Microsoft Teams Development Bootcamp Labs

DISCLAIMER

Microsoft, Microsoft Edge, Office 365, PowerShell, Visual Studio, and Windows PowerShell are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

No part of the text or software included in this training package may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission from Microsoft. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

To obtain authorization for uses other than those specified above, please visit the Microsoft Copyright Permissions Web page at https://www.microsoft.com/en-us/legal/intellectualproperty/permissions

This content is proprietary and is intended only for users described in the content provided in this document. This content and information are provided to you under a non-disclosure agreement and cannot be distributed. Copying, disclosing all or any portion of the content and/or information included in this document is strictly prohibited.

Microsoft Teams Development Bootcamp Lab

The rapid growth of Microsoft Teams and new sets of Teams platform capabilities have opened great opportunities for ISVs, system integrators, managed solution providers, and services partners to integrate, extend, and tailor Teams to meet their customer's needs. This virtual hands-on lab comes with pre-installed software and an Office 365 tenant to get you started on developing Teams apps. During these labs, you will have the opportunity to gain hands-on experience building a Teams app that makes use of the Teams Developer Platform (Tabs, bots, connectors, message extensions, and more). The lab will guide you through building the app for the following scenario:

- HR Talent App Track (Teams app development)
  • Quickly hire new talent into their organization
  • Provide immediate interview feedback
  • Schedule interview loops and improve the overall hiring process of new employees

Lab: Building an HR hiring app solution for Microsoft Teams

During this lab, you will learn how to build and deploy a Microsoft Teams app in Office 365 that will be used by the Human Resources department within their Microsoft Teams clients. The app will facilitate the department's hiring of new talent into the organization, provide immediate interview feedback, schedule interview loops, and improve the overall hiring process of new employees. The lab is divided into several exercises that will help you understand how to transform hiring and candidate management flow of new talent and make it more interactive and responsive for HR teams and interviewees.

Estimated time to complete: ***3 hours 30 minutes***

Before you begin

To complete the exercises in this lab, you will create Microsoft Office 365 tenant with an Enterprise E5 trial subscription.

What you will learn

After completing the exercises in this lab, you will be able to:

- Deploy an app in Office 365
- Understand how to define information in the app's manifest JSON file
- Review and update an app package and verify Teams' specific content is present
- Deploy your app package to Microsoft Teams
- Design and extend your Microsoft Teams app with tabs, bots, adaptive cards, connectors, and messaging extensions
- Test the capabilities of your Microsoft Teams app

Scenario

You are developing apps for your company's Microsoft Teams deployment. You've spent many hours developing apps that are currently deployed in SharePoint Online and you want to reuse them as much as possible. Using the skills you already have, you are ready to begin creating apps, bots, connectors, and messaging extensions in Microsoft Teams. You also want to become familiar with using Graph Explorer to interact with Microsoft Teams.

Exercise 1: Setting up your development environment and creating a new app

In this exercise, you will familiarize yourself with the development environment as well as the demo tenant and start building the scaffolding of the app using Teams App Studio.

Tasks

1. **Prepare your Office 365 environment**
    1. **IMPORTANT!** You must have an Office 365 tenant to complete this lab. If you don't have one, you can sign up for an Office 365 developer subscription by following the instructions here.
    2. **IMPORTANT!** Because of limitations in Teams App Studio, a Microsoft Teams free tenant will **not** work for this lab.

2. □ **Sign into the Microsoft 365 admin center**
    1. Open Microsoft Edge and then browse to **https://admin.microsoft.com**.

       In this lab, references to Microsoft Edge are specific to the browser used inside the virtual machine. If the browser on your computer is required to perform a task, the lab will specify that you use the browser on the host computer.

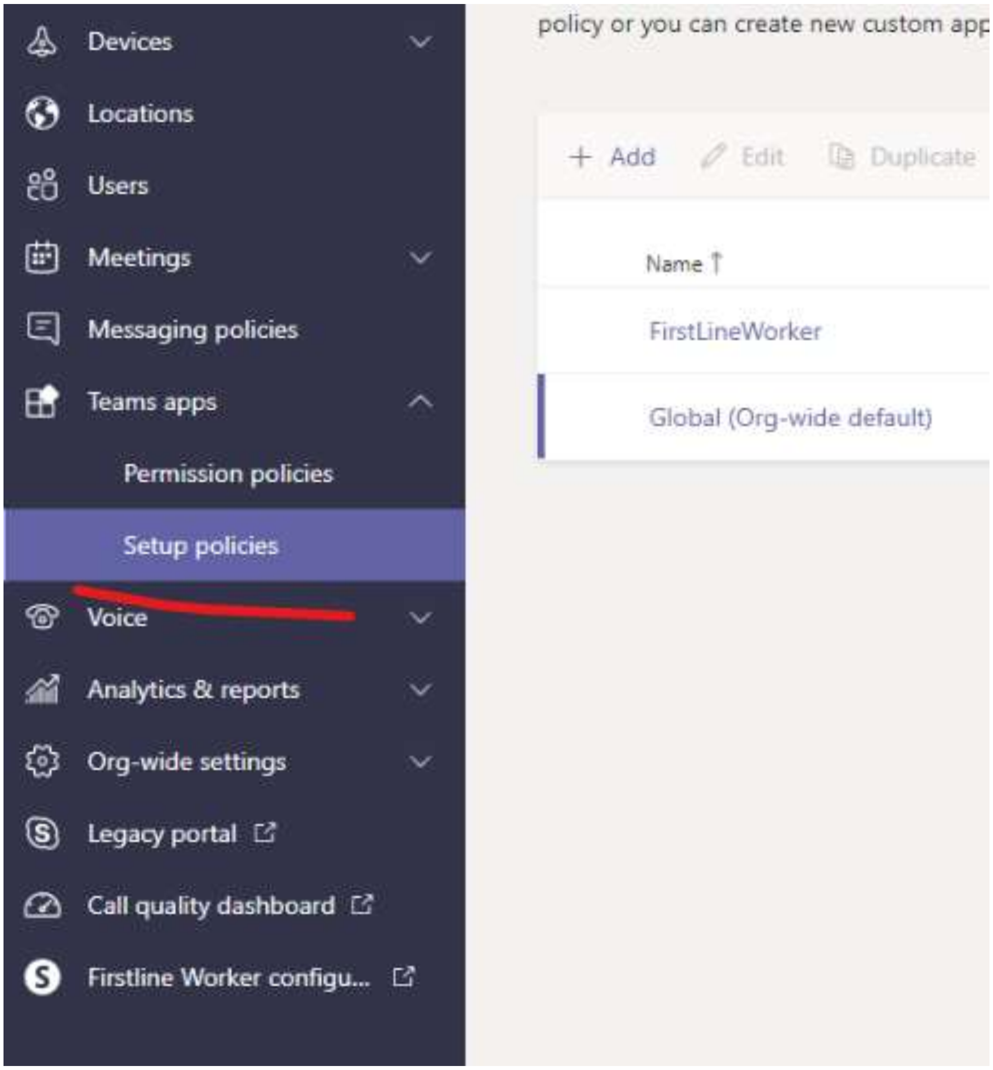
    2. In the Sign in window, enter **admin credential**
    3. Select **Next**.
    4. In the **Password** box, enter **[PASSWORD]**.
    5. Select **Sign in**.

        Although saving credentials is not required for the lab, it may help if your browser or session times out.

    6. In the Microsoft 365 admin center, on the right, verify that **The new admin center** switch is enabled.
       If this option is not enabled, you may need to select the **Try the new admin center** switch.

        You may need to change the browser Zoom from 100% to 75% if there is no option for the new admin center or preview of the new admin center. If you are unable to find the option, continue with the lab. The new admin center may be fully deployed and the old admin center no longer available.

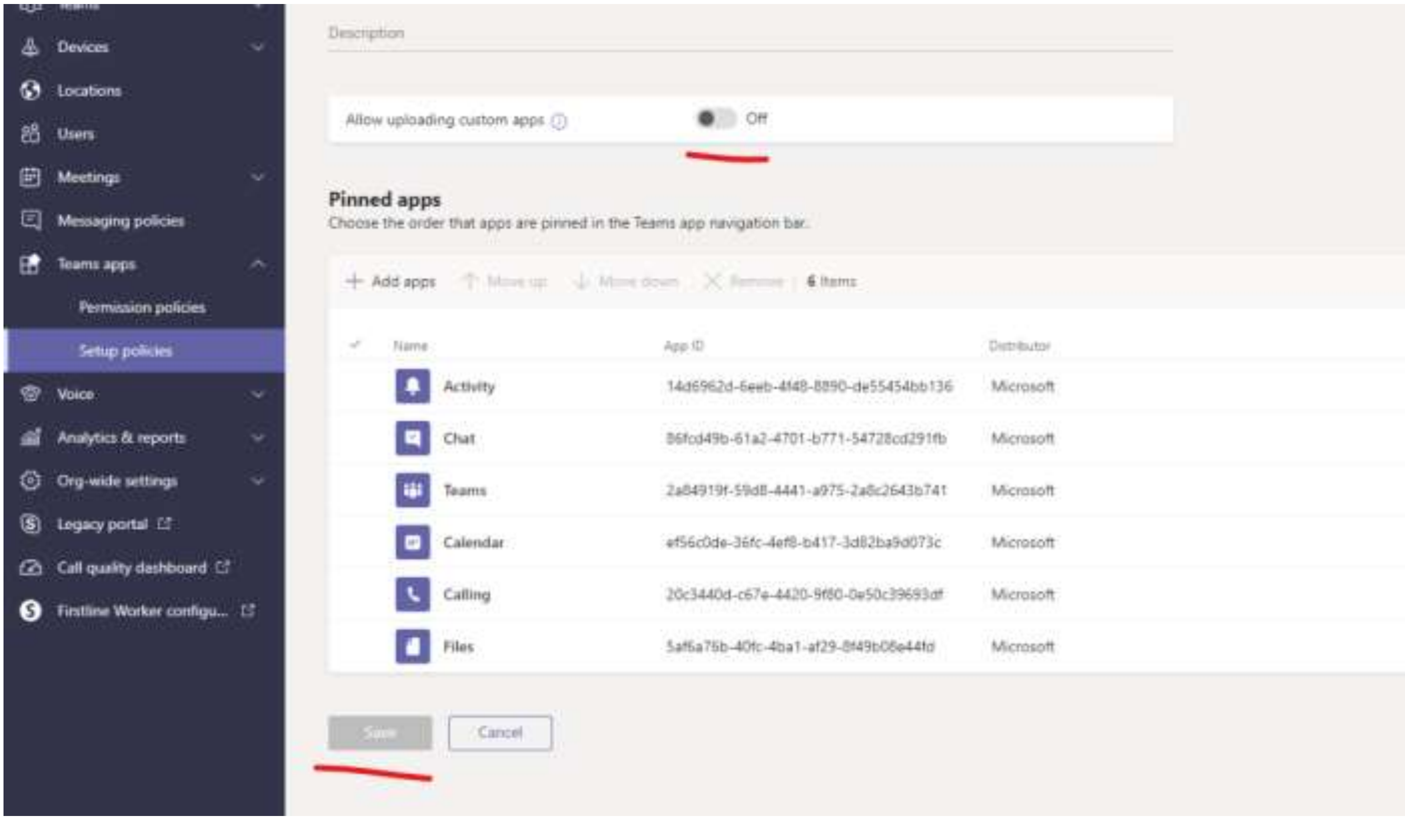
3. □ **Enable uploading custom apps**
    1. You will first need to enable side loading for Teams within your Office 365 environment. Open the Admin Center by visiting from your browser. Login using your admin credentials
    2. Click on ***Setup Policies*** under ***Teams App***.



policy or you can create new custom app

Name ↑
FirstLineWorker
Global (Org-wide default)

3. Click on ***Global (Org-wide default)***
    1. Turn ON ***Allow uploading custom apps***
    2. Click ***Save***



Allow uploading custom apps: Off

Pinned apps

Choose the order that apps are pinned in the Teams app navigation bar.

6 items

Name	App ID	Distributor
Activity	14d6962d-6eeb-4f48-8890-de55454bb136	Microsoft
Chat	86fcd49b-61a2-4701-b771-54728cd291fb	Microsoft
Teams	2a84919f-59d8-4441-a975-2a8c2643b741	Microsoft
Calendar	ef56c0de-36fc-4ef8-b417-3d82ba9d073c	Microsoft
Calling	20c3440d-c67e-4420-9f80-0e50c39693df	Microsoft
Files	5af6a76b-40fc-4ba1-af29-8f49b08e44fd	Microsoft

1.

   □ **Run the HRHiringAppLab.exe script**

   1. Run the **HRHiringAppLab.exe (located in the repo)**
   2. In the **User Account Control** dialog box, select **Yes** and then wait for the application to open.
   3. In the **Office 365 Admin username** box, enter **[USERNAME]**.
   4. In the **Office 365 Admin password** box, enter **[PASSWORD]** and then select **Verify Credentials**.
   5. Ensure the credentials and licenses were verified and then select **Run Script**.
      The script will create resources in your Office 365 subscription that are used in the lab.
   6. In the Lab Provisioning Tool, review the information in the output text box. You may continue with the lab and return to this step later to ensure the script has completed.
      When the script has completed, select **Finish**.
      If the script takes longer than 15 minutes to complete, select **Cancel** and run the script again. Close File Explorer.

□ **Install App Studio from the Store**

1. In Microsoft Teams, in the app bar on the left, select the **More added apps** ellipsis icon and then select **More apps**.
   The **More added apps** ellipsis icon is in the app bar on the left under **Files**.

   2. On the Apps page, in the **Search all** box, enter **App Studio**.
   3. In the results, select **App Studio**.
   4. In the **App Studio** dialog box, review the information and then select **Add**.
      If you have already installed App Studio as part of another lab, you do not need to install again.
   5. If prompted, close the **App Studio is now available to you** dialog box.

6. □ **Create a new app named Contoso HR Talent App**
   1. In Microsoft Teams, in the app bar, select the **More added apps** ellipsis icon and then select **App Studio**.
   2. In App Studio, select the **Manifest editor** tab.
   3. Under **Welcome**, select **Create a new app**.
   4. In the left navigation, under **Complete these steps** verify that **App details** is selected.
   5. On the App details page, in the **Short name** box, enter **Contoso HR Talent App**.
   6. In the **Long name** box, enter **Human Resources New Hiring App**.
   7. Under **Identification** review the information and then select **Generate**.
   8. Review the unique identity given to your new app.
   9. Select the **App ID** and then press Ctrl+C.
   10. In the following text box, paste or type the App ID.
       This will automatically add the App Id to this lab document.

       You may also want to save the App ID in Notepad.
   11. In the **Package Name** box, enter **<TenantName>.hr.com.microsoft.com**.
   12. In the **Version** box, enter **1.0.0**.
   13. Under **Descriptions**, in the **Short description** box, enter **Assist HR talent hiring**.
   14. In the **Long description** box, enter **This Human Resources app is designed to aid the HR department with their new hire acquisitions.**
   15. Under **Developer information**, in the **Name** box, enter your name.
   16. In the **Website** box, enter **https://<TenantName>.sharepoint.com/sites/ContosoHR**.
   17. Under **App URLs**, in the **Privacy statement** box, enter **https://<TenantName>.sharepoint.com/privacystatement**.
   18. In the **Terms of use** box, enter **https://<TenantName>.sharepoint.com/servicesagreement**.
   19. Under **Branding**, under **Full color**, select **Update**.
   20. Select **FullColorBrand.png** and then select **Open**.
   21. Under **Transparent outline**, select **Update**.
   22. Select **TransparentBrand.png** and then select **Open**.
   23. On the App Studio page, under the tabs, in the breadcrumb navigation, select **Home**.
   24. In the Manifest editor tab, verify that the new app is shown.
       The app is automatically saved.

7. □ **Create a forwarding URL for your app**
    1. Download ngrok.exe to your C:\LabFiles (pre-create this folder in advance)
    2. Right-click or tap and hold **Start** and then select **Windows PowerShell (Admin)**.
    3. In the **User Account Control** dialog box, select **Yes**.
    4. In Windows PowerShell, enter the following and then press Enter:

       ```
       cmd
       ```

    5. In Windows PowerShell, enter the following and then press Enter:

       ```
       cd C:\LabFiles
       ```

    6. In Windows PowerShell, enter the following and then press Enter:

       ```
       ngrok http 5000 --host-header=localhost:5000
       ```

    7. Review the output of the command. Note the **Session Expires** timer.

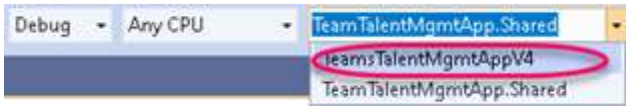
       > As Microsoft Teams is an entirely cloud-based product, it requires all services it accesses be available from the cloud using HTTPS endpoints. To enable the exercises in this lab to work within Microsoft Teams, a tunneling application is required. This lab uses ngrok for tunneling publicly-available HTTPS endpoints to a web server running locally on the developer workstation.
       > Ngrok exposes local services that are behind NAT networks and firewalls to the public internet over secure tunnels. The free tunnel used in this lab will expire after a period of 8 hours. After it expires, or if you quit or close the command window, the tunnel will need to be recreated. The new tunnel will have a new URL. The new URL will need to be updated wherever the previous URL/app URL is published.

    8. Select the FQDN in the forwarding URL, right-click or tap the selected URL.
       This will copy the text to the clipboard.
       For example, if the forwarding URL is https://123456.ngrok.io, select ***123456.ngrok.io***.
    9. In the following text box, paste or type the URL.
       This will automatically add the URL to this lab document.

       You may also want to save the URL in Notepad.
    10. Leave the command window open or minimize the window.
        This will keep a remote connection open to the local app instance.

        > Do not close the console window. The console session keeps the tunnel connection open to the app you are running locally.

