Implementing OneDrive unified web picker SDK (JavaScript) – Preview

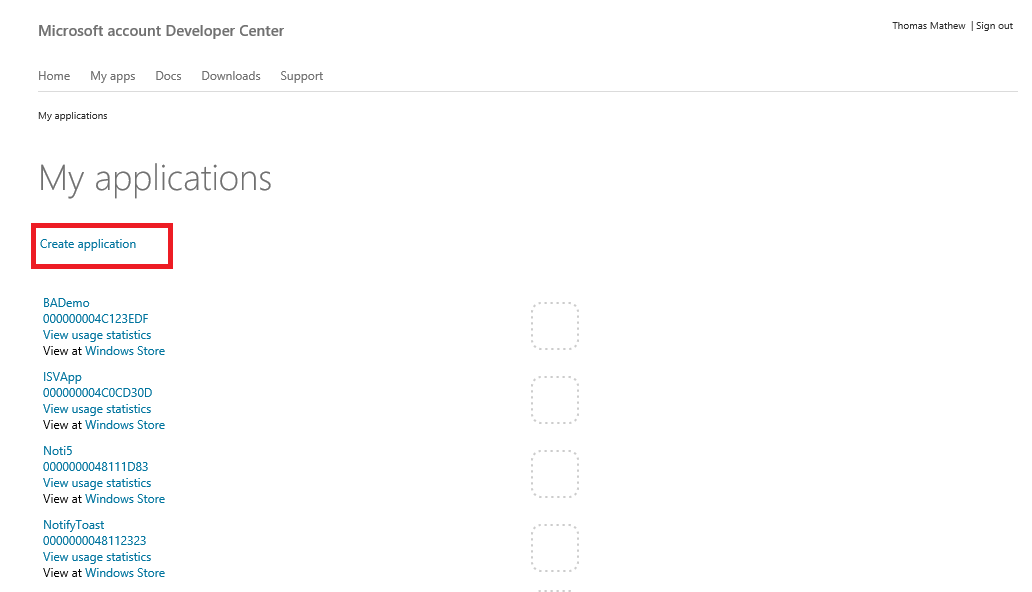
Step by Step Guide

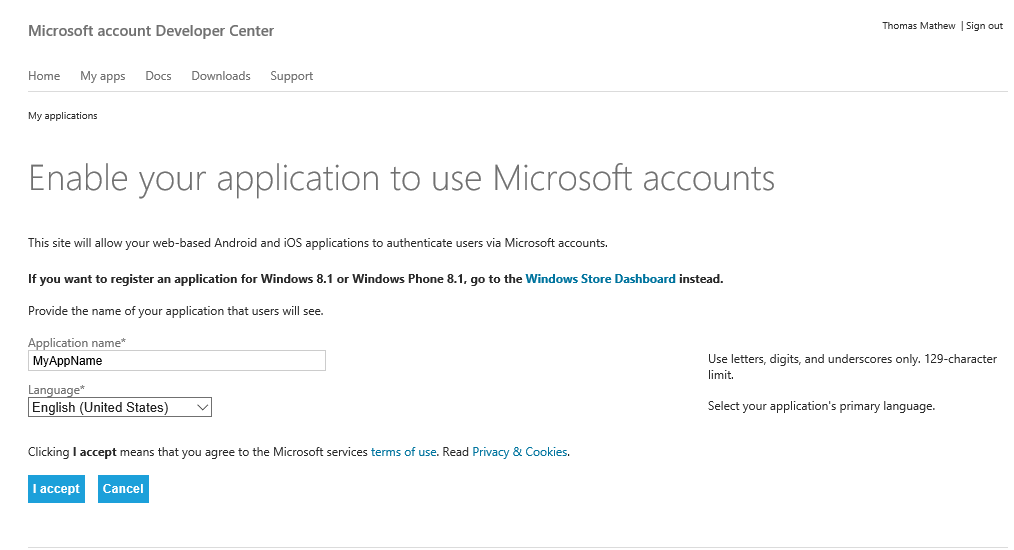
The OneDrive picker and saver SDK for web apps is the fastest way to integrate OneDrive into your web app. Quickly open and save files to OneDrive consumer and OneDrive for Business with just a few lines of code and no need to manage authentication or the user experience.

