



Developing Microsoft Teams Teamwork Solutions

DISCLAIMER

© 2019 Microsoft Corporation. All rights reserved.

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 is 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.

Table of Contents

Developing Microsoft Teams Teamwork Solutions

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

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

Exercise 3: Building a personal tab in the Contoso HR Talent app

Exercise 4: Building a team tab in the Contoso HR talent app

Exercise 5: Creating a bot in the Contoso HR Talent app

Exercise 6: Installing a messaging extension in the Contoso HR Talent app

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

Exercise 8: Exploring task module invocation

Exercise 9: Reviewing activity feed alert configurations

Exercise 10: Using Graph Explorer

Conclusion

Lab: Developing Microsoft Teams Teamwork Solutions

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 be provided Microsoft Office 365 tenant with an Enterprise E5 trial subscription.

The type text  feature 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 shown.

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.

[< Table of Contents](#)

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. Verify or sync the lab virtual machine's time

In some cases, the hosted virtual machine will not update or sync its clock when launched. The following steps will ensure the clock is set to the correct time for the associated time zone.

- a. In the virtual machine, sign in as **T Lab User** with password **T labpass@word1**.
This is the only virtual machine in this lab and is named CLIENT01. You may see reference to this virtual machine's name during the lab. All work in this lab is expected to be performed inside this virtual machine. Activation notifications can be ignored.
- b. In the System tray, right-click or tap and hold **Date/Time** and then select **Adjust date/time**.
- c. On the Date & time page, select the **Set time automatically** switch and verify it is set to **Off**.
- d. Select the **Set time automatically** switch again and verify it is set to **On**.
You may see the virtual machine's clock update.

2. Browse to the Microsoft Teams web app

- a. Open Microsoft Edge and then browse to **T https://teams.microsoft.com**.
- b. In the Sign in window, enter **T [USERNAME]**.
- c. Select **Next**.
- d. In the **Password** box, enter **T [PASSWORD]**.
- e. Select **Sign in**.
- f. In the **Let Microsoft Edge save and fill your password for this site next time** banner, select **Save**.
- g. In the **Stay signed in** dialog box, select the **Don't show this again** check box and then select **Yes**.



Saving the credentials is not mandatory but will make it easier to sign in and out throughout the lab.

- h. In the Microsoft Teams web app, complete or close the introduction wizard and close any other notification windows.

3. Install and sign in to the Microsoft Teams desktop app

- a. In the Microsoft Teams web app, in the left navigation, select **Get app**.
 - b. At the bottom of the browser, in the banner, select **Run**.
 - c. In the **Login to Microsoft Teams** dialog box, sign in as **[T] [USERNAME]** with password **[T] [PASSWORD]**.
 - d. Leave Microsoft Teams open.
- 4. Run the TeamworkLab.exe script as administrator**
- a. Open File Explorer and then browse to **C:\Scripts**.
 - b. Right-click or tap and hold **TeamworkLab.exe** and then select **Run as administrator**.
 - c. In the **User Account Control** dialog box, select **Yes**.
 - d. In the **Office 365 Administrator** dialog box, enter **[T] [USERNAME]** with password **[T] [PASSWORD]** and then select **OK**.
 - e. Continue with the lab. You do not need to wait for the script to complete.
The script will create user accounts and teams in your subscription that will be used in the lab.
Later, when the script completes, in the **Script Complete** dialog box, select **OK**.
If the script takes longer than 15 minutes to complete, close the PowerShell window and run the script again.
 - f. Close File Explorer.
- 5. Create a replacement token for the Office 365 domain name**
- a. In the following FQDN, select the first name and then right-click or tap and hold and then select **Copy**:
- @lab.CloudCredential(134).Tenant**
- For example, if the FQDN is MODx321.onmicrosoft.com, you will copy **MODx321**
- b. In the following text box, paste or type the name.
This will automatically add the tenant name to this lab document.

- 6. Enable sideloading of external apps**
- a. Switch to Microsoft Edge and then browse to **[T] <https://admin.microsoft.com>**.
 - b. You should automatically be signed in.
If not, sign in as **[T] [USERNAME]** with password **[T] [PASSWORD]**
 - c. In the left navigation, select the **Show more** ellipsis icon.
 - d. Select **Settings > Services & add-ins**.
 - e. On the Services & add-ins page, select **Microsoft Teams**.

 Using the scroll wheel on your mouse may skip too far in the virtual machine. If you are having trouble locating items when scrolling, try using the scroll bar instead.

- a. Under **Tenant-wide settings**, select **Apps**.
- b. Under **External Apps**, to the right of **Allow sideloading of external apps**, select the toggle switch and verify it is set to **On**.
- c. At the bottom of the page, select **Save**.
- d. In the Microsoft Teams pane, select **Close**.

7. Verify the Teams and channels created by the lab script

- a. Switch to the Microsoft Teams desktop app.
- b. In the left navigation verify the **Teams** feature is selected.
If it is not selected, select **Teams**.
- c. In the list of teams, verify that the following teams and custom channels exist:
 - Team - **HR Hiring**
 - Channel - **HR activities**
 - Channel - **HR interview**
 - Team - **HR New Orientation**
 - Channel - **New hire materials**
 - Channel - **Ramp-up**
 - Team - **HR Talent Development**
 - Channel - **New talent training**
 - Channel - **Training mentoring**

 If the teams or team channels do not exist, continue with the lab. The Teams may still be provisioning and replicating in Office 365.
Keep in mind that when you reach a task that requires one of the new teams or channels, you may have to wait until it is available.