8. □ **Open and build the TeamsTalentMgmtApp using Microsoft Visual Studio**
   This is the Contoso Talent app that will be deployed.
   1. Install Visual Studio 2019 Community edition if you don’t have Visual Studio 2019 installed
   2. Select **Start** and then select **Visual Studio 2019**.
   3. In Visual Studio **Welcome** dialog box, select **Sign in**.
   4. In the **Sign in** dialog box, in the **Email, phone, or Skype** box, enter **[USERNAME]** and then select **Next**.
   5. In the **Enter password** dialog box, in the **Password** box, enter **[PASSWORD]** and then select **Sign in**.
   6. In the **Visual Studio** dialog box, select **Start Visual Studio**.
      The General development settings will be used for this lab.
   7. In the Visual Studio window, under **Community 2019**, select **Check for an updated license**.
   8. After the license is updated, select **Close**.
   9. In the Visual Studio window, under **Get started**, select **Open a project or solution**.
   10. In the Open Project/Solution window, browse to **msteams-sample-contoso-hr-talent-app**/**TeamsTalentMgmtApp** (make sure you clone the repo to your machine)
   11. Select **TeamsTalentMgmtApp.sln** and then select **Open**.
   12. In the **Security Warning for TeamTalentMgmtApp.Shared** dialog box, clear the **Ask me for every project in this solution** check box.
   13. Select **OK**.
       Wait for the solution to load. When loaded, you will see content on the right in Solution Explorer and you will see **Restore completed…** at the bottom of the window.
   14. On the menu, select **Build** and then select **Build Solution**.
       Monitor the Output pane at the bottom of the screen and wait for the build to complete. This may take 1-2 minutes.
9. □ **Update the appsettings file with the app URL and app ID**
   1. On the right, in Solution Explorer under **src** > **TeamsTalentMgmtAppV4**, select **appsettings.json**.
   2. In the appsettings file, locate **BaseUrl**.
   3. Replace the current value of **PASTE_YOUR_NGROK_URL_HERE** with **https://<appURL>**.
   4. In the appsettings file, locate **TeamsAppId**.
   5. Replace the current value of **PASTE_YOUR_TEAMS_APP_ID_HERE** with **<appId>**.
   6. On the menu bar, select the **Save** icon or press Ctrl+S.
10. □ **Run debugging using IIS Express**
    1. In Visual Studio, on the menu bar, select the **Startup Projects** menu and verify **TeamsTalentMgmtAppV4** is selected. If necessary, select **TeamsTalentMgmtAppV4**.



2. On the menu, select **Debug** > **Start Debugging**.
   Alternatively, you can press F5 on your keyboard.
3. In the **Microsoft Visual Studio** dialog box, review the information and then select **Yes**.
4. In the **Security Warning** dialog box, review the information and then select **Yes**.
   This is the self-signed certificate being used by Visual Studio.
5. Microsoft Edge will be opened automatically and a new tab named localhost will open.
6. The page should read **Status Code: 404; Not found**. This is expected.
7. After the page successfully loads, close the **localhost** tab.
8. Leave Visual Studio open with the debugger running.
   This will keep the application available for use later.

< Table of contents

Exercise 2: Deploying a SharePoint web part as a Microsoft Teams tab

In this exercise, you will deploy a SharePoint web part as a Microsoft Teams tab into your HR Hiring channel. First you will focus on making sure your SharePoint Solution is aware of the Microsoft Teams context. Then you will work on packaging your SharePoint Solution and deploying it as a client-side web part to a SharePoint page. The SharePoint web part contains a list of current candidates, positions they are applying for, the candidates’ status, and assigned members of the hiring team who will be conducting the interviews.

Tasks

1. □ **Add the service URL to the Contoso Talent webpart settings.json file**
   1. Install Visual Studio Code
   2. In your PC, select **Start** > **Visual Studio Code** > **Visual Studio Code**.
   3. Close the any open tabs.
      For example, Welcome or Release Notes tab.
   4. In Visual Studio Code, on the menu, select **File** > **Open File**.
   5. In the Open File window, browse to **SharePointTalentMgmtWebPart**\src\\webparts\\contosoTalent.
   6. Select **settings.json** and then select **Open**.
   7. To the right of **ServiceURL:**, replace **PASTE_YOUR_NGROK_URL_HERE** with **https://<appURL>**.
   8. On the menu, select **File** > **Save**.

2. □ **Open ContosoTalentWbPart.ts**
    1. In Visual Studio Code, on the menu, select **File** > **Open File**.
    2. In the Open File window, browse to **SharePointTalentMgmtWebPart \src\webparts\contosoTalent**
    3. Select **ContosoTalentWbPart.ts** and then select **Open**.
3. □ **Verify the teamsContext private variable exists inside the ContosoTalentWebPartProps class**
    1. In ContosoTalentwebParts.ts, locate **export default class ContosoTalentWebPartProps**.
       This will be on line 34.
    2. Under **export default class ContosoTalentWebPartProps**, verify that the **private _teamsContext: microsoftTeams.Context;** variable exists.
       This variable stores the Microsoft Teams context information.
    3. No changes need to be made.
4. □ **Verify the onInit method is present**
    1. In ContosoTalentWebParts.ts, locate **protected onInit()**.
       This will be on line 43.
    2. Make sure the solution is aware of the Microsoft Teams context, when it's used as a tab. Verify the following code is present:

```
protected onInit(): Promise<any> {
let retVal: Promise<any> = Promise.resolve();
if (this.context.microsoftTeams) {
retVal = new Promise((resolve, reject) => {
    this.context.microsoftTeams.getContext(context => {
    this._teamsContext = context;
    resolve();
       });
   });
   }
   return retVal;
}
```

    14. No changes need to be made.
5. □ **Review the render method**
    1. In ContosoTalentWebParts.ts, locate **let baseClass**.
       This will be on line 70.
    2. Review the code used for the baseClass.
       Notice that content will be rendered differently when the code is rendered as a tab in Microsoft Teams or as a web part in SharePoint.
    3. No changes need to be made.
6. □ **Start a new terminal session using Visual Studio Code**
    1. In Visual Studio Code on the menu, select **Terminal** > **New Terminal**.
    2. At the PowerShell prompt, enter the following and then press Enter:

```
cd C:\LabFiles\SharePointTalentMgmtWebPart
```

You may need to select inside the terminal pane and press Enter to see the prompt.

To save time in the lab, the npm package manager and gulp package have already been installed using the *npm install* command.

7. **Install SharePoint Framework Development environment**
    1. Follow steps from [this document]() to install all required tools for SharePoint Framework development

8. □ **Create a release build of the solution**
    1. In the terminal pane, at the PowerShell prompt, enter the following and then press Enter:

        ```
        gulp bundle --ship
        ```

    2. Wait for the build to complete.

        A message stating that the *build failed because a task wrote output to stderr* can be safely ignored. This is a bug in gulp and does not cause the application package to fail.

9. □ **Package the Contoso Talent web part**
    1. In the terminal pane, at the PowerShell prompt, enter the following and then press Enter:

        ```
        gulp package-solution --ship
        ```

    2. Review the output of the command.

        This will create the **contoso-talent.sppkg** file in the C:\LabFiles\SharePointTalentMgmtWebPart\sharepoint\solution folder.

10. □ **Verify the debugger is still running in Visual Studio**
    1. Switch to Microsoft Visual Studio.
    2. On the right, verify the Diagnostic Tools pane is displayed and showing session diagnostics.
    3. If Diagnostic Tools are not running:
        - In Visual Studio, on the menu bar, select the **Startup Projects** menu and verify **TeamsTalentMgmtAppV4** is selected.
        - On the menu, select **Debug** > **Start Debugging**.
          Alternatively, you can press F5 on your keyboard.
    4. Leave Visual Studio open with the debugger running.

11. □ **Create a new app catalog for your SharePoint Online tenant**
    1. Switch to Microsoft Edge.

2. Open a new tab and then browse to **https://<TenantName>-admin.sharepoint.com**.
   You should automatically be signed in. If not, sign in as **[USERNAME]** with password **[PASSWORD]**.
3. In the left navigation, select **Classic features**.
4. In the Classic features window, under **Apps**, select **Open**.
5. On the Apps page, select **App Catalog**.
   If the **Create a new app catalog site** option is not displayed, then the app catalog is already configured and you can skip to the next task.
6. On the App Catalog Site page, verify **Create a new app catalog site** is select and then select **OK**.
7. On the Create App Catalog Site Collection page, in the **Title** box, enter **AppCatalog**.
8. In the **Web Site Address** box, to the right of **/sites/**, enter **appcatalog**.
9. In the **Administrator** box, enter **[USERNAME]** and then select the **Check Names** icon.
10. Verify your administrator name resolves and then select **OK**.
11. Close the **Apps** tab and then switch to the **SharePoint admin center** tab.
12. In the SharePoint admin center in the left navigation under **Sites**, select **Active sites**.
13. Under **Site name**, locate **AppCatalog**.
    If the site is not listed, refresh the page. You may need to periodically refresh the page until the AppCatalog site has been created. This may take 2-3 minutes.

12. □ **Upload the Contoso Talent package to the app catalog**
    1. In the **Active sites** list, to the right of **AppCatalog** in the URL column, select the **…/sites/appcatalog** link.
    2. On the App Catalog page, in the left navigation, select **Apps for SharePoint**.
    3. On the Apps for SharePoint page, select **Upload**.
    4. In the **Add a document** dialog box, next to **Choose a file**, select **Browse**.
    5. In the Open windows, browse to **C:\LabFiles\SharePointTalentMgmtWebPart\sharepoint\solution**.
    6. Select **contoso-talent.sppkg** and then select **Open**.

       > If the package is not present, return to the *Create a release build of the solution* task and verify or perform the required steps to create the package.

    7. In the **Add a document** dialog box, select **OK**.
    8. In the **Do you trust contoso-talent-client-side-solution** dialog box, review the information.
    9. Select the **Make this solution available to all sites in the organization** check box and then select **Deploy**.

13. □ **Verify there are no errors for the app package**
    1. On the Apps for SharePoint page, in the **All Apps** list, locate the **contoso-talent-client-side-solution** app.

2. To the right, under the **App Package Error Message** column, verify that **No errors** is displayed.
   You may need to scroll to the right to see the column.

   > If you see an error in the column, delete the failed upload by selecting the package, select the **More** menu and then select **Delete**. Wait 1-2 minutes, and then upload the app again.

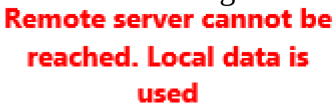
14. □ **Add the Contoso Talent app to a new page in the Contoso HR site**

1. In Microsoft Edge, browse to **https://<TenantName>.sharepoint.com/sites/ContosoHR**.
   This site was created by the script executed at the beginning of this exercise.
2. In the top right, select the **Settings** menu gear icon and then select **Add a page**.
3. Close any open introduction wizard.
4. On the new page, select **Name your page** and then enter **Contoso Talent**.
5. On the Contoso Talent page, in the middle, select the **Add a new web part in column one** ⊕ icon.
6. Under **All A to Z**, select **Contoso Talent**.

   > If the Contoso Talent app is not listed, you may need to wait 3-5 minutes until it has been made available in SharePoint Online.

7. Wait for the web part to load.

   > Your web part will display information that is hosted by the running debugger in Visual Studio. You can switch to the PowerShell console running ngrok and see the connection that has been made.
   >
   > If you see an error, similar to the one below, you must verify all your connection settings before continuing and update your apps if necessary.



Remote server cannot be reached. Local data is used

   > You may run into this error later if the ngrok session times out. Use the instructions in [Appendix B](#) to reconnect your app if ngrok times out or has been reset.

8. In the top right, select **Publish**.
9. In the Help others find your page pane, select **Add page to navigation** and then close the pane.

15. □ **Review the webpart**

1. In Microsoft Edge, refresh the Contoso Talent page and then review the webpart.
2. Select the different tabs in the web part and review the information and functionality.

16. □ **Create a compressed archive from the Teams folder content**

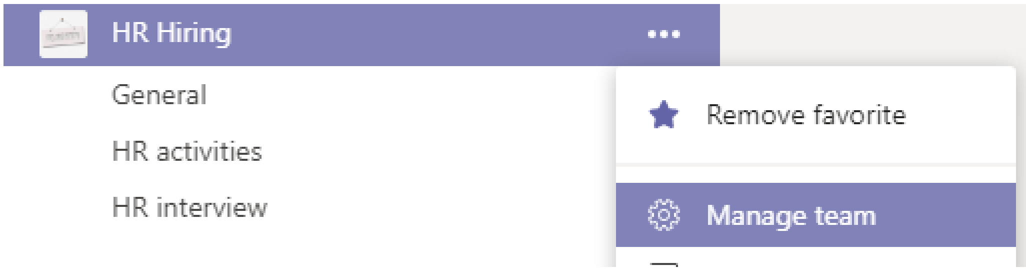
1. In YOUR PC, right-click or tap and hold **Start** and then select **Microsoft PowerShell (Admin)**.
   For this lab, PowerShell is being used to create the compressed (zip) file. Alternative methods, such as the GUI or third-party apps can be used as well.
2. In the **User Account Control** dialog box, select **Yes**.
3. In Windows PowerShell, enter the following and then press Enter:

```
Compress-Archive -Path
C:\LabFiles\SharePointTalentMgmtWebPart\teams\* -DestinationPath
C:\LabFiles\Teams.zip
```

4. Wait for the command to complete and then close Windows PowerShell.

17. □ **Upload a custom app to the HR Hiring Team**

1. Switch to the Microsoft Teams desktop app signed in as the tenant administrator.
2. In the app bar, select **Teams**.
3. In the Teams pane, to the right of **HR Hiring**, select the **More options** ellipsis icon and then select **Manage team**.



4. On the HR Hiring page, on the menu, select the **Apps** tab.
5. In the lower-right corner, select **Upload a custom app**.

If *Upload a custom app* is not visible in the lower right corner of the Apps tab, custom app upload (sideloading of external apps) may not be fully provisioned in your subscription. You can perform the following steps as a work around, however, you may still have problems uploading a custom app later if provisioning is still happening.
- Select **More apps**.
- In the Apps window, select **Upload a custom app** > **Upload for Contoso**.
- In the Open window, browse to **C:\LabFiles\**
- Select **Teams.zip** and then select **Open**. - In the Apps window, select **Contoso Hiring Board**.
- In the **Contoso Hiring Board** dialog box, review the information and then select **Add to a team**.
- In the **Select a channel to start using Contoso Hiring Board** dialog box, select the **Search** box, enter **HR activities** and then select **HR activities HR Hiring**.
- Select **Set up a tab**.

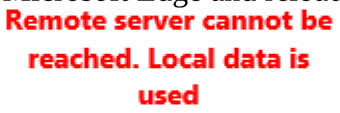
- In the **Contoso Hiring Board** dialog box, select **Save**.
- In the Contoso Talent pane, review the Service URL. This is the same URL you updated in the settings.json file. Close the Contoso Talent pane.
- Review the app and notice that it is the same app used in the Contoso HR site. Changes made in the Microsoft Teams channel tab are reflected in the Contoso HR SharePoint site and changes in the Contoso HR SharePoint site will be reflected in the Microsoft Teams channel.
- Skip the remaining steps and proceed to the next exercise.

6. In the Open window, browse to **C:\LabFiles**.
7. Select **Teams.zip** and then select **Open**.
8. Review the **Apps** list and verify the **Contoso Hiring Board (Custom app)** is listed.

18. □ **Add the Contoso Hiring Board tab to the HR activities channel**

1. In the Microsoft Teams desktop app, on the HR Hiring page, select the **Channels** tab.
2. In the **Channels** list, select **HR activities**.
3. In the HR activities channel, on the menu, select the **Add a tab +** icon.
4. In the Add a tab window, select **Contoso Hiring Board**.
5. Select **Save**.
6. In the Contoso Talent pane, review the Service URL. This is the same URL you updated in the settings.json file.
   Close the Contoso Talent pane.
7. Review the app and notice that it is the same app used in the Contoso HR site. Changes made in the Microsoft Teams channel tab are reflected in the Contoso HR SharePoint site and changes in the Contoso HR SharePoint site will be reflected in the Microsoft Teams channel.

If you see the error shown below, switch to Visual Studio and verify or restart the debugger. After the debugger has been restarted, return to Microsoft Edge and reload the page.



Remote server cannot be reached. Local data is used

More information about the tasks in this exercise can be found using these Microsoft documents:
Building Microsoft Teams tab using SharePoint Framework
https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/get-started/using-web-part-as-ms-teams-tab
Using a Microsoft Teams tab as a SharePoint Framework web part
https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/tabs/tabs-in-sharepoint

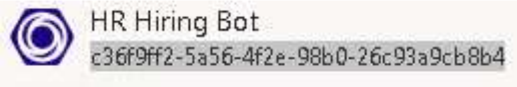
< Table of contents

Exercise 3: Creating a bot in the Contoso HR Talent app

In this exercise, you will build and connect an intelligent bot and then integrate it into your Contoso HR Talent App. It is a simple commands-based bot that will enrich your experience within the app.

Tasks

1. □ **Open the Contoso HR Talent app in the manifest editor**
    1. In Microsoft Teams, in the app bar on the left, select the **More added apps** ellipsis icon and then select **App Studio**.
    2. On the App Studio page, select the **Manifest editor** tab.
    3. Under **Recently created apps**, select **Contoso HR Talent App**.
2. □ **Create a new bot**
    1. In the Contoso HR Talent App, under step **2 Capabilities**, select **Bots**.
    2. Under **Bots**, select **Set up**.
    3. In the Set up a bot window, select the **New bot** tab.
    4. In the **Name** box, enter **HR Hiring Bot**.
    5. Under **Messaging bot** select the **My bot supports uploading and downloading files** check box.
    6. Under **Scope**, select the **Personal** and **Team** check boxes.
    7. Select **Create bot**.
3. □ **Save the bot's app ID**
    1. On the Bots page, under **HR Hiring Bot**, select the **App Id** text located under the bot name.

        HR Hiring Bot
        c36f9ff2-5a56-4f2e-98b0-26c93a9cb8b4

    2. Press Ctrl+C.
    3. In the following text box, paste or type the bot's App Id.
        This will automatically add the bot password to this lab document.

        You may also want to save the bot app Id in Notepad.

        Remove any blank spaces before or after the bot's App Id.
4. □ **Generate a new password**
    1. On the Bots page, under **App passwords**, select **Generate a new password**.
    2. Select the new password and then press Ctrl+C.

   3. In the following text box, paste or type the bot password.
      This will automatically add the bot password to this lab document.

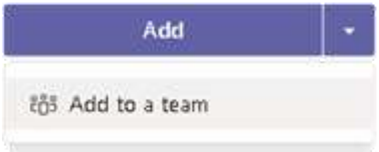
      You may also want to save the password in Notepad.
   4. Select **OK**.

5. □ **Add the Messaging endpoint**
   1. On the Bots page under **Messaging endpoint**, in the **Bot endpoint address** box, enter **https://<appURL>/api/messages**.

6. □ **Add commands to the bot**
   1. On the Bots page under **Commands**, select **Add**.
   2. In the **New command** dialog box, in the **Command text** box, enter **help**.
   3. In the **Help text** box, enter **Find out what I can do**.
   4. Under **Scope**, select the **Personal** and **Team** check boxes.
   5. Select **Save**.
   6. Under **Bot menus**, notice the new command has been added.

7. □ **Edit the appsettings.json file with the new app ID and password information**
   1. Switch to Visual Studio and the appsettings tab.
   2. On the menu bar, select the **Stop Debugging** ⏸ ⏹ ↻ icon or press Shift+F5.
   3. In the appsettings file, locate **MicrosoftAppId** on line 4.
   4. Replace the **PASTE_YOUR_BOT_APPID_HERE** text with **<BotappId>**.
   5. Locate **MicrosoftAppPassword** on line 5.
   6. Replace the **PASTE_YOUR_BOT_APP_PASSWORD_HERE** text with **<BotPassword>**.
   7. On the menu bar, select **Save**.
   8. On the menu, select **Debug** > **Start Debugging** or press F5.
   9. In Microsoft Edge, wait for the localhost page to load and then close the localhost tab.