Note: The unified OneDrive web picker SDK (JavaScript) is in preview and may not work exactly as documented. Do not assume that the current behavior won't deviate from this documentation.

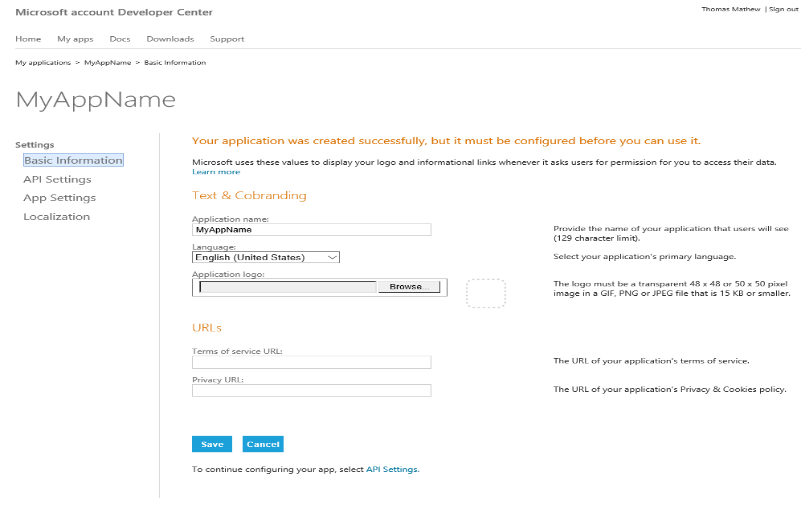
You can fine additional information here - <https://dev.onedrive.com/sdk/unified-javascript-picker-saver.htm>

**Setting up OneDrive Consumer**

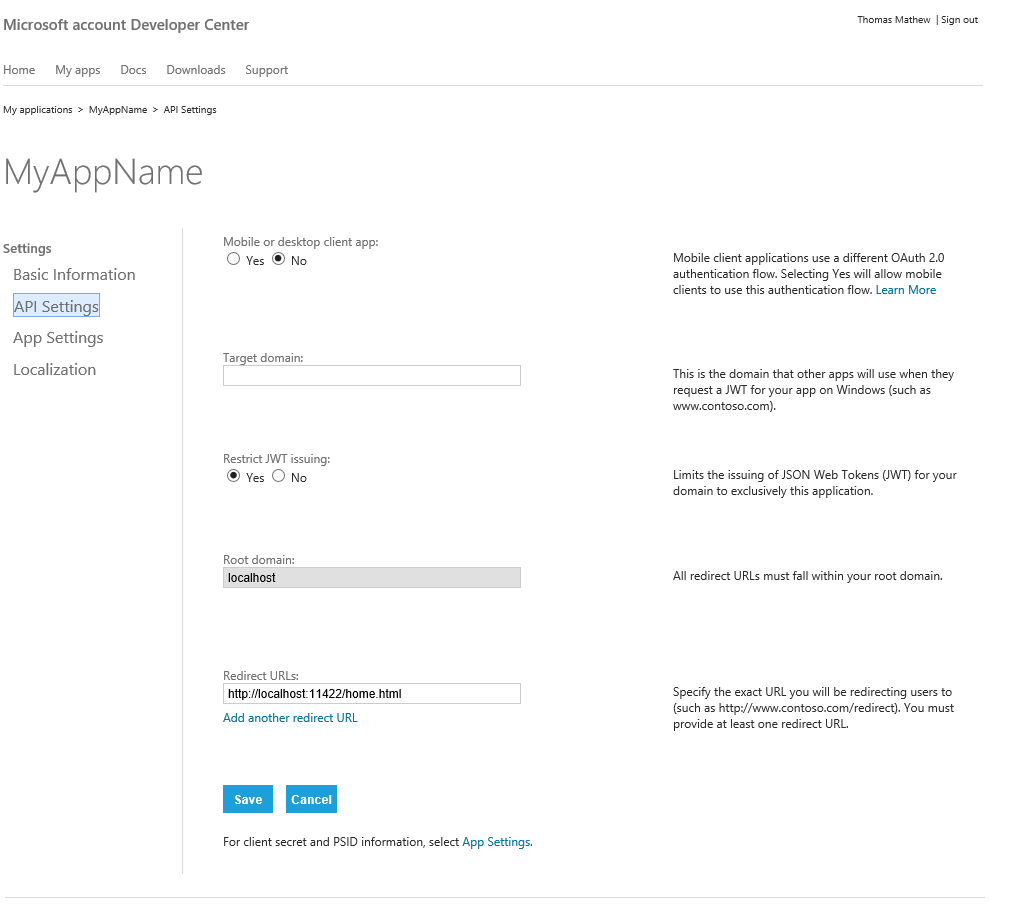
1. Register your App to get the Client ID
   1. Go to <https://account.live.com/developers/applications>
   2. Select Create Application
   3. Enter Application Name and select Language from the drop down and Click “I accept”