8. Install App Studio from the Store

- a. In Microsoft Teams, in the left navigation, at the bottom above **Help**, select **Store**.
- b. On the Store page, in the **Search all** box, enter  **App Studio**.
- c. In the results, select **App Studio**.
- d. In the **App Studio** dialog box, review the information and then select **Install**.
- e. Close the **App Studio is now available to you** dialog box.

9. Create a new app named Contoso HR Talent App

- a. In Microsoft Teams, in the left navigation, select the **More apps** ellipsis icon and then select **App Studio**.
- b. In App Studio, select the **Manifest editor** tab.
- c. Under **Welcome**, select **Create a new app**.
- d. In the left navigation, under **Complete these steps** verify that **App details** is selected.
- e. On the App details page, in the **Short name** box, enter **Contoso HR Talent App**.
- f. In the **Long name** box, enter **Human Resources New Hiring App**.
- g. Under **Identification** review the information and then select **Generate**.
- h. Review the unique identity given to your new app.
- i. Select the **App ID** and then press **Ctrl+C**.
- j. 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.
- k. In the **Package Name** box, enter **<TenantName>.hr.com.microsoft.com**.
- l. In the **Version** box, enter **0.0.1**.
- m. Under **Descriptions**, in the **Short description** box, enter **Assist HR talent hiring**.
- n. In the **Long description** box, enter **This Human Resources app is designed to aid the HR department with their new hire acquisitions.**
- o. Under **Developer information**, in the **Name** box, enter your name.
- p. In the **Website** box, enter **https://<TenantName>.sharepoint.com/ContosoHR**.
- q. Under **App URLs**, in the **Privacy statement** box, enter **https://privacy.microsoft.com/en-us/privacystatement**.
- r. In the **Terms of use** box, enter **https://www.microsoft.com/en-us/servicesagreement**.
- s. Under **Branding**, under **Full color**, select **Update**.
- t. In the Open window, browse to **C:\LabFiles**.
- u. Select **FullColorBrand.png** and then select **Open**.
- v. Under **Transparent outline**, select **Update**.
- w. In the Open window, in **C:\LabFiles**, select **TransparentBrand.png** and then select **Open**.
- x. On the App Studio page, under the tabs in the breadcrumb navigation, select **Home**.

y. In the Manifest editor tab, verify that the new app is shown.

The app was automatically saved.

10. **Create a forwarding URL for your app hosted on CLIENT01**

a. Right-click or tap and hold **Start** and then select **Windows PowerShell (Admin)**.

b. In the **User Account Control** dialog box, select **Yes**.

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

```
cmd
```

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

```
cd C:\LabFiles
```

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

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

f. Review the output of the command.



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.

g. 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 <http://123456.ngrok.io>, select **123456.ngrok.io**.

h. 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.

i. 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.

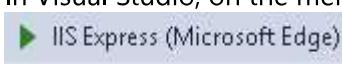
11. **Open and build the TeamsTalentMgmtApp using Microsoft Visual Studio** This is the Contoso Talent app that will be deployed.

- a. Select **Start** and then select **Visual Studio 2017**.
- b. In Visual Studio **Welcome** dialog box, select **Sign in**.
- c. In the **Sign in** dialog box, in the **Email, phone, or Skype** box, enter **T [USERNAME]** and then select **Next**.
- d. In the **Enter password** dialog box, in the **Password** box, enter **T [PASSWORD]** and then select **Sign in**.
- e. In the **Visual Studio** dialog box, review the default settings and then select **Start Visual Studio**.
The General development settings will be used for this lab.
- f. On the Start Page tab, under **Open** select **Open Project / Solution**.
- g. In the Open Project window, browse to **C:\LabFiles\Solutions\TeamsTalentMgmtApp**.
- h. Select **TeamsTalentMgmtApp.sln** and then select **Open**.
Wait for the solution to load.
- i. On the menu, select **Build** and then select **Build Solution**.
Monitor the Output pane and wait for the build to complete. This may take 1-2 minutes.

12. **Update the web.config file with the app URL and app ID**

- a. On the right, in Solution Explorer, select **Web.config**.
- b. In the Web.config file, under **configuration > appSettings**, locate **BaseUrl**.
- c. Replace the current value of **PASTE_YOUR_NGROK_URL_HERE** with **T https://<appURL>**.
- d. In the Web.config file, under **configuration > appSettings**, locate **TeamsAppId**.
- e. Replace the current value of **PASTE_YOUR_TEAMS_APP_ID_HERE** with **T <appId>**.
- f. On the menu bar, select the **Save** icon or press **Ctrl+S**.

13. **Run debugging using IIS Express (Microsoft Edge)**

- a. In Visual Studio, on the menu bar, select **IIS Express (Microsoft Edge)**

- b. In Microsoft Edge, verify that a new tab has opened and that the **Team Talent Management App** page is loaded.
It may take a minute to build and load the page.
- c. After the page successfully loads and you've confirmed that the solution is ready, close the **Help** tab.
- d. 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**
 - a. In CLIENT01, select **Start > Visual Studio Code > Visual Studio Code**.
 - b. Close the **Welcome** tab.
 - c. In Visual Studio Code, on the menu, select **File > Open File**.
 - d. In the Open File window, browse to **C:\LabFiles\Solutions\SharePointTalentMgmtWebPart\src\webparts\contosoTalent**
 - e. Select **settings.json** and then select **Open**.
 - f. To the right of **ServiceURL":**, replace **PASTE_YOUR_NGROK_URL_HERE** with  **https://<appURL>**.
 - g. On the menu, select **File > Save**.
- 2. **Open ContosoTalentWbPart.ts**
 - a. In Visual Studio Code, on the menu, select **File > Open File**.
 - b. In the Open File window, browse to **C:\LabFiles\Solutions\SharePointTalentMgmtWebPart\src\webparts\contosoTalent**
 - c. Select **ContosoTalentWbPart.ts** and then select **Open**.
- 3. **Verify the teamsContext private variable exists inside the ContosoTalentWebPartProps class**
 - a. In ContosoTalentwebParts.ts, locate **export default class ContosoTalentWebPartProps**.
This will be on line 34.
 - b. Under **export default class ContosoTalentWebPartProps**, verify that the **private _teamsContext: microsoftTeams.Context;** variable exists.
This variable stores the Microsoft Teams context information.
 - c. No changes need to be made.

