# GetOneDrive files SharePoint 2013 sample for Visual Studio 2013

Learn how to use the SharePoint managed API to get a list of folders and files from your OneDrive Pro site.

This sample *provider-hosted* app demonstrates how to use the SharePoint managed APIs to get the folder collection under the Shared Documents folder.

## Description

The project is organized into view (pages folder), view model (ViewModel folder), and model (DataModel folder). The code that uses the managed SharePoint APIs is in the **OneDriveProDocs**.cs file. The Scripts folder contains the ChromeLoader.js file for showing SP chrome on the UI.

The view model is composed of an **Item** class that describes an individual file from OneDrive Pro and an **Items** class that encapsulates an observable collection of **Item** that is consumed by the view layer.

In the model layer, each OneDrive Pro folder is queried for contained files. The combined collection of files are displayed in a web UI. The logic in this solution works with your OneDrive Pro site and your team site without any code change.

The UI logic instantiates the view model **items** class by calling the **items** constructor. The constructor calls the model **\_OneDriveModel.Run()** method. **Run** connects to OneDrive Pro and gets the files and folders to fill the Items observable collection. A list on the UI is bound to the observable collection and fills when the collection fills.

## Prerequisites

This sample requires the following:

* A SharePoint 2012 development environment.
* An Office 365 developer license and Office 365 development site.
* Visual Studio 2012 and Office 2012 tools for Visual Studio 2012.
* Azure account with a website set up to host the GetOneDriveFiles web app.

## Configure, build, and publish the sample

Follow these steps to configure the GetOneDriveFiles sample app.

1. Update the Site Url property of the solution with the URL of the home page of your Office 365 developer site.
2. Obtain an app id and secret for this app and populate the values for the ClientId and ClientSecret keys inside the Web.config file of the getOnedrivefilesweb project.
3. In the getOnedrivedocsWeb project properties page, set the custom web server URL to the Azure web site URL.
4. Download a provider-host publish profile from Azure for the hosting web site.
5. Publish the remote web app to Azure
6. Publish the app to the SharePoint OneDrive Pro app catalog

For more complete configure, build, and publish details, see the “GetOneDriveFiles – Building and deployment” topic.