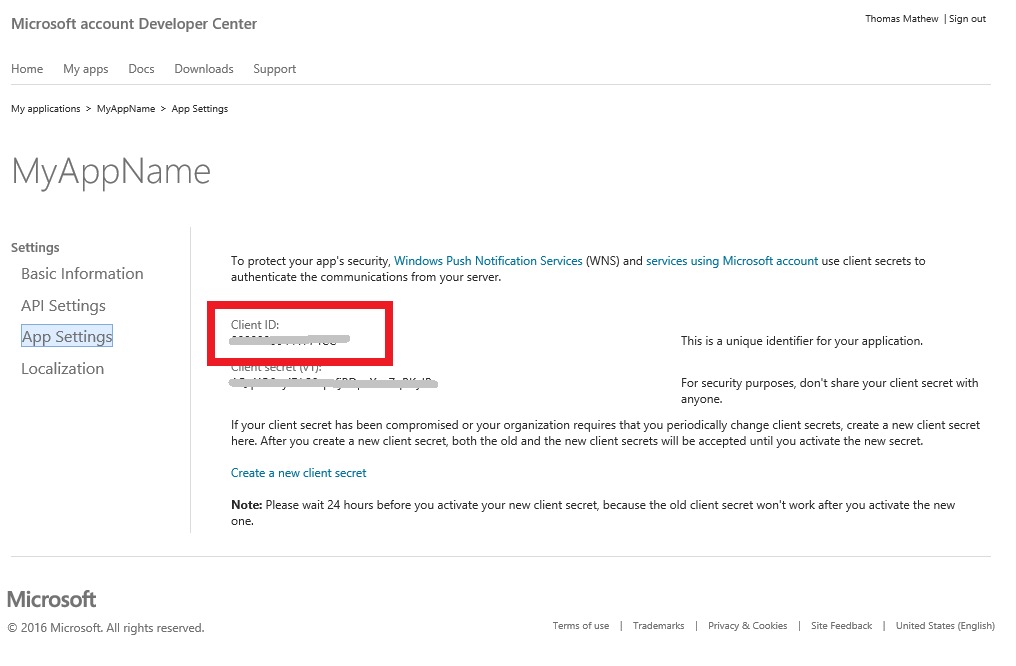


* 1. (Optional step) – Under Settings – Basic Information Tab - You can upload an Application Logo and enter the Terms of Service URL and Privacy URL for your application and hit Save button. OneDrive will display the Application Logo in the picker and saver experience.



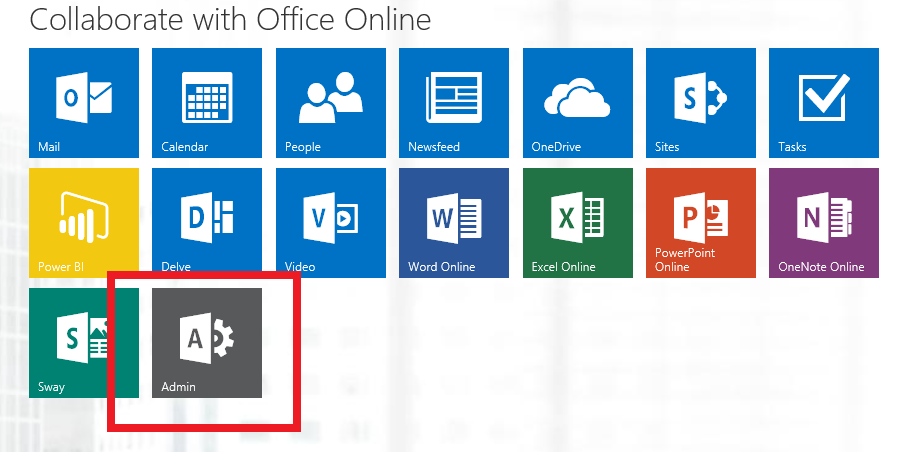
* 1. Under – API Settings tab, enter Root Domain and the Redirect URL fields. For the Redirect URL, use the URL for the web page that is going to reference the SDK.

