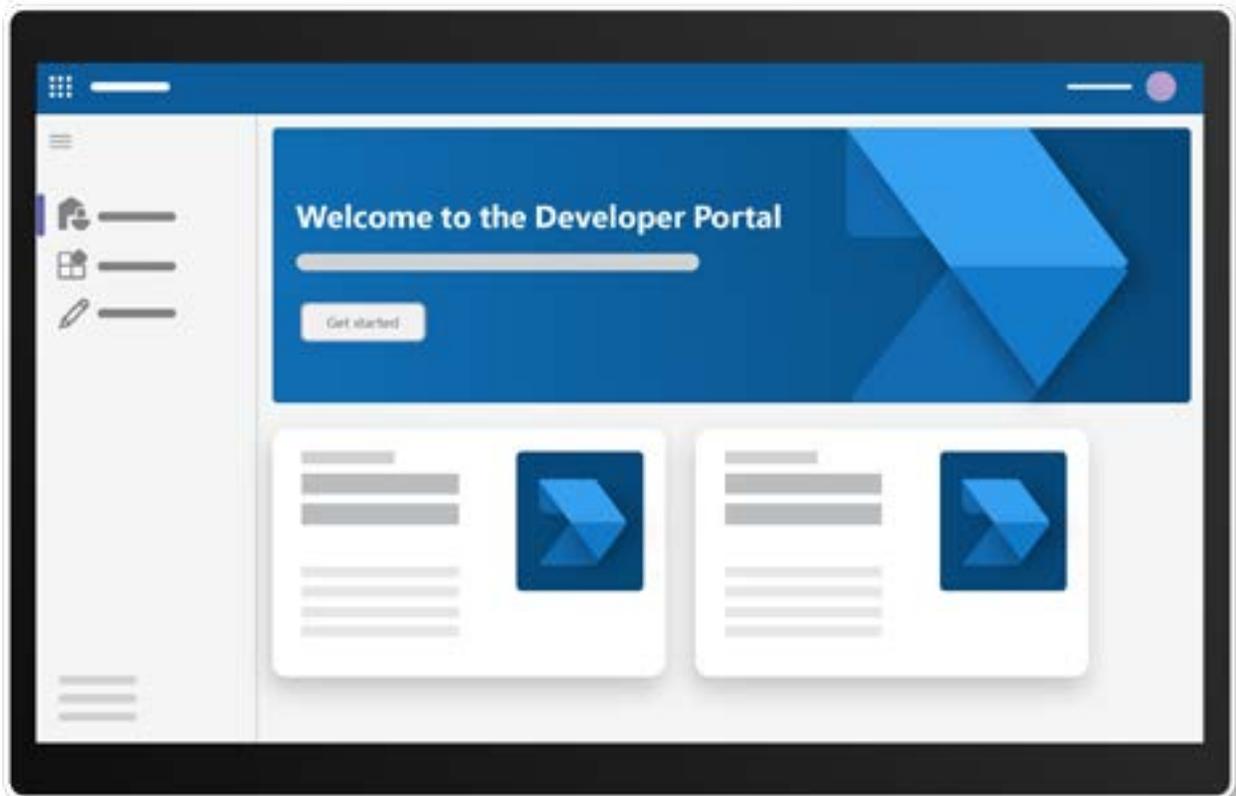
Developer Portal for Teams

The Developer Portal offers a tool for configuring, distributing, and managing Teams apps. It provides a dedicated app management console where you can register, configure, and distribute your app to your organization or the Teams store. You can also evaluate and debug your app, analyze usage insights, and access other tools, including:

- **Scene studio** to help you design custom Together Mode scenes for Teams meetings

- An **Adaptive Cards editor** to help you create and preview Adaptive Cards to include with the build apps

- **Microsoft identity platform management** to register apps with Azure Active Directory to help users sign in and provide access to APIs

- **Teams store app validation** to check the app package against the test cases that Microsoft uses to review your app

- **Bot management** to add conversational bots to an app that communicate with users, respond to their questions, and proactively notify them about changes and other events

05
Bring Teams into your app

When building Teams apps, you can expose your app inside Teams—but you can also bring some Teams functionality to your app with a rich set of features. Azure Communication Services, which power Teams, enable real-time audio and video conferences in your existing app. Similarly, the Microsoft Graph API brings Teams (chats and channels) collaboration to your app. By bringing Teams into your app, you can improve users' experience by helping them get all the relevant information they need for their work, enabling them to stay in their flow and collaborate seamlessly without having to switch back and forth across apps.

