Contents

[SharePoint 2013: Use multiple SharePoint lists in a Windows Phone app (sample 1) 2](#_Toc329688032)

[Description of the sample 2](#_Toc329688033)

[Prerequisites 2](#_Toc329688034)

[Key components of the sample 2](#_Toc329688035)

[Configure the sample 3](#_Toc329688036)

[Run and test the sample 3](#_Toc329688037)

[Troubleshooting 3](#_Toc329688038)

[Change log 4](#_Toc329688039)

[Related content 4](#_Toc329688040)

# SharePoint 2013: Use multiple SharePoint lists in a Windows Phone app (sample 1)

This documentation is preliminary and is subject to change.

Summary:  Create Windows Phone apps that use data from multiple SharePoint lists.

If you have two SharePoint lists based on the same schema, you can take advantage of the classes implemented by the Windows Phone SharePoint List Application template and create objects of those classes that are specific to each list. You can use multiple SharePoint lists in your app in several ways. When you create a Windows Phone app based on the Windows Phone SharePoint List Application template, you specify a single target SharePoint list, but the architecture of the resulting project is extensible enough to accommodate the integration of multiple lists.

## Description of the sample

Suppose you have two SharePoint lists based on the Contacts list template. One list, named Marketing Team, contains members of a marketing team at your organization; the other list, Engineering Team, contains members of an engineering team. If you create a project by using the Windows Phone SharePoint List Application template, and you specify the Marketing Team list as the target list on which to base the project, an instance of the ListDataProvider class is created (named DataProvider by default) in the implementation of the App class in the App.xaml.cs file in the project. This object represents the list (that is, the Marketing Team list) as a data source for the app, providing operations to access and manipulate items in the list. An instance of the ListViewModel class is also created for the list on which the app is based. This object has a property member (which also happens to be named DataProvider) that can be set to a given instance of the ListDataProvider class, establishing the data source for the ListViewModel class instance.

You can create an additional instance of the ListDataProvider class in the project to serve as the data source for the second list (Engineering Team) in the App.xaml.cs file.

## Prerequisites

This sample requires the following:

* Visual Studio 2010 Express with the new SharePoint templates
* An installation of SharePoint 2013 Preview with administrative privileges

## Key components of the sample

The MultipleSPListAppSingleProject project connects to a single list (Marketing Team) and provides the ability to create, edit, and delete the items in the list. You can follow the instructions in **How to: Use multiple SharePoint 2013 lists in a Windows Phone app** to update this solution and to understand how to use the same listdataprovider.cs file for two lists that have the same schema and for two lists that have different schema.

The sample contains the following:

* MultipleSPListAppSingleProject project
* App.xaml.cs is autogenerated by Windows Phone SharePoint List Application template. The App.xaml file represents the overall Windows app. The associated code-behind file, App.xaml.cs, includes procedural code to handle life-cycle events for the app. App.xaml.cs contains references to SharePoint Server and the list title.
* Listdataprovider.cs contains code for accessing data on SharePoint Server and provides access to the query syntax on which the various list views of the application are based.

## Configure the sample

This code sample covers two scenarios:

* Creating a solution involving SharePoint lists based on the same schema
* Creating a solution involving SharePoint lists based on different schemas

To learn more about the configuring the solution for different scenarios, see **How to: Use multiple SharePoint 2013 lists in a Windows Phone app**.

Follow these steps to configure the sample.

1. Update the value of TargetSiteUrl in the App.xaml.cs file of the MultipleSPListAppSingleProject solution with the URL of the home page of your SharePoint.
2. Update the value of ListTitle in the App.xaml.cs file of the MultipleSPListAppSingleProject solution with the title of your SharePoint list.

## Run and test the sample

Users of your app can navigate between the Marketing Team list and the Engineering Team list. Because the underlying list schemas are the same, the default DataProvider and MainViewModel objects generated by the template and the added SecondaryDataProvider and SecondaryViewModel objects handle all the data transactions without requiring any modifications to the ListDataProvider.cs file.

1. Choose the F5 key to build and run the app.

## Troubleshooting

The following table lists common configuration and environment errors that prevent the sample from running or deploying properly and how to solve them.

| Problem | Solution |
| --- | --- |
| While running the SharePoint List wizard from Visual Studio 2010 Express, an error may occur if developer does not have sufficient privilege on SharePoint site. | Give sufficient privilege to the user account with which developer is running the wizard. |
| Form-based authentication error. | Form-based authentication is not enabled by default. To enable basic form-based authentication for the web application, follow these steps.   * Navigate to Central Administration and ensure you have administrator rights on the server. * Under Application Management, choose Manage Web Applications. * Choose your web application (on which you have your SharePoint site, which you are accessing from your mobile app). * From the ribbon, choose Authentication Providers. * In the Authentication Provider dialog box, choose Default to edit the authentication. * In the Edit Authentication Model window under Claims Authentication Types, choose Basic Authentication. |

## Change log

| Version | Date |
| --- | --- |
| First version | July 16, 2012 |

## Related content

* **How to: Use multiple SharePoint 2013 lists in a Windows Phone app**
* **How to: Create a Windows Phone SharePoint 2013 list app**
* **Overview of Windows Phone SharePoint 2013 application templates in Visual Studio**