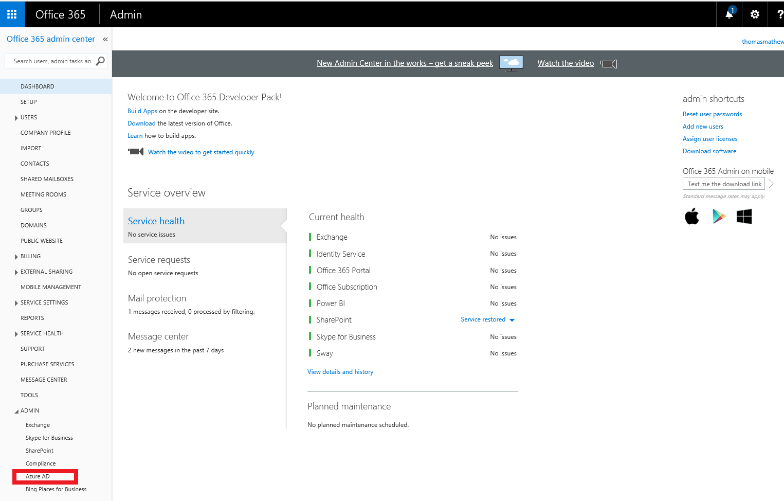


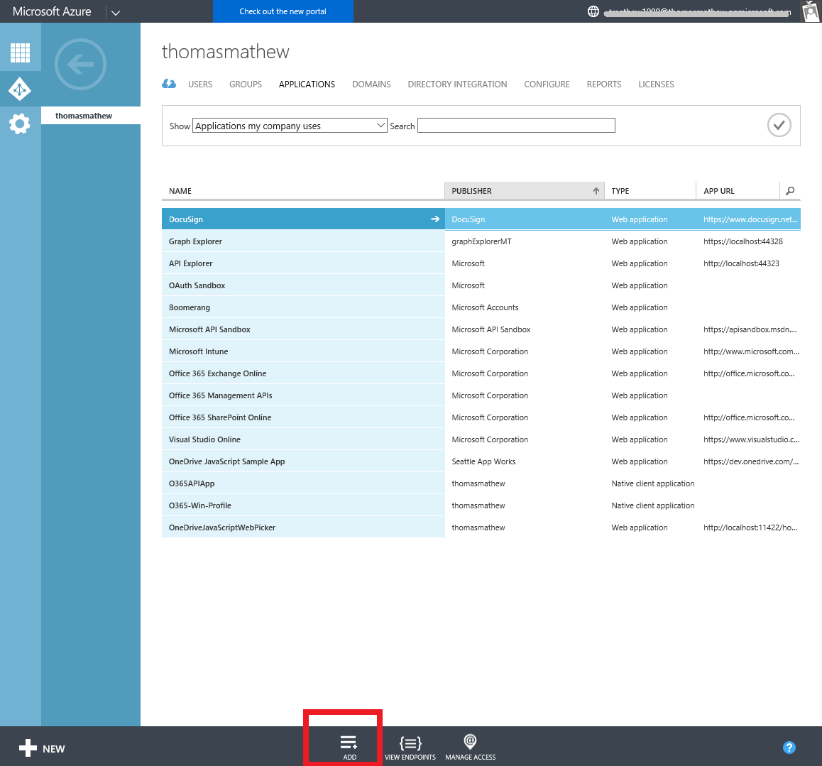
* + 1. Under App Settings, copy the Client ID value. We need this value in our application to enable access to OneDrive for Consumer. 