8. □ **Distribute and test the new bot**
   1. Switch to the Microsoft Teams desktop app.
   2. In App Studio, in the left navigation under step **3 Finish**, select **Test and distribute**.
   3. On the Test and Distribute page, select **Install**.
   4. In the **Contoso HR Talent App** dialog box, review the information.
   5. Select the **Add** menu and then select **Add to a team**.



   6. In the **Select a channel to start using Contoso HR Talent App** dialog box, select the **Search** box and then select **General HR Talent Development**.
   7. Select **Set up a bot**.

If you see a message that reads **Uploading custom apps is not allowed**, the Teams setup policy is still being updated. This is a service related issue and may take up to 24 hours to finish provisioning.

8. On the General channel Conversations tab, at the bottom of the window under **What can I do?**, select **help**.
9. Press Enter.
   It may take a few moments to return the request. If the request is not returned, continue with the lab. The bot may still be provisioning in the service.
10. Review the results of the request.
11. In the **Start a new conversation** box, enter **@Contoso** and then select **Contoso HR Talent App**.
    If the Contoso HR Talent App is not listed in the resolved @ mentions, in the HR Hiring Bot response to **Help**, select **Reply**.

    Alternatively, if the Contoso HR Talent App bot is not resolving:
    - In the app bar, select **Chat**.
    - Select the **New chat** ✎icon.
    - In the **To** box, enter **HR** and then select **HR Hiring Bot**.
    - Under **What can I do?**, select **help**, and then press Enter.

    If the Contoso HR Talent App bot is not responding, continue to the next exercise and return after 5-15 minutes and then try the Contoso HR Talent App bot again.

12. To the right of HR Hiring Bot, add a space using your spacebar, enter **List all your open positions** and then press Enter.
    You may need to press Enter a second time.
13. Review the results.

More information about bots can be found using this Microsoft document
https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/bots/bots-overview

< Table of contents

Exercise 4: Building a personal tab in the Contoso HR Talent app

In this exercise, you will build a Personal tab that will allow you to display rich interactive web content and then integrate that tab into your Contoso HR Talent app. The tab will be scoped to your personal experience and you will be able to interact with it outside the context of a team or channel.

Tasks

1. □ **Update the bot app registration in Azure Active Directory**
   Here you will update the bot's authentication and API permission settings.
   1. Switch to Microsoft Edge and the Microsoft 365 admin center tab.
      If you closed the tab, browse to **https://admin.microsoft.com** and sign into the Microsoft 365 admin center as **[USERNAME]** with password **[PASSWORD]**.
   2. In the left navigation, under **Admin centers**, select **Azure Active Directory**.
      If necessary, select **Show all** to see the Admin centers.
   3. In the Azure Active Directory admin center, in the sidebar on the left, select **Azure Active Directory**.
   4. In the left navigation, under **Manage**, select **App registrations**.
   5. In the App registrations blade, select **HR Hiring Bot**.
   6. In the App registrations blade at the top, using the mouse, point to the **Directory (tenant) ID** and then select the **Copy to clipboard** icon.
   7. In the following text box, paste or type the Directory (tenant) ID.
      This will automatically add the Directory (tenant) ID to this lab document.
      □

      > If Copy to clipboard does not work, try highlighting the ID and then using Ctrl+C. If you are unable to paste the ID in the lab document, open Notepad and create and paste the ID in a new text document. Do this for any copy/paste issue you may experience. Be sure to identify what the ID is used for. There will be multiple IDs used in this exercise.

   8. In the HR Hiring Bot blade, under **Manage**, select **Authentication**.
   9. Under **Redirect URIs**, in the **TYPE** menu, verify that **Web** is selected.
   10. In the **REDIRECT URI** box, enter **https://<appURL>/StaticViews/LoginResult.html**.
   11. Under **Advanced settings** > **Implicit grant**, select the **Access tokens** and **ID tokens** check boxes.
   12. In the menu at the top, select **Save**.
   13. In the side rail on the left, under **Manage**, select **API permissions**.
   14. Under **API Permissions**, select **Add a permission**.
   15. Under **Select an API**, on the Microsoft APIs tab, select **Microsoft Graph**.
   16. Under **Microsoft Graph**, select **Delegated permissions**.
   17. Under **PERMISSION**, select the following check boxes:
       - **email**
       - **openid**
       - **profile**
   18. At the bottom of the page, select **Add permissions**.
2. □ **Open the Contoso HR Talent App in the Manifest editor**

    1. Switch to the Microsoft Teams desktop app.
    2. In the app bar, select the **More added apps** ellipsis icon and then select **App Studio**.
    3. On the App Studio page, select the **Manifest editor** tab.
    4. Under **Recently created apps**, select **Contoso HR Talent App**.
3. □ **Add a personal tab to the Contoso HR Talent App**
    1. In the Contoso HR Talent App, in the left navigation under step **2 Capabilities**, select **Tabs**.
    2. On the Tabs page, under **Add a personal tab**, select **Add**.
    3. In the **Personal tab** dialog box, in the **Name** box, enter **Potential candidates**.
    4. In the **Entity ID** box, enter **OpenPositionsTab**.
    5. In the **Content URL** box, enter **https://<appURL>/StaticViews/OpenPositionsPersonalTab.html?v=1**.
    6. In the **Website URL** box, enter **https://<appURL>/StaticViews/OpenPositionsPersonalTab.html?v=1&web=1**.
    7. Select **Save**.
4. □ **Distribute and test the new personal tab**
    1. In the left navigation, under step **3 Finish**, select **Test and distribute**.

        The tunneling site warning can be safely ignored for this lab.

    2. On the Test and Distribute page, select **Install**.
    3. In the **Contoso HR Talent App** dialog box, select **Add**.
    4. In the Contoso HR Talent App, select the **Potential candidates** tab.
    5. In the Potential Candidates tab, select **Sign In**.
    6. In the **Enter password** dialog box, verify **[USERNAME]** is shown.
    7. In the **Password** box, enter **[PASSWORD]**, and then select **Sign in**.
    8. In **Permissions requested** dialog box, review the information, select the **Consent on behalf of your organization** check box and then select **Accept**.
    9. Review the information in the new personal tab.
        You can return to the personal tab later by selecting the **More added apps** ellipsis icon and then selecting **Contoso HR Talent App**.
5. □ **Uninstall the previous Contoso HR Talent App personal tab**
    1. In the app bar, select **Teams**.
    2. In the app bar, select the **More added apps** ellipsis icon.
    3. Right-click or tap and hold **Contoso HR Talent App** and then, select **Uninstall**.
    4. In the **Uninstall Contoso HR Talent App** dialog box, select **Uninstall**.

More information about static tabs can be found using this Microsoft document
https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/tabs/tabs-static

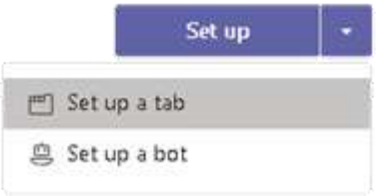
< Table of contents

Exercise 5: Building a team tab in the Contoso HR talent app

In this exercise, you will build a team tab that will allow you to display rich interactive web content. You will then integrate new tab into your Contoso HR Talent App. The tab will be scoped to your team and will provide quick access to team information and resources.

Tasks

1. □ **Open the Contoso HR Talent App in the Manifest editor**
    1. In Microsoft Teams, in the app bar, select the **More added apps** ellipsis icon and then select **App Studio**.
    2. On the App Studio page, select the **Manifest editor** tab.
    3. Under **Recently created apps**, select **Contoso HR Talent App**.
2. □ **Add a Team tab**
    1. In the Contoso HR Talent App, in the left navigation under step **2 Capabilities**, select **Tabs**.
    2. On the Tabs page, under **Team tab**, select **Add**.
    3. In the **Team tab** dialog box, in the **Configuration URL** box, enter **https://<appURL>/StaticViews/TeamTabConfig.html?v=1**.
    4. Verify the **Can update configuration** check box is selected.
       If necessary, select the check box.
    5. Under **Scope**, select the **Team** and **Group chat** check boxes.
    6. Select **Save**.
3. □ **Distribute and test the new Team tab**
    1. In the left navigation, under step **3 Finish**, select **Test and distribute**.
    2. On the Test and Distribute page, select **Install**.
    3. In the **Contoso HR Talent App** dialog box, select the **Add** menu and then select **Add to a team**.
    4. In the **Select a channel to start using Contoso HR Talent App** dialog box, select the **Search** box and then select **General HR Hiring**.
    5. Select the **Set up** menu and then select **Set up a tab**.



6. In the **Contoso HR Talent app** dialog box, in the **Tab name** box, enter **Senior Designer position**.
   You may need to wait for the dialog box to open.
7. Select the **Select job posting** menu and then select **Senior Designer**.
8. Select **Save**.
9. In the Senior Designer position tab, review the newly added content.

More information about configurable tabs can be found using this Microsoft document https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/tabs/tabs-configurable

< Table of contents

Exercise 6: Adding a messaging extension to the Contoso HR Talent app

In this exercise, you will set up a messaging extension and configure it to be part of your Contoso HR Talent app. Messaging extensions provide a new and powerful way for users to engage with your app within Microsoft Teams. At the end of this exercise users will be able to query for additional information from your service and post that information in the form of cards directly into channel conversations.

Tasks

1. □ **Open the Contoso HR Talent app in the manifest editor**
   1. In the Microsoft Teams app, in the app bar, select the **More added apps** ellipsis icon and then select **App Studio**.
   2. In App Studio, select the **Manifest editor** tab.
   3. In Manifest editor, select **Contoso HR Talent App**.
2. □ **Add messaging extensions to the app**
   1. In the Contoso HR Talent App, under step **2 Capabilities**, select **Messaging extensions**.
   2. Under **Messaging extensions**, select **Set up**.
   3. In the Messaging Extension window, select the **Use existing bot** tab.
   4. Under **Bot ID**, select **Select from one of my existing bots**.
   5. Select the **Choose One** menu and then select **HR Hiring Bot**.
   6. In the **Bot name** box, enter **HR Hiring Bot**.
   7. Clear the **Can update configuration** check box and then select **Save**.
3. □ **Verify the messaging endpoint**

    1. Under **Messaging endpoint**, in the **Bot endpoint address**, verify that **https://<appURL>/api/messages** is shown.
       This is the messaging endpoint used when the bot was created earlier.

4. □ **Add a new command to the messaging extension for position searches**
    1. Under **Command**, select **Add**.
    2. In the **New command** dialog box, select **Allow users to query your service for information and insert that into a message**.
    3. In the New command window, the **Command Id** box, enter **searchPositions**.
    4. In the **Title** box, enter **Positions**.
    5. In the **Description** box, enter **Search positions by keyword**.
    6. Verify the **Initial run**, **Command Box**, and **Compose Box** check boxes are selected.
    7. Under **Parameter**, in the **Name** box, enter **searchText**.
    8. In the **Title** box, enter **Keywords**.
    9. In the **Description** box, enter **Search by keywords**.
    10. In the **Set the type of input required** menu, verify it is set to **text**.
    11. Select **Save**.

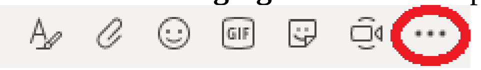
        > If the command fails to save and close, you may need to sign out of Microsoft Teams, sign in again, and then try to create the messaging extension command again.
        > If the creation of the command fails again, browse to **https://teams.microsoft.com**, sign in as **[USERNAME]** with password **[PASSWORD]**. Use the Microsoft Teams web client to create the messaging extension command. When complete, return to the Microsoft Teams desktop client.

    12. On the Messaging Extensions page, scroll down and, under **Command**, notice the new **Positions** command is displayed.

5. □ **Add a new command to the messaging extension for candidate searches**
    1. Under **Command**, select **Add**
    2. In the **New command** dialog box, select **Allow users to query your service for information and insert that into a message**.
    3. In the New command window, in the **Command Id** box, enter **searchCandidates**.
    4. In the **Title** box, enter **Candidates**.
    5. In the **Description** box, enter **Search candidates by name**.
    6. Verify the **Initial run**, **Command Box**, and **Compose Box** check boxes are selected.
    7. Under **Parameter**, in the **Name** box, enter **searchText**.
    8. In the **Title** box, enter **Name**.
    9. In the **Description** box, enter **Search by name**.
    10. In the **Set the type of input required** menu, verify it is set to **text**.
    11. Select **Save**.

6. □ **Add a new command to the messaging extension for new position creation and initiate an action from the composed message area**
    1. Under **Command**, select **Add**
    2. In the **New command** dialog box, select **Allow users to trigger actions in external services while inside of Teams**.
       Be sure you've selected the correct type of command. This command is using a different type than the other commands.
    3. In the **New command** dialog box, select **Fetch a dynamic set of parameters from your bot**.
    4. In the New command window, in the **Command Id** box, enter **openNewPosition**.
    5. In the **Title** box, enter **Create position**.
    6. In the **Description** box, enter **Search candidates by name**.
    7. Clear the **Command box** check box.
    8. Verify the **Initial run** and **Compose Box** check boxes are selected.
    9. Select the **Message** check box.

        > To initiate actions from messages, you'll need to add the "message" context to your messaging extension's command.

    10. Select **Save**.
7. □ **Update the Message Handlers**
    1. On the Messaging Extensions page, scroll down and locate **Message Handlers**.
    2. In the **Enter a valid domain** box, enter ***.ngrok.io**.
    3. Select **Add**.
8. □ **Distribute and test the new messaging extension**
    1. In the left navigation, under step **3 Finish**, select **Test and distribute**.
    2. On the Test and Distribute page, select **Install**.
    3. In the **Contoso HR Talent App** dialog box, select the **Add** menu and then select **Add to a team**.
    4. In the **Select a channel to start using Contoso HR Talent App** dialog box, select the **Search** box and then select **General HR Talent Development**.
    5. Select **Set up**.
    6. Close the **Contoso HR Talent app** dialog box.
       The messaging extension will be installed.
    7. On the General channel Conversations tab, under **Start a new conversation**, select the **Messaging extensions** ellipsis icon.
       

    8. Under **Suggestions**, select **Contoso HR Talent App**.
    9. Select the **Candidates** tab and review the results.
    10. In the **Search by name** box, enter **Deb** and review the results.
    11. Select **Debora Morse**.
    12. Review the information and available options, clear any existing text under the card and then press Enter.
9. □ **Verify message handlers (link unfurling)**

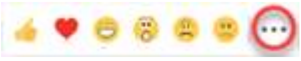
    1. In **General**, select the **Conversations** tab.
    2. In the **Start a new conversation** box, enter **https://<appURL>/StaticViews/OpenPositionsPersonalTab.html?positionId=1** and then, on your keyboard, press Space.

        The additional space/whitespace character is being added to trigger the link. The type text feature may not simulate a pasted link or manually typed text and without the additional space, the link will not be produced.

    3. Review the displayed card. In the top right of the card, select the **Expand preview** ⤢icon.
    4. Review the full card and then press Enter.

10. □ **Create a new position using messaging extension actions**
    1. Under **Start a new conversation**, select the **Messaging extensions** ellipsis icon.
    2. Under **Suggestions**, select **Contoso HR Talent App**.
    3. In the Contoso HR Talent App, select the **Actions** ⊕icon and the select **Create position**.
    4. In the **Contoso HR Talent App**, in the **Title** box, enter **IT Operations manager**.
    5. Select the **Location** menu and then select **Dallas**.
    6. In the **Description** box, enter **Manages the team that maintains the operational integrity of the technologies and services of the organization. The manager's team uses monitoring applications to track performance. When outages occur or are imminent, the team must promptly respond to support the needs of the business users.**
    7. Select **Create posting**.
    8. In the **Contoso HR Talent App** dialog box, review the posting and then select **Confirm posting**.
    9. In the conversation, review the card and then press Enter.
    10. Under **Start a new conversation**, select the **Messaging extensions** ellipsis icon.
    11. Select **Contoso HR Talent App**.
    12. Review the **Positions** tab and notice the new position that has been created.
    13. Press Esc or select an area in the Conversation window to close the app.

11. □ **Test new position creation from within an existing conversation**
    1. In the new IT Operations manager conversation, move your mouse over the conversation and then select the **More options** ellipsis icon.

    2. Select **More actions** > **Create position**.
    3. This is the same Create new job posting card you recently completed, close the card.

More information about messaging extensions can be found using this Microsoft document https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/messaging-extensions/messaging-extensions-overview

< Table of contents

Exercise 7: Creating an Office 365 connector in the Contoso HR Talent app

In this exercise, you will build a new connector to include in your Contoso HR Talent app. You will package and publish your connector within your Contoso HR Talent app. This connector will provide you updates on current status changes of people that are currently part of the hiring process.

Tasks

1. □ **Open the Contoso HR Talent app in the manifest editor**
    1. In the Microsoft Teams desktop app, in the app bar, select the **More added apps** ellipsis icon and then select **App Studio**.
    2. In App Studio, select the **Manifest editor** tab.
    3. In Manifest editor, select **Contoso HR Talent App**.
2. □ **Add a new Connector to the Contoso Talent app**
    1. In the Contoso HR Talent App, under step **2 Capabilities**, select **Connectors**.
    2. On the Your connector page, under **New Connector**, select **Register**.
       You will automatically be switched to Microsoft Edge and a new tab will open to the Connectors Developer Dashboard.
       If necessary, sign in as **[USERNAME]** with password **[PASSWORD]**.

        > If Microsoft Edge opens to the **Connectors for "** page or **Something went wrong** page, perform the following steps:
        > - Close the tab and switch to the Microsoft Teams desktop app.
        > - Close the **Connector** dialog box.
        > - Under **New connector**, select **Register**.

    3. In the Connectors Developer Dashboard window, in the **Connector name** box, enter **Contoso Talent Connector**.
    4. Under **Logo**, select the **Edit** icon.
    5. In the Open window, browse to **C:\LabFiles\HRsolution**.
    6. Select **HRConnector.png** and then select **Open**.
    7. In the **Short description of your app (10 words or less)** box, enter **Talent app for Contoso managers and recruiters**.
    8. In the **Detailed description of what your Connector does (3-5 sentences)** box, enter **This sample recruiting and talent app showcases many of the**

    capabilities that Microsoft Teams supports. App content is hosted for illustrative purposes only.

