

# Oracle Fusion Cloud Sales Automation

---

**How do I integrate Microsoft Teams  
with Redwood User Experience?**



Oracle Fusion Cloud Sales Automation  
How do I integrate Microsoft Teams with Redwood User Experience?

F89626-02

Copyright © 2024, Oracle and/or its affiliates.

Author: Oracle

# Contents

## **Get Help** i

---

<b>1</b>	<b>Get Started with Integrating Microsoft Teams with Redwood User Experience</b>	<b>1</b>
	Overview of Microsoft Teams Collaboration	1
	Integrate Redwood UX with Microsoft Teams	1
	Create the Microsoft Azure Application	3
	Register the Microsoft Azure Application in Oracle Sales	4
	Register a REST Web Service Connection	5
	Create a Custom Formula Field	6
	Create an Object Function Server Script	8
	Configure Smart Action Functions	10



# Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

## Get Help in the Applications

Use help icons  to access help in the application. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons.

## Get Support

You can get support at [My Oracle Support](#). For accessible support, visit [Oracle Accessibility Learning and Support](#).

## Get Training

Increase your knowledge of Oracle Cloud by taking courses at [Oracle University](#).

## Join Our Community

Use [Cloud Customer Connect](#) to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest [ideas](#) for product enhancements, and watch events.

## Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the [Oracle Accessibility Program](#). Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

## Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to [oracle\\_fusion\\_applications\\_help\\_ww\\_grp@oracle.com](mailto:oracle_fusion_applications_help_ww_grp@oracle.com).

Thanks for helping us improve our user assistance!



# 1 Get Started with Integrating Microsoft Teams with Redwood User Experience

## Overview of Microsoft Teams Collaboration

Salespeople who are members of a sales team need to collaborate with each other, and with other users outside their team, when pursuing an opportunity in their Sales application.

Microsoft Teams is a collaboration tool that enables salespeople to easily collaborate among sales team members to further advance their sales pipeline.

### Oracle Fusion Applications Integration with Microsoft Teams

Oracle provides a sample solution to integrate with Microsoft Teams that works consistently across Fusion applications such as Oracle Procurement and Oracle Sales. For example, salespeople can create a team in Microsoft Teams for Oracle Fusion business objects, such as requisitions for Oracle Procurement and opportunities for Oracle Sales, and view Microsoft Teams conversations within Fusion application records. See [Oracle Fusion Applications Integration with Microsoft Teams](#) which provides more details about the solution as well as the next steps to take.

**Note:** Oracle recommends to use this preferred solution to integrate with Microsoft Teams across Fusion applications.

### Oracle Sales Extensibility Solution for the Integration with Microsoft Teams

If you don't require the ability to view conversations within the Oracle Sales interface, then Oracle Sales has an alternative solution that allows salespeople to simply click the Create Team in Microsoft Teams action from the opportunity page to create a dedicated team in Microsoft Teams. Once the team is created, salespeople can then use the Go to Microsoft Teams action to navigate to the team and collaborate with others to pursue the opportunity. With this solution, salespeople can start to collaborate directly in Microsoft Teams for the opportunity and are provided with an option to easily navigate to it.

**Note:** This playbook provides the setup steps required to implement this solution and configure the Create Team in Microsoft Teams and Go to Microsoft Teams user actions.

## Integrate Redwood UX with Microsoft Teams

As an administrator, you manage the Oracle Microsoft Teams integration for users using Sales for Redwood user experience. This topic explains how to enable the Microsoft Teams integration to create a team in Microsoft Teams for an opportunity. However, you can also use these steps to create a team and access Microsoft Teams for custom objects as well.

**Note:** The steps to setup the integration for custom objects are very similar, except when creating the formula field and object functions. The sample opportunity setup scripts provided contain **Optyld**. This value must be replaced with the ID field for the custom object in all the sample scripts.

Here are the high-level steps to set up the integration of Oracle Sales Redwood UX and Microsoft Teams:

1. Create the Microsoft Azure application.
2. Register the Microsoft Azure application in Oracle Sales
3. Register a REST Web Services connection.
4. Create a custom formula field.
5. Create an object function server script.
6. Define the **Create Team in Microsoft Teams** smart action.
7. Define the **Go to Microsoft Teams** smart action.

## Before You Start

You need two users to complete the setup:

- A setup user in Sales
- An administrator user for the Microsoft Azure

For this integration to work, make sure:

- Your Oracle Sales Redwood UX supports fragments.
- That the email address in the resource record for the user using Sales for Redwood UX matches the user's email address in Microsoft Teams.
- The opportunity team owner must be a Microsoft Teams user and also be the owner of the team in Microsoft Teams.
- To add the opportunity team members to the resource team of the record before the Microsoft Teams team is created. This ensures that these resources can be automatically added to the team in Microsoft Teams when the team is created.