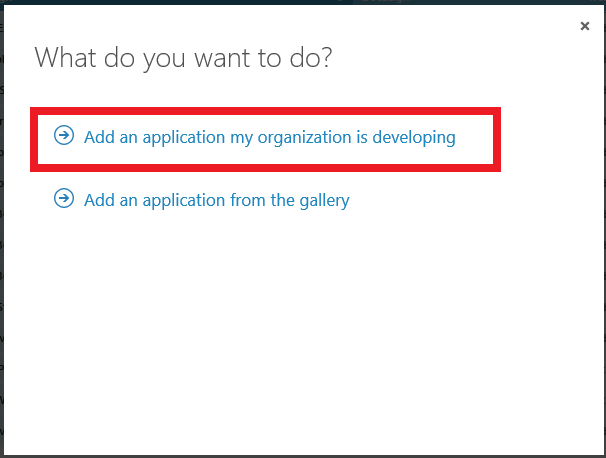
**Setting up OneDrive for Business**

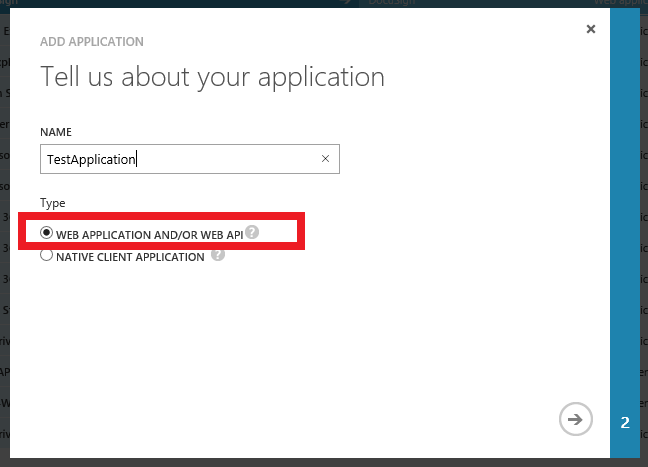
1. **Register You Application within Azure AD to get CLIENT ID**
   1. Sign in to your Office 365 tenant (<https://portal.office.com/home>) and go to the Office 365 admin center by selecting Admin as shown below