    9. In the **Company website** box, enter **https://<TenantName>.sharepoint.com/sites/ContosoHR**.
    10. In the **Configuration page for your Connector** box, enter **https://<appURL>/StaticViews/ConnectorConfig.html**.
    11. In the **Valid domains** box, enter **<appURL>**.
        For Connectors, an FQDN must be used. A wildcard is not valid and will cause the installation of the connector to fail.
    12. In the **Do you want to enable actions on your Connector cards** dialog box, select **No**.
    13. Select the **I accept the terms and conditions of the App Developer Agreement** check box and then select **Save**.

> If the Connector fails to be created, in Microsoft Edge, close the tab and return to the Microsoft Teams desktop app. Close the **Connector** dialog box, select **Register**, and then try to create the Connector again.

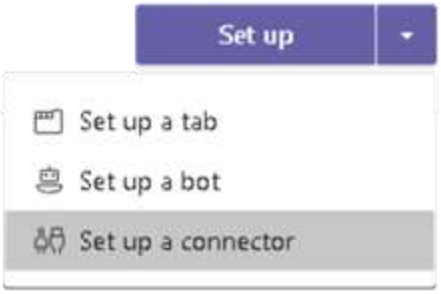
3. □ **Download the existing app manifest**
    1. Under **Test your Connector**, select **Download Manifest**.
    2. In the banner, select the **Save** menu and then select **Save as**.
    3. In the Open window, browse to **C:\LabFiles\HRsolution** and then select **Save**.
4. □ **Record the Connector Id**
    1. In Microsoft Edge, in the banner, select **Open folder**.
    2. In C:\LabFiles\HRsolution, right-click or tap and hold **manifest.json** and then select **Open with Code**.
    3. In Visual Studio Code, locate the connector **id** on line 4.
       This Connector Id is also located on line 23.
    4. Select the id inside the quotation marks and then press Ctrl+C.
    5. In the following text box, paste or type the Connector Id.
       This will automatically add the Connector Id to this lab document.

You may also want to save the Connector Id in Notepad.

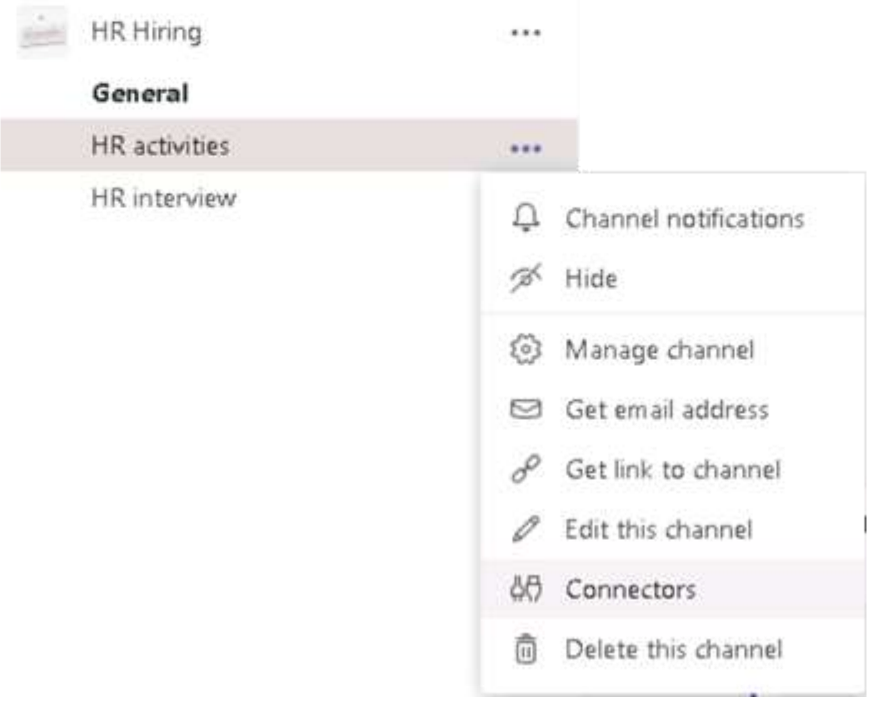
    6. Close Visual Studio Code.

5. □ **Complete the Connector setup in Microsoft Teams**
    1. Switch to the Microsoft Teams desktop app.
    2. In the **Connector** dialog box, in the **Name** box, enter **Contoso Talent Connector**.
    3. In the **Connector ID** box, enter **<ConnectorId>**.
    4. In the **Configuration URL** box, enter **https://<appURL>/StaticViews/ConnectorConfig.html**.
    5. Select **Save**.
6. □ **Distribute and test the new connector**

1. In the left navigation, under step **3 Finish**, select **Test and distribute**.
2. On the Test and Distribute page, select **Install**.
3. In the **Contoso HR Talent App** dialog box, select the **Add** menu and then select **Add to a team**.
4. In the **Select a channel to start using Contoso HR Talent App** dialog box, in the **Search**, enter **HR activities** and then select **HR activities HR Hiring**.
5. Select the **Set up** menu and then select **Set up a connector**.



If the Connectors dialog box does not open, under the **HR Hiring** team, to the right of **HR Activities**, select the **More options** ellipsis icon and then select **Connectors**.



6. In the **Connectors for "HR activities" channel in "HR Hiring" team** dialog box, in the **Search** box, enter **Contoso**.
7. In the results, locate **Contoso Talent Connector** and then select **Configure**.
8. In the **Contoso HR Talent App** dialog box, review the information and then select **Save**.
9. Close the **Connectors for "HR activities" channel in "HR Hiring" team** dialog box.

If the **Connectors for "HR activities" channel in "HR Hiring" team** dialog box does not close, you will need to quit and restart Microsoft Teams:
- In the System tray, right-click or tap and hold the **Microsoft Teams** icon and then select **Quit**. Wait for Microsoft Teams to close.
- On the Desktop, double-click or double-tap **Microsoft Teams**.

10. In the HR Hiring team, select the **HR Activities** channel.
11. In the HR activities channel, select the **Contoso Hiring Board** tab.
12. Under **Senior Software Engineer**, locate **Lynda Trevino**.
13. To the right of **Lynda Trevino**, in the **Current Stage** column, select **Interviewing** and then select **Offer**.
    Offer will update to **Offered** in the Current Stage column.
14. In the HR activities channel, select the **Conversations** tab.
15. Notice the Contoso Talent Connector has updated the conversation with a change of status message about Lynda Trevino.

More information about connectors can be found using this Microsoft document
https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/connectors/connectors

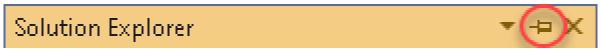
< Table of contents

Exercise 8: Exploring task module invocation

In this exercise, you will explore task module invocation through tabs, bots, messaging extensions and deep linking. This task module will allow you to create a popup experience in your Contoso HR Talent App.

Tasks

1. □ **Open and review the CandidateFeedback.html file in Visual Studio**
   1. Switch to Microsoft Visual Studio.
   2. On the menu bar, select **View** and then select **Solution Explorer**.
   3. In Solution Explorer, on the top right, select the **Auto Hide** pin icon.

   4. In Solution Explorer, expand **StaticViews** and then select **CandidateFeedback.html**.
   5. Review the html, style sheet, and JavaScript used in the file.
2. □ **Review how modules can be invoked outside of Teams (deep linking)**

   1. In CandidateFeedback.html, scroll down to line 189.
   2. Review the code located between lines 189 to 203.
      This code demonstrates how to create a task module deep link that can be shared and used outside of Teams using a serialization of the TaskInfo object.

3. □ **Review how channel tabs and personal tabs can invoke a module**
   1. In Solution Explorer, under **StaticViews**, select **TeamTab.html**.
   2. Scroll down to line 268.
   3. Review the code located between lines 268 to 273.
      This code shows how to invoke a task module from a tab using microsoftTeams.tasks.startTask() from the tab and passing a TaskInfo object.

4. □ **Invoke a task module from JavaScript**
   1. Switch to the Microsoft Teams desktop app.
   2. Select the **HR Hiring** > **General** channel.
   3. Select the **Add a tab** + icon.
   4. In the **Add a tab** dialog box, select **Contoso HR Talent App**.
   5. In the Contoso HR Talent App window, in the **Tab name** box, enter **Task module invoked from JavaScript**.
   6. Select the **Select job posting** menu and then select **Full Stack Developer**.
   7. Select **Save**.
   8. Under **Applicants**, select **Scotty Cothran**.
   9. In the Contoso HR Talent App window, to the right of **Feedback for Scotty Cothran**, select the **Copy to Clipboard** ⧉ icon.
   10. Close the Contoso HR Talent App window.
   11. Select the **Conversations** tab.
   12. Right-click or tap and hold in the **Start a new conversation** box, select **Paste**, and then select Enter.
   13. In the conversation, select the link you just posted.
   14. Review the task module that was open via deep linking and then close the window.

5. □ **Review how bots can invoke a module**
   1. Switch to Visual Studio.
   2. In Solution Explorer, expand **Services** > **Templates** and then select **CandidateToTemplate.cs**.
   3. Scroll down to line 177.
   4. Review the code located between lines 177 to 189.
      This example shows how you can use the Action.OpenUrl card action for Adaptive cards to invoke task module

6. □ **Review how messaging extensions can invoke a module**
   1. In Solution Explorer, expand **Services** and then select **InvokeActivityHandler.cs**.
   2. Scroll down to line 134.
   3. Review the code located between lines 134 to 167.
      This shows an example of the HTTP response to the invoke message. There's a TaskInfo object embedded in a wrapper object that Teams uses to display the task module.

More information about task modules can be found using this Microsoft document https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/task-modules/task-modules-overview

< Table of contents

Exercise 9: Reviewing activity feed alert configurations

In this exercise, you will configure activity feed notifications leveraging your existing integration with the Bot Framework APIs.

Tasks

1. □ **Review activity feed alert configurations**
    1. In Visual Studio, in Solution Explorer, expand **Services** and then select **BotService.cs**.
    2. Scroll down to line 61.
    3. Review the code located between lines 61 to 101.
       This code demonstrates how to capture a conversation reference and then using it later to initialize outbound (proactive) messages.
    4. In Solution Explorer, under **Services**, select **NotificationService.cs**.
    5. Scroll down to line 186.
    6. Review the code located between lines 186 to 229.
       This example shows how to construct a message and send the notification to a user.
2. □ **Test activity feed updates**
    1. Switch to Microsoft Teams.
    2. To the right of **HR Talent Development** select the **More options** ellipsis icon and then select **Manage team**.
    3. On the Members tab, verify **Mindy Cooper** is listed and is a team owner.
    4. Select the **Apps** tab.
    5. To the right of **Contoso HR Talent App (Custom app)** select the **Uninstall** icon.
    6. In the **Uninstall Contoso HR Talent App (Custom app)** dialog box, review the information and then select **Uninstall**.
       The app is being uninstalled and reinstalled…
    7. In Microsoft Teams, in the app bar, select the **More added apps** ellipsis icon and then select **App Studio**.
    8. On the App Studio page, select the **Manifest editor** tab.
    9. Under **Recently created apps**, select **Contoso HR Talent App**.
    10. In the left navigation, under step **3 Finish**, select **Test and distribute**.
    11. On the Test and Distribute page, select **Install**.

    12. In the **Contoso HR Talent App** dialog box, select the **Add** menu and then select **Add to a team**.
    13. In the **Select a channel to start using Contoso HR Talent App** dialog box, select the **Search** box and then select **General HR Talent Development**.
    14. Select the **Set up** menu and then select **Set up a tab**.
    15. Close the Window.

        A new tab does not need to be created. The purpose was to install the app in the channel.

3. □ **Sign into Teams web app as Mindy Cooper and review the activity feed**
    1. On the taskbar, right-click or tap and hold Microsoft Edge and then select **New InPrivate window**.
    2. In the InPrivate window, browse to **https://teams.microsoft.com**.
    3. In the Sign in window, enter **MindyC@[TENANTNAME]**.
    4. Select **Next**.
    5. In the **Password** box, enter **labpass@word1**.
    6. Select **Sign in**.
    7. In the Microsoft Teams window, select **Use the web app instead**.
    8. In the Microsoft Teams web app, complete or close the introduction wizard and close any other notification windows.
    9. In the app bar, select **Activity** and review the existing activity notifications.
    10. In the app bar, select **Teams**.

        Leave Microsoft Teams web app open.

4. □ **Test the activity feed update**
    1. Switch to the Microsoft Teams desktop app.
    2. In the HR Talent Development General channel, select the **Conversations** tab.
    3. In the **Start a new conversation** box, enter **@** and then select **Get bots**.
    4. In the **Add a bot** dialog box, select **Contoso HR Talent App**.
    5. To the right of **Contoso HR Talent App**, add a space, enter **new job posting**, and then press Enter.
    6. In the new job posting card, in the **Title** box, enter **Senior QA**.
    7. Select the **Hiring manager** menu, select **Mindy Cooper**, and then select **Create posting**.
    8. Switch to the Microsoft Edge InPrivate window and the Microsoft Teams web app.
    9. In the app bar, select **Activity** and review the new activity notifications.
    10. When complete, close the InPrivate browser window.

        More information about activity feeds can be found using this Microsoft document https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/activity-feed

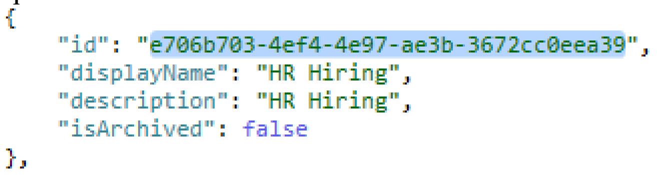
< Table of contents

Exercise 10: Using Graph Explorer

In this exercise, you will create a new HR Recruitment channel and a chat thread in your existing team via Graph Explorer. Then you will review how to use a bot that uses Graph APIs for Teams to dynamically create a new team.

Tasks

1. □ **Sign into Graph Explorer**
    1. Switch to Microsoft Edge and browse to **https://developer.microsoft.com/en-us/graph/graph-explorer**.
       You can use and existing tab or open new tab.
    2. In the left navigation, under **Authentication**, select **Sign in with Microsoft**.
    3. In the **Pick an account** dialog box, select **[USERNAME]**.
    4. If prompted for a password, enter **[PASSWORD]**.
    5. In the **Permissions requested** dialog box, review the information, select the **Consent on behalf of your organization** check box, and then select **Accept**.
2. □ **Modify permissions for read and write to users and groups**
   The queries and changes made in this exercise require the addition of certain permissions for Graph Explorer.
    1. In the left navigation, under **Authentication**, select **modify permissions**.
    2. In the **Modify Permissions** dialog box, select the check box for each of the following permissions:
        - **Group.Read.All**
        - **Group.ReadWrite.All**
        - **User.Read.All**
        - **User.ReadWrite.All**
    3. Review the information and then select **Modify Permissions**.
       You will be signed out.
    4. In the **Pick an account** dialog box, select **[USERNAME]**.
    5. If necessary, enter **[PASSWORD]** and then select **Sign in**.
    6. In the **Permissions requested** dialog box, review the information, select the **Consent on behalf of your organization** check box, and then select **Accept**.
3. □ **Add sample queries to the console**
    1. In the left navigation, under **Sample Queries**, select **show more samples**.
    2. In the Sample Categories pane, to the right of **Microsoft Teams** and **Microsoft Teams (beta)** select the toggle switches and verify they are set to **On**.
    3. Close the Sample Categories pane.
4. □ **Retrieve an existing Microsoft Teams Id**
    1. In the left navigation, under **Microsoft Teams**, select **GET my joined teams**.
    2. Locate the **HR Hiring** Team.
    3. Above the displayName, to the right of **id**, select the unique id inside the quotation marks.

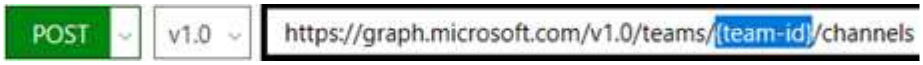


```
{
    "id": "e706b703-4ef4-4e97-ae3b-3672cc0eea39",
    "displayName": "HR Hiring",
    "description": "HR Hiring",
    "isArchived": false
},
```

4. Right-click or tap and hold the id and then select **Copy**.
   Do not select the quotation marks.
5. In the following text box, paste or type the Team id.
   This will automatically add the HR Hiring Team id to this lab document.

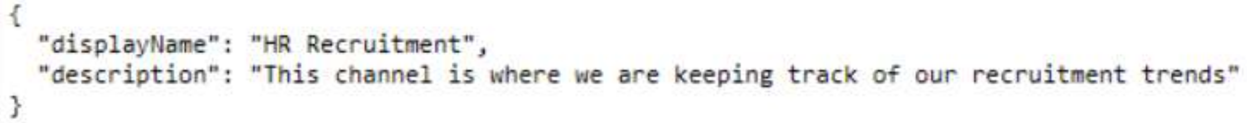
   You may also want to save the Team id in Notepad.

5. □ **Create a new channel in an existing Team**
   1. In the left navigation, under **Microsoft Teams**, select **POST create channel**.
   2. At the top of the page, under the **Query** box, select **Request Headers**.
   3. Verify that the **Key** box contains **Content-type** and that the **Value** box contains **application/json**.
      If there are no values in the boxes, refresh the page and then in the left navigation, under **Microsoft Teams**, select **POST create channel**.
   4. At the top of the page, in the **Query** box, locate **{team-id}**.

      POST | v1.0 | https://graph.microsoft.com/v1.0/teams/{team-id}/channels

   5. Replace **{team-id}** with **<HRHiringTeamId>**.
   6. Under the query, select **Request Body**.
   7. On the Request Body tab, locate **displayName**.
   8. Replace the existing display name with **HR Recruitment**.
   9. Replace the existing description with **This channel is where we are keeping track of our recruitment trends**.
   10. When complete, the request body content should look like this:

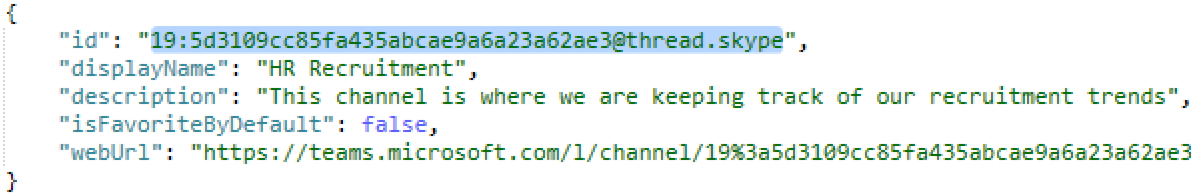
       ```
       {
         "displayName": "HR Recruitment",
         "description": "This channel is where we are keeping track of our recruitment trends"
       }
       ```

   11. Select **Run Query**.
   12. Verify the banner displayed shows **Success** and then, under **Response Preview**, review the output of the query.
6. □ **Create a chat thread using Graph Explorer**
   1. At the top of the page, to the left of the **Query** box, select the **v1.0** menu and then select **beta**.
   2. In the left navigation, under **Microsoft Teams (beta)**, select **POST create chat thread**.
   3. In the middle of the page, review the information banner.
   4. At the top of the page, to the left of the **Query** box, select the **POST** menu and then select **GET**.
   5. In the **Query** box, enter:

      ```
      https://graph.microsoft.com/beta/teams/<HRHiringTeamId>/channels
      ```

   6. Select **Run Query**.

7. Locate the **HR Recruitment** channel.
8. Above the displayName, to the right of **id**, select the unique id inside the quotation marks.



```
{
    "id": "19:5d3109cc85fa435abcae9a6a23a62ae3@thread.skype",
    "displayName": "HR Recruitment",
    "description": "This channel is where we are keeping track of our recruitment trends",
    "isFavoriteByDefault": false,
    "webUrl": "https://teams.microsoft.com/l/channel/19%3a5d3109cc85fa435abcae9a6a23a62ae3
}
```

9. Right-click or tap and hold the id and then select **Copy**.
   Do not select the quotation marks.
10. In the following text box, paste or type the channel id.
    This will automatically add the HR Recruitment id to this lab document.

