**Build and Deploy multi-tab Microsoft Teams personal tab using SharePoint Framework(SPFx)**

## **Overview**

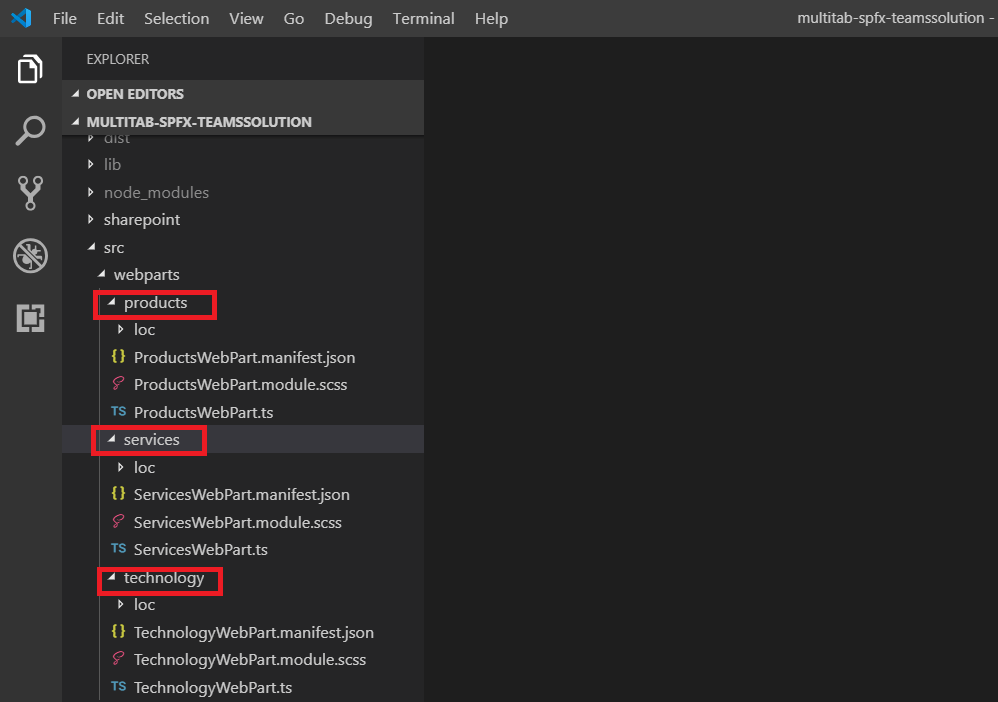
Creating and distributing an app built on the Microsoft Teams Platform involves deciding what to build, building your web services, creating an app package, and distributing that package to your target end users. SharePoint Framework is spreading its wings and is now ready to go beyond SharePoint and develop solutions targeting the Office 365 platform. SPFx already supports hosting the SPFx web part as tabs in MS Teams and create personal tabs in MS Team. A personal app is an app with a personal scope. As an app developer, you have the option to provide a version of your app that is built for the individual user. This way, you’re able to create a one-on-one interaction with your users.

In this exercise, we will explore how can build and deploy multi-tab Microsoft Teams personal tab  with SharePoint Framework (SPFx).

Follow the below steps to develop the multi-tab Microsoft Teams personal tab  with SharePoint Framework solution:-

## **Step 1: Create three Web Part’s under the same solution named:**

1. Products
2. Services
3. Technology

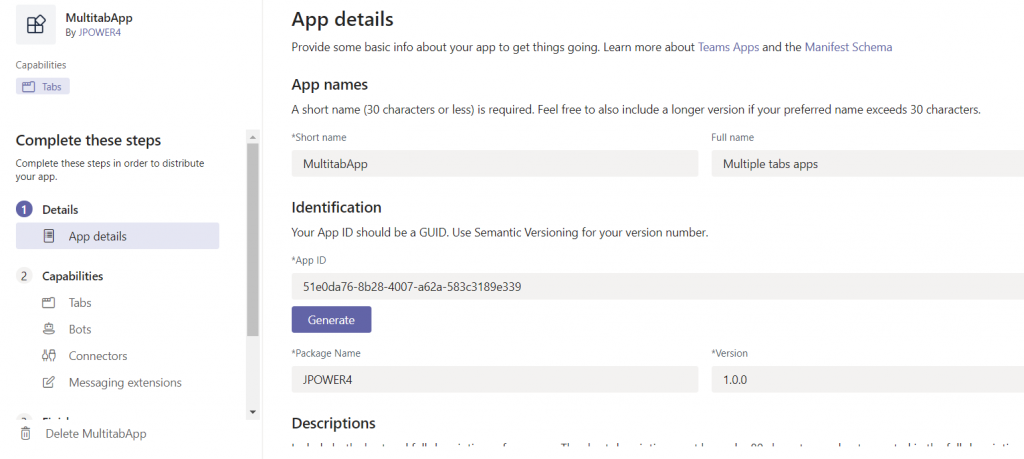


Update the style as per your requirement

Update the package-solution.json file with **“skipFeatureDeployment”: true** to deploy the web parts in tenant level