4. Verify the **onInit** method is present

- a. In ContosoTalentWebParts.ts, locate **protected onInit()**.
This will be on line 43.
- b. 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;
}
```

- c. No changes need to be made.

5. Review the **render** method

- a. In ContosoTalentWebParts.ts, locate **let baseClass**.
This will be on line 70.
- b. 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.
- c. No changes need to be made.

6. Start a new terminal session using Visual Studio Code

- a. In Visual Studio Code on the menu, select **Terminal > New Terminal**.
- b. At the PowerShell prompt, enter the following and then press Enter:

```
📄 cd C:\LabFiles\Solutions\SharePointTalentMgmtWebPart
```

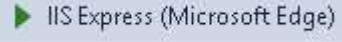
⚠ You may need to select inside the terminal pane and press Enter to see the prompt.

💡 To save time in the lab, the npm package manager has already been installed using the *npm install* command.

7. Create a release build of the solution

- a. In the terminal pane, at the PowerShell prompt, enter the following and then press Enter:

```
📄 gulp bundle --ship
```

- b. Wait for the build to complete.
- 8. Package the Contoso Talent web part**
- a. In the terminal pane, at the PowerShell prompt, enter the following and then press Enter:
 `gulp package-solution --ship`
 - b. Review the output of the command.
This will create the **contoso-talent.sppkg** file in the `C:\LabFiles\Solutions\SharePointTalentMgmtWebPart\sharepoint\solution` folder.
- 9. Verify the debugger is still running in Visual Studio**
- a. Switch to Microsoft Visual Studio.
 - b. On the right, verify the Diagnostic Tools pane is displayed and showing session diagnostics.
 - c. If Diagnostic Tools are not running, on the menu bar, select **IIS Express (Microsoft Edge)**

 - d. In Microsoft Edge, verify that a new tab has opened and that the **Team Talent Management App** page is loaded.
It may take a minute to build and load the page.
 - e. After the page successfully loads, close the **Help** tab.
 - f. Leave Visual Studio open with the debugger running.
- 10. Create a new app catalog for your SharePoint Online tenant**
- a. Switch to Microsoft Edge.
 - b. 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]**
 - c. In the left navigation, select **apps**.
 - d. On the apps page, select **App Catalog**.
If the app catalog page for your tenant is displayed, then the app catalog is already configured and you can skip to the next task.
 - e. On the App Catalog Site page, verify **Create a new app catalog site** is selected and then select **OK**.
 - f. On the Create App Catalog Site Collection page, in the **Title** box, enter  **AppCatalog**.
 - g. In the **Web Site Address** box, to the right of `/sites/`, enter  **appcatalog**.
 - h. In the **Administrator** box, enter  **[USERNAME]** and then select the **Check Names** icon.
 - i. Verify your administrator name resolves and then select **OK**.

- j. On the site collections page, wait for the new appcatalog site to finish provisioning.
If necessary, refresh the page to view the current provisioning status.

11. Upload the Contoso Talent package to the app catalog

- a. In Microsoft Edge, on the site collections page, select
<https://<TenantName>.sharepoint.com/sites/appcatalog>.
- b. In the **site collection properties** dialog box, select
<https://<TenantName>.sharepoint.com/sites/appcatalog>.
- c. On the App Catalog page, in the left navigation, select **Apps for SharePoint**.
- d. On the Apps for SharePoint page, select **Upload**.
- e. In the **Add a document** dialog box, next to **Choose a file**, select **Browse**.
- f. In the Open windows, browse to **C:\LabFiles\Solutions\SharePointTalentMgmtWebPart\sharepoint\solution**.
- g. Select **contoso-talent.sppkg** and then select **Open**.
- h. In the **Add a document** dialog box, select **OK**.
- i. In the **Do you trust contoso-talent-client-side-solution** dialog box, review the information.
- j. Select the **Make this solution available to all sites in the organization** check box and then select **Deploy**.

12. Verify there are no errors for the app package

- a. On the Apps for SharePoint page, in the **All Apps** list, locate the **contoso-talent-client-side-solution** app.
- b. 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, wait 1-2 minutes, and then upload the app again.

13. Add the Contoso Talent app to a new page in the Contoso HR site

- a. 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.
- b. In the top right, select the **Settings menu** gear icon and then select **Add a page**.
- c. Close any open introduction wizard.
- d. On the new page, select **Name your page** and then enter  **Contoso Talent**.

- e. On the Contoso Talent page, in the middle, select the **Add a new web part in column one**  icon.
- f. Under **All A to Z**, select **Contoso Talent**.
- g. 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.

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

14. Review the webpart

- a. On the Contoso Talent page, review the webpart.
- b. Select the different tabs in the web part and review the information and functionality.

15. Create a compressed archive from the Teams folder content

- a. In CLIENT01, 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.

- b. In the **User Account Control** dialog box, select **Yes**.
- c. In Windows PowerShell, enter the following and then press Enter:

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

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

16. Upload a custom app to the HR Hiring Team

- a. Switch to the Microsoft Teams desktop app signed in as the tenant administrator.
- b. In the left navigation, select **Teams**.
- c. In the Teams pane, to the right of **HR Hiring**, select the **More options** ellipsis icon and then select **Manage team**.



- d. On the HR Hiring page, on the menu, select the **Apps** tab.
- e. In the lower-right corner, select **Upload a custom app**.
- f. In the Open window, browse to **C:\LabFiles**.
- g. Select **Teams.zip** and then select **Open**.
- h. Review the **Apps** list and verify the **Contoso Hiring Board (Custom app)** is listed.

17. Refresh the Contoso Talent page in Microsoft Edge and verify connectivity

- a. Switch to Microsoft Edge and the Contoso Talent tab.
- b. In Microsoft Edge, select the **Refresh** icon or press F5 to reload the page.
- c. Verify the page is correctly displayed.

! 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**

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

- a. Switch to the Microsoft Teams desktop app.
- b. On the HR Hiring page, select the **Channels** tab.
- c. In the **Channels** list, select **HR activities**.
- d. In the HR activities channel, on the menu, select the **Add a tab +** icon.
- e. In the Add a tab window, select **Contoso Hiring Board**.
- f. In the Contoso Hiring Board window, in the **Service URL** box, verify that **https://<appURL>** is shown.

! If the Service URL box is not displayed, select **Back** and then try the following to correct the issue:

- a. In the Add a tab window, select **SharePoint**.
- b. In the **SharePoint** dialog box, select **Home** and then select **Save**.