## Enable the Archive Option for Microsoft Teams

After you've successfully created the Microsoft Azure application and registered the app in Oracle Sales, follow these steps to enable the archive option so that when an opportunity closes, the corresponding Microsoft Team gets archived automatically.

1. In the **Setup and Maintenance** work area, go to:
  - Offering: Sales
  - Functional Area: Integrations
  - Task: Manage Microsoft 365
2. On the Manage Microsoft 365 page, go to the Microsoft Teams tab.
3. Select the **Archive Opportunity Team in Microsoft Teams Enabled** check box.
4. Click **Save and Close**.



## Create the Microsoft Azure Application

To create the Microsoft Teams add-in, you must first create the Microsoft Azure application. To get your host URL follow these steps:

1. Sign in to the Microsoft Azure portal at <https://portal.azure.com>.
2. Click **Admin** on the home page.

The Admin center page appears.

3. Click **Microsoft Entra ID** under Azure Services.

The Microsoft Entra ID Admin Center page appears.

**Note:** If you're a first-time user, then you must sign up for your Microsoft Entra ID.

4. Click **App registrations**.

The Applications list page appears.

- a. Click **+ New registration**.

The Register an application dialog box appears.

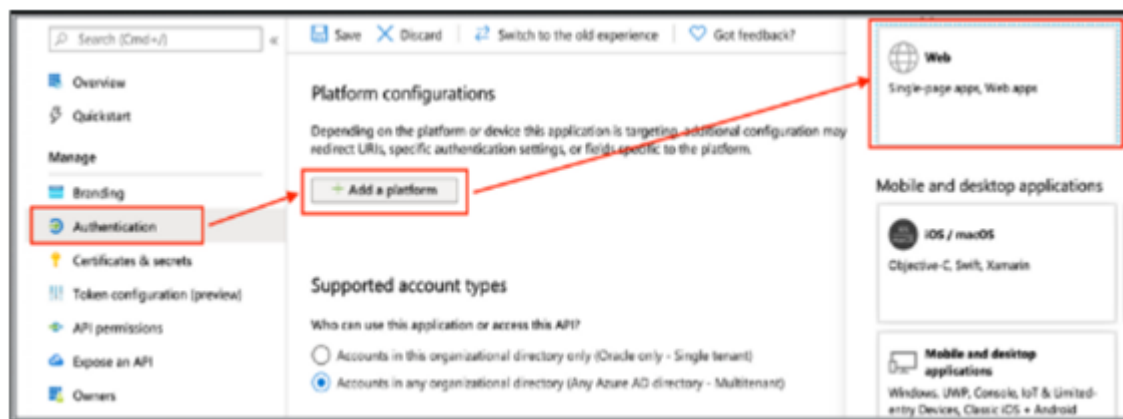
- b. For Name, enter a name for the application, for example, Microsoft Teams Collaboration.
- c. Under Supported account types, select **Accounts in this organizational directory only (Oracle only - single tenant)**.
- d. Leave Redirect URI (optional) blank.
- e. Click **Register** to create the Microsoft Teams add-in application.

The application appears on the App registrations page.

The Application details page displays the Application (client) ID and Directory (tenant) ID.

**Note:** Keep a note of these values because you insert these in the Manage Microsoft 365 task page later on. See the *Register the Microsoft Azure Application in Oracle Sales* section for more information.

5. Next click **Authentication** and from the Authentication page, click **Add a platform** and select **Web** as shown in this screenshot:



6. In the **Redirect URIs** section of the Configure Web dialog box, add the URI of your environment (for example, `https://sample-pod.oraclecloud.com`).
7. In the Implicit grant section, select **Access tokens** and **ID tokens**.
8. Click **Configure**.
9. Next click **Certificates & secrets**, and from the Add a client secret section, click **+ New client secret**.
10. Select a duration for the expiration of the Client Secret. For example, if you select **12 months**, the Client Secret becomes invalid after 12 months, and the automatic refresh stops. The sync process will start only after you sign in to the client again.  
  
If you select **Custom**, enter the start and end dates, which must be within 24 months. After the end date, you must sign in to the client again, to start the sync.
11. Click **Save**.  
  
The application generates a value for the key. Keep a note of this value because you'll use this as the Application Key in the Manage Microsoft 365 task page later on.
12. Click **API permissions** and click **Add a permission**. The Request API permissions pane appears.
13. Scroll to the end of the pane and click **Microsoft Graph Commonly used Microsoft APIs**.
14. Click **Application permissions** and select the following permissions:
  - o `Directory.Read.All`
  - o `Directory.ReadWrite.All`
  - o `Group.Read.All`
  - o `Group.ReadWrite.All`
15. Click **Add permissions**.
16. Click **Grant admin consent for Oracle** and then click **Yes** on the confirmation dialog box.