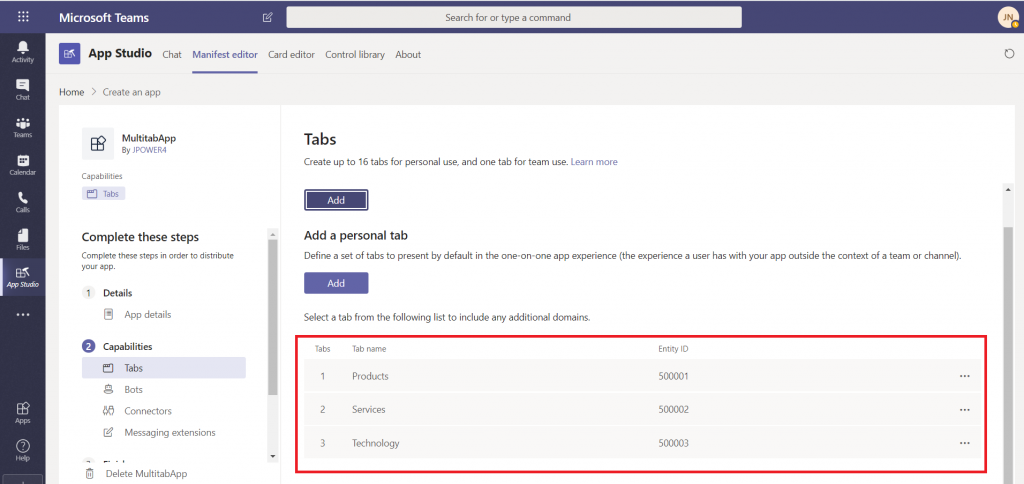
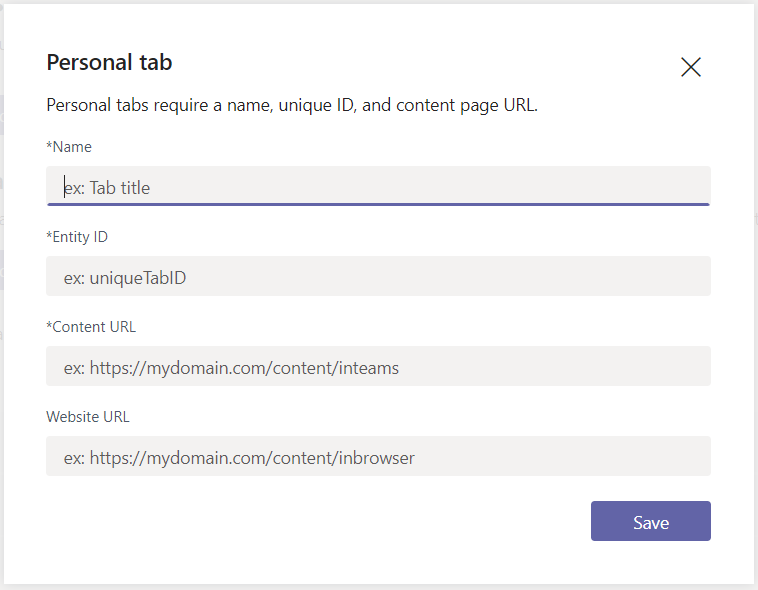
## **Step 2: Create App Manifest using App Studio with three tabs**