If the SharePoint tab's pages and lists are not displayed, select **Back** and try again. In some cases, you may need to sign out and sign in to the Microsoft Teams desktop app as **[T] [USERNAME]** with password **[T] [PASSWORD]** and then retry the steps.

You may also try to add the Excel tab and then try to add the SharePoint tab. You do not need to complete the addition of the Excel tab. This may have to be done in the lab environment to allow the UI to show the tab settings dialog when the screen refresh is delayed.

- c. In the HR activities channel, select the **Add a tab** icon.
- d. In the Add a tab window, select **Contoso Hiring Board**.

g. Select **Save**.

h. 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.



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: 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. Open the Contoso HR Talent App in the Manifest editor
 - a. In Microsoft Teams, in the left navigation, select the **More apps** ellipsis icon and then select **App Studio**.
 - b. On the App Studio page, select the **Manifest editor** tab.
 - c. Under **Recently created apps**, select **Contoso HR Talent App**.
- 2. Add a personal tab to the Contoso HR Talent App
 - a. In the Contoso HR Talent App, in the left navigation under step **2 Capabilities**, select **Tabs**.
 - b. On the Tabs page, under **Add a personal tab**, select **Add**.
 - c. In the **Personal tab** dialog box, in the **Name** box, enter **Potential candidates**.
 - d. In the **Entity ID** box, enter **<appId>**.
This is your App ID.
 - e. In the **Content URL** box, enter **https://<appURL>/StaticViews/OpenPositionsPersonalTab.html?v=1**.
 - f. In the **Website URL** box, enter **https://<appURL>/StaticViews/OpenPositionsPersonalTab.html?v=1&web=1**.
 - g. Select **Save**.
- 3. Distribute and test the new Team tab
 - a. In the left navigation, under step **3 Finish**, select **Test and distribute**.
 - b. On the Test and Distribute page, select **Install**.
 - c. In the **Contoso HR Talent App** dialog box, select **Install**.
 - d. In the **Contoso HR Talent App** dialog box, review the information and then select **Open**.
 - e. Review the information in the new personal tab. You can return to the personal tab later by selecting the **More apps** ellipsis icon and then selecting **Contoso HR Talent App**.

 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 4: 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
 - a. In Microsoft Teams, in the left navigation, select the **More apps** ellipsis icon and then select **App Studio**.
 - b. On the App Studio page, select the **Manifest editor** tab.
 - c. Under **Recently created apps**, select **Contoso HR Talent App**.
- 2. Add a Team tab
 - a. In the Contoso HR Talent App, in the left navigation under step **2 Capabilities**, select **Tabs**.
 - b. On the Tabs page, under **Team tab**, select **Add**.
 - c. In the **Team tab** dialog box, in the **Configuration URL** box, enter **[T https://<appURL>/StaticViews/TeamTabConfig.html?v=1]**.
 - d. Verify the **Can update configuration** check box is selected.
If necessary, select the check box.
 - e. Under **Scope**, select the **Team** and **Group chat** check boxes.
 - f. Select **Save**.
- 3. Distribute and test the new Team tab
 - a. In the left navigation, under step **3 Finish**, select **Test and distribute**.
 - b. On the Test and Distribute page, select **Install**.
 - c. In the **Contoso HR Talent App** dialog box, review the information.
 - d. Next to **Add to a team or chat**, select **Search**.
 - e. In the **Teams suggestions** list, select **HR Hiring**.
 - f. Select **Install**.
 - g. In the **Contoso HR Talent App is now available for HR Hiring** dialog box, select the **First, pick the channel where you want to use the app** menu and review the available channels in the Team.
 - h. Select **General**.

- i. To the right of **Tab**, select **Set up**.
- j. In the **Contoso HR Talent app** dialog box, in the **Tab name** box, enter **Senior Designer position**.
- k. Select the **Select job posting** menu and then select **Senior Designer**.
- l. Select **Save**.
- m. In the Senior Designer position tab, review the newly added content.



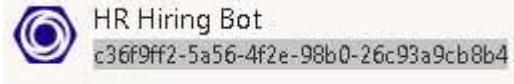
More information about configurable tabs can be found using this Microsoft document
<https://docs.microsoft.com/en-us/microsoftteams/platform/concepts/tabs/tabs-configurable>

< Table of Contents

Exercise 5: 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
 - a. In Microsoft Teams, in the left navigation, select the **More apps** ellipsis icon and then select **App Studio**.
 - b. On the App Studio page, select the **Manifest editor** tab.
 - c. Under **Recently created apps**, select **Contoso HR Talent App**.
- 2. Create a new bot
 - a. In the Contoso HR Talent App, under step **2 Capabilities**, select **Bots**.
 - b. Under **Bots**, select **Set up**.
 - c. In the Set up a bot window, select the **New bot** tab.
 - d. In the **Name** box, enter **HR Hiring Bot**.
 - e. Under **Messaging bot** select the **My bot supports uploading and downloading files** check box.
 - f. Under **Scope**, select the **Personal** and **Team** check boxes.
 - g. Select **Create bot**.
- 3. Save the bot's app ID
 - a. On the Bots page, under **HR Hiring Bot**, select the **App Id** text located under the bot name.

The screenshot shows a card for the "HR Hiring Bot". The card has a blue circular icon on the left. To its right, the bot's name "HR Hiring Bot" is displayed in a large, bold, black font. Below the name is a smaller, gray text box containing the bot's unique identifier: "c36f9ff2-5a56-4f2e-98b0-26c93a9cb8b4".

b. Press **Ctrl+C**.
 - c. 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

- a. On the Bots page, under **App passwords**, select **Generate a new password**.
- b. Review the password.
If the password contains an ampersand "&" symbol, close the **New password generated** dialog box and then select **Generate a new password** again.
The password will be used later in the web.config file and the ampersand special character will cause failures.

c. Select the new password and then press **Ctrl+C**.

d. 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.

e. Select **OK**.

5. Add the Messaging endpoint

- a. On the Bots page under **Messaging endpoint**, in the **Bot endpoint address** box, enter  <https://<appURL>/api/messages>.

6. Add commands to the bot

- a. On the Bots page under **Commands**, select **Add**.
- b. In the **New command** dialog box, in the **Command text** box, enter  **help**.
- c. In the **Help text** box, enter  **Find out what I can do**.
- d. Under **Scope**, select the **Personal** and **Team** check boxes.
- e. Select **Save**.
- f. Under **Bot menus**, notice the new command has been added.

7. Edit the web.config file with the new app ID and password information

- a. Switch to Visual Studio and the Web.config tab.
- b. On the menu bar, select the **Stop Debugging**  icon.
- c. In the Web.config file, locate **MicrosoftAppId** on line 12.
- d. Replace the **PASTE_YOUR_BOT_APPID_HERE** text with  **<BotappId>**.
- e. Locate **MicrosoftAppPassword** on line 13.
- f. Replace the **PASTE_YOUR_BOT_APP_PASSWORD_HERE** text with  **<BotPassword>**.