## Register the Microsoft Azure Application in Oracle Sales

In Oracle Sales, use the Manage Microsoft 365 task to register the Microsoft application you previously set up in the Microsoft Azure Directory.

1. Sign in as a setup user or as the sales administrator.
2. Navigate to:
  - o Offering: Sales
  - o Functional Area: Integrations
  - o Task: Manage Microsoft 365
3. Click the **Microsoft Teams** tab.
4. The first time you set up the integration, a usage agreement notice displays for your information:

**Usage Agreement:** This Oracle-Microsoft integration or plug-in may enable you to link to or transmit your content or third-party content to, or otherwise access, or retrieve content from, Microsoft and its platforms, users, or services. Oracle doesn't control and isn't responsible for Microsoft sites or platforms or services, the performance or availability of the services, or any content received or sent. You bear any and all risks associated with access to and use of Microsoft sites, platforms, and services and are solely responsible for entering into and compliance with separate terms between you and Microsoft. Oracle isn't responsible for the security, protection or confidentiality of such content (including obligations in the Hosting and Delivery Policies and Data Processing Agreement and Oracle's Privacy Policy) that's transmitted to such Microsoft sites or platforms or services. You're solely responsible for obtaining or having any required consents or other legal basis for your

use of this integration or plug-in. Oracle reserves the rights to terminate your connection or integration if your use of this integration or plug-in violates the terms of your Agreement(s) with Oracle.

5. Enter the Application ID, the Application Key, and the Tenant ID. See [Create the Microsoft Azure Application](#) for information about how to generate these details.

Let's say you enter the name as Oracle Microsoft Teams Integration. When you sign in to your Microsoft 365 account, you can see Oracle Microsoft Teams Integration as an add-in that you can select. Here's a sample screenshot:

**Set Up Microsoft Teams Integration** ⓘ

Enter the application ID and application key of the application registered in Azure Active Directory.

\* Application ID

\* Application Key

\* Tenant ID

Archive Opportunity Team in Microsoft Teams Enabled

## Register a REST Web Service Connection

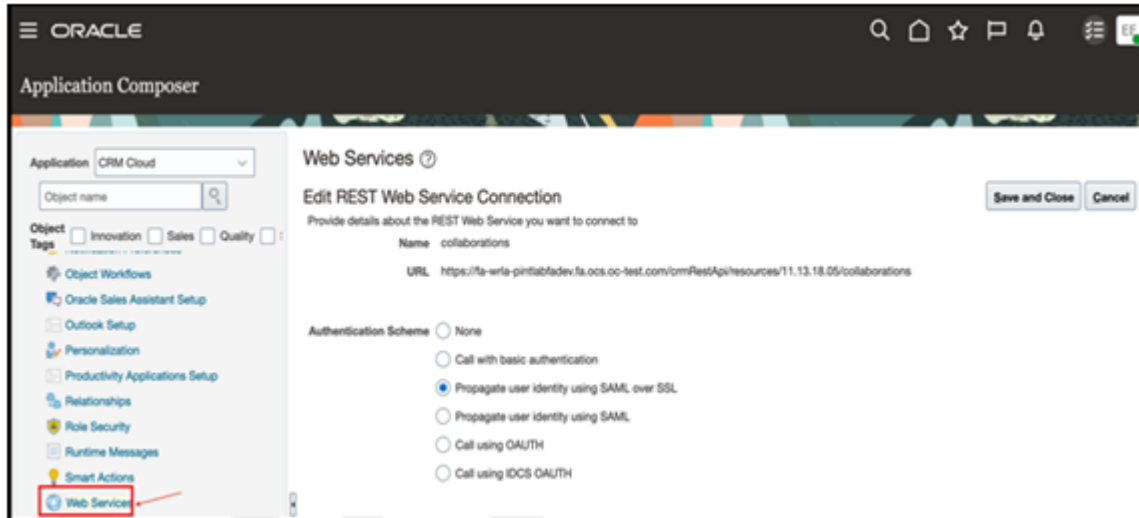
To register a web service for use in your scripts, you must first select Web Services in the Common Setup pane in Application Composer.

The collaboration API is used to determine if a Microsoft Team is already created for the opportunity for the **Create Team in Microsoft Teams** action. It is also used to retrieve the Microsoft Teams URL for the team associated with the opportunity for the **Go to Microsoft Teams** action.

Complete the following steps to register a REST Web Service Connection to use for your Sales for Redwood UX integration with Microsoft Teams.