Add App details

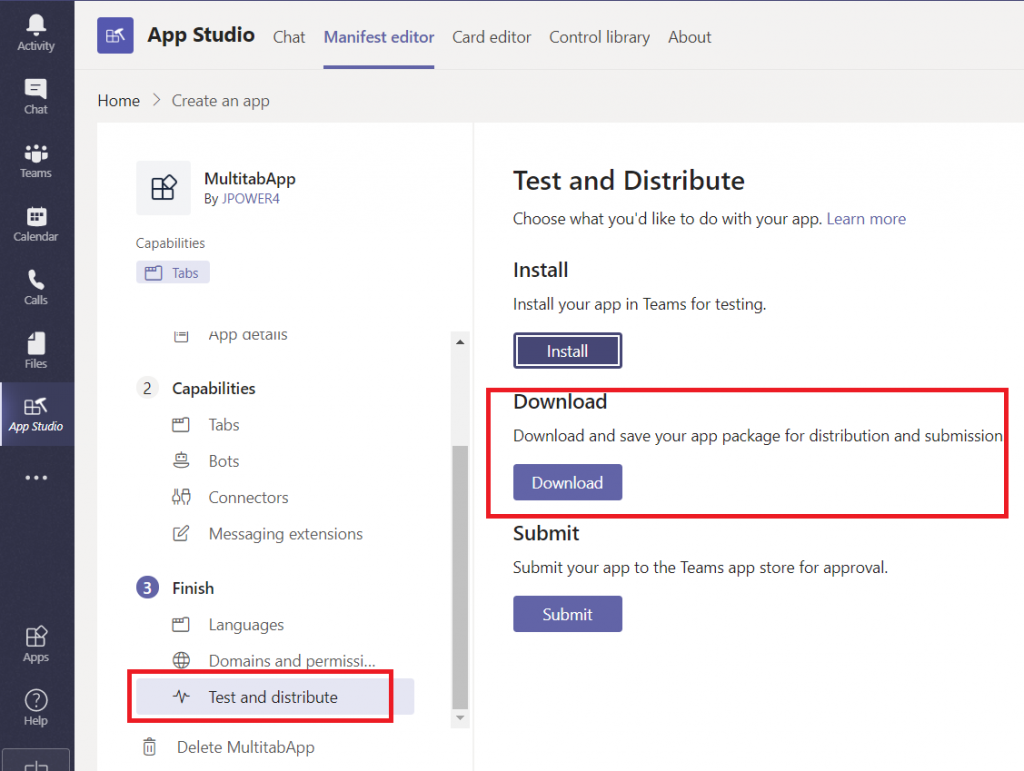


Capabilities –> Tab –> Add personal tabs

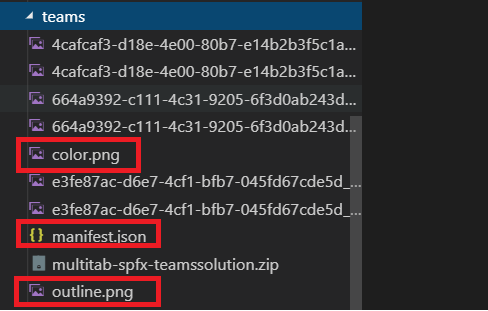
1. Name
2. Entity ID (Ex:10001)
3. Content Url – Add any https url later we change in manifest.json file
4. Website Url



Click Test and distribute and Download the package i.e zip file with manifest.json and two icons



Unzip the file and copy the three files and paste under teams folder



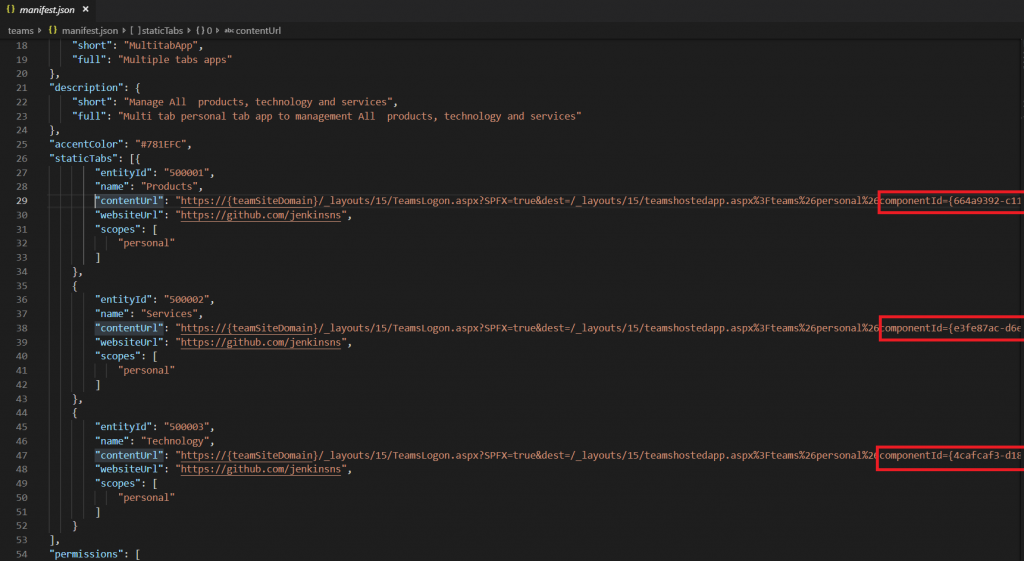
## **Step 3: Modify the manifest.json file**

refer here https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/guidance/creating-team-manifest-manually-for-webpart

Open manifest.json file and replace the content url for all three tabs

"contentUrl": "https://{teamSiteDomain}/\_layouts/15/TeamsLogon.aspx?SPFX=true&dest=/\_layouts/15/teamshostedapp.aspx%3Fteams%26personal%26componentId={{**SPFX\_COMPONENT\_ID**}}%26forceLocale={locale}",

