# SharePoint 2013 workflow: Integrate with Netflix

This documentation is preliminary and is subject to change.

Summary:  This SharePoint 2013 Preview sample demonstrates using Visual Studio 2012 and Windows Azure Workflow to create a workflow that uses a web service workflow activity with an OData query to access the Netflix service.

## Description of the Netflix integration sample

This sample addresses a scenario in which a user creates a workflow-driven app for SharePoint that contains lists of movie titles from Netflix. Each time the user adds a new movie title, the workflow populates a list field, "TopMovies" with the top movies obtained using the Netflix OData query.

## Prerequisites

This sample requires the following;

SharePoint 2013 Preview

Windows Azure Workflow

Microsoft Visual Studio 2012, either the Ultimate or Professional version

SharePoint development tools in Visual Studio 2012

## Create and build the sample

The following steps and procedures enable you to create, build, and run the sample. The sample solution file is provided for reference.

alert_caution Important

After creating up your Visual Studio workflow project, you must set the **Site URL** property to the server on which the workflow will run.

**To create the app and workflow**

1. Using Visual Studio 2012, create an App for SharePoint 2013 project, and name it Netflix.
2. In the Netflix app, add a list and name it MovieList. For list type, choose Default (Blank).
3. Set the list column1 to name="Title"; type=”single line of text”; required=”true”.
4. Set the list column2 to name="TopMovies"; type=”single line of text”.
5. Add a workflow to the app project, and complete the workflow creation wizard:
6. Name the workflow NetflixWorkflow, and for workflow type, choose List workflow.
7. Ensure that the Associate the workflow check box is selected.
8. In the drop-down list of workflows to associate with the list, choose MovieList.
9. In the second drop-down list, choose Workflow History as the history list (if any).
10. In the third drop-down list, choose Workflow Tasks as the task list (if any).
11. For workflow start, choose A user manually starts the workflow.
12. When you finish the workflow creation wizard, the workflow SPI is created and the workflow.xaml file is opened in the Workflow Designer.
13. In the Workflow Designer, create the following variables for Sequence scope:
14. topItems (Int32)
15. videoStr (String)
16. searchKeyword (String)
17. title (String)
18. videoDV (DynamicValue)
19. Add the LookupCurrentSPListItemStringProperty activity; set its PropertyName to "Title" and its Result field to "searchKeyboard".
20. Add a WriteLIne activity; set the Text value to "Keyword is:" + searchKeyword.
21. Add the Assign activity; set To as topItems and Value to 3.
22. Add the HttpGet activity; set ResponseContent to videoDV, and set Uri to "http://odata.netflix.com/Catalog/Titles?$filter=substringof('" & searchKeyword & "', Name)&$format=json&$top=" & topItems & "&$select=Name"
23. Add the While activity and set it to loop through the output from HttpGet activity, as follows:
24. Create a variable for the while scope; name it item (Int32) and set default to 0.
25. Set a condition such that item < topItems.
26. In the body, use GetDynamicProperty activity to extract the Title property from videoDV, and then use videoStr to hold the concatenated extracted movie titles.
27. Add a Sequence activity.
28. Add a GetDynamicValueProperty<String> activity.
29. Set PropertyName to "d(" & item.ToString() & ")/Name".
30. Set Result to title.
31. Set Source to videoDV.
32. Add an Assign activity; set To to videoStr; set Value to videoStr & title & ","
33. Add an Assign activity; set To to item; set Value to item + 1.
34. Add a SetField activity; set FieldName to TopMovies; set FieldValue to videoStr.
35. Add a WriteLine activity; set Text to "After:::" + videoStr.
36. In the designer, open the Pages node, and then open Home.aspx. In the PlaceHolderMain part, add the following HTML: <a href="../lists/MovieList"> MovieList </a>
37. Open the file workflow.xaml and set breakpoints at the HttpGet and SetField activities.

## Debug the sample

To debug and test the workflow sample, do the following:

1. Start the Test Service Host console.

The Home.aspx page should start to load.

1. Click MovieList to navigate to the list page.
2. Click Add a new item to add a list item. Enter Happiness in the title box, and then click Save.
3. Right-click the list item and choose Workflow.
4. Choose the NetflixWorkflow – WorkflowStart association to run the workflow. Both of the breakpoints that you set should halt execution at the appropriate points. Press F10/F5 to continue.
5. Observe that the test from the WriteLine activities are displayed in Test Service Host console.
6. Refresh the SharePoint MovieList webpage, and observe that the TopMovies column has updated text reading "Happiness, The Inn of the Sixth Happiness."
7. Close the web browser to stop debugging, and close the Test Service Host process.
8. Verify that the app gets retracted from the SharePoint server.

## Change log

| Version | Date |
| --- | --- |
| First release. | July 16, 2012 |

## Related content

[SharePoint 2013 workflow samples](http://msdn.microsoft.com/en-us/library/ffaccd6b-426d-4ca0-b62f-bc7b14641a49)