1. Navigate to **Configuration > Sandboxes** to create and enter a sandbox with Application Composer.
2. From the sandbox, open Application Composer from the **Tools** menu.
3. In Application Composer, select **Common Setup**, and click **Web Services**.
4. Click **REST** to create a new connection. The **Edit REST Web Service Connection** page displays.

5. Provide the details about the REST Web Service to connect to as per the following:
  - a. **Name:** collaborations
  - b. **URL:** https://<env>/crmRestApi/resources/11.13.18.05/collaborations  
(replace <env> with your environment URL, such as http://localhost:3000-test.com/crmRestApi/resources/11.13.18.05/collaborations)



6. In the **Authentication Schema**, select **Propagate user identity using SAML over SSL**.
7. Configure the GET method as follows:
  - o Request Payload: **Default (Schema URL)**
  - o Response Payload: **Code Sample**

Select the **Code Sample** option and copy the following payload code.

```
{ "CollabId": "300100575968676", "CreatedBy": "MHoope", "CreationDate": "2023-07-24T03:38:27.103+00:00",  
  "https://teams.microsoft.com/l/team/19%3aH7m-L-14hQrN-J8Ych4c3fEAxb09AoknST_z4dV1ghY1%40thread.tacv2/  
  conversations?groupId=e512e688-d8b8-4a5a-b34e-8f508675e8aa&tenantId=33b71f2d-7f52-4370-  
  bd2c-08efb530b1c3", "ObjectId": "300100574254353", "ObjectName": "opportunities", "Name": "OPPTY  
  - Test Opty 000", "Description": "This team is specifically for the Test Opty 000 opportunity:  
  IBM", "TeamRelationshipName": null, "links": [ { "rel": "self", "href": "https://<env>/  
  crmRestApi/resources/11.13.18.05/collaborations/300100575968676", "name": "collaborations",  
  "kind": "item" }, { "rel": "canonical", "href": "https://<env>/crmRestApi/resources/11.13.18.05/  
  collaborations/300100575968676", "name": "collaborations", "kind": "item" } ] }
```

8. Configure the POST request and response to use **Default (Schema URL)**.
9. Click **Save and Close**.

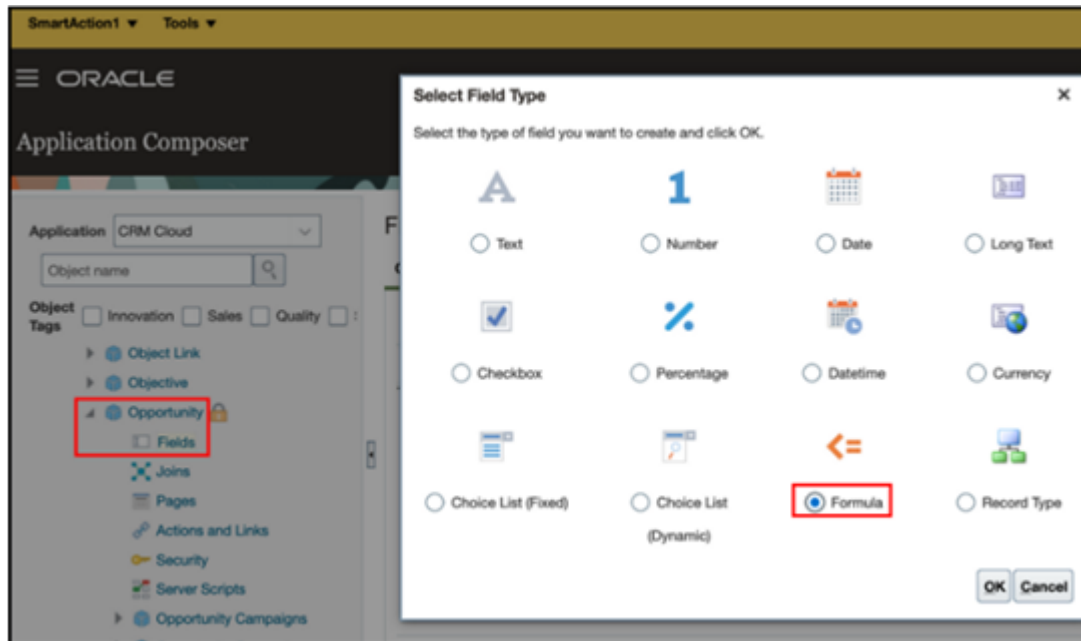
## Create a Custom Formula Field

A formula field is required to identify whether a Microsoft Teams already exists for the record. To create a formula field, follow these steps:

1. Navigate to **Configuration > Sandboxes** to create and enter a sandbox with Application Composer.
2. From the sandbox, open Application Composer from the **Tools** menu.
3. In Application Composer, expand the **Standard Objects** node, and then expand **Opportunity**.

