title: Create a Teams app tab author: laujan description: A guide to building a tab keywords: teams tabs group channel configurable ms.topic: conceptual ms.author:

Create a Teams app tab

Enhance your app experience by serving web content that you host to your channel, group chat, and personal users via a custom tab. Tabs provide efficient access to designated team information and resources. You'll need to complete the following steps to create a tab:

1. Prepare your development environment.
2. Create your tab content.
3. Integrate your code with Teams.
4. Host your app service.
5. Create your app package and upload to Microsoft Teams.

Prepare your development environment

- You'll need an Office 365 tenant and a team configured with *Allow uploading custom apps* enabled. To learn more, see Manage Microsoft Teams settings for your organization.

- If you don't currently have an Office 365 account, you can sign up for a free subscription through the Office 365 Developer Program. The subscription will remain active as long as you're using it for ongoing development.

Create your tab content

Whether you present your tab within the personal or channel/group scope, it will essentially be an HTML page that you host—the difference is how your tab content URL is set.

☑ You must allow your pages to be served in an IFrame, via X-Frame-Options and/or Content-Security-Policy HTTP response headers.

- Many standard websites block their pages from loading in an IFrame. However, Teams provides a unique webview option for users to view embedded content within the desktop client.

☑ Your authentication logic needs to use a method other than redirect (e.g., use token-based or cookie-based authentication).

- Typically, as a safeguard against click-jacking, login pages don't render in IFrames.

☑ Your cross-domain navigation logic should allow the Teams client to validate the origin against a static validDomains list in the app manifest when loading or communicating with the tab.

- Browsers adhere to a same-origin policy restriction that prevents a webpage from making requests to a different domain than the one that served a web page. However, you may need to redirect the configuration or content page to a another domain or subdomain.

☑ Style your tabs based on the Teams client's theme, design, and intent (see Content and conversations, all at once using tabs).

- Tabs work best when they're built to address a specific needs and focus on a small set of tasks or a subset of data that is relevant to the tab's channel location.

Integrate your code with Teams

☑ Within your content page, add a reference to Microsoft Teams JavaScript client SDK using script tags.

☑ Following your page load, make a call to `microsoftTeams.initialize()` in the Teams client SDK.

- Your script has to include the `microsoftTeams.initialize()` method call for your page to be displayed.

Host your app service

Your content needs to be available on a hosted URL available from the cloud using HTTPS endpoints. For testing, you can use a reverse proxy, such as ngrok, to expose your local port to an internet-facing URL.

Create your app package and upload to Microsoft Teams

Your app manifest is a `json` file that defines your app’s capabilities and references your content URL.

The app package is a zip folder containing the following:

- Your **full color icon** measuring 192 x 192 pixels.
- Your **transparent outline icon** measuring 32 x 32 pixels.
- Your **manifest.json** file that specifies the attributes of your app.

Create your app package manually and upload directly to Teams

Teams requires that your manifest is named "manifest.json" and corresponds with the latest version of the Manifest schema for Microsoft Teams hosted at https://developer.microsoft.com/json-schemas/teams/v1.5/MicrosoftTeams.schema.json. To create your manifest, complete the property values required for all apps and, based upon your tab's scope and capabilities, complete the tab reference values:

Personal Tabs

The displayed content for personal tabs is the same for all users and is set directly in your manifest by the `contentUrl` property in the `staticTabs` array:

```
...
"staticTabs": [
    {
      "entityId": "idForPage",
      "name": "Display name for tab",
      "contentUrl": "https:// yourURL.com/content ",
      "websiteUrl": "https://yourURL.com/website",
      "scopes": [ "personal" ]
    }
...
```

Channel/group tabs

Channel/group tabs allow users to set the tab content from a configuration page typically based upon URL query string parameters(see Create a Content Page for Your Custom Tab):

```

...
"configurableTabs": [
    {
      "configurationUrl": "https://yourURL.com/configure",
      "canUpdateConfiguration": true,
      "scopes": [ "team", "groupchat" ]
    }
...
```

Once you have completed your `manifest.json` bundle it in a zip folder along with your two required images.

Upload app package directly to a team

1. Open the Microsoft Teams client. If you use the web based version you can inspect your front-end code using your browser's developer tools.

2. In the *YourTeams* panel on the left, select the `...` menu next to the team that you're using to test your tab and choose **Manage team**.

3. In the main panel select **Apps** from the tab bar and choose **Upload a custom app** located in the lower right-hand corner of the page.

4. Open your project directory, browse to the **./package** folder, select the app package zip folder and choose **Open**. Your tab will upload into Teams.

Upload your app package with App Studio

If you have an app package you can select the App Studio Manifest editor tab and choose the **Import an existing app** tile to upload your zip file.

If your files are not in an app package zip folder, select the Manifest editor tab and choose the **Create a new app** tile and follow the steps below:

1. If you do not have App studio installed in Teams, select **Apps** Store App at the bottom-left corner of the Teams app, and search for App Studio. Once you find the tile, select it and choose install in the pop-up window dialog box.

2. Open the Microsoft Teams client. Opening the web based version will enable you to inspect your front-end code.

3. Open App Studio and select the **Manifest editor** tab.

4. Choose the **Create a new app** tile.

5. Add your app details (see Manifest schema for Microsoft Teams](foo.md) for full description of each fields).

6. In the capabilities section select **Tabs**.

    - For a personal tab, choose *Add a personal tab* and select **Add**. You will be presented with a pop-up dialogue window where you can add your tab details.

    - For a channel/group tab, under *Team Tab* select **Add** and complete the tab details fields in the Team tab pop-up window. Make sure the *can update configuration? Team*, and *Group chat* boxes are checked and select **Save**.

7. In the *Domains and permissions* section, the *Domains from your tabs* field should contain your host or ngrok URL without the HTTPS prefix.

8. From the **Finish** => **Test and distribute** tab you can **Download** your app package, **Install** the package into a team, or **Submit** to the Teams app store for approval. *If you are using a reverse proxy you will get a warning in the **Description** field on the right. The warning can be ignored while testing your tab*.

> [!NOTE] Using the Teams developer platform, you can can choose your app's level of distribution:
>
> - The Teams Store offers apps that any organization using Office 365 can access.
> - Your tenant’s Teams App Catalog has apps that are only available to your organization. These are known as line-of-business apps.

View your tab in Teams

- View your personal tab:
    - In the navbar located at the far-left of the Teams client, select the `...` menu and choose your app from the list.
- View your channel/group tab:
    - Return to your team, choose the channel where you would like to display the tab, select + from the tab bar, and choose your tab from the gallery.
    - Follow the directions for adding a tab. Note that there's a custom configuration dialog for your channel/group tab. Select **Save** and your tab will be added to the channel's tab bar.

Modify or remove a tab

You can enable users to modify, reconfigure, or rename a group/channel tab by setting your manifest's `canUpdateConfiguration` property to `true`. Supported removal options can further refine the user experience. You can designate what happens to the content when a tab is removed by including a removal options page in your app and setting a value for the `removeUrl` property in the `setSettings()` configuration (see below). Personal tabs can't be modified but can be uninstalled by the user.

Microsoft Teams setSettings() configuration:

```
microsoftTeams.settings.setSettings({
    contentUrl: "add content page URL here",
    entityId: "add unique name here",
    suggestedDisplayName: "add name to display on tab here",
    websiteUrl: "required for mobile clients",
    removeUrl: "add removal page URL here"
});
```

Learn more

- Create a content page for your tab
- Create a configuration page for your tab
- Update or remove a tab