Lab 29 – SPFx to Teams

Create a Microsoft Teams tab project

Create a new project directory in your favorite location

md teams-tab

cd teams-tab

Create a new solution by running the Yeoman SharePoint Framework Generator

yo @microsoft/sharepoint

What is your solution name?: teams-tab

Do you want to allow the tenant admin the choice of being able to deploy the solution to all sites immediately without running any feature deployment or adding apps in sites?: Yes

Which type of client-side component to create?: WebPart

What is your Web part name?: MyFirstTeamsTab

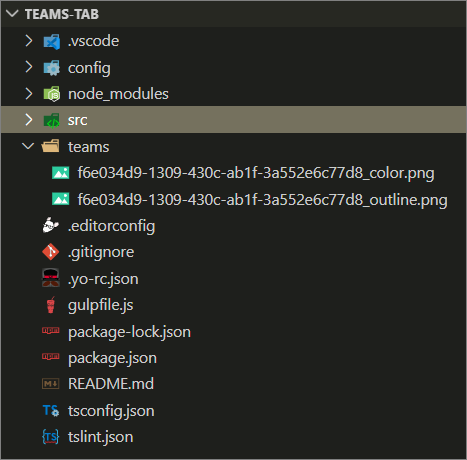
What is your Web part description?: My first Teams tab

Which framework would you like to use?: No JavaScript web framework

Open the project in VS code

code .

SharePoint Framework v1.8 and later projects will include the ./teams folder in the solution structure:



**Update the web part manifest to make it available for Microsoft Teams**

Locate the ./src/webparts/\*\*manifest.json file for the web part you want to make available to Teams and modify the supportedHosts properties to include "TeamsTab" as in the following example.

{

"$schema": "https://developer.microsoft.com/json-schemas/spfx/client-side-web-part-manifest.schema.json",

//...

"supportedHosts": ["SharePointWebPart", "TeamsTab"],

//...

}

**Update the render() method as follows.**

let title: string = "";

    let subTitle: string = "";

    let description:string = "";

    if(this.context.sdks.microsoftTeams)

    {

      title = "Welcome to Teams!";

      subTitle = "This is the SPfx webpart as tab in Teams";

      description = "Team Name: " + this.context.sdks.microsoftTeams.context.teamName + " Channel Name : " + this.context.sdks.microsoftTeams.context.channelName;

    }

    else{

      title = "Welcome to SharePoint!";

      subTitle = "Customize SharePoint experiences using Web Parts. ";

      description = "Site Name : " + this.context.pageContext.web.title;

    }

    this.domElement.innerHTML = `

      <div class="${ styles.firstTeamsWebpart }">

        <div class="${ styles.container }">

          <div class="${ styles.row }">

            <div class="${ styles.column }">

              <span class="${ styles.title }">${title}</span>

              <p class="${ styles.subTitle }">${subTitle}</p>

              <p class="${ styles.description }">${description}</p>

            </div>

          </div>

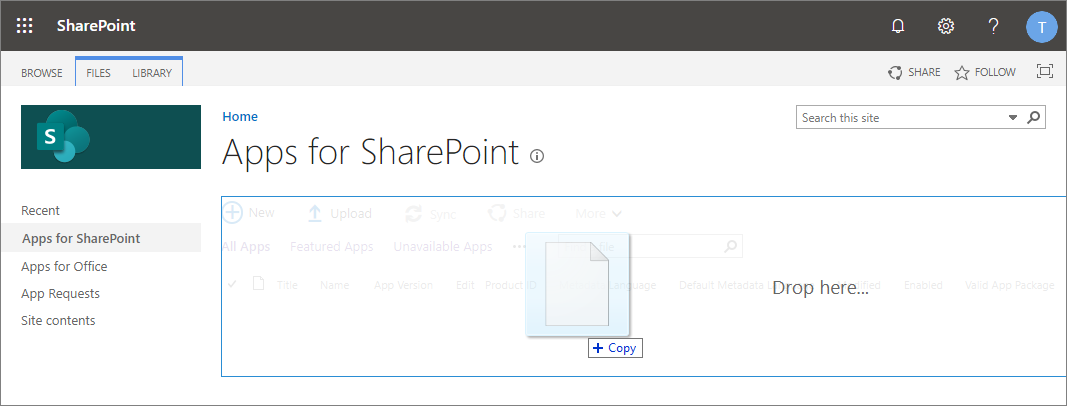
        </div>

      </div>`;