**Note:** You can also use these steps to create a team and access Microsoft Teams for other objects such as custom objects as well.

4. Click **Fields** as shown in this example screenshot.



5. Click the **Custom Fields** tab.
6. On the Custom Fields tab, click **New**.
7. Select Formula as the field type with the following details.
  - a. **Formula Type:** Text
  - b. **Display Label:** microsoftTeamCreated
8. Click **Next**.
9. Copy and paste the following groovy code:

```
def conn = adf.webServices.collaborations;
try
{
    // Provide query parameter for the account object you want to receive
    def queryParams = ['finder':'GetCollaborationsForObjectFinder;ObjectId='+OptyId+',Validate=true']
    conn.dynamicQueryParams = queryParams
    def collabResponse = conn.GET()
    if (collabResponse.count == 0 ) {
        return "NO";
    }
    if (collabResponse.count > 0 ) {
        return "YES";
    }
    return "ERROR";
}
catch(Exception ex)
{
    println(conn.statusCode+"");
    println(conn.httpErrorResponse+"");
    return "ERROR";
}
```

}

**Note:** To set up the integration for custom objects you can copy the scripts provided but you must replace the Optyld value with the ID field for the custom object in all the sample scripts.

10. Click **Submit**.

## Create an Object Function Server Script

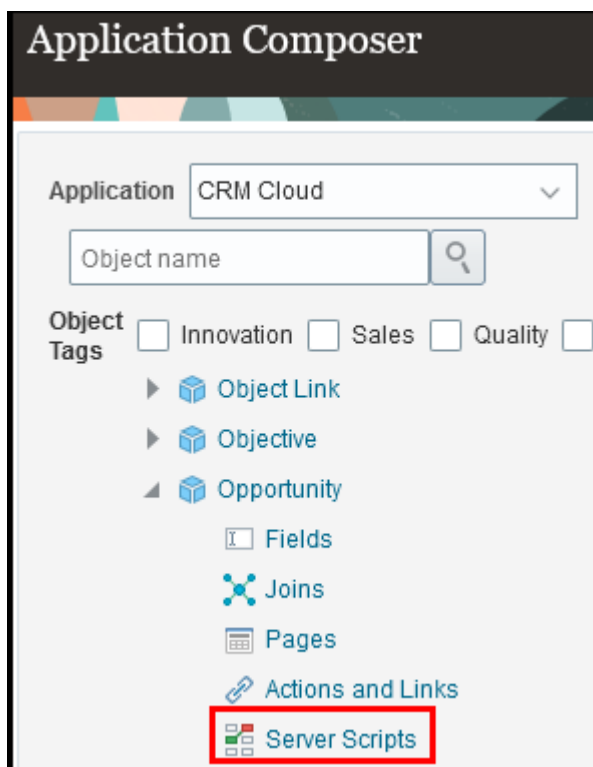
As an administrator, you must define the object function server script to be used by the Go to Microsoft Teams smart action to retrieve the Microsoft Teams URL associated with the record.

Here's how to define the GoToTeams function server script in Application Composer.

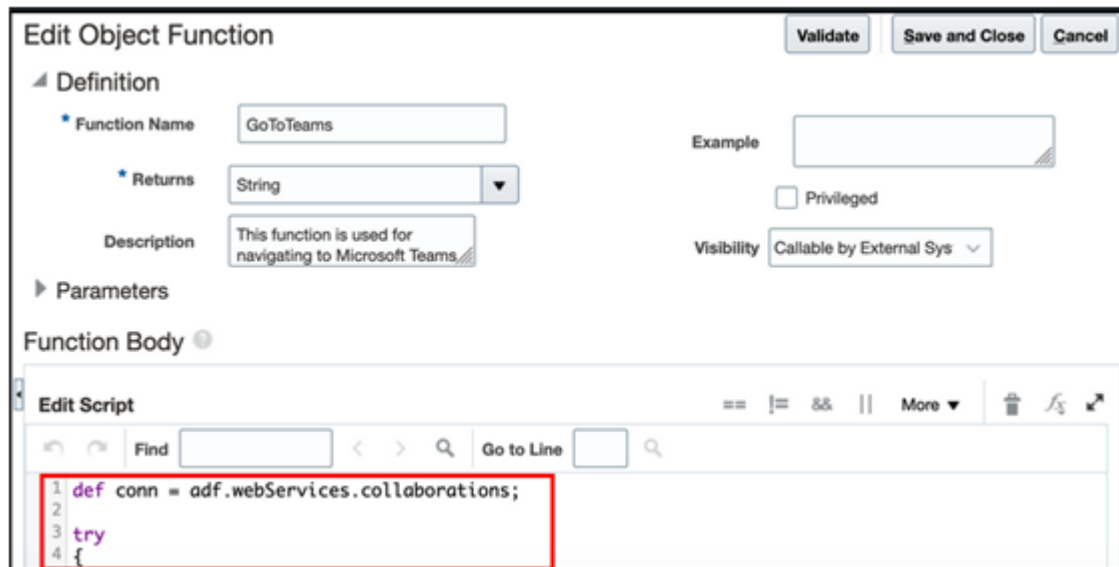
1. Navigate to **Configuration > Sandboxes** to create and enter a sandbox with Application Composer.
2. From the sandbox, open Application Composer from the **Tools** menu.
3. In Application Composer, expand the **Standard Objects** node, and then expand **Opportunity**.