    You may also want to save the Team id in Notepad.

11. At the top of the page, to the left of the **Query** box, select the **GET** menu and then select **POST**.
12. In the **Query** box, enter `https://graph.microsoft.com/beta/teams/<HRHiringTeamId>/channels/<HRRecruitChannelId>/chatThreads`
13. Select **Run Query**.
14. In the banner, verify **Success** is displayed and then, under **Response Preview**, review the response.

7. □ **Review the new channel and chat thread created using Microsoft Graph**
   1. Switch to the Microsoft Teams desktop app.
   2. In the app bar, select **Teams**.
   3. Under **HR Hiring**, select **1 hidden channel** > **HR Recruitment**.
   4. In the Conversations tab, notice the **Hello world** chat.

8. □ **Review Graph API services configurations**
   1. Switch to Visual Studio.
   2. In Solution Explorer, expand **Services** and then select **GraphApiService.cs**.
   3. Scroll down to line 103.
   4. Review the code located between lines 103 to 155.
      This code demonstrates how to implement a bot that uses the Microsoft Graph APIs for Teams to dynamically create a new team, channels, add members, and configure a new channel tab.
   5. Locate line 157.
      This creates channels in the team.
   6. Scroll down to line 169.
   7. Review the code located between lines 169-176.
      This code installs an app to the team.
   8. Locate lines 178.
   9. Review the code located between lines 178-193.
      This adds and configures a tab to the team's recently created channel.

< Table of contents

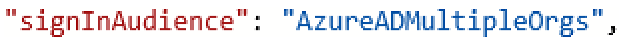
Exercise 11: Creating proactive messages using the Graph API

In this exercise, you will install bot via Graph Explorer for all hiring managers. Then you will review how welcome message can be sent to users.

This exercise requires a Microsoft Azure subscription. A Microsoft Azure trial or trial promotion code may be provided. If a code is not provided, you must sign up for a Microsoft Azure trial subscription. Use the instructions in **Appendix A** to sign up for a trial.

Tasks

1. □ **Update the bot app registration in Azure Active Directory**
    1. Switch to Microsoft Edge and the HR Hiring Bot -API Permissions tab.
       If you closed or browsed away from the HR Hiring Bot page:
        - Browse to **https://aad.portal.azure.com**.
        - If necessary, sign into the Azure Active Directory admin center admin center as **[USERNAME]** with password **[PASSWORD]**.
        - In the Azure Active Directory admin center, in the sidebar on the left, select **Azure Active Directory**.
        - In the Overview blade, under **Manage**, select **App registrations**.
        - In the App registrations blade, select **HR Hiring Bot**.
    2. In the HR Hiring Bot blade, under **Manage**, select **Manifest**.
    3. In the manifest editor, scroll down and locate **signInAudience**.
    4. In the quotations, change the *AzureADandPersonalMicrosoftAccount* value to **AzureADMultipleOrgs**.
       This change will allow the Chat.Read permission to be added later in this task.

       ```
       "signInAudience": "AzureADMultipleOrgs",
       ```

    5. Verify the change is correct and then select **Save**.
    6. In the HR Hiring Bot blade, under **Manage**, select **Authentication**.
    7. Under **Redirect URIs**, in the second row, the **TYPE** column, verify that **Web** is listed.
    8. In second row, in the **REDIRECT URI** box, enter **https://token.botframework.com/.auth/web/redirect**.
    9. In the top menu, select **Save**.
    10. In the HR Hiring Bot blade, under **Manage**, select **API permissions**.
    11. Under **API Permissions**, select **Add a permission**.
    12. Under **Select an API**, on the Microsoft APIs tab, select **Microsoft Graph**.

13. Under **Microsoft Graph**, select **Delegated permissions**.
14. Under **PERMISSION**, select the following check boxes:
    - **AppCatalog** > **AppCatalog.ReadWrite.All**
    - **Chat** > **Chat.Read**
    - **Directory** >
        - **Directory.AccessAsUser.All**
        - **Directory.Read.All**
        - **Directory.ReadWrite.All**
    - **Group** > **Group.ReadWrite.All**
    - **User** >
        - **User.Read**
        - **User.ReadWrite.All**
15. At the bottom of the page, select **Add permissions**.
16. On the API permissions page, verify that **Microsoft Graph** has 11 permissions assigned.
    There will be three permissions assigned from a previous exercise and the eight you assigned in the previous task.
17. Under **Grant consent**, select **Grant admin consent for Contoso**.
18. At the top of the page, review the **Do you want to grant consent for the requested permissions for all accounts in Contoso?** notification and then select **Yes**.

2. □ **Register the bot service with Microsoft Azure**
    1. In Microsoft Edge, browse to **https://dev.botframework.com**.
    2. In the top right, select **Sign in**.
    3. At the top of the page, select **My bots**.
    4. In the **Terms of service** dialog box, select the **I agree that my use of the Bot Framework service is subject to the same terms under which I use the Azure Bot Service through Azure.** check box, and then select **Save**.
    5. In the **My bots** list, to the right of **HR Hiring Bot**, select **Migrate**.
    6. In the **Migrate** dialog box, verify the correct Azure Subscription and location are selected. Change these settings if necessary.
    7. Select **Migrate**.
    8. In the **Terms of Service** dialog box, review and select the two check boxes and then select **Agree**.
       Registration of the bot service in Azure may take 1-2 minutes.
    9. In the **Migration completed** dialog box, select **Open the new bot**.
3. □ **Update the bot settings**
    1. In the Bot Channels Registration blade, under **Bot management**, select **Settings**.
    2. At the bottom of the page, under **OAuth Connection Settings**, select **Add Setting**.
    3. In the New Connection Settings blade, in the **Name** box, enter **<TenantName>-AADv2OAuth**.
    4. Select the **Service Provider** menu and then select **Azure Active Directory v2**.
    5. In the **Client id** box, enter **<BotappId>**.
       This is the Bot application ID.
    6. In the **Client secret**, enter **<BotPassword>**.

    7. In the **Tenant ID** box, enter **<TenantID>**.
    8. In the **Scope** box, enter **AppCatalog.ReadWrite.All Chat.Read Directory.AccessAsUser.All Directory.Read.All Directory.ReadWrite.All Group.ReadWrite.All User.Read User.ReadWrite email openid profile**
    9. Select **Save**.

4. □ **Update the appsettings.json file in Visual Studio**
    1. Switch to Microsoft Visual Studio.
    2. On the menu bar, select the **Stop Debugging** ⏸ ⏹ ↻ icon or press Shift+F5.
    3. In the appsettings file, locate **OAuthConnectionName**.
       If necessary, in the Solution Explorer, under **src** > **TeamsTalentMgmtAppV4**, select **appsettings.json**.
    4. In the appsettings file, locate **OAuthConnectionName**.
    5. Replace the current value of **PASTE_YOUR_CONNECTION_NAME** with **<TenantName>-AADv2OAuth**.
    6. On the menu bar, select the **Save** icon or press Ctrl+S.
    7. On the menu, select **Debug** > **Start Debugging** or press F5.
    8. In Microsoft Edge, wait for the localhost page to load and then close the localhost tab.

5. □ **Update the apps domains and permissions**
    1. Switch to the Microsoft Teams desktop app.
    2. In the app bar, select the **More added apps** ellipsis icon and then select **App Studio**.
    3. On the App Studio page, select the **Manifest editor** tab.
    4. Under **Recently created apps**, select **Contoso HR Talent App**.
    5. In the left, under **3 Finish**, select **Domains and permissions**.
    6. On the Valid domains page, in the **Enter a valid domain** box, enter **token.botframework.com** and select **Add**.

6. □ **Install the updated HR Hiring Bot for the team and organization**
    1. In the left navigation, under step **3 Finish**, select **Test and distribute**.

       The 'validDomains' warning displayed in the description can safely be ignored for this lab.

    2. Under **Download**, select **Download**.
       Wait for the download to complete.
    3. On the Test and Distribute page, select **Install**.
    4. In the **Contoso HR Talent App** dialog box, select the **Add** menu and then select **Add to a team**.
    5. In the **Select a channel to start using Contoso HR Talent App** dialog box, select the **Search** box and then select **General HR Talent Development**.
    6. Select the **Set up** menu and then select **Set up a bot**.
    7. In the app bar, select the **More added apps** ellipsis icon and then select **More apps**.
    8. In the **Apps** category list, select **Upload a custom app** and then select **Upload for Contoso**.

9. In the Open window, under **Quick Access**, select **Downloads**.
10. Select **ContosoHRTalentApp.zip** and then select **Open**.
    Wait for the install to complete.
11. In the **Apps** category list, select **Built for Contoso**.
    You may only see **Contoso** until the Apps are refreshed.
12. Verify the **Contoso HR Talent App** is listed.

7. □ **Install the bot for hiring managers**
    1. In the app bar, select **Chat**.
    2. Select the **New chat** icon.
    3. In the **To** box, enter **HR Hiring Bot**.
    4. In the **Type a new message box**, enter **install bot** and then press Enter.
    5. In the **Please sign in to proceed** OAuth card, select **Sign in**.
    6. If prompted, sign in as **[USERNAME]** with password **[PASSWORD]**.
    7. In the **Permissions requested** dialog box, select the **Consent on behalf of your organization** check box and then select **Accept**.
    8. Review the response and in the **Type a new message box**, enter **install bot** and then press Enter.
       Wait for the bot to install.
    9. Review the response and notice that the hiring bot has been installed for Mindy Cooper and Henry Wickham.

       You may receive the message "Bot wasn't installed to any hiring manager". This may happen if the Contoso HR Hiring App has not been installed for the organization or if the Chat.Read permission was unable to be installed. You will test the installation of the app in the next step.

8. □ **Verify the HR Hiring Bot has been installed for Mindy Cooper**
    1. In Microsoft Teams, in the top right, select the user icon and then select **Sign out**.
    2. In the Microsoft Teams window, in the **Sign-in address** box, enter **MindyC@<TenantName>.onmicrosoft.com** and then select **Sign in**.
    3. In the **Password** box, enter **labpass@word1** and then select **Sign in**.
    4. Close the introduction wizard and any other notifications.
    5. In the app bar, select the **More added apps** ellipsis icon.
    6. Select **Contoso HR Talent App**.
    7. In the **Type your questions here** box, enter **open positions** and then press Enter.
    8. Review the response from the HR Hiring Bot.

       If the HR Hiring bot is not responding:
       - In the app bar, select **Chat**.
       - Select the **New chat** icon.
       - In the **To** box, enter **HR Hiring Bot**.
       - Under **What can I do?**, select **open positions**, and then press Enter.
       - Review the response from the HR Hiring Bot.
       If the HR Hiring Bot is not responding, you may need to wait 5-15 minutes and then try the HR Hiring bot again.

9. If you are continuing to the next lab, sign out of all applications and close all open windows.

More information about the tasks in this exercise can be found using these Microsoft documents:
Use the Microsoft Graph API to work with Microsoft Teams
https://docs.microsoft.com/en-us/graph/api/resources/teams-api-overview?view=graph-rest-beta
Microsoft Graph blog: Announcing "30 Days of Microsoft Graph" blog series
https://developer.microsoft.com/en-us/graph/blogs/announcing-30-days-of-microsoft-graph-blog-series/#

< Table of contents

Conclusion: Building an HR hiring app solution for Microsoft Teams

After completing this lab, you should have acquired solid foundational skills for developing responsive apps. Beginning with the creation of an app manifest file, you learned to extend your Microsoft Teams app with personal/team tabs, bots, messaging extensions, adaptive cards, connectors, and SharePoint web parts. In addition, you now have hands-on experience interacting with Graph Explorer and using Graph APIs to provision a team, channels, and tabs. We hope you will be able to use your newly acquired skills to build end-to-end business solutions for your customers by taking advantage of all the new Teams extensibility platform capabilities, and to reuse your existing investments in SharePoint.

< Table of contents

Lab: Building a low code IT helpdesk app for Microsoft Teams

During this lab, you will gain hands-on experience using the Microsoft Teams developer platform. You will build a desktop solution using Microsoft Teams, QnA Maker, the Microsoft Bot Framework and PowerApps. Additionally, you will use Microsoft Flow and app templates.

Estimated time to complete: ***1 hours 30 minutes***

Before you begin

To complete the exercises in this lab, you will be provided Microsoft Office 365 tenant with an Enterprise E5 trial subscription. You will also need a Microsoft Azure subscription. A Microsoft Azure trial or trial promotion code may be provided. If a code is not provided, you must sign up for a Microsoft Azure trial subscription. Instructions for signing up for a trial are provided in the lab.

The type text T and copy/paste ⎘ features used in this lab will send the specified text string to the active window in the virtual machine. Always compare the text in the lab document with the typed text in the virtual machine and verify the expected text is displayed.

What you will learn

After completing the exercises in this lab, you will be able to:

- Create a team and add apps
- Create and deploy a bot
- Use QnA Maker to setup a sample knowledge base
- Deploy a bot to a team and train the bot
- Create a sample IT ticketing app using PowerApps
- Deploy an app to Teams
- Create and deploy an Azure AD application
- Create an adaptive card
- Create a flow in PowerApps

Scenario

You are responsible for developing and deploying apps in your company's Microsoft Teams organization. The help desk has requested an app or apps that can help them better assist its customers. You will start by creating a team and channel for interactions with the help desk. You will add a bot that will can answer common questions in the help desk's knowledge base. You will then add a custom tab to a team channel that will help with password resets and an app to allow people to create a support case. Finally, you will automate the delivery of support case update messages sent to the employee.

< Table of contents

Exercise 1: Setting up your lab environment

In this exercise, you will configure app setup policies, create an Azure resource group and add QnA Maker as a resource. You will also setup the PowerApps environment.

Tasks

1. □ **Run the UpdateTime.exe script**
   Occasionally, the hosted virtual machine's clock does not update as expected. Time synchronization is being performed to ensure the virtual machine's time is set correctly.
   1. On YOUR PC, this is the virtual machine you are currently using, sign in as **Lab User** with password **labpass@word1**.
   2. Open File Explorer and then browse to **C:\Scripts**.
   3. Double-click or double-tap **UpdateTime.exe**.
   4. In the **User Account Control** dialog box, select **Yes**.
   5. A console window will open and force the virtual machine to synchronize its time.
      Wait for the process to complete in the console window.
2. □ **Create a replacement token for the Office 365 domain name**
   When filled out, this token's text will automatically update in the lab document.
   1. In the following FQDN, select the first name, right-click or tap and hold, and then select **Copy**:

      [TENANTNAME]

      For example, if the FQDN is LODjx321.onmicrosoft.com, you will copy ***LODjx321***
   2. In the following text box, paste or type the name.
      This will automatically add the tenant name to this lab document.
      □
3. □ **Sign into the Microsoft 365 admin center**
   1. Open Microsoft Edge and then browse to **https://admin.microsoft.com**.

      In this lab, references to Microsoft Edge are specific to the browser used inside the virtual machine. If the browser on your computer is required to perform a task, the lab will specify that you use the browser on the host computer.

   2. In the Sign in window, enter **[USERNAME]**.
   3. Select **Next**.
   4. In the **Password** box, enter **[PASSWORD]**.
   5. Select **Sign in**.

      > Although saving credentials is not required for the lab, it may help if your browser or session times out.

   6. In the Microsoft 365 admin center, on the right, verify that **The new admin center** switch is enabled.
      If this option is not enabled, you may need to select the **Try the new admin center** switch.

      > If there is no option for the new admin center or preview of the new admin center, continue with the lab. The new admin center may be fully deployed and the old admin center no longer available.

4. □ **Enable uploading custom apps**
   This was previously referred to as sideloading of external apps.
   1. In the left navigation, select **Show all**.
   2. Under **Admin centers**, select **Teams**.
   3. In the Microsoft Teams admin center, in the left navigation, select **Teams apps** > **Setup policies**.
   4. On the App setup policies page, select **Global (Org-wide default)**.
   5. Next to **Upload custom apps**, select the toggle switch and verify it is set to **On**.

      If you have completed the *Building an HR hiring app solution for Microsoft Teams* lab, this setting will already be enabled.

   6. At the bottom of the page, select **Save**.
   7. In Microsoft Edge, close the Microsoft Teams admin center tab.
5. □ **Create a resource group in Microsoft Azure**
   1. In the browser, open a new tab and then browse to **https://portal.azure.com**.
   2. In the Microsoft Azure portal, in the sidebar on the left, select **Resource groups**.
   3. On the Resource groups blade, select **+ Add**.

      If you are prompted to sign up for a free trial, follow the prompts to create a free trial.
      If needed, step-by-step instruction can be found in **Appendix A**. When complete, return to this task.

   4. Under **Project details**, verify a subscription is shown.
      This subscription should be your Free Trial, Azure Pass, MSDN subscription, etc.
   5. In the **Resource group** box, enter **TeamsITAppGroup**.
   6. If necessary, change the **Region** to a location closest to you.
   7. Select **Review + create**.
   8. Verify the **Validation passed** banner is displayed and then, at the bottom of the blade, select **Create**.
   9. In the Resource group blade, on the menu, select **Refresh** and verify the new TeamsITAppGroup is displayed.
6. □ **Create a new resource for QnA Maker**
   1. In the Microsoft Azure portal, above the **FAVORITES** list on the left, select **+ Create a resource**.
   2. In the New blade, in the **Search the Marketplace** box, enter **QnA Maker** and then press Enter.
   3. Under **QnA Maker**, select **Create**.
   4. In the Create blade, in the **Name** box, enter **TeamsITAppQnA-<TenantName>**.
      This name must be unique across Microsoft Azure.
   5. Select the **Pricing tier** menu and then select **F0**.
      For this lab, a greater pricing tier will not be necessary.
   6. Select the **Resource group** menu and then select **TeamsITAppGroup**.