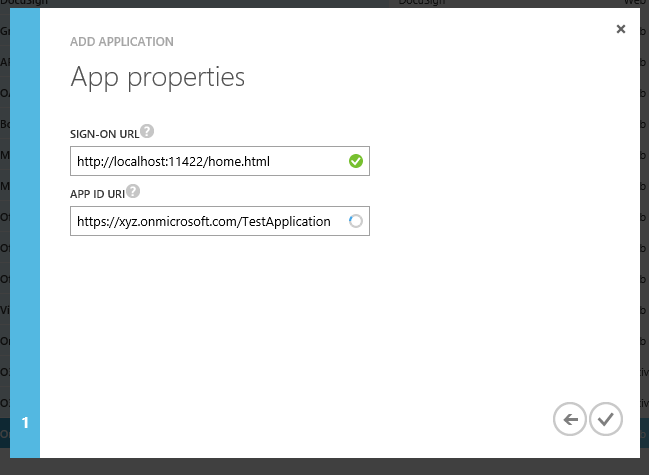
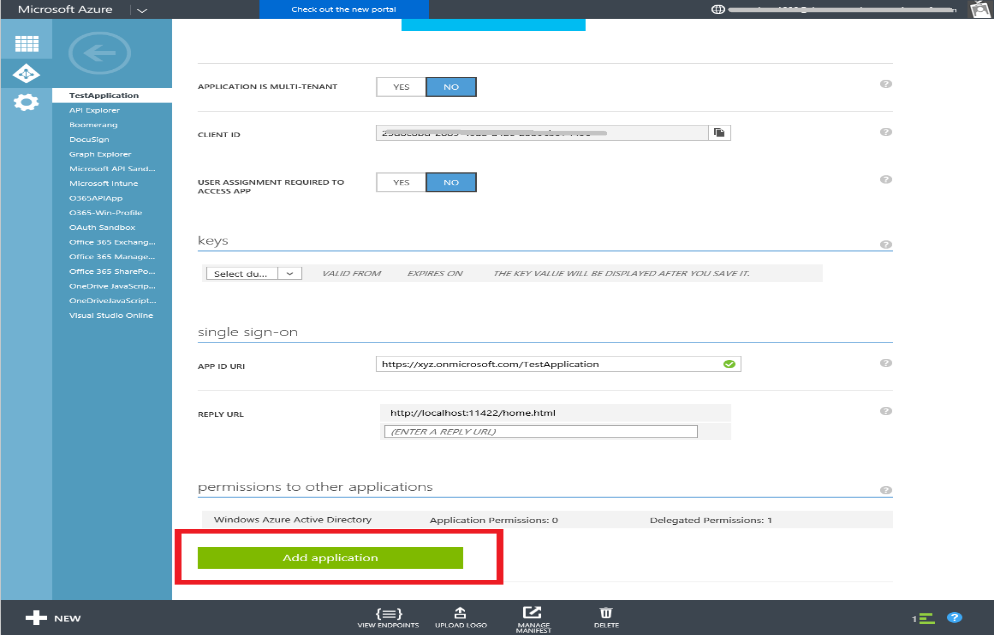
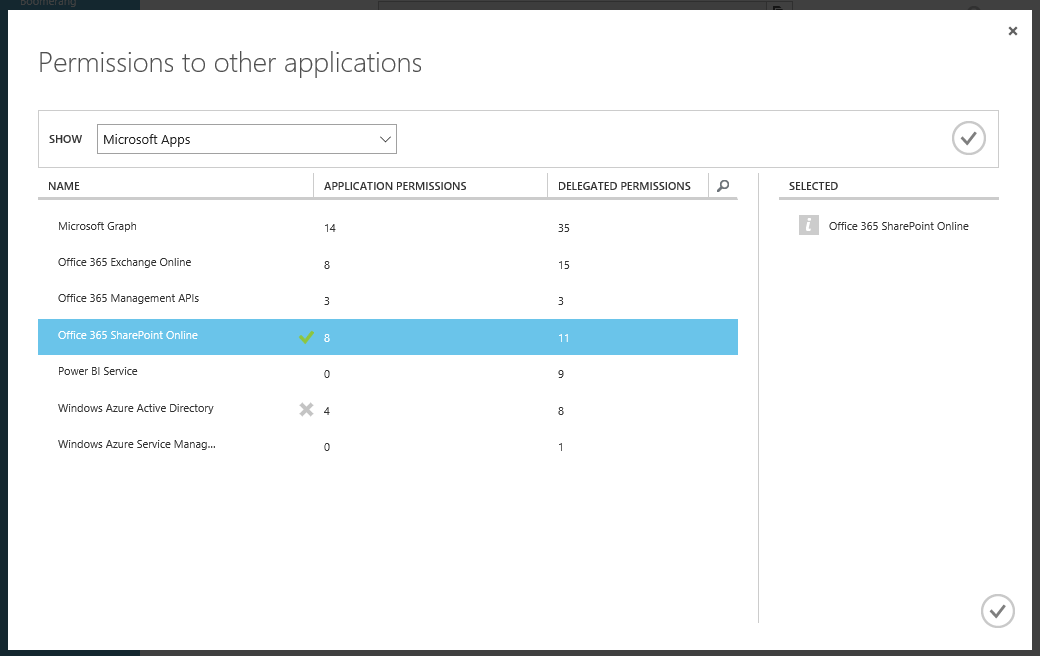


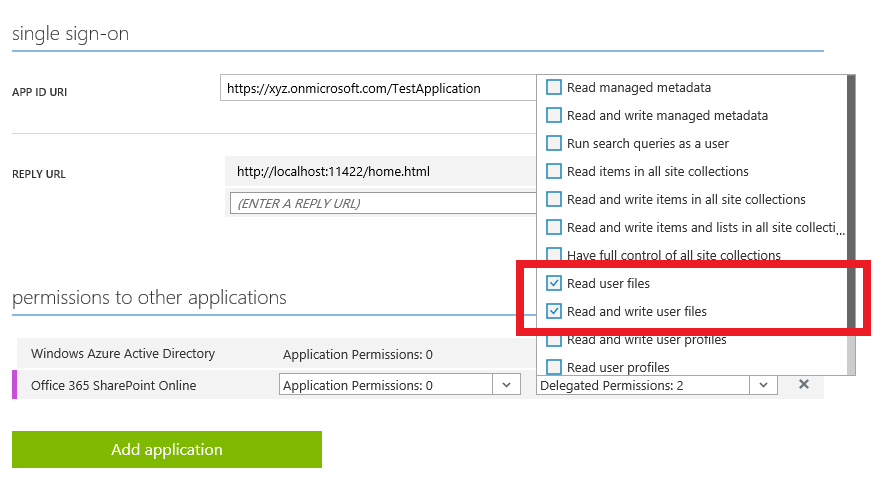
* 1. On the Office 365 Admin Center Screen, select Azure AD on the left menu as shown.
  2. 
  3. In the Azure Portal, click your Organizational Directory and select APPLICATIONS menu item from the top. Then Click ADD button from the Menu below to add a new Application.