**Note:** You can also use these steps to create a team and access Microsoft Teams for other objects such as custom objects as well.

4. Click **Service Scripts** as shown in this example screenshot.



5. Click the **Object Functions** tab to display a screen similar to the following:



The screenshot shows the 'Edit Object Function' interface. At the top right are buttons for 'Validate', 'Save and Close', and 'Cancel'. The 'Definition' section includes fields for 'Function Name' (GoToTeams), 'Returns' (String), and 'Description' (This function is used for navigating to Microsoft Teams). There is also an 'Example' field, a 'Privileged' checkbox, and a 'Visibility' dropdown (Callable by External Sys). The 'Parameters' section is collapsed. The 'Function Body' section contains a code editor with the following Groovy code:

```
1 def conn = adf.webServices.collaborations;
2
3 try
4 {
```

6. Enter required details as follows:
- **Function Name:** GoToTeams
  - **Returns:** String
  - **Visibility:** Callable by External Systems

7. Enter the following groovy sample code:

```
def conn = adf.webServices.collaborations;
try
{
def queryParams = ['q':'ObjectId='+OptyId]
conn.dynamicQueryParams = queryParams
def collabResponse = conn.GET()
def weburl = ""
if (collabResponse.count > 0 ) {
weburl = collabResponse.items[0].Weburl;
}
return weburl;
}
catch(Exception ex)
{
println(conn.statusCode+""); //for diagnostic logging
println(conn.httpErrorResponse+""); //for diagnostic logging
throw ex;
}
```

**Note:** You can copy the scripts provided to set up the integration for custom objects, but you must replace the **OptyId** value with the ID field for the custom object in all the sample scripts.

8. Click **Validate**.
9. Click **Save and Close**.

## Configure Smart Action Functions

Salespeople can define a smart action to take them directly to Microsoft Teams by entering a smart action in the Action Bar from their Opportunity page. This action will only be applicable if a team already exists for the selected opportunity record.

Salespeople can choose to create a team in Microsoft Teams by entering a smart action in the Action Bar from their Opportunity page in their sales application.

**Note:** While the example refers to the opportunity object, the following set up steps are similar for other objects such as custom object as well.

As an administrator, you must define the Create Team in Microsoft Teams and Go to Microsoft Teams smart actions in Application Composer.

### Define the Create Team in Microsoft Teams Smart Action

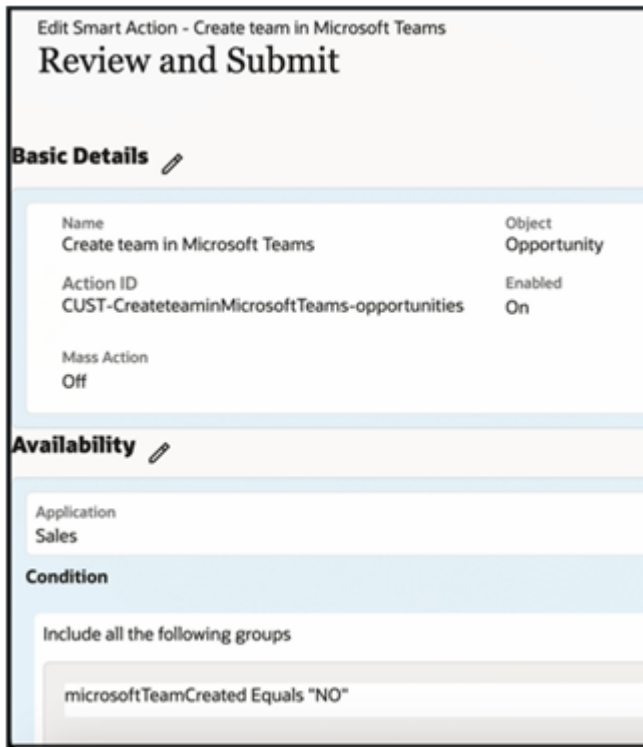
Here's how to define the Create Team in Microsoft Teams smart action function in Application Composer.