Go command prompt

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

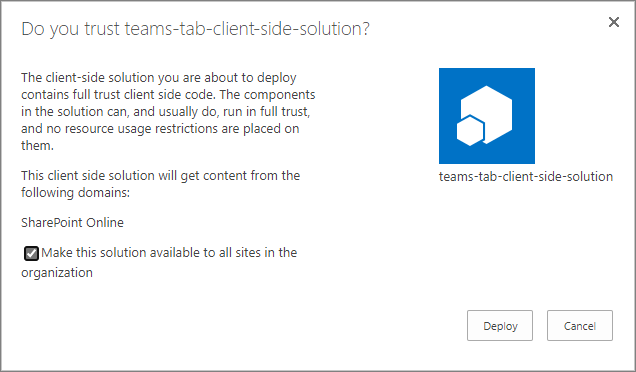
Upload or drag and drop the **teams-tab-webpart.sppkg** to the App Catalog.



1. This deploys the client-side solution package. Because this is a full trust client-side solution, SharePoint displays a dialog and asks you to trust the client-side solution to deploy.

Notice how the **domain** list in the prompt includes **SharePoint Online**. This is because the content is either served from the Microsoft 365 CDN or from the App Catalog, depending on the tenant settings.

Ensure that the **Make this solution available to all sites in the organization** option is selected, so that the web part can be used from the Microsoft Teams side.

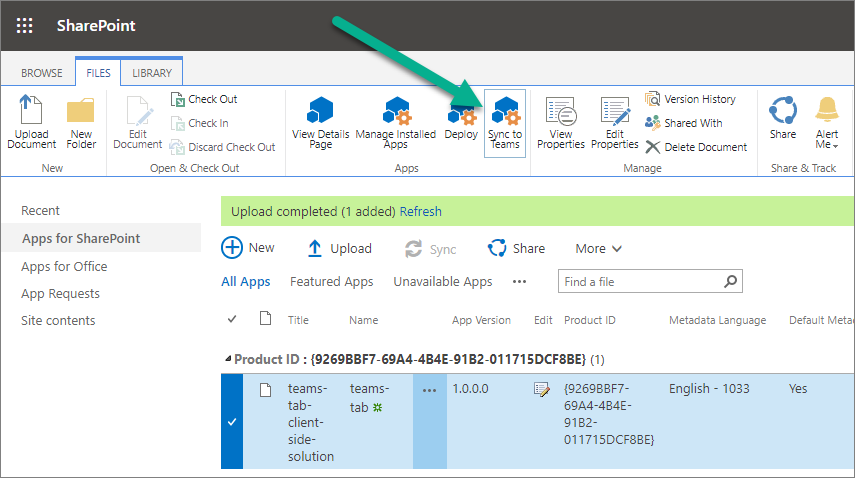


1. Select **Deploy**.

**Make the web part available in Microsoft Teams**

To make your web part available in Microsoft Teams, you'll have synchronize your solution with teams.

1. Select the **teams-tab-client-side-solution** package in the SharePoint tenant App Catalog and select the **Sync to Teams** button at in the ribbon in the **Files** tab.



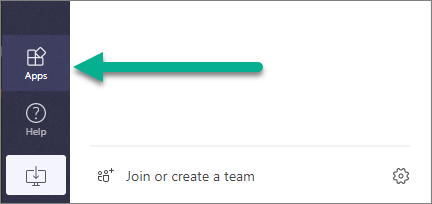
1. Confirm that you can see the status message on the top-right corner.