- g. On the menu bar, select **Save**.

h. On the menu, select **Debug** > **Start Debugging** or press F5.

💡 If the debug session fails, review the build error. You may need to generate a new password. If you generate a new password, be sure to replace the old password with the new password in the lab document as well as the Web.config file.

i. In Microsoft Edge, wait for the localhost page to load and then close the Help tab.

8. Distribute and test the new bot

- a. Switch to the Microsoft Teams desktop app.
- b. In App Studio, in the left navigation under step **3 Finish**, select **Test and distribute**.
- c. On the Test and Distribute page, select **Install**.
- d. In the **Contoso HR Talent App** dialog box, review the information.
- e. Next to **Add to a team or chat**, select **Search**.
- f. In the **Teams suggestions** list, select **HR Hiring**.
- g. Select **Install**.
- h. In the **Contoso HR Talent App is now available for HR Hiring** dialog box, to the right of **Bot**, select **Set up**.
The default **General** channel will be used.

- i. On the General channel Conversations tab, at the bottom of the window under **What can I do?**, select **help**.
- j. 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.
- k. Review the results of the request.

l. In the **Start a new conversation** box, enter **T @HR** and then select **HR Hiring Bot**.

⚠️ If the HR Hiring bot is not resolving, in the left navigation, select **Chat**, and create a new chat with the HR Hiring Bot.
If the HR Hiring Bot is not responding, continue to the next exercise and return after 5-15 minutes and then try the HR Hiring bot again.

- m. To the right of HR Hiring Bot, enter **T List all your open positions** and then press Enter.
You may need to press Enter a second time.
- n. 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 6: Installing a messaging extension in 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**
 - a. In the Microsoft Teams app, in the left navigation, select the **More options** ellipsis icon and then select **App Studio**.
 - b. In App Studio, select the **Manifest editor** tab.
 - c. In Manifest editor, select **Contoso HR Talent App**.
- 2. Add messaging extensions to the app**
 - a. In the Contoso HR Talent App, under step **2 Capabilities**, select **Messaging extensions**.
 - b. Under **Messaging extensions**, select **Set up**.
 - c. In the Messaging Extension window, select the **Existing** tab.
 - d. In the **Name** box, enter **HR Talent App Messaging Extension**.
 - e. Under **Bot ID**, select **Select from one of my existing bots**.
 - f. Select the **Choose One** menu and then select **HR Hiring Bot**.
 - g. Leave the **Can update configuration** check box set to the default selection.
 - h. Select **Save**.
- 3. Verify the messaging endpoint**
 - a. 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 commands to the messaging extension**
 - a. Under **Command**, select **Add**.
 - b. In the **New command** dialog box, in the **Command Id** box, enter **searchPositions**.
 - c. In the **Title** box, enter **Positions**.

- d. In the **Description** box, enter T **Search positions by keyword**.
- e. Verify the **Initial run** check box is selected.
- f. Under **Parameter**, in the **Name** box, enter T **searchText**.
- g. In the **Title** box, enter T **Keywords**.
- h. In the **Description** box, enter T **Search by keywords**.
- i. 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 T <https://teams.microsoft.com>, sign in as T **[USERNAME]** with password T **[PASSWORD]**. Use the Microsoft Teams web client to create the messaging extension command. When complete, return to the Microsoft Teams desktop client.

- j. Notice the new command is displayed under the **Command** section.
- k. Under **Command**, select **Add**.
- l. In the **New command** dialog box, in the **Command Id** box, enter T **searchCandidates**.
- m. In the **Title** box, enter T **Candidates**.
- n. In the **Description** box, enter T **Search candidates by name**.
- o. Verify the **Initial run** check box is selected.
- p. Under **Parameter**, in the **Name** box, enter T **searchText**.
- q. In the **Title** box, enter T **Name**.
- r. In the **Description** box, enter T **Search by name**.
- s. Select **Save**.

5. Distribute and test the new messaging extension

- a. In the left navigation, under step **3 Finish**, select **Test and distribute**.
- b. On the Test and Distribute page, select **Install**.
- c. In the **Contoso HR Talent App** dialog box, review the information.
- d. Next to **Add to a team or chat**, select **Search**.
- e. In the **Teams suggestions** list, select **HR Hiring**.
- f. Select **Install**.

- g. In the **Contoso HR Talent App is now available for HR Hiring** dialog box, to the right of **Messaging**, select **Set up**.
The default **General** channel will be used.
- h. On the General channel Conversations tab, review the **Contoso HR Talent App** search box.
- i. Select the **Candidates** tab and review the results.
- j. In the **Search by name** box, enter **T Deb** and review the results.
- k. Select **Debora Morse**.
- l. Review the information and available options, and then press Enter.
6. **Switch the Microsoft Teams desktop app to Developer preview mode**
- a. In the Microsoft Teams desktop app, in the top right, select the profile picture or profile initials.
 - b. Select **About > Developer preview**.
 - c. In the **Developer preview** dialog box, review the information and then select **Switch to developer preview**.
 - d. In the **Sign in** dialog box, enter **T [USERNAME]** and then select **Next**.
Microsoft Teams may open behind other windows.
 - e. In the **Enter password** dialog box, enter **T [PASSWORD]** and then select **Sign in**.
7. **Download the existing app manifest**
- a. In the Microsoft Teams app, in the left navigation, select the **More options** ellipsis icon and then select **App Studio**.
 - b. In App Studio, select the **Manifest editor** tab.
 - c. In Manifest editor, select **Contoso HR Talent App**.
 - d. In the left navigation, under step **3 Finish**, select **Test and distribute**.
 - e. Select **Download**.
This will download the package to the default download folder.
8. **Extract the ContosoHRTalentApp.zip**
- a. Open File Explorer and, under **Quick access**, select **Downloads**.
 - b. In the Downloads folder, right-click or tap and hold **ContosoHRTalentApp.zip** and then select **Extract All**.
 - c. In the **Select a Destination and Extract Files** dialog box, enter **T C:\LabFiles\ContosoHRTalentApp**.
 - d. Verify **Show extracted files when complete** is selected.
If necessary, select the check box.

- e. Select **Extract**.
- 9. Update the manifest.json file with a new command**
- In this task, you will add code to initiate actions from a messaging extension and use a task module response to collect information from a user.
- a. In File Explorer, in the ContosoHRTalentApp folder, right-click or tap and hold **manifest.json** and then select **Open with Code**.
 - b. In Visual Studio Code, locate the **composeExtensions** key on line 82.
 - c. Inside the composeExtensions key, at the end of line 112 to the right of the closed curly bracket, add the following code:

```
,  
{  
  "id": "openNewPosition",  
  "title": "Create position",  
  "description": "Open a new position",  
  "type": "action",  
  "fetchTask": true,  
  "parameters": [  
    {  
      "name": "empty",  
      "title": "empty"  
    }  
  ]  
}
```

d. When complete, your changes should look like this:

```
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
```

e. Select **File > Save**.

10. **Create a compressed archive from the ContosoHRTalentApp folder content**

- a. In CLIENT01, right-click or tap and hold **Start** and then select **Microsoft PowerShell (Admin)**.
- b. In the **User Account Control** dialog box, select **Yes**.
- c. In Windows PowerShell, enter the following and then press Enter:

```
powershell Compress-Archive -Path C:\LabFiles\ContosoHRTalentApp\* -DestinationPath C:\LabFiles\UpdatedContosoHRTalentApp.zip
```

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

11. **Uninstall the previous Contoso HR Talent App personal and Team tab**

- a. Switch to Microsoft Teams.

- b. In the left navigation, select the **More apps** ellipsis icon.
- c. Highlight **Contoso HR Talent App** and then, to the right, select the ellipsis icon.
- d. Select **Uninstall**.
- e. In the **Uninstall Contoso HR Talent App** dialog box, select **Uninstall**.
- f. In the left navigation, select **Teams**.
- g. To the right of **HR Hiring**, select the **More options** ellipsis icon and then select **Manage Team**.
- h. On the menu, select the **Apps** tab.
- i. In the **Apps** list, locate **Contoso HR Talent App (Custom app)**.
- j. In the **Apps** list, to the right of **Contoso HR Talent App (Custom app)**, select the **Uninstall** icon.
- k. In the **Uninstall Contoso HR Talent App (Custom app)** dialog box, select **Uninstall**.

12. **Upload the updated Contoso HR Talent App**

- a. In the lower-right corner, select **Upload a custom app**.
- b. In the Open window, browse to **C:\LabFiles**.
- c. Select **UpdatedContosoHRTalentApp.zip** and then select **Open**.
- d. Review the **Apps** list and verify the **Contoso HR Talent App (Custom app)** is listed.

13. **Verify the new messaging extensions actions**

- a. In the left navigation, select **Teams**.
- b. Select **HR Hiring > General**.
- c. On the Conversations tab, under **Start a new conversation**, select the **Messaging extensions** ellipsis icon.

- d. Under **Suggestions**, select **Contoso HR Talent App**.

14. **Create a new position using the messaging extensions action**

- a. In the Contoso HR Talent App, select the **Actions**  icon and the select **Create position**.
- b. In the **Contoso HR Talent App**, in the **Title** box, enter  **IT Operations manager**.
- c. Select the **Location** menu and then select **Dallas**.
- d. 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.

- e. Select **Create posting**.
- f. In the **Contoso HR Talent App** dialog box, review the posting and then select **Confirm posting**.
- g. On the Conversations tab, press Enter.
This will post a new message about the added position.
- h. On the Conversations tab, under **Start a new conversation**, select the **Messaging extensions** ellipsis icon.
- i. Select **Contoso HR Talent App**.
- j. Review the **Positions** tab and notice the new position that has been created.



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
 - a. In the Microsoft Teams desktop app, in the left navigation, select the **More options** ellipsis icon and then select **App Studio**.
 - b. In App Studio, select the **Manifest editor** tab.
 - c. In Manifest editor, select **Contoso HR Talent App**.
- 2. Add a new Connector to the Contoso Talent app
 - a. In the Contoso HR Talent App, under step **2 Capabilities**, select **Connectors**.
 - b. 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 Microsoft Edge opens to the **Connectors for** " page, perform the following steps:
 - a. Close the tab and switch to the Microsoft Teams desktop app.
 - b. Close the **Connector** dialog box.
 - c. Under **New connector**, select **Register**.
 - c. In the Connectors Developer Dashboard window, in the **Connector name** box, enter **Contoso Talent Connector**.
 - d. Under **Logo**, select the **Edit** icon.
 - e. In the Open window, browse to **C:\LabFiles**.
 - f. Select **HRConnector.png** and then select **Open**.
 - g. In the **Short description of your app (10 words or less)** box, enter **Talent app for Contoso managers and recruiters**.
 - h. 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.**
 - i. In the **Company website** box, enter **T**
https://<TenantName>.sharepoint.com/sites/ContosoHR.

- j. In the **Configuration page for your Connector** box, enter <https://<appURL>/StaticViews/ConnectorConfig.html>.
- k. In the **Valid domains** box, enter <appURL>.
- l. In the **Do you want to enable actions on your Connector cards** dialog box, select **No**.
- m. 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

- a. Under **Test your Connector**, select **Download Manifest**.
- b. In the banner, select the **Save** menu and then select **Save as**.
- c. In the Open window, browse to **C:\LabFiles** and then select **Save**.

4. Record the Connector Id

- a. In Microsoft Edge, in the banner, select **Open folder**.
- b. In C:\LabFiles, right-click or tap and hold **manifest.json** and then select **Open with Code**.
- c. Locate the **Connector Id** on line 4.
This Connector Id is also located on line 23.
- d. Select the id located inside the quotation marks and then press Ctrl+C.
- e. 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.