1. Navigate to **Configuration > Sandboxes** to create and enter a sandbox with Application Composer.
2. From the sandbox, open Application Composer from the **Tools** menu.
3. Click **Smart Actions**, available from **Common Setup**.
4. Fromn the **Kind of Actions** page, choose **REST-based action**, and then click **Continue**.



5. Create the smart action with the following **Basic Details**:
  - o **Name**: Create Team in Microsoft Teams
  - o **Object**: Opportunity
  - o **Enabled**: On
  - o **Mass Action**: Off
  - o For **Availability**, enter the following: **Application**: Sales
  - o For **Condition**, enter `microsoftTeamCreated equals "NO"` in the **Include all the following groups** field.

Here's an example of what the page looks like for the Opportunity object.



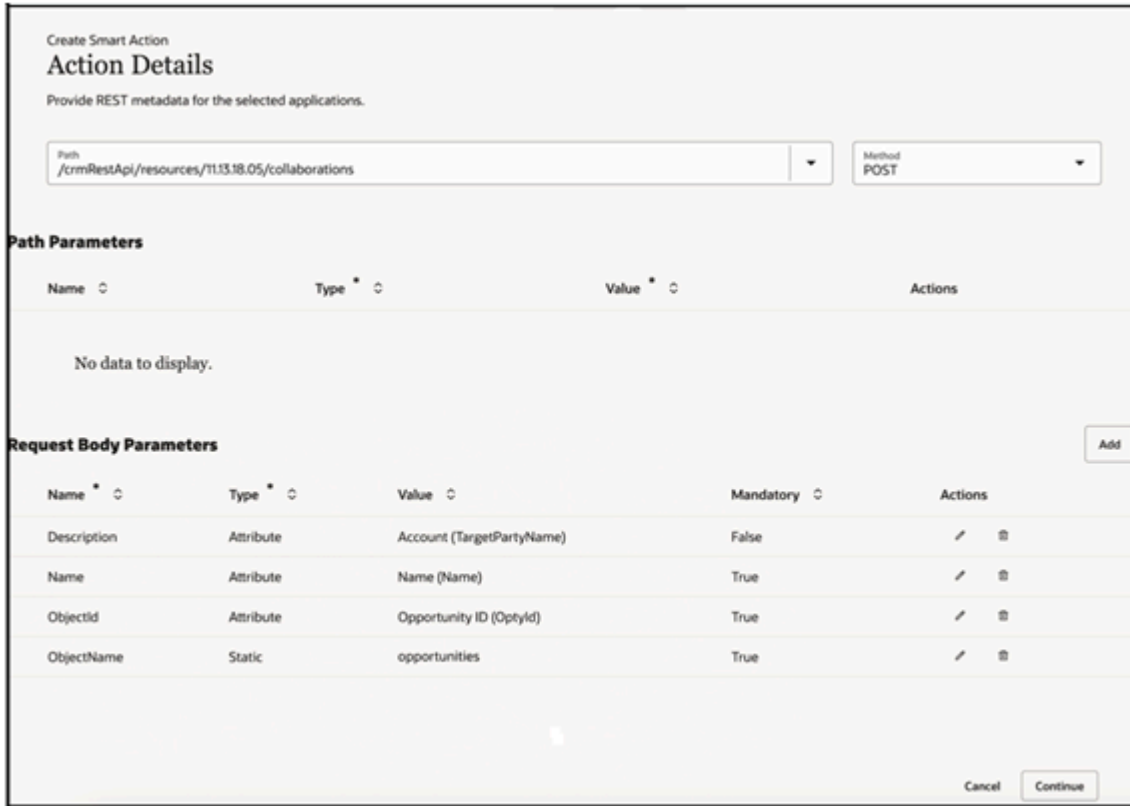
6. On the **Action Details** page, enter the REST metadata in the **Path** field of the selected smart action. / `crmRestApi/resources/11.13.18.05/collaborations` .
7. From the **Method** drop-down list, select **POST**.
8. Click **Add** and enter the following Request Body Parameters for the **Create Team in Microsoft Teams** smart action:

Object	Name	Type	Value	Mandatory
Opportunity	ObjectName	Static	Opportunities	True
	ObjectId	Attribute	Opportunity ID (OpptyId)	True
	Name	Attribute	Name(Name)	True
	Description	Attribute	Account(TargetPartyName)	False