7. Select the **Azure Search pricing tier** menu and then select **F (3 Indexes)**.
   For this lab, a greater search pricing tier will not be necessary.
8. If necessary, select the **Website location** and change it to a location closest to you.
9. Under App insights, verify **Enable** is selected and then select **Create**.

   > You may be warned that Microsoft Search is not enabled. This can happen if you have just created your Microsoft Azure trial subscription. Select **Create** again and the resource should be deployed.

7. □ **Publish the IT Spend Analysis Sample PBIX using Microsoft Power BI**
   1. Select **Start** and then select **Microsoft Power BI Desktop**.
   2. On the Welcome to Power BI Desktop page, select **Already have a Power BI account? Sign in**.
      You may need to scroll down to see the sign in link.
   3. In the **Sign in** dialog box, enter **[USERNAME]** and then select **Sign in**.
   4. In the **Password** box, enter **[PASSWORD]** and then select **Sign in**.

      > The tenant administrator account has already been assigned an Office 365 E5 license. The license includes Power BI Pro.

   5. In the welcome window, on the left, select **Open other reports**.
   6. In the Open window, browse to **C:\LabFiles\ITapp**, select **IT Spend Analysis Sample PBIX.pbix**, and then select **Open**.
   7. Close any additional open windows.

      > The pbix file was downloaded from this Microsoft document.

   8. On the ribbon bar, select **Publish**.
   9. In the **Publish to Power BI** dialog box, select **My workspace** and then select **Select**.
   10. Wait for publishing to complete and then close the **Publishing to Power BI** dialog box.
   11. Close the Power BI Desktop app.

8. □ **Create a new PowerApps environment for the IT help desk app**
   1. Switch to Microsoft Edge and then browse to **https://admin.powerapps.com**.
      You should be signed in automatically. If necessary, sign in as **[USERNAME]** with password **[PASSWORD]**.
   2. In the top right, select **+ New environment**.
   3. In the New environment window, in the **Environment name** box, enter **TeamsITapp**.
   4. If necessary, select the **Region** menu and choose a region that you want the environment to serve. Your database and resources will be tied to that region.
   5. Select the **Environment type** menu and then select **Production**.
   6. Select **Create environment**.

7. In the **You've created an environment** dialog box, review the information and then select **Skip**.
   A database does not need to be created for this lab. Later, an Excel spreadsheet will be uploaded and used as the data source.
8. In Microsoft Edge, close the PowerApps Admin center tab.
   Leave the Microsoft 365 admin center open.

< Table of contents

Exercise 2: Using Microsoft Teams to create a team and installing the Power BI app

In this exercise, you will create the team and team channels used by the help desk. You will also add the Power BI app to your Teams desktop client and review the sample data you uploaded using the Power BI Desktop app.

Tasks

1. □ **Sign into Microsoft Teams and create a new team named Help Desk**
    1. In YOUR PC, minimize all open windows.
    2. On the desktop, double-click or double-tap **Microsoft Teams**.
    3. In the Microsoft Teams window, in the **Sign-in address** box, enter **[USERNAME]** and then select **Sign in**.
    4. In the **Password** box, enter **[PASSWORD]** and then select **Sign in**.
    5. Close the introduction wizard and any other notifications.
    6. In the app bar on the left, select **Teams**.
    7. At the bottom of the Teams list, select **Join or create a team**.
    8. On the Join or create a team page, select **Create team**.
    9. In the **Create your team** dialog box, select **Build a team from scratch**.
    10. On the What kind of team will this be page, select **Org-wide**.
    11. On the Some quick details about your private team page, in the **Team name** box, enter **Help Desk**.
    12. Select **Create**.
2. □ **Edit the Help Desk team's member permissions**
    1. In the channel list, under **Help Desk**, to the right of **General**, select the ellipsis and then select **Manage channel**.
    2. On the Channel settings tab, under **Permissions**, select **Only owners can post messages**.
3. □ **Edit the Help Desk team's member permissions**
    1. In the channel list, to the right of **Help Desk**, select the ellipsis and then select **Manage team**.
    2. On the Help Desk page, select the **Settings** tab.

    3. Select **Member permissions**.
    4. Clear all the selected check boxes.

4. □ **Add a channel named Announcements to the team**
    1. Select the **Channels** tab.
    2. Select **Add channel**.
    3. In the Create a channel for "Help Desk" team window, in the **Channel name** box, enter **Announcements**.
    4. Select the **Automatically show this channel in everyone’s channel list** check box and then select **Add**.

5. □ **Add the Power BI app to the Help Desk General channel**
    1. In the teams list, under **Help Desk**, select **General**.
    2. Select the **Add a tab** + icon.
    3. In the **Add a tab** dialog box, select **Power BI**.
    4. In the **Power BI** dialog box, under **Select Power BI reports to show in this tab**, select **IT Spend Analysis Sample PBIX** and then select **Save**.
    5. On the Power BI tab, at the bottom of the report, select the **IT Spend Trend** tab and verify that information is shown.

< Table of contents

Exercise 3: Creating and deploying a question and answer bot

In this exercise, you will set up a knowledge base using QnA Maker and create and deploy a bot to your help desk team using App Studio. You will then test and train the bot. You will also add alternate phrasing to the knowledge base to help train the QnA Maker model.

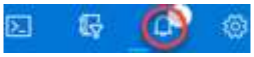
Tasks

1. □ **Sign into Microsoft Teams and create a new team named Help Desk**
    1. Switch to Microsoft Edge and then browse to **https://www.qnamaker.ai**. You should be signed in automatically. If necessary, sign in as **[USERNAME]** with password **[PASSWORD]**.
    2. At the top of the QnA Maker page, select **Create a knowledge base**.
    3. On the Create a knowledge base page, next to **Step 2**, select the **Microsoft Azure Directory ID** menu and then select your ID.
    4. Select the **Azure subscription name** menu and then select your subscription.
    5. Select the **Azure QnA service** menu and then select **TeamsITAppQnA-<TenantName>**.
    6. Next to **Step 3** in the **Name** box, enter **IT Help Desk FAQ**.
    7. Next to **Step 4**, under **File name**, select **+ Add file**.

8. In the Open window, browse to **C:\Labfiles\ITapp**, select **IT Helpdesk Knowledge Base.xlsx** and then select **Open**.
9. Under **Chit-chat**, review the description and available options.
   Select **Caring** or select another option.
10. Next to **Step 5**, select **Create your KB**.
    Wait for the knowledge base to be created.
11. On the Knowledge base page, scroll down and review the newly added content created from the Excel spreadsheet.
12. In the top right menu, select **Save and train**.
13. After save and train is complete, in the top right menu, select **PUBLISH**
14. On the IT Help Desk FAQ page, review the information and then select **Publish**.

2. □ **Create a bot**
   1. On the Success! Your service has been deployed. What's next? page, select **Create Bot**.
      A new tab will open to the Microsoft Azure portal.
   2. In the **Web App Bot** blade, in the **Bot handle** box, delete the existing handle and then enter **ITHelpDeskBot-<TenantName>**.
   3. Verify the automatically added subscription, resource group, and location.
      If necessary, change any of the settings to match your deployment.
   4. Select the **Pricing tier** menu and then select **F0**.
   5. Review the remaining settings and, at the bottom of the blade, select **Create**.
      Wait for the bot to be created. It may take 1-2 minutes to complete.

      > If you receive a warning stating that "Resource provider 'Microsoft.Storage' not registered for the subscription", wait for 5-10 seconds and then select **Create** again. This may happen with newly provisioned Azure subscriptions.

3. □ **Deploy the bot**
   1. If not already displayed, on the menu, select the **Notification** icon and then select **Go to resource**.
      
   2. In the **ITHelpDeskBot-<TenantName>** blade, under **Bot management**, select **Channels**.
   3. On the Connect to channels page, under **Add a featured channel**, select the **Configure Microsoft Teams channel** icon.
   4. On the Configure MSTeams page, select **Save**.
   5. In the **Terms of Service** dialog box, review the information, select the **I agree to the Microsoft Channel Publication Terms and the Microsoft Privacy Statements for my deployment to the Microsoft Teams channel.** check box and then select **Agree**.
   6. On the left, under **Bot management**, select **Settings**.
   7. On the Bot profile page, under **Icon**, select the folder icon.
   8. In the Open window, browse to **C:\LabFiles\ITapp**, select **BotIcon32.png** and then select **Open**.

    9. In the **Display name** box, change the name to **IT Help Desk Bot**.
    10. In the top menu, select **Save**.

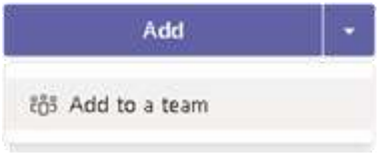
4. □ **Install App Studio in the Microsoft Teams desktop client**

    If you are continuing this lab from a previous lab and App Studio is already installed, you can skip to the next task.

    1. Switch to Microsoft Teams signed in as your tenant administrator.
    2. In the Teams app bar on the left, select **More added apps** ellipsis icon and then select **More apps**.
    3. On the Apps page, in the **Search all** box, enter **App Studio**.
    4. In the results, select **App Studio**.
    5. In the **App Studio** dialog box, review the information and then select **Add**.
    6. Close the **App Studio is now available to you** dialog box.

5. □ **Import the help desk manifest and generate a new App ID**
    1. In the Teams app bar on the left, select **More added apps** ellipsis icon and then select **App Studio**.
    2. In App Studio, select the **Manifest editor** tab.
    3. Under **Welcome**, select **Import an existing app**.
    4. In the Open window, browse to **C:\LabFiles\ITapp**, select **HelpDeskBotManifest.zip** and then select **Open**.
    5. Under **Recently create apps**, select **Help Desk Bot**.
    6. On the App details page, under **Identification**, select **Generate**.
       This will generate a new App ID for your deployment.

6. □ **Install the Help Desk bot**
    1. On the left under **Complete these steps**, under **Capabilities**, select **Bots**.
    2. Under **Bots**, to the right of **Imported Bot**, select **Delete**.
    3. In the **Delete "Help Desk Bot"** dialog box, select the **I understand that everything will be deleted** check box and then select **Delete**.
       This will remove the imported bot and enable you to set up the new bot for your lab.
    4. Under **Bots**, select **Set up**.
    5. In the **Set up a bot** dialog box, select the **Existing bot** tab.
    6. In the **Name** box, enter **Help Desk Bot**.
    7. Under **Bot ID**, select **Select from one of my existing bots**.
    8. Select the **Choose One** menu and then select **IT Help Desk Bot**.
    9. Under **Scope**, select the **Personal** and **Team** check boxes.
    10. Select **Save**
    11. On the left under **Complete these steps**, under step **3 Finish**, select **Test and distribute**.
    12. On the Test and Distribute page, under **Install**, select **Install**.
    13. In the **Help Desk Bot** dialog box, review the information.
    14. Select **Add** menu and then select **Add to a team**.



15. In the **Select a channel to start using Contoso HR Talent App** dialog box and then select **General Help Desk**.
16. Select **Set up a bot**.

    For this lab, you will install the bot to a channel. In a real world scenario doing so would probably not be the best choice because there would be too many conversations in the channel. This type of bot would work best in 1:1 chat sessions.

7. □ **Test the Bot**
    1. In the app bar, select **Teams**.
    2. Under **Help Desk**, select **General**.
    3. On Conversations tab in the **Start a new conversation** box, enter **@** and then select **Get bots**.
    4. In the **Add a bot** dialog box, select **Help Desk Bot**.
    5. On your keyboard, press the spacebar, enter **Hello** and then press Enter.
    6. Review the response.
       The response you receive will depend on the chit-chat dataset you chose.
    7. In the response, select **Reply**, enter **How are you feeling?** and then press Enter.
    8. Review the response.
       The response you receive will depend on the chit-chat dataset you chose.
    9. In the **Start a new conversation** box, enter **@Help** and then select **Help Desk Bot**.
    10. On your keyboard, press the spacebar, enter **Why can't I see anything?** and then press Enter.
    11. Review the response.
        In this case, the bot does not provide a response to a common question you've received at the help desk regarding a person's monitor.
8. □ **Train the model**
    1. Switch to Microsoft Edge and the QnA Maker tab.
       If you have closed this tab or browsed away from it, browse to **https://www.qnamaker.ai** and sign in using your tenant administrator credentials.
    2. In the top menu, select **My knowledge bases**.
    3. In the **My knowledge bases** list, select **IT Help Desk FAQ**.
    4. On the Knowledge base page, in the **Search the knowledge base** box, enter **monitor** and then press Enter.
    5. In the results, in the **Question** column, locate **What do I do if display on the monitor is black?**.
    6. Under the question, select **+ Add alternative phrasing**.
    7. In the **Alternative phrasing** box, enter **Why can't I see anything?**.

8. In the top right menu, select **Save and train**.
9. After the knowledge base is saved, select **PUBLISH**.
10. On the IT Help Desk FAQ page, select **Publish**.

9. □ **Test the new model**
    1. Switch to Microsoft Teams and the Help Desk Bot chat.
    2. In the **Start a new conversation** box, enter **@Help** and then select **Help Desk Bot**.
    3. On your keyboard, press the spacebar, enter **Why can't I see anything?** and then press Enter.
    4. Review the response and notice the updated model now provides an expected result.

< Table of contents

Exercise 4: Creating and deploying a help desk ticket app using PowerApps

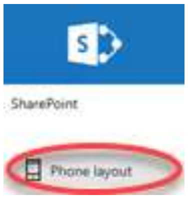
In this exercise, you will upload and verify your sample help desk ticket data to the Help Desk team's SharePoint Online site. You will then create your app using Power Apps and connect the app to your ticket data. After you've set up your app, configured data fields, and tested it, you will deploy your app to Microsoft Teams.

Tasks

1. □ **Upload a list to the Help Desk document library in SharePoint Online**
    The script used in this task will create a list named SampleTicketData in SharePoint Online and then upload data to the list.
    1. On YOUR PC, open File Explorer and then browse to **C:\Scripts**.
    2. Double-click or double-tap **ITAppLabSPList.exe**.
    3. In the **User Account Control** dialog box, select **Yes** and then wait for the application to open.
    4. In the **Office 365 Admin username** box, enter **[USERNAME]**.
    5. In the **Office 365 Admin password** box, enter **[PASSWORD]** and then select **Verify Credentials**.
    6. Ensure the credentials were verified and then select **Run Script**.
        The script will create a new SharePoint Online list and populate it with sample data.
    7. In the Wadeware Lab Provisioning Tool, review the information in the output text box and then select **Finish**.
    8. Close File Explorer.
2. □ **Review the sample ticket data**

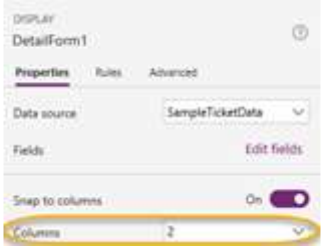
1. Switch to Microsoft Edge and then browse to **https://<TenantName>.sharepoint.com/sites/HelpDesk/Lists/SampleTicketData**.
   You should be signed in automatically. If necessary, sign in as **[USERNAME]** with password **[PASSWORD]**.
2. Close any open welcome messages or notifications.
3. Review the list of sample support tickets and their associated fields. You will be using this data for the rest of the exercises to build a low code solution for the IT help desk.

3. □ **Choose a template for your app**
   1. In Microsoft Edge, browse to **https://make.powerapps.com**.
   2. In the top right, select the **Environment** menu and then select **TeamsITapp**.
   3. On the Build business apps, fast page, under **Make your own app**, select **All templates**.
   4. On the Make apps like these page, select the **Start from data** tile.
   5. In the **Start from data** dialog box, select **Create**.
4. □ **Connect your app**
   1. If prompted, in the dialog box, review the country/region setting and then select **Get started**.
   2. On the Create an app in TeamsITapp page, on the **SharePoint** tile, select **Phone layout**.

   3. Under **How do you want to connect to your data?**, verify **Connect directly (cloud services)** is selected and then select **Create**.
   4. On the Connect to a SharePoint site page, in the **Enter the SharePoint URL for the location of your list** box, enter **https://<TenantName>.sharepoint.com/sites/HelpDesk/Lists/SampleTicketData**.
   5. Select **Go**.
   6. Under **Choose a list**, select **SampleTicketData**.
   7. In the lower right corner, select **Connect**.
      Wait for the app to be built.
   8. If prompted, in the **Welcome to PowerApps Studio** dialog box, select **Skip**.
5. □ **Configure the app layout**
   1. In the top left menu, select **File** and then select **App settings**.
   2. In the **App settings** list, select **Screen size + orientation**.
   3. On the Screen size + orientation page, under **Orientation**, select **Landscape**.
   4. Under **Advanced settings**, under **Lock aspect ratio**, select the **On** switch and verify it is now set to **Off**.

5. In the lower right, select **Apply**.
6. In the top left, select the **Back** arrow icon.
7. On the left, under **Tree view**, expand **DetailScreen1** and then select **DetailForm1**.
8. On the right, in the DISPLAY pane, change the value of **Columns** to **2**.



DISPLAY

DetailForm1

Data source: SampleTicketData

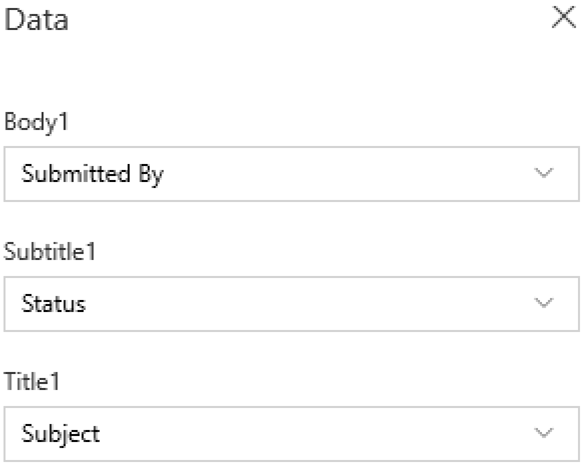
Fields: Edit fields

Snap to columns: On

Columns: 2

9. Under **Tree view**, expand **EditScreen1** and then select **EditForm1**.
10. On the right, in the EDIT pane, change the value of **Columns** to **2**.

6. □ **Configure data fields**
   1. In the Tree view, under **BrowseScreen1**, select **BrowseGallery1**.
   2. On the right, in the GALLERY pane, next to **Fields**, select **Edit**.
   3. In the Data pane, select the **Body1** menu and then select **Submitted By**.
      You may need to wait and select the menu again if the data fields take longer than expected to load.
   4. Select the **Subtitle1** menu and then select **Status**.
   5. Select the **Title1** menu and then select **Subject**.
      This may already be set.
   6. When complete, your Data settings should look like the following:



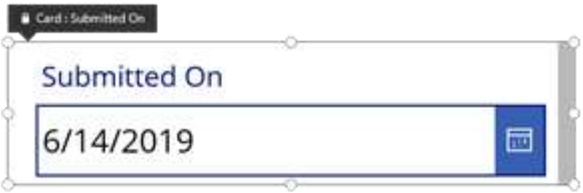
Data

Body1: Submitted By

Subtitle1: Status

Title1: Subject

7. □ **Set default field values**
   1. On the left in the Tree view under **EditScreen1**, select **EditForm1**.
   2. In the form preview, select **Submitted On**.