Then update the {**SPFX\_COMPONENT\_ID**} based on the Webpart ID from webpart.json file Ex: ProductsWebPart.manifest.json file

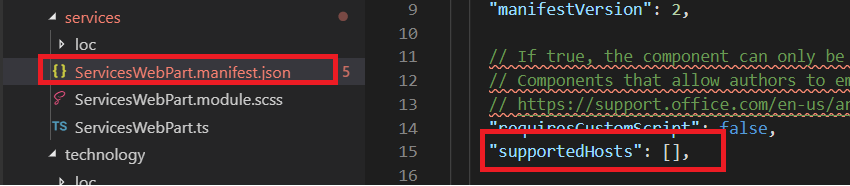


Then go to valid Domains section and update the domains

"\*.login.microsoftonline.com", "\*.sharepoint.com", "\*.sharepoint-df.com", "spoppe-a.akamaihd.net", "spoprod-a.akamaihd.net", "resourceseng.blob.core.windows.net", "msft.spoppe.com"

## **Step 4: update the deployment option in all three webparts**

“supportedHosts”: []



Save all and zip the manifest.json, color.png and ouline.png files

Select all three files and compress it to create zip file and give proper name for the zip file

## **Step 5: Deploy the App**

go to command line

gulp bundle --ship

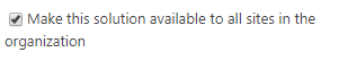
It will convert the Typescript to java script and bundle it

gulp package-solution --ship

It will ship the bundle to office 365 tenant cdn

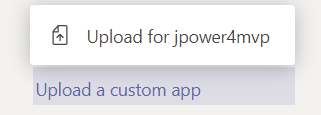
Open appcatlog site for your tennant and deploy the app

Select the ‘Make this solution available to all sites in the organization’ checkbox to deploy this app in tenant scoped

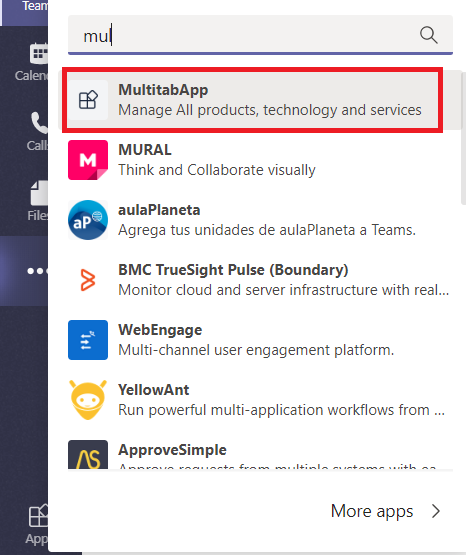


## **Step 6: Deploy teams app in Microsoft Teams**

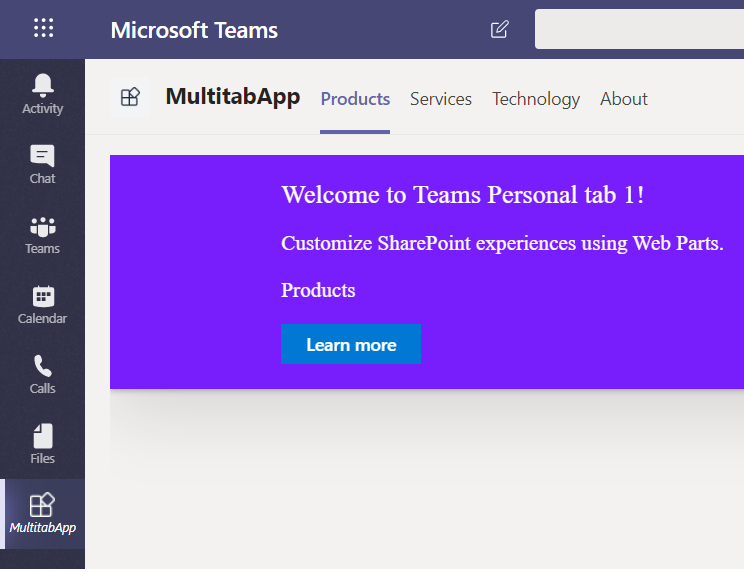
* Go to teams.microsoft.com
* upload a custom app



After upload go to left hand menu … and select your app and Add it



Now we got the solution with three tabs and all three webparts loaded in the tabs based on webpart ID



## **Summary**

This post shows how to use Microsoft Teams App Studio to create a manifest for SharePoint Framework solution which can provide a multi-tab experience and potentially to include also a bot on the personal solution. This setup is aligning with the design guidance for the Microsoft Teams personal apps to have standard set of tabs for end users. Post demonstrates creation of the solution manifest file manually, rather than taking advantage of the standard “Sync to Teams” synchronisation for getting SharePoint Framework solutions also available in Microsoft Teams.