Sync to Teams status message

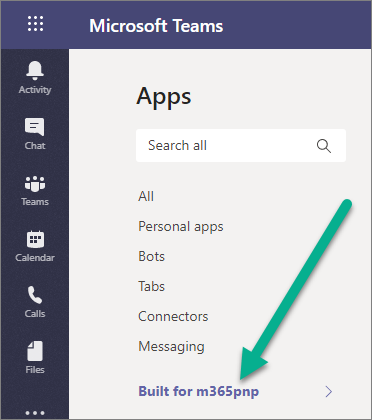
**Important**

If the synchronization isn't successful, please ensure that you have deleted possible previous version from the Microsoft Team App Catalog. This is the most common reason why the synchronization would fail as existing solution is not overridden currently.

1. Move to the Microsoft Teams side and check the App Catalog status by selecting **Apps** from the left navigation



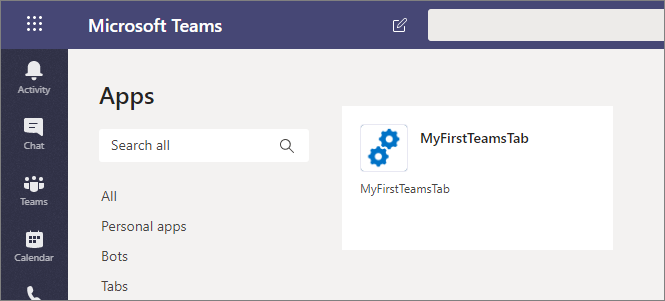
1. Wait until your tenant-specific apps selection is visible:



**Note**

This can take few minutes based due to caching.

1. Confirm that you can see **MyFirstTeamsTab** app in the catalog:



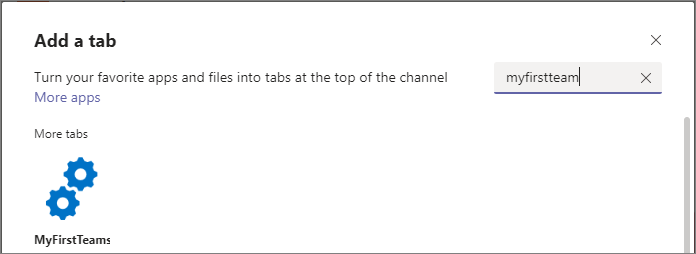
**Note**

This can take few minutes based due to caching.

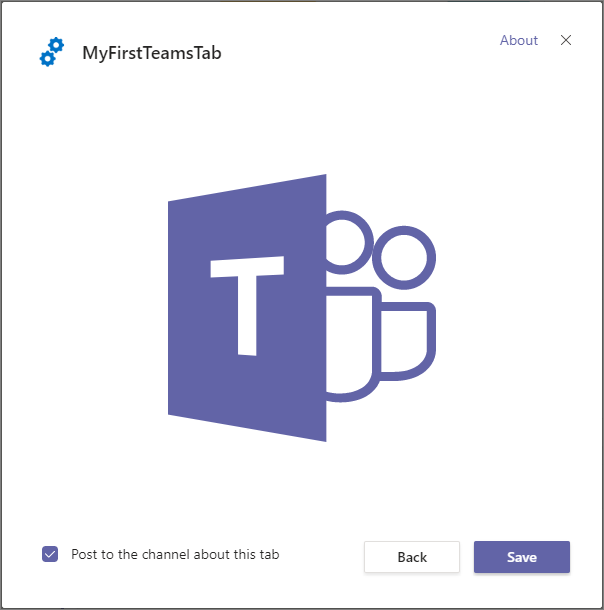
1. Move to a channel in a team where you want to test the solution. In the below picture, we have activated **General** channel in **Team**:



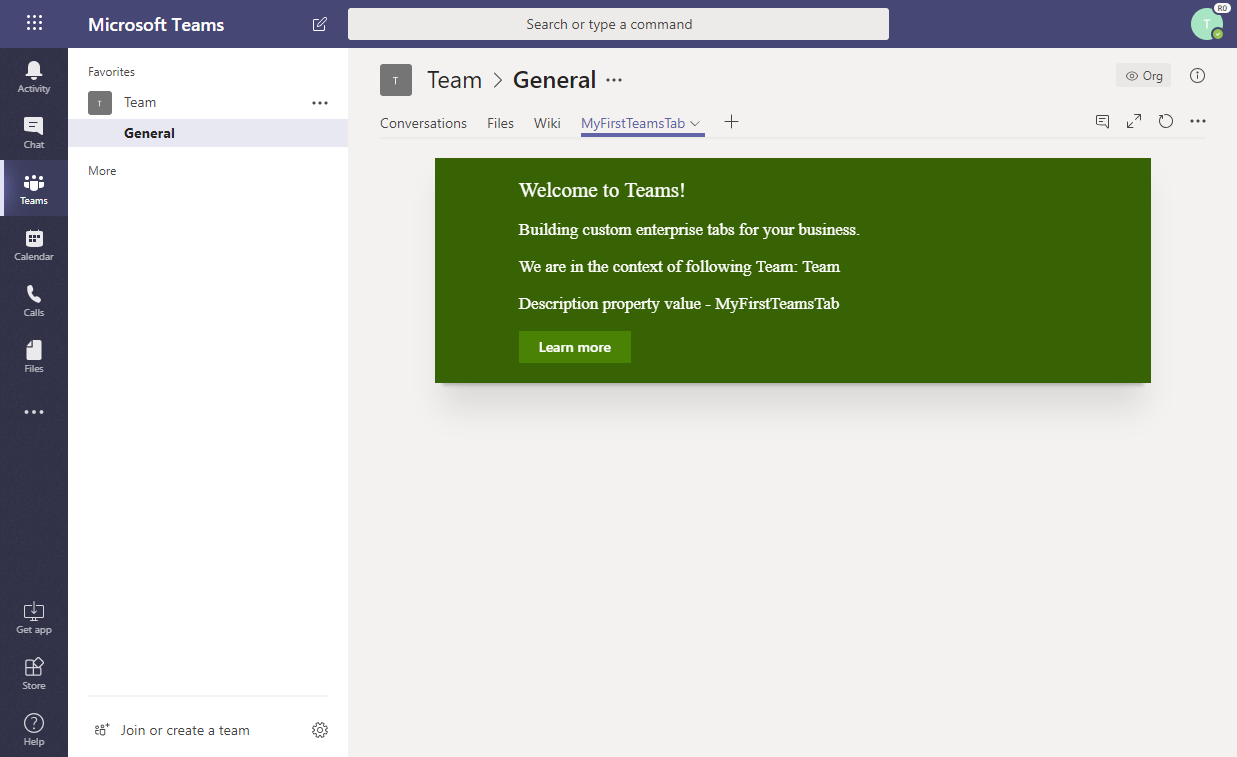
1. Select **+** to add a new tab on the channel.
2. Select your custom tab called **MyFirstTeamTab** in the list. You can use the search capability if needed to find your custom tab:



1. Select **Add** to confirm the tab to be selected.
2. Select **Save** to confirm the tab to be installed on the channel:



Your custom tab has been added on the Microsoft Teams channel and you can see how the code is reacting that it's in Microsoft Teams context. Theme of the web part is by default coming from the underlying SharePoint site.



**Note**

You can control the initial configuration option visibility from the web part manifest by setting canUpdateConfiguration property as false. By default this setting is true which is taken into account when the Teams manifest is automatically created as part of the *Sync* process.

**Update your app package**

When you make changes to your app and create a new package, you might find an error when selecting **Sync to Teams**.

A notification **Failed to sync solution to teams** might appear on the top right of your page.

If this happens follow these steps to delete your app from Microsoft Teams, and then try to sync it again:

1. Open Microsoft Teams.
2. When viewing your team channel, click + to **Add a tab**.
3. Select the **More apps** link at the top.
4. Find your app in the list, and click on the **...** menu.
5. Select **Delete** to remove the app from Microsoft Teams.

You can now sync your new version to Microsoft Teams.