Microsoft Graph exposes REST APIs and client libraries to access data in Microsoft 365 and Teams. It easily extends your app with Teams functionality (like video calling or chat) using the building blocks in Microsoft 365. Plus, Microsoft is introducing new capabilities—like enabling chats with federated users, identifying which messages are read and unread by the current user, and subscribing to user chats and membership changes—with Microsoft Graph APIs.

Azure Communication Services are cloud-based services with REST APIs and client library SDKs available to help developers integrate communication (voice and video calling, telephony, chat, SMS, and email notifications) into their apps. Azure Communication Services' interoperability with Teams enables you to develop apps that interact with external customers and Teams users by streamlining voice, video, telephony, and chat capabilities. You can also add SMS and email notification capabilities to extend the value of Teams to further suit specific needs. Plus, with the sample app builder, you can easily build and deploy a sample app for virtual appointments with Azure Communication Services in just a few minutes—no coding needed.
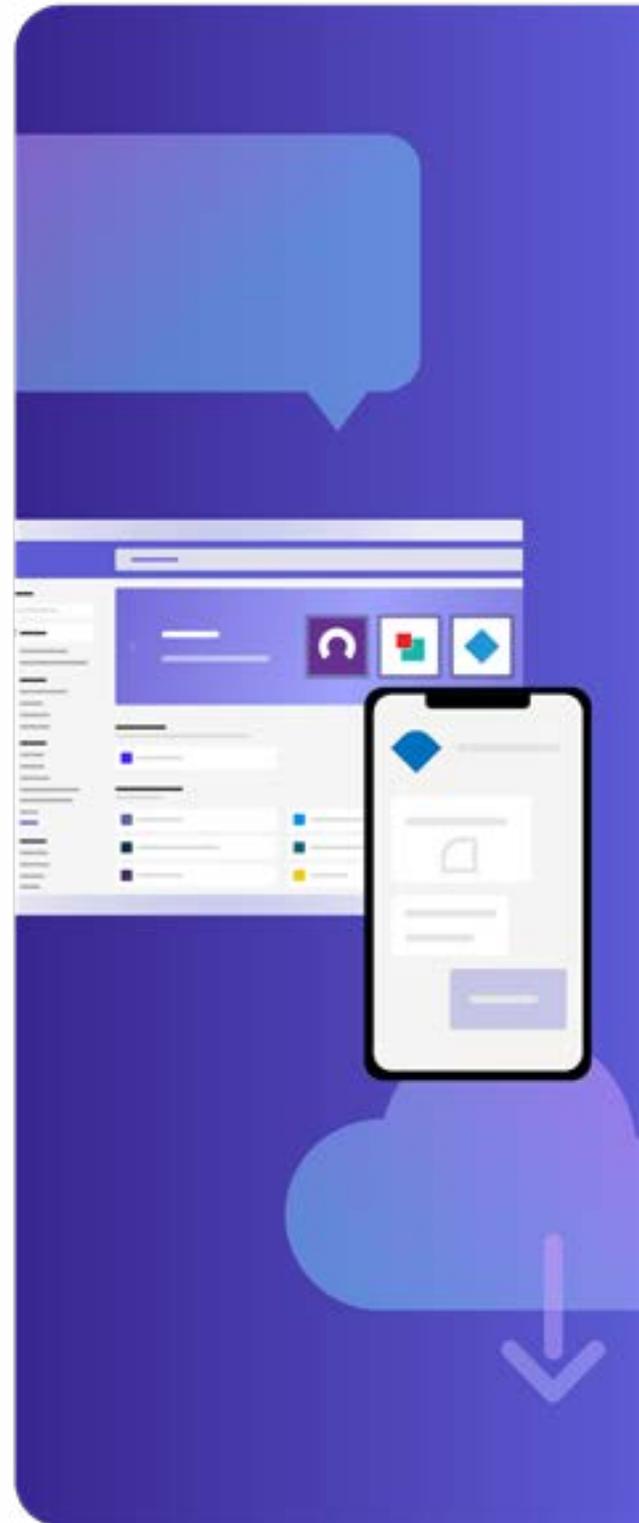
06
Publish and certify your app

