Hands-On Lab

Searching and Sharing

Lab version: 1.0.0

Last updated: 6/20/2012



Contents

[Overview 3](#_Toc326842832)

[Exercise 1: Add Recipe Sharing 4](#_Toc326842833)

[Task 1 – Invoke the Share Charm 4](#_Toc326842834)

[Task 2 – Implement Recipe Sharing 4](#_Toc326842835)

[Task 3 – Implement Recipe Image Sharing 6](#_Toc326842836)

[Exercise 2: Add Recipe Search 8](#_Toc326842837)

[Task 1 – Invoke the Search Charm 8](#_Toc326842838)

[Task 2 – Add Search Support 8](#_Toc326842839)

[Task 3 – Refine the Search Results Page 11](#_Toc326842840)

[Task 4 – Add Search Suggestions 12](#_Toc326842841)

[Summary 13](#_Toc326842842)

Overview

* 1. One of the key features of a Windows 8 user experience is the charms bar that slides in from the right side of the screen in response to swipes or presses of Win-C. The buttons (“charms”) in the charms bar provide a means for Metro style apps to expose commonly used features in a manner that is consistent across apps. For example, if you want to perform a search in a Metro style app, you select the Search charm and type a search term into the search pane. The UI – and the actions required to invoke that UI – are the same in every app. To share data with another app, you use the Share charm. An app that supports sharing can then share data – for example, a drawing in a paint application or a recipe in Contoso Cookbook – with other apps that support sharing.
  2. In this lab, you will add support for searching and sharing to Contoso Cookbook. You will get first-hand experience implementing searching and sharing contracts, and learn how these contracts provide a higher level of integration between either two Metro style apps or an app and Windows itself.

# Objectives

* 1. This lab will show you how to:
  + Implement sharing in a Metro style app
  + Implement search in a Metro style app
  + Implement search suggestions

# System Requirements

* 1. You must have the following items to complete this lab:
  + Microsoft Windows 8 Release Preview
  + Microsoft Visual Studio 2012 RC

# Setup

* 1. You must perform the following steps to prepare your computer for this lab:
  2. Install Microsoft Windows 8 Release Preview
  3. Install Microsoft Visual Studio 2012 RC

# Exercises

* 1. This Hands-On Lab comprises the following exercises:
  2. Add Recipe Sharing
  3. Add Recipe Search
  4. Estimated time to complete this lab: **30 to 40 minutes**.

Exercise 1: Add Recipe Sharing

1. In Exercise 1, you will add sharing support to Contoso Cookbook so recipes can be shared with other applications. You will share two types of data for each recipe: textual data that includes the recipe name, ingredients, and directions, and image data containing a pictorial representation of the recipe.

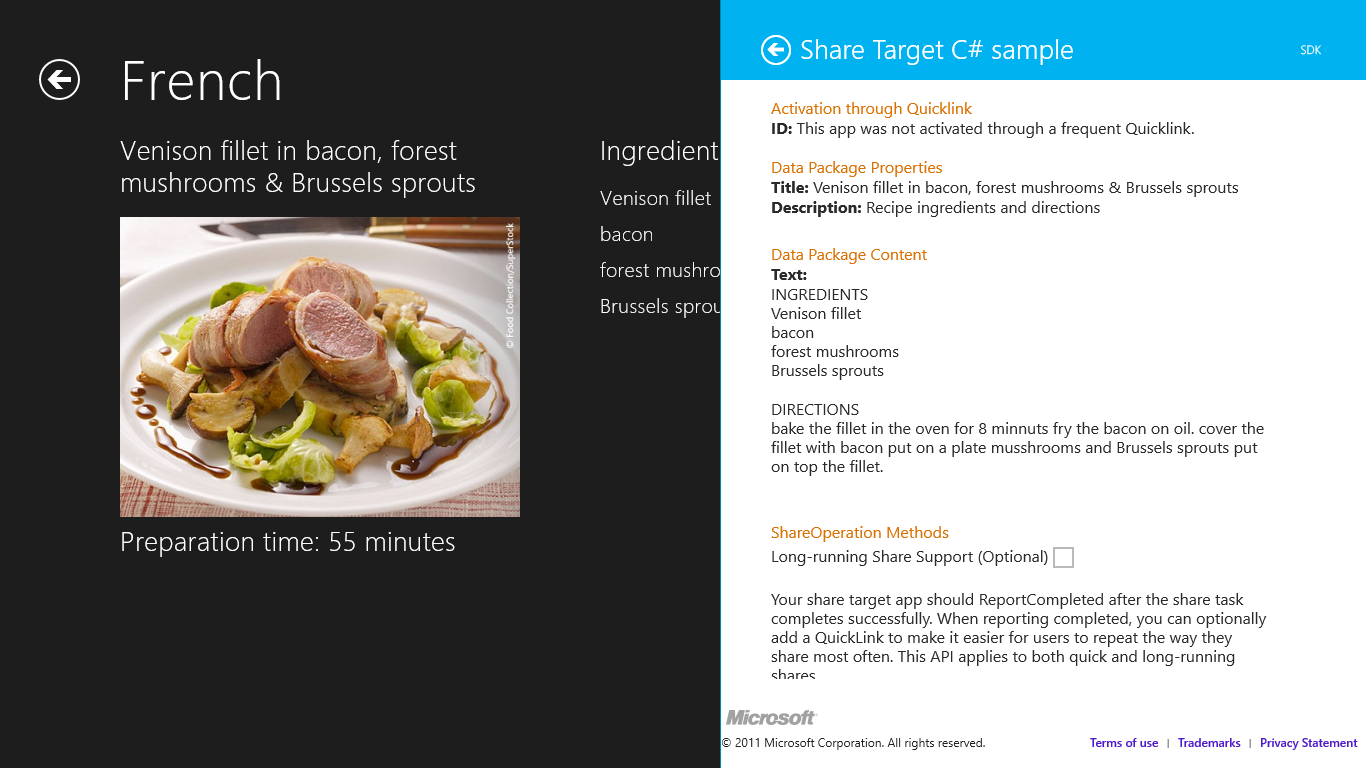
Task 1 – Invoke the Share Charm