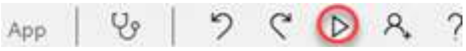
Card : Submitted On

Submitted On

6/14/2019

   3. On the right, in the CARD pane, select the **Advanced** tab.
   4. Select **Unlock to change properties**.
   5. Under **DATA**, select **More options**.
   6. In the **Default** box, delete the existing content and then enter **Today()**
   7. In the form preview, select **Submitted By**.
   8. On the right, on the **Advanced** tab, select **Unlock to change properties**.
   9. Under **DATA**, in the **Default** box, delete the existing content and then enter **User().FullName**

8. □ **Test the new app**
   1. In the Tree view, select **BrowseScreen1**.
   2. In the upper right menu, under the signed in user icon, select **Preview the app**.

   3. In the **Search items** box, enter **Problem**.
   4. Review the discovered ticket and then select **Problem with Keyboard**.
   5. Review the **Submitted By** and **Submitted On** fields.
   6. In the top right, select the **Edit item** icon.
   7. Review the **Submitted By** and **Submitted On** fields and notice they have been updated to reflect the default value changes made earlier.
   8. In the top right, select the **Submit item** check mark icon.
   9. Review the update and then, in the top right, select the **Close preview** icon. Alternatively, you can press Esc on your keyboard.

9. □ **Save your app**
   1. On the top left menu, select **File** and then select **Save**.
   2. In the Save as window, verify that **The cloud** is selected.
   3. In the **App** box, enter **Help Tickets**.
   4. In the lower right, select **Save**.

10. □ **Add PowerApps as a tab in the Help Desk General channel**
    1. Switch to the Microsoft Teams desktop client signed in as your tenant administrator.
    2. In the app bar, select **Teams**.
    3. In the Teams list, select **Help Desk** > **General**.
    4. Select the **Add a tab** + icon.
    5. In the Add a tab window, in the **Search** box, enter **PowerApps**.
    6. In the results, select **PowerApps**.
    7. In the **PowerApps** dialog box, select **Add**.
    8. In the list of apps, select **Help Tickets** and then select **Save**.
    9. Review the added app.

< Table of contents

Exercise 5: Configuring and deploying a list search app

In this exercise, you will register two Azure AD applications in your tenant's directory: one for the list search messaging extension and another for the configuration app. You will configure your list search app to use the sample ticket data list you created earlier. Finally, you will deploy a list search app in Microsoft Teams and verify the app is connecting to your list search app and successfully retrieving sample ticket data.

Tasks

1. □ **Register the List Search app Microsoft Azure AD**
    1. Switch to Microsoft Edge and then browse to **https://portal.azure.com**.
       You should be signed in automatically. If necessary, sign in as **[USERNAME]** with password **[PASSWORD]**.
    2. In the Microsoft Azure portal, in the sidebar on the left, select **Azure Active Directory**.
    3. In the Azure Active Directory blade, under **Manage**, select **App registrations**.
    4. On the menu, select **New registration**.
    5. On the Register an application blade, in the **Name** box, enter **listsearchapp**.
    6. Under **Supported account types**, verify that **Accounts in this organization directory only** is selected.
       If necessary, change the selection.
    7. At the bottom of the blade, select **Register**.
       The Redirect URI will not be used.
    8. In the listsearchapp blade, using the mouse, point to the **Application (client) ID** and then select the **Copy to clipboard** icon.
    9. In the following text box, paste or type the Application (client) ID.
       This will automatically add the Application (client) ID to this lab document.
       □
    10. In the listsearchapp blade, using the mouse, point to the **Directory (tenant) ID** and then select the **Copy to clipboard** icon.
    11. In the following text box, paste or type the Directory (tenant) ID.
        This will automatically add the Directory (tenant) ID to this lab document.
        □
    12. In the side rail on the left, under **Manage**, select **Certificate & secrets**.
    13. Under **Client secrets**, select **New client secret**.
    14. In the Add a client secret window, in the **Description** box, enter **app secret**.
    15. Under **Expires**, select **Never**.
    16. Select **Add**.

17. Under **Client secrets**, to the right of **app secret**, point to the secret under the **VALUE** column and then select the **Copy to clipboard** icon.
18. In the following text box, paste or type the app secret.
    This will automatically add the app secret value to this lab document.

    > You must copy the app secret now. This value will not be visible later and will need to be regenerated if you do not copy and save the value.

19. In the side rail on the left, under **Manage**, select **API permissions**.
20. Under **API Permissions**, select **Add a permission**.
21. Under **Select an API**, on the Microsoft APIs tab, select **Microsoft Graph**.
22. Under **Microsoft Graph**, select **Delegated permissions**.
23. Under **PERMISSION**, select the **offline access** check box.
24. Scroll down and then expand **Sites**.
25. Select the **Sites.Read.All** check box and then, at the bottom of the page, select **Add permissions**.
    If continuing from the previous lab, you may see **Update permissions** instead.
26. Under **Grant consent**, select **Grant admin consent for** *your organization*.
27. In the **Do you want to grant consent for the requested permissions for all accounts** notification, select **Yes**.

2. □ **Register the list search configuration app in Microsoft Azure AD**
    1. In the Microsoft Azure portal, in the sidebar on the left, select **Azure Active Directory**.
    2. In the Azure Active Directory blade, under **Manage**, select **App registrations**.
    3. On the menu, select **New registration**.
    4. On the Register an application blade, in the **Name** box, enter **listsearchconfiguration**.
    5. Under **Supported account types**, verify that **Accounts in this organization directory only** is selected.
       If necessary, change the selection.
    6. At the bottom of the blade, select **Register**.
       The Redirect URI will not be used.
    7. In the listsearchconfiguration blade, using the mouse, point to the **Application (client) ID** and then select the **Copy to clipboard** icon.
    8. In the following text box, paste or type the Application (client) ID.
       This will automatically add the configuration app's Application (client) ID to this lab document.

3. □ **Create a custom deployment in Microsoft Azure for the list search app**
    1. In Microsoft Edge, browse to **https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FOfficeDev%2Fmicrosoft-teams-list-search-app%2Fmaster%2FDeployment%2Fazuredeploy.json**

    This will create a custom deployment in Microsoft Azure based on the ARM template identified in the URL.

2. In the Custom deployment blade, under **BASICS**, to the right of **Resource group**, select **Create new**.
3. In the **Name** box, enter **listsearchresgroup** and then select **OK**.
4. Select the **Location** menu and set the location to **(US) Central US**.
   The resource group location must be in a datacenter that supports Application Insights, Azure Search, QnA Maker, and Logic Apps.
5. Under **SETTINGS**, in the **Base Resource Name** box, enter **<TenantName>-listsearch**
6. In the **Messaging Extension Client Id** box, enter **<ListSearchAppAppID>**
7. In the **Messaging Extension Client Secret** box, enter **<ListSearchAppSecret>**
8. In the **Config App Client Id** box, enter **<ListSearchConfigAppID>**
9. In the **Config Admin UPN List** box, enter **[USERNAME]**
10. In the **Tenant Id** box, enter **<ListSearchAppDirID>**
11. Scroll down and, under **TERMS AND CONDITIONS**, review the terms and conditions and then select the **I agree to the terms and conditions stated above** check box.
12. Select **Purchase**.
13. Wait for the deployment to finish. Do not continue until this process has completed. In some cases, the deployment may take up to 10 minutes. If this is a new subscription, provisioning may take 20-30 minutes.
    You can follow the deployment progress by selecting the **Notification** icon in the top menu.

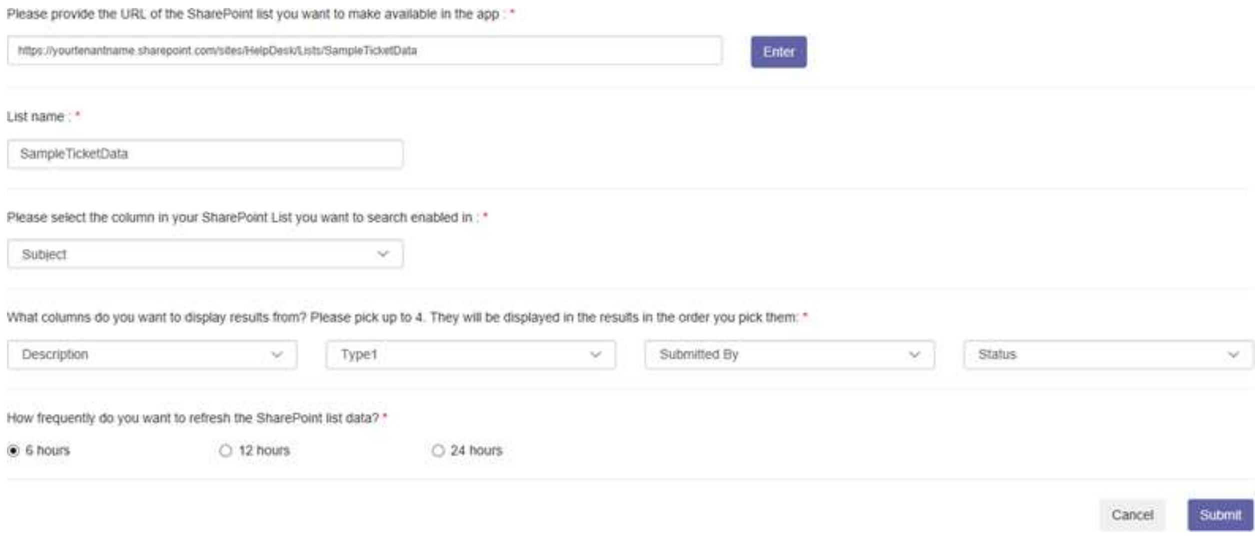
4. □ **Complete the configuration app set up**
   1. In the Microsoft Azure portal, in the sidebar, select **Resource groups**.
   2. In the Resource groups blade, select **listsearchresgroup**.
   3. In the listsearchresgroup blade, in resources list, select **<TenantName>-listsearch-config** that has a resource **TYPE** of **App Service**.
      There are several resources named <TenantName>-listsearch-config. Be sure to select the <TenantName>-listsearch-config resource that is type *App Service*.
   4. In the <TenantName>-listsearch-config blade, in the overview information at the top, point to the **URL** and then select the **Copy to clipboard** icon.
   5. In the following text box, paste or type the config URL.
      This will automatically add the config URL to this lab document.
      ▭
   6. In the Microsoft Azure portal, in the sidebar, select **Azure Active Directory**.
   7. In the Azure Active Directory blade, under **Manage**, select **App registrations**.
   8. On the Owned applications tab, select **listsearchconfiguration**.
   9. Under **Manage**, select **Authentication**.
   10. Under **Redirect URIs**, in the **TYPE** column, verify that **Web** is listed.
   11. In the **REDIRECT URI** box, enter **<ListSearchConfigUrl>**
   12. Under **Implicit grant**, select **ID Tokens**.
   13. In the top menu, select **Save**.
   14. In the Microsoft Azure portal, in the sidebar, select **Azure Active Directory**.
   15. In the Azure Active Directory blade, under **Manage**, select **App registrations**.

16. On the Owned applications tab, select **listsearchapp**.
17. Under **Manage**, select **Authentication**.
18. Under **Redirect URIs**, in the **TYPE** column, verify that **Web** is listed.
19. In the **REDIRECT URI** box, enter **<ListSearchConfigUrl>**.
20. Under **Implicit grant**, select **ID Tokens**.
21. In the top menu, select **Save**.

5. □ **Add the SampleTicketData SharePoint Online list to the list search configuration app**
    1. In Microsoft Edge, browse to **<ListSearchConfigUrl>**
       This is the listsearch-config URL you saved earlier.
    2. Sign in as **[USERNAME]** with password **[PASSWORD]**.
    3. In the **Permissions requested** dialog box, select the **Consent on behalf of your organization** check box and then select **Accept**.
    4. On the Home page, select **Set SharePoint user**.
       The List Search app is not able to access SharePoint until it gets an access token.
    5. Sign in as **[USERNAME]** with password **[PASSWORD]**.

       If you are unable to successfully sign in, verify that you have set the redirect URI correctly for both applications. Both applications use the same redirect URI.

    6. Select **Add a new SharePoint list**.
    7. In the **Please provide the URL of the SharePoint list you want to make available in the app** box, enter **https://<TenantName>.sharepoint.com/sites/HelpDesk/Lists/SampleTicketData**
    8. Select **Enter**.
    9. Verify the **List name** is **SampleTicketData**.
    10. Select the **Please select the column in your SharePoint List you want to search enabled in:** menu and then select **Subject**.
    11. In each of the **What columns do you want to display results from** menus, select **Description**, **Type1**, **Submitted By**, and **Status**.



Please provide the URL of the SharePoint list you want to make available in the app : *

https://yourtenantname.sharepoint.com/sites/HelpDesk/Lists/SampleTicketData

List name : *

SampleTicketData

Please select the column in your SharePoint List you want to search enabled in : *

Subject

What columns do you want to display results from? Please pick up to 4. They will be displayed in the results in the order you pick them: *

Description | Type1 | Submitted By | Status

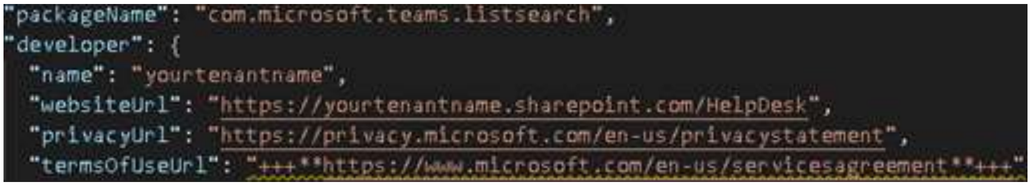
How frequently do you want to refresh the SharePoint list data? *

● 6 hours ○ 12 hours ○ 24 hours

12. Leave the refresh frequency set to the default value and then select **Submit**.
13. Do not browse away from this page. Wait for the process to complete. The application will fetch the list data and index it, which can take several minutes, especially if the list contains a large amount of data.

6. ☐ **Deploy the teams app package**

1. Open File Explorer and then browse to **C:\LabFiles\ITapp\ListSearchManifest**.
2. Double-click or double-tap **manifest.json**.
3. If prompted, in the **How do you want to open this file** dialog box, select **Visual Studio Code** and then select **OK**.
4. Under **developer**, replace the values for **name**, **websiteUrl**, **privacyUrl**, and **termsofUseUrl** with the following values:
   "name": "**<TenantName>**"
   "websiteUrl": "**https://<TenantName>.sharepoint.com/HelpDesk**"
   "privacyUrl": "**https://privacy.microsoft.com/en-us/privacystatement**"
   "termsOfUseUrl": "**https://www.microsoft.com/en-us/servicesagreement**"
5. When complete, your updates should be similar to the following:



```
"packageName": "com.microsoft.teams.listsearch",
"developer": {
  "name": "yourtenantname",
  "websiteUrl": "https://yourtenantname.sharepoint.com/HelpDesk",
  "privacyUrl": "https://privacy.microsoft.com/en-us/privacystatement",
  "termsOfUseUrl": "+++**https://www.microsoft.com/en-us/servicesagreement**+++"
```

6. Under **composeExtensions**, locate **botId**.
7. In **botId** replace **< botid >** with **<ListSearchAppAppID>**.
   The is the listsearchapp Application (client) ID.
8. Under **validDomains** replace **< app domain >** with **<TenantName>-listsearch.azurewebsites.net**.
   This is the list search app's domain.
9. On the menu, select **File** > **Save**.
10. Close Visual Studio Code.

    11. Right-click or tap and hold **Start** and then select **Microsoft PowerShell (Admin)**. For this lab, PowerShell is being used to create the compressed (zip) file. Alternative methods, such as the GUI or third-party apps can be used as well.
    12. In the **User Account Control** dialog box, select **Yes**.
    13. In Windows PowerShell, enter the following and then press Enter:

        ```
        Compress-Archive -Path C:\LabFiles\ITapp\ListSearchManifest\* -
        DestinationPath C:\LabFiles\ITapp\ListSearchManifest.zip
        ```

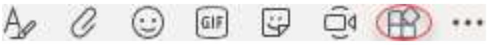
    14. Wait for the command to complete and then close Windows PowerShell.

7. □ **Upload the custom app into Microsoft Teams**
    1. Switch to Microsoft Teams.
    2. To the right of **Help Desk**, select the **More options** ellipsis icon and then select **Manage Team**.
    3. Select the **Apps** tab.
    4. In the lower right, select **Upload a custom app**.
    5. In the Open window, browse to **C:\LabFiles\ITapp**, select **ListSearchManifest.zip** and then select **Open**.

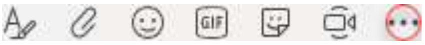
        If the app fails to install, delete the ListSearchManifest.zip file, open the manifest.json file again and verify you did not include extra characters or mistakes. There is a copy of the ListSearchManifest folder in C:\Scripts\Backup if needed. Be sure to copy the manifest.json file from the backup location to the LabFiles location to preserve the backup.

    6. In the **Apps** list, verify that **List Search (Custom app)** is shown.

8. □ **Use the list search app in Microsoft Teams**
    1. In the **Help Desk** team, select the **General** channel.
    2. At the bottom of the Conversations tab, select the **List Search** app icon.

        If the List search icon is not pinned to the app tray, select the **Messaging extensions** ellipsis icon and then select **List Search**.

        You may need to wait ~1 minute for the app to load.

    3. In the List Search app, in the **Search by Subject** box, enter **keyboard**.
    4. Review the results of the search and select **Problem with Keyboard**. This will surface item information as an adaptive card in the conversation.
    5. Select **Share**.
    6. In the message, enter **@Help** and then select **Help Desk**. If the adaptive card is not displayed, delete or send the existing message and then

open the app and try sharing again.
If *Help Desk* is not shown, continue with the lab and return later to try again.
7. On your keyboard, press the spacebar, enter **Thanks for this!** and press Enter.

< Table of contents

Exercise 6: Creating a flow for the help desk ticketing app using PowerApps

In this exercise, you will add a new flow to the help tickets app you created in PowerApps. This flow will create a post in the Help Desk Announcements channel.