Object	Name	Type	Value	Mandatory
	TeamRelationshipName	Static	<p>To get the value, follow these steps:</p> <ol style="list-style-type: none"> <li>In a sandbox, go to <b>Application Composer</b>.</li> <li>Click <b>Relationships</b>.</li> <li>Copy the name of the relationship for the <b>Source Object</b> = Object of interest and <b>Target Object</b> = Resource</li> </ol> <p>The TeamRelationshipName parameter value is used to retrieve the team members on the record. It is only required if you're using a custom relationship for the standard objects or if you're setting up this integration for a custom object.</p>	False
<b>Custom</b>	ObjectName	Static	<p>Use API name of custom object.</p> <p>To get the value, follow these steps:</p> <ol style="list-style-type: none"> <li>In a sandbox, go to <b>Application Composer</b>.</li> <li>Expand node, then Custom Objects node. <b>Objects</b> node, then <b>Custom Objects</b> node.</li> <li>Click the custom object you want.</li> <li>On the <b>Overview</b> page for the Custom Object, copy the <b>API Name</b> value to use as the ObjectName</li> </ol>	True
	ObjectId	Attribute	Custom object ID (ID field for the object)	True
	Name	Attribute	Name(Name)	True
	Description	Attribute	Account(TargetPartyName)	False
	TeamRelationshipName	Static	<p>To get the value, follow these steps:</p> <ol style="list-style-type: none"> <li>In a sandbox, go to <b>Application Composer</b>.</li> <li>Click <b>Relationships</b>.</li> <li>Copy the name of the relationship for the <b>Source Object</b> = Object of interest and <b>Target Object</b> = Resource</li> </ol>	False

Object	Name	Type	Value	Mandatory

Here's a sample screenshot of what the Create Smart Action UI looks like:



9. On the **Confirmation Message** page, you can define primary, secondary, and confirmation messages. Enter the following:
  - o **Primary Message:** Your team will be created in Microsoft Teams in a few seconds.
  - o **Secondary Message:** You can continue to use the application while it's being created. Please don't submit the action again.
  - o **Continue Button Label:** OK
  - o **Cancel Button Label:** Cancel
  - o **Confirmation Message:** Your team OPPTY-{ATTRIBUTE\_VALUE\_NAME} in Microsoft Teams is created. Use the Go to Microsoft Teams action to go to the team .
10. Check the details on the **Review and Submit** page.
11. Click **Submit**.

## Define the Go to Microsoft Teams Smart Action

Here's how to define the Go to Microsoft Teams smart action function in Application Composer.

1. Navigate to **Configuration > Sandboxes** to create and enter a sandbox with Application Composer.
2. From the sandbox, open Application Composer from the **Tools** menu.
3. Click **Smart Actions**, available from **Common Setup**.

4. Create the smart action with the following **Basic Details**:
  - o **Name**: Go To Microsoft Teams
  - o **Object**: Opportunity
  - o For **Availability**, enter the following: **Application**: Sales
  - o For **Condition**, enter `microsoftTeamCreated equals "YES"` in the **Include all the following groups** field.
5. On the **Action Details** page, for **Object function action** field, select **GoToTeams** which you previously defined in the Create an Object Function Server Script topic.
6. Select the **Object function returns a URL** toggle.
7. In the **Browser Window Target** field, select **New Tab** from the drop-down list as shown in this sample screenshot of what the Create Smart Action UI looks like.

The screenshot shows the 'Action Details' form with the following configuration:

- Object function action**: GoToTeams
- Object function returns a URL**:  (highlighted with a red box)
- Browser Window Target**: New tab (highlighted with a blue box)
- Request Body Parameters**: A table with columns Name, Type, Value, Mandatory, and Actions. The table is currently empty, displaying 'No data to display.'
- Buttons**: Cancel and Continue (highlighted with a blue box)

8. Click **Continue**.

9. Check the details on the **Review and Submit** page as shown:

The screenshot shows a 'Review and Submit' page for an action configuration. The page is divided into several sections:

- Basic Details**: A table with three columns: Name (GoToMicrosoftTeams), Object (Opportunity), and Action ID (CUST-GoToMicrosoftTeams-opportunities).
- Availability**: A section with a sub-section for 'Application' set to 'Sales'.
- Condition**: A section with a sub-section for 'Include all the following groups' containing a single condition: 'microsoftTeamCreated Equals "YES"'. The 'Submit' button at the bottom right is highlighted with a red box.

10. Click **Submit**.

Salespeople can now add all existing members of their opportunity to a team in Microsoft Teams and start the collaboration directly from the Opportunities UI. This allows users of Microsoft Teams to interact with opportunity team members from their Sales application and add any required external resources with expertise to help win the deal. Using the **Create Team in Microsoft Teams** action in the opportunity, a dedicated opportunity team is created in Microsoft Teams for the opportunity team members. Once the team is created, users can use the **Go to Microsoft Teams** action to begin collaboration with the team.