* 1. Select “Add an application my organization is developing” when prompted.
  2. Enter Application Name and select Web Application API.



* 1. Enter SIGN-ON URL and APP ID URI and Complete button to complete the step. 
  2. Once the Application has been added, select the CONFIGURE menu item from the top.
  3. Scroll to the bottom of the page to the “permissions to other applications” section and click “Add Application”
  4. Select Office 365 SharePoint Online and select the Complete button
  5. Select “Read user files” **and/or** select “Read and write user files” from the list of Delegates Permissions. Your app has read permissions if your app only opens from OneDrive and write permissions if your app saves to OneDrive



* 1. Select SAVE bottom from the bottom menu.
  2. Ensure that the Reply URL is correct. The Redirect URL is the web page that is going to reference the SDK.
  3. Your app must be set to use the implict flow. You can do that by downloading your app's manifest, modifying it so that oauth2AllowImplicitFlow is set to true, and uploading it.
  4. Copy the CLIENT ID value. We need this value in our application to enable access to OneDrive for Business.

**Putting it all together**

1. In your web page referencing the OneDrive.js library enter the **OneDrive for Consumer CLIENT ID** and the **OneDrive for Business CLIENT ID** separated by a comma. The CLIENT IDs are copied from steps above.

<script type="text/javascript" src="https://js.live.net/v6.0/OneDrive.js" id="onedrive-js" client-id=*OneDriveConsumerClientID,OneDriveforBusinessClientID*"></script>

As an example

<script type="text/javascript" src="https://js.live.net/v6.0/OneDrive.js" id="onedrive-js" client-id="12345678445555XX,xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"></script>

1. As a sample try out the following code for a web page

<!DOCTYPE html>

<html>

<head>

<title> Test OneDrive</title>

<script type="text/javascript" src="https://js.live.net/v6.0/OneDrive.js" id="onedrive-js" client-id="12345678445555XX,xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"></script>

</head>

<body>

<script type="text/javascript">

var pickerOptions = {

success: function (files) {

// Handle returned file object(s)

alert("You picked " + files.values[0].fileName);

},

cancel: function () {

// Handle when the user cancels picking a file

},

error: function (e) {

// Handle when there is an error getting a link to the selected file

},

linkType: "downloadLink", // or webLink

multiSelect: false, // or true

openInNewWindow: true // or false

}

function launchOneDrivePicker() {

OneDrive.open(pickerOptions);

}

</script>

<h1>Open Files From OneDrive</h1>

<p>

<button onClick="javascript:launchOneDrivePicker();" title=""Open from OneDrive">

<img style="vertical-align: middle; height: 16px;" src="https://js.live.net/v5.0/images/SkyDrivePicker/SkyDriveIcon\_white.png">