Tasks

1. □ **Create an adaptive card in Microsoft Teams**
    1. In the Teams app bar on the left, select **More added apps** ellipsis icon and then select **App Studio**.
    2. Select the **Card editor** tab.
    3. Under **Welcome**, select **Create a new card**.
    4. In the **Create a new card** dialog box, under **Adaptive Card**, read the information and then select **Create**.
    5. To the left of the code, under **Manifest** in the **Name** box, enter **New Ticket Card**.
    6. On the json tab, select all the existing code and then delete it.
       You can highlight all the code using Ctrl+A or manually highlight it and then press Delete.
       This json code will be replaced in the next steps.
    7. Open File Explorer and then browse to **C:\LabFiles\ITapp**.
    8. Double-click or double-tap **NewTicketCard.txt**.
    9. In Notepad, select all the text and then press Ctrl+C.
       Alternatively, you can select all the text and then use the right-click or Edit menu options to copy the text to the clipboard.
    10. Switch to Microsoft Teams and App Studio.
    11. On the json tab, paste the text you copied to the clipboard.
        You can use Ctrl+V or right-click in the code window and then select **Paste**.
        Your changes will be saved automatically.

        If you are unable to see the existing json tab contents, resize the Teams app to full screen and try again.

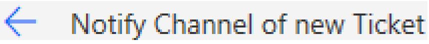
2. □ **Add a new flow to the Help Tickets app created in PowerApps**
    1. Switch to Microsoft Edge and then browse to **https://make.powerapps.com**.
       If you already have the PowerApps browser tab open, you can also switch to that and then, in the left navigation, select **Home**.

   You should be signed in automatically. If necessary, sign in as **[USERNAME]** with password **[PASSWORD]**.

2. In PowerApps, in the left navigation, select **Apps**.
   This app was created earlier in Exercise 4: Creating and deploying a help desk ticket app using PowerApps.
3. In the Apps in TeamsITapp page, to the right of **Help Tickets**, select the **More Commands** ellipsis icon and then select **Edit**.
4. On the left in the Tree view, select **EditScreen1**.
5. In the form, select the **Submit** ✓button.
6. On the PowerApps page, in the top menu, select **Action** and then select **Flows**.
7. In the Data pane, select **Create a new flow**.

3. □ **Configure the new flow**

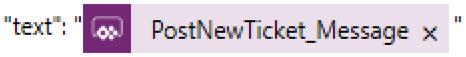
   1. On the Flow page, in the top, verify you are signed in. If not, select **Sign in**.
      You should be signed in automatically. If necessary, sign in as **[USERNAME]** with password **[PASSWORD]**.
   2. If prompted, in the Welcome to Microsoft Flow window, select **Get started**.
   3. In the top right, select the tenant administrator name or user icon and then select **TeamsITapp**.
      This is the environment you created earlier.
   4. In the **Templates** list, select **PowerApps button**.
   5. In the top left, select the text **PowerApps button**.
      This is editable text.
   6. Replace the text with **Notify Channel of new Ticket**.

      
      Notify Channel of new Ticket

   7. On the PowerApps page, select **+ New step**.
   8. Under **Choose an action**, select **Microsoft Teams**.
   9. In the **Actions** list, locate and then select **Post your own adaptive card as the Flow bot to a channel**.
      You may see "preview" in the action name.
   10. In the action, to the right of **Post your own adaptive card as the Flow bot to a channel**, select the **Menu** ellipsis and then select **Rename**.
   11. In the **Name** box, delete the existing name and then enter **PostNewTicket**.
   12. Select the **Team** menu and then select **Help Desk**.
   13. Select the **Channel** menu and then select **Announcements**.
   14. Switch to the NewTicketCard.txt file open in Notepad.
       If you have closed Notepad, browse to **C:\LabFiles\ITapp** and open **NewTicketCard.txt**.
   15. In Notepad, select all the text and then press Ctrl+C.
   16. Switch to Microsoft Edge and Flow.
   17. In the **Message** box, paste the NewTicketCard text.
   18. In the message content, in the **body** section, locate **"text": "[[Title]]"**.
   19. Select **[[Title]]**, scroll down and locate the **Add dynamic content from the apps and connections used in this flow** dialog box.

Be sure to only select the text inside the quotation marks and do not include the quotation marks in the selection.

20. On the **Dynamic content** tab, select **Ask in PowerApps**.
21. The window may automatically scroll to the top of the message box.
22. The message should now display the PowerApps dynamic content.

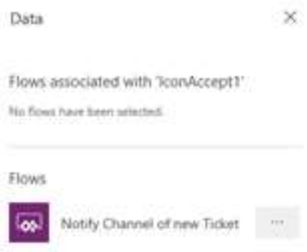


"text": " PostNewTicket_Message × "

23. In the message, locate and select **[[SubmittedBy]]**.
24. Scroll down and in the **Add dynamic content from the apps and connections used in this flow** dialog box, select **Ask in PowerApps**.
If *Ask in PowerApps* is not shown, select **See more** and then select **Ask in PowerApps**.
25. In the message, locate and select **[[SubmittedOn]]**.
26. Scroll down and then select **Ask in PowerApps**.
27. In the message, locate and select **[[Description]]**.
28. Scroll down and then select **Ask in PowerApps**.
29. At the bottom of the page, select **Save**.

4. □ **Finish adding the flow to the Help Ticket PowerApp**

1. In Microsoft Edge, switch to the **Help Tickets - PowerApps** tab.
2. In the Data pane, select **Notify channel of new Ticket**.



Data ×

Flows associated with 'IconAccept1'

No flows have been selected.

Flows

Notify Channel of new Ticket ···

If you closed the Data pane, in the Tree view, under **EditScreen1**, select **IconAccept1**. In the top menu, select **Action** and then select **Flows**.
If the *Notify Channel of new Ticket* flow is not listed, return to the previous task and verify that you changed from your tenant environment to the **TeamsITapp** environment.

3. In the **IconAccept1** pane on the right, select the **Advanced** tab.
4. On the Advanced tab, in the **OnSelect** box, or in the **formula bar** above the form, replace the text with the following:

```
SubmitForm(EditForm1);NotifyChannelofnewTicket.Run(EditForm1.Last
Submit.Subject,EditForm1.LastSubmit.'Submitted
By',EditForm1.LastSubmit.'Submitted
On',EditForm1.LastSubmit.Description)
```

    5. On the top left menu, select **File** and then, on the Save page, select **Save**.
    6. After the changes have been saved, select **Publish**.
    7. In the **Publish** dialog box, select **Publish this version**.

5. □ **Test the newly created flow**
    1. Switch to Microsoft Teams.
    2. In the app bar, select **Teams** and then select the **Help Desk** > **General** channel.
    3. Select the **Help Tickets** tab.
       If prompted, in the **Almost there** dialog box, select **Allow**.
    4. Select the **Create new item** ⊞icon.
    5. In the new ticket, in the **Subject** box, enter **Printer is jammed**.
    6. In the **Description** box, enter **It doesn't print** and then select the **Submit** check mark icon.
    7. In the Help Desk team, select the **Announcements** channel.
    8. Review the new card created that displays the details of your new ticket.
       If a new conversation in the Announcements channel has not been created, create another ticket using the steps above and verify a new conversation has been created in the Announcements channel.

< Table of contents

Conclusion: Building a low code IT helpdesk app for Microsoft Teams

After completing this lab, you should be familiar with Microsoft Team's capabilities as an Application Platform. You learned how to use solution accelerators like PowerApps, Microsoft Flow, QnA Maker, and App Templates to easily build and deploy an app on Microsoft Teams. We hope you will be able to leverage these resources and acquired skills to build solutions for your customers' unique business needs.

< Table of contents

Appendix A

1. □ **Sign up for a Microsoft Azure trial subscription**
    1. If necessary, open Microsoft Edge, browse to **https://admin.microsoft.com**, and then sign into Microsoft 365 as **[USERNAME]** with password **[PASSWORD]**.
    2. If you are already in the Microsoft Azure portal, on the Create a free account page, select **Start free** and then skip the next step.
    3. Open a new browser tab and then browse to **https://azure.microsoft.com/en-us/free**.
       Be sure to use the same browser session you used to sign into Microsoft 365. This

will ensure you are associating the trial account with your lab's Microsoft 365 account.

4. Select **Start free**.
5. At the top of the page, verify you are signed in using your tenant administrator account.
   Being signed in will ensure that the Azure trial is associated with your lab's Microsoft 365 subscription.
6. Complete the sign up form and accept the agreement.

   You will be asked for a credit card number. This will be used to verify your identity. You won’t be charged unless you upgrade.

< Table of contents

Appendix B

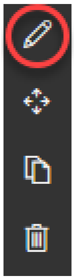
If your ngrok session was reset or restarted, you will have a new URL that will need to be used in your SharePoint Online app.

1. □ **Record the ngrok connection URL**
   1. Using the instructions in Exercise 1, task 8. Create a forwarding URL for your app hosted on YOUR PC, run the ngrok application.
   2. Review the output of the command.
   3. Select the FQDN in the forwarding URL, right-click or tap the selected URL.
      This will copy the text to the clipboard.
      For example, if the forwarding URL is https://123456.ngrok.io, select ***123456.ngrok.io***.
   4. In the following text box, paste or type the URL.
      This will automatically add the URL to this lab document.
      □
      You may also want to save the URL in Notepad.
   5. Leave the command window open or minimize the window.
      This will keep a remote connection open to the local app instance.

      > Do not close the console window. The console session keeps the tunnel connection open to the app you are running locally.

2. □ **Update your SharePoint Online app's service URL**
   It is expected that the app is open in Visual Studio 2019 and that the debugger is running. If this is not the case, review the tasks in Exercise 2 to start the debugger.
   1. In Microsoft Edge, browse to **https://<TenantName>.sharepoint.com/sites/ContosoHR/SitePages/Contoso-Talent.aspx**
      This is the talent app page you should have already created.

2. In the upper right, select **Edit**.
3. In the Contoso Talent web part, select any whitespace around the app elements. For example, select the whitespace to the right of the **Offer** tab.
4. In the edit web part menu, select **Edit web part**.

5. In the **Contoso Talent** pane, in the **Service URL** box, enter **https://<appURL>**.
6. Close the Contoso Talent pane.
7. Select **Republish**.
8. Verify that you do not see any connection errors.
9. Additionally, you may need to uninstall, update the URLs used by the app in Manifest editor, and then reinstall any app that needs to be updated.
   Be sure to update any app registrations in Azure Active Directory with the new ngrok URL.

< Table of contents

Appendix C

1. □ **Cancel your Azure trial subscription**
   If you do not intend to use your Azure trial subscription, use the following steps to cancel the trial subscription.
    1. Open Microsoft Edge and browse to **https://portal.azure.com**.
    2. Sign in as **[USERNAME]** with password **[PASSWORD]**.
    3. In the top bar, in the **Search resources, services, and docs** box, enter **Subs** and then select **Subscriptions**.
    4. In the **Subscriptions** list, select **Free Trial**.
    5. In the Free Trial blade, select **Cancel subscription**.
    6. In the Are you sure you want to cancel subscription Free Trial? blade, review the information and then select **Ignore and cancel**.
    7. In the **TYPE THE SUBSCRIPTION NAME** box, enter **Free Trial**.
    8. Select the **Reason for cancellation** menu and select an appropriate reason.
    9. Optionally, include a description.
    10. Select **Cancel subscription**.
    11. Close Microsoft Edge and any open tabs.

< Table of contents

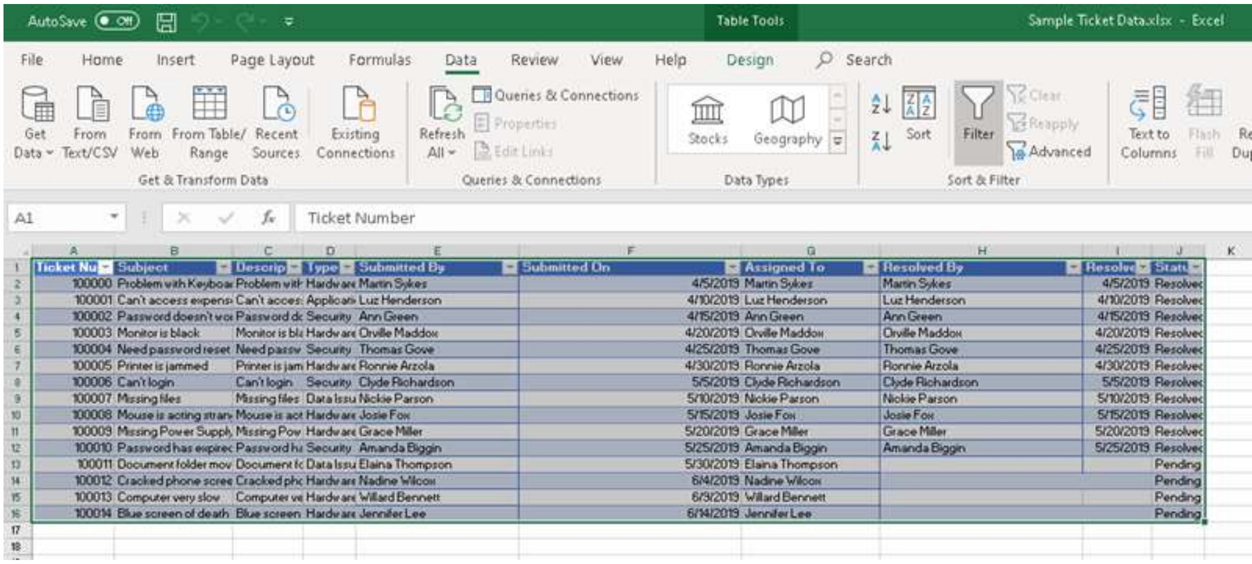
Appendix D

1. □ **Export support ticket data from Excel into a SharePoint Online list**
   If the ITAppLabSPList.exe script fails to run or if you are using a different name for your Help Desk. The instructions will reference the Help Desk team created in the lab - **Building a low code IT helpdesk app for Microsoft Teams > Exercise 2: Using Microsoft Teams to create a team and installing the Power BI app**.
    1. Open Microsoft File Explorer and browse to **C:\Scripts\Backup**.
    2. Double-click or double-tap **Sample Ticket Data.xlsx**.

       If necessary, accept the Office 365 ProPlus license and sign in as **[USERNAME]** with password **[PASSWORD]**.

    3. With the spreadsheet open, on your keyboard, press Ctrl+A.
       This will select cell A1.
    4. On your keyboard, press Ctrl+Shift+End.
       This will select all cells from A1 to the last used cell.

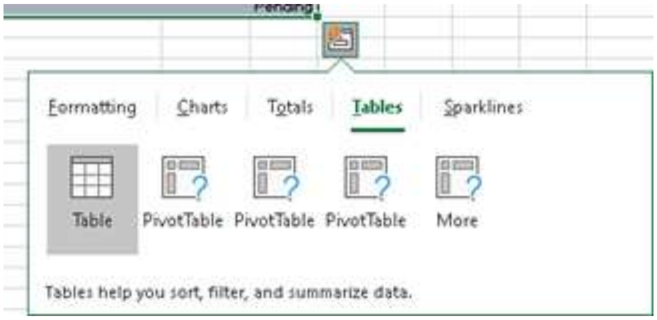
       Alternatively, you can use your mouse or touch device to select the required cells.



A1 Ticket Number

Ticket Nu	Subject	Descrip	Type	Submitted By	Submitted On	Assigned To	Resolved By	Resolve	Statu
100000	Problem with Keyboa	Problem with	Hardware	Martin Sykes	4/5/2019	Martin Sykes	Martin Sykes	4/5/2019	Resolved
100001	Can't access expens	Can't acces:	Applicati	Luz Henderson	4/10/2019	Luz Henderson	Luz Henderson	4/10/2019	Resolved
100002	Password doesn't wo	Password dc	Security	Ann Green	4/15/2019	Ann Green	Ann Green	4/15/2019	Resolved
100003	Monitor is black	Monitor is bla	Hardware	Orville Maddox	4/20/2019	Orville Maddox	Orville Maddox	4/20/2019	Resolved
100004	Need password reset	Need passw	Security	Thomas Gove	4/25/2019	Thomas Gove	Thomas Gove	4/25/2019	Resolved
100005	Printer is jammed	Printer is jam	Hardware	Ronnie Arzola	4/30/2019	Ronnie Arzola	Ronnie Arzola	4/30/2019	Resolved
100006	Can't login	Can't login	Security	Clyde Richardson	5/5/2019	Clyde Richardson	Clyde Richardson	5/5/2019	Resolved
100007	Missing files	Missing files	Data Issu	Nickie Parson	5/10/2019	Nickie Parson	Nickie Parson	5/10/2019	Resolved
100008	Mouse is acting stran	Mouse is act	Hardware	Josie Fox	5/15/2019	Josie Fox	Josie Fox	5/15/2019	Resolved
100009	Missing Power Supply	Missing Pow	Hardware	Grace Miller	5/20/2019	Grace Miller	Grace Miller	5/20/2019	Resolved
100010	Password has expired	Password ha	Security	Amanda Biggin	5/25/2019	Amanda Biggin	Amanda Biggin	5/25/2019	Resolved
100011	Document folder mov	Document fc	Data Issu	Elaina Thompson	5/30/2019	Elaina Thompson			Pending
100012	Cracked phone scree	Cracked phc	Hardware	Nadine Wilcox	6/4/2019	Nadine Wilcox			Pending
100013	Computer very slow	Computer ve	Hardware	Willard Bennett	6/9/2019	Willard Bennett			Pending
100014	Blue screen of death	Blue screen	Hardware	Jennifer Lee	6/14/2019	Jennifer Lee			Pending

5. Point to the lower right corner of the select data and then select the **Quick Analysis** icon or press Ctrl+Q.



6. In the Quick Analysis tool, select the **Tables** tab and then select **Table**. This will treat the selected data within the sheet as a Table.
7. In the ribbon bar, select **Design**.
8. In the ribbon bar, select **Export** > **Export Table to SharePoint List**.
9. In the Export Table to SharePoint List window - Step 1 of 2, in the **Address** box, enter **https://<TenantName>.sharepoint.com/sites/HelpDesk**.

   > If you created a team using a different name than *Help Desk*, change the Address to match the name of the team you created. Be aware that all lab instructions will use the *Help Desk* site name.

10. In the **Name** box, enter **SampleTicketData**.

    > If you use a name other than *SampleTicketData*, remember to use that name when the lab instructions refer to *SampleTicketData*.

11. Select **Next**.
12. Sign in as **[USERNAME]** with password **[PASSWORD]**.
13. In the **Use this account everywhere on your device** dialog box, clear the **Allow my organization to manage my device** check box.
14. Select **This app only**.
15. In the Export Table to SharePoint List - Step 2 of 2 window, review the information and then select **Finish**.
16. In the **Microsoft SharePoint Foundation** dialog box, review the information. The URL shown should match **https://<TenantName>.sharepoint.com/sites/HelpDesk/SampleTicketData** or match the site and/or name you chose to use.
17. Select the link.
18. If necessary, sign in as **[USERNAME]** with password **[PASSWORD]**.
19. Review the newly created list.

< Table of contents