5. Complete the Connector setup in Microsoft Teams

- a. Switch to the Microsoft Teams desktop app.
- b. In the **Connector** dialog box, in the **Name** box, enter **Contoso Talent Connector**.
- c. In the **Connector ID** box, enter <ConnectorId>.
- d. In the **Configuration URL** box, enter
<https://<appURL>/StaticViews/ConnectorConfig.html>.
- e. Select **Save**.

6. Distribute and test the new connector

- a. In the left navigation, under step **3 Finish**, select **Test and distribute**.
- b. On the Test and Distribute page, select **Install**.
- c. In the **Contoso HR Talent App** dialog box, review the information.
- d. Next to **Add to a team or chat**, select **Search**.
- e. In the **Teams suggestions** list, select **HR Hiring**.
- f. Select **Install**.
- g. In the **Contoso HR Talent App is now available for HR Hiring** dialog box, select the **First, pick the channel where you want to use the app** menu and then select **HR activities**
- h. To the right of **Connector**, select **Set up**.
You may need to wait for the dialog box to open. If the dialog box does not open, close the **Connectors for "HR activities" channel in "HR Hiring" team** dialog box and try to install the connector again.
- j. Close the **Connectors for "HR activities" channel in "HR Hiring" team** dialog box.
- k. In the HR activities channel, select the **Contoso Hiring Board** tab.
- l. Under **Senior Software Engineer**, locate **Lynda Trevino**.
- m. To the right of **Lynda Trevino**, in the **Current Stage** column, select **Interviewing** and then select **Offered**.
- n. In the HR activities channel, select the **Conversations** tab.
- o. Notice the Contoso Talent Connector has updated the conversation with the change of status for 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**
 - a. Switch to Microsoft Visual Studio.
 - b. On the menu bar, select the **Stop Debugging**  icon.
 - c. On the right in Solution Explorer, expand **StaticViews** and then select **CandidateFeedback.html**.
 - d. Review the html, style sheet, and JavaScript used in the file.
- 2. Review how channel tabs and personal tabs can invoke a module**
 - a. In Solution Explorer, under **StaticViews**, select **TeamTab.html**.
 - b. Scroll down to line 268.
 - c. Review the code located between lines 268 to 272.
This code shows how to invoke a task module from a tab using `microsoftTeams.tasks.startTask()` from the tab and passing a `TaskInfo` object.
- 3. Review how bots can invoke a module**
 - a. In Solution Explorer, expand **TypeConverters** and then select **CandidateToAdaptiveCardTypeConverter.cs**.
 - b. Scroll down to line 102.
 - c. Review the code located between lines 102 to 108.
This example shows how you can use the `Action.OpenUrl` card action for Adaptive cards to invoke task module
- 4. Review how messaging extensions can invoke a module**
 - a. In Solution Explorer, expand **Services > MessagingExtension** and then select **MessagingExtensionActionsService.cs**.
 - b. Scroll down to line 45.
 - c. Review the code located between lines 45 to 62.
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.
- 5. Review how modules can be invoked outside of Teams (Deep linking)**

- a. In Solution Explorer, under **StaticViews**, select **CandidateFeedback.html**.
- b. Scroll down to line 189.
- c. Review the code located between lines 189 to 204.
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.



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

a. In Visual Studio, in Solution Explorer, expand **Services** and then select **BotService.cs**.

b. Scroll down to line 91.

c. Review the code located between lines 91 to 106.

This code demonstrates how to capture a conversation reference and then using it later to initialize outbound (proactive) messages.

d. In Solution Explorer, under **Services**, select **NotificationService.cs**.

e. Scroll down to line 74.

f. Review the code located between lines 74 to 112.

This example shows how to construct a message and send the notification to a user.



More information about task modules 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 in to Graph Explorer**
 - a. Switch to Microsoft Edge and open a new tab.
 - b. Browse to T <https://developer.microsoft.com/en-us/graph/graph-explorer>.
 - c. In the left navigation, under **Authentication**, select **Sign in with Microsoft**.
 - d. In the **Pick an account** dialog box, select **[USERNAME]**.
 - e. If prompted for a password, enter T **[PASSWORD]**.
 - f. 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.

 - a. In the left navigation, under **Authentication**, select **modify permissions**.
 - b. In the **Modify Permissions** dialog box, select the check box for the following permissions:
 - **Group.Read.All**
 - **Group.ReadWrite.All**
 - **User.Read.All**
 - **User.ReadWrite.All**
 - c. Review the information and then select **Modify Permissions**.
You will be signed out.
 - d. In the **Pick an account** dialog box, select **[USERNAME]**.
 - e. If necessary, enter T **[PASSWORD]** and then select **Sign in**.
 - f. 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**
 - a. In the left navigation, under **Sample Queries**, select **show more samples**.

- b. 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**.
- c. Close the Sample Categories pane.

4. Retrieve an existing Microsoft Teams Id

- a. In the left navigation, under **Microsoft Teams**, select **GET my joined teams**.
- b. Locate the **HR Hiring** Team.
- c. 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  
},
```