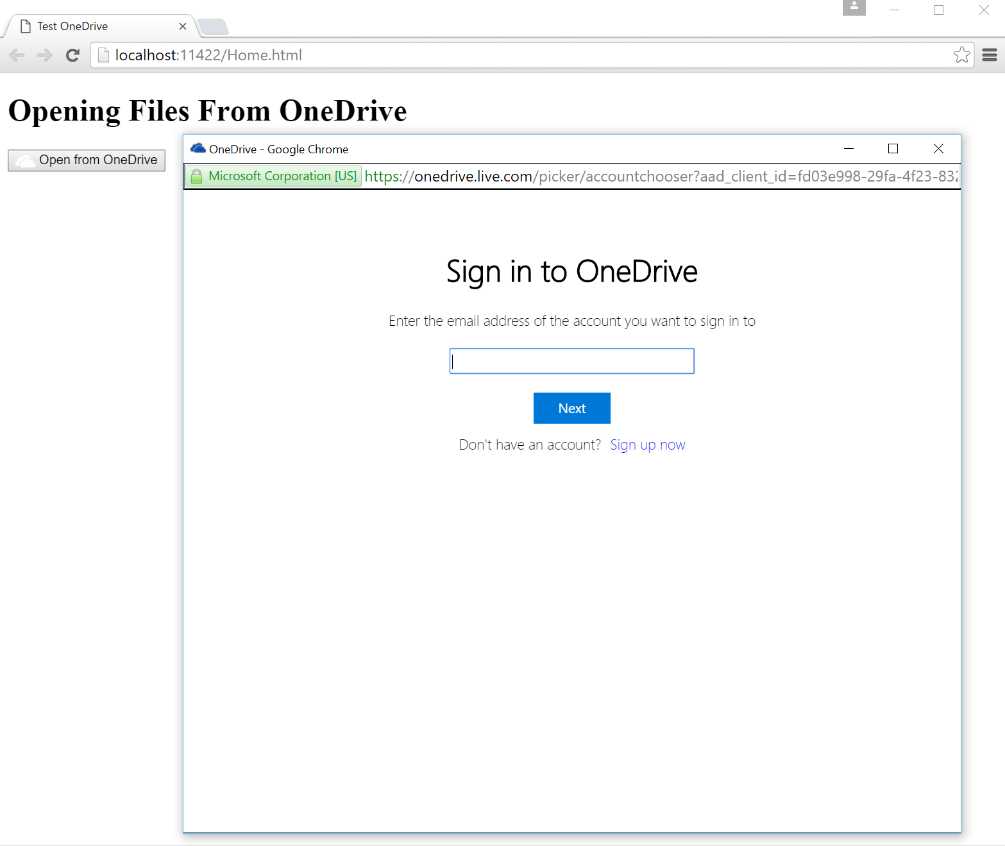
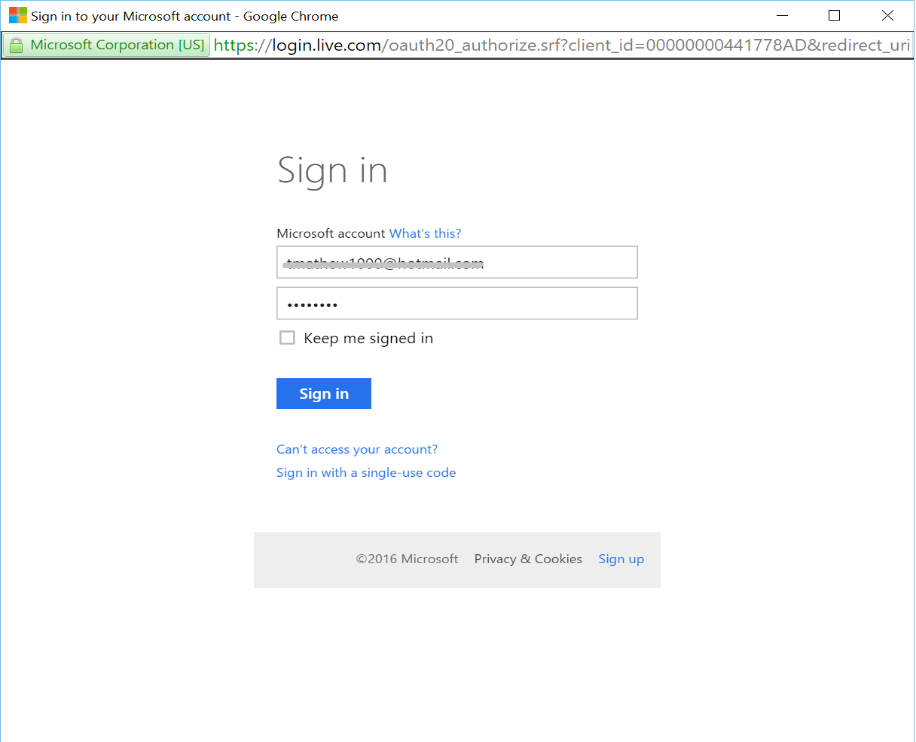
<span class="oneDriveButton">Open from OneDrive</span>

</button>

</p>

</body>

</html>

1. Prior to testing the solution, always clear the browser cookies as the Access token is cached.
2. When running the above sample code and when selecting the “Open from OneDrive” button, you are first prompted to enter the login credentials. You can use either Microsoft account (formerly knowns as Windows Live ID) or O365 Tenant Credentials). 
3. You will be redirected to enter in the full credentials
4. Once authenticated, you will finally be presented with the File Picker to choose your file(s). 