To publish your app to the Teams store, you need to review the Teams store validation guidelines and submit your app for publishing based on these parameters. This validation helps users and admins verify the authenticity of your app. The Microsoft 365 App Compliance Program is designed to evaluate, validate, and highlight your app security, privacy, and data trustworthiness based on industry standards, including the Service Organization Control Report (SOC 2) and the Payment Card Industry Data Security Standard (PCI DSS). With the App Compliance Automation tool, developers can automate various tasks to achieve the certification faster and more easily. The tool produces reports that developers can share with IT to gain visibility into app security and compliance.

07
Microsoft 365 Developer Program

The Microsoft 365 Developer Program offers a free, instant sandbox for you to prototype your apps and solutions. The sandbox comes pre-provisioned and configured for development with sample Teams data so you can start quickly, saving time so you can focus on what matters: exploring and learning the capabilities of the Microsoft 365 platform. The Teams sample data pack includes:

- User, mail, and calendar data with 16 sample users, plus user data and content, to help you model your solutions. The data includes user mailboxes, calendar events, and integration into a Teams developer environment with simulated chats and team memberships.

- A customized developer environment featuring pre-configured Teams app sideloading and five sample teams, channels, tabs, and chat sessions built around the 16 sample users.

- A developer portal that's pre-installed and pinned to offer easy access to configure and manage apps that developers build.

- The ability to choose from the SharePoint sample data and six site templates that can be installed and evaluated for your organization (with one pre-installed in your instant sandbox).
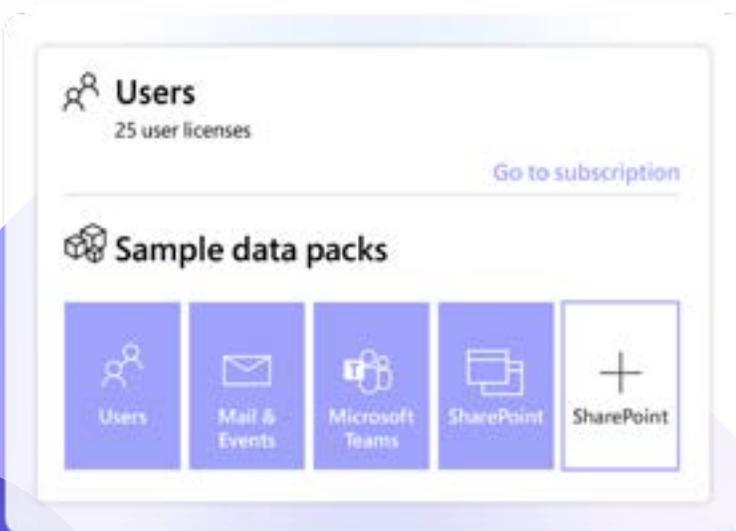
08

Next steps

Teams is collaborative by design. It's uniquely suited to build and deploy collaborative apps into every corner of your organization, empowering employees, optimizing operations, and fully engaging your customers. Do you have an app idea that can harness the power of Teams for your organization or your customers? Take advantage of Teams' rich and flexible set of extensibility points, UI constructs, and APIs to build powerful collaborative apps with your existing development skills and the tools provided by Microsoft 365 and Azure.

> Get started by joining the
> Microsoft 365 Developer Program

> Learn more with
> Teams developer documentation

> Download the infographic:
> Build collaborative apps with Microsoft Teams