Contents

[SharePoint 2013: Implement business logic and data validation 2](#_Toc329688043)

[Description of the sample 2](#_Toc329688044)

[Prerequisites 3](#_Toc329688045)

[Key components of the sample 3](#_Toc329688046)

[Configure the sample 3](#_Toc329688047)

[Run and test the sample 4](#_Toc329688048)

[Troubleshooting 4](#_Toc329688049)

[Change log 5](#_Toc329688050)

[Related content 5](#_Toc329688051)

# SharePoint 2013: Implement business logic and data validation

This documentation is preliminary and is subject to change.

Summary:  Implement data validation in a Windows Phone app created by using the Windows Phone SharePoint List Application template.

In a Windows Phone app intended for production use, you likely need to validate data entered by users to, for example, enforce business logic, or to ensure appropriate formatting of entered values, or simply to catch mistakes before saving values to a SharePoint list. Projects based on the Windows Phone SharePoint List Application template include default data-validation logic, but such projects also provide a mechanism for developers to implement custom data validation.

## Description of the sample

Some data types for fields in SharePoint lists are associated by default with simple formatting or data validation. If you enter an invalid URL for a field based on the Hyperlink or Picture field type in a SharePoint list and attempt to save your changes, you see a message indicating that the address you entered is invalid. If you enter a customer name as a value for a field based on the Date and Time field type, you receive a message directing you to enter a date within a valid range for the field. Some of these basic validation rules are also enforced by default in a Windows Phone app created from the Windows Phone SharePoint List Application template. If you enter anything other than a date value in a field that is bound to a SharePoint field of Date and Time type in the Edit form of a Windows Phone app based on a SharePoint list, you see a validation error message when the focus shifts from the TextBox control associated with the field.

You can validate data entered by users of your Windows Phone app in several ways. A project created by using the Windows Phone SharePoint List Application template includes classes that serve as intermediaries between the forms (that is, the views) of the data in the Windows Phone app (for example, the EditForm.xaml file) and the data itself in the SharePoint list on which the app is based. These classes can be considered implementations of the ViewModel component of the [Model-View-ViewModel design pattern](http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx). (For more information about how the Windows Phone SharePoint List Application template conforms to the MVVM software design pattern, see **Architecture of the Windows Phone SharePoint List Application template**.) In applications designed based on the MVVM pattern, data validation is often handled in the data layer (that is, in the Model component). In projects created from the Windows Phone SharePoint List Application template, an extensible mechanism for data validation has been "pushed up" a layer and implemented in the ViewModel component, to make it easier for developers to manage data validation. In projects based on the template, therefore, the most suitable place for custom code that validates user input or otherwise manages data is in these ViewModel classes. In terms of data validation, the EditItemViewModel class and the NewItemViewModel class (the classes associated with the forms most likely to involve editing and updating list data) both provide an open implementation of a validation method (named Validate) that overrides the base validation method in the class from which these two classes are derived.

## 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 sample contains the following:

* SPListAppDataValidation project, which contains the App.xaml.cs file and the CreateProductOrdersList.ps1 Windows PowerShell script.
* App.xaml.cs   This file is autogenerated by the 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 file contains references to SharePoint Server and the list title
* CreateProductOrdersList.ps1   You can run this Windows PowerShell script from the SharePoint Management Shell to create the SharePoint list on which this project is based.
* The data validation rules implemented in this sample are as follows. (For more information about data validation rules in this code sample, see [SharePoint 2013: Implement business logic and data validation](#BMb10abb0f9a75482080b68acd39ef8da8).)
* The fulfillment date of an order must be later than the date on which the order was placed.
* Phone numbers must be in one of the common phone number formats; for example, "(555) 555-5555". (This rule is necessary because the field type for phone numbers is "Single line of text" (that is, Text), which can be any text up to 255 characters.)

## Configure the sample

This sample assumes that you are working in the context of a Windows Phone app project that was created from the Windows Phone SharePoint List Application template and that your app is based on a Product Orders list created from the Custom List template on the server and containing the columns and field types shown in Table 1.

Table 1. Sample Product Orders list

| Column | Type | Required |
| --- | --- | --- |
| Product (i.e., Title) | Single line of text (Text) | Yes |
| Description | Single line of text (Text) | No |
| Quantity | Number | Yes |
| Order Date | Date and Time (DateTime) | No |
| Fulfillment Date | Date and Time (DateTime) | No |
| Contact Number | Single line of text (Text) | No |

Follow these steps to configure the sample.

1. Update the value of TargetSiteUrl in the App.xaml.cs file of the SPListAppDataValidation solution with the URL of the home page of your SharePoint website.
2. Update the value of TargetListTitle in the App.xaml.cs file of the SPListAppDataValidation solution with the title of the targeted SharePoint list.

## Run and test the sample

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

* [SharePoint 2013: Implement business logic and data validation](#BMb10abb0f9a75482080b68acd39ef8da8)
* **How to: Create a Windows Phone SharePoint 2013 list app**
* **Overview of Windows Phone SharePoint 2013 application templates in Visual Studio**