- d. Right-click or tap and hold the id and then select **Copy**.
Do not select the quotation marks.

- e. 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

- a. In the left navigation, under **Microsoft Teams**, select **POST create channel**.
- b. At the top of the page, under the **Query** box, select **Request Headers**.
- c. 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**.
- d. At the top of the page, in the **Query** box, locate **{team-id}**.
- e. Replace **{team-id}** with T **<HRHiringTeamId>**.
- f. Under the query, select **Request Body**.
- g. On the Request Body tab, locate **displayName**.
- h. Replace the existing display name with T **HR Recruitment**.
- i. Replace the existing description with T **This channel is where we are keeping track of our recruitment trends**.

j. 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"  
}
```

k. Select **Run Query**.

l. 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

a. At the top of the page, to the left of the **Query** box, select the **v1.0** menu and then select **beta**.

b. In the left navigation, under **Microsoft Teams (beta)**, select **POST create chat thread**.

c. In the middle of the page, review the information banner.

d. At the top of the page, to the left of the **Query** box, select the **POST** menu and then select **GET**.

e. In the **Query** box, enter

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

f. Select **Run Query**.

g. Locate the **HR Recruitment** channel.

h. 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  
}
```

i. Right-click or tap and hold the id and then select **Copy**.

Do not select the quotation marks.

j. 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.

k. At the top of the page, to the left of the **Query** box, select the **GET** menu and then select **POST**.

l. In the **Query** box, enter

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

m. Select **Run Query**.

- n. 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**
 - a. Switch to the Microsoft Teams desktop app.
 - b. In the left navigation, select **Teams**.
 - c. Under **HR Hiring**, select **1 more channel > HR Recruitment**.
 - d. In the Conversations tab, notice the **Hello world** chat.
- 8. Review Graph API services configurations**
 - a. Switch to Visual Studio.
 - b. In Solution Explorer, expand **Services** and then select **Graph ApiService.cs**.
 - c. Scroll down to line 30.
 - d. Review the code located between lines 30 to 51.

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.
 - e. Locate line 48.

This creates channels in the team.
 - f. Scroll down to line 59.
 - g. Review the code located between lines 59-67.

This code installs an app to the team.
 - h. Locate lines 69.
 - i. Review the code located between lines 69-80.

This adds and configures a tab to the team's recently created channel.



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

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 also to reuse your existing investments in SharePoint.

[< Table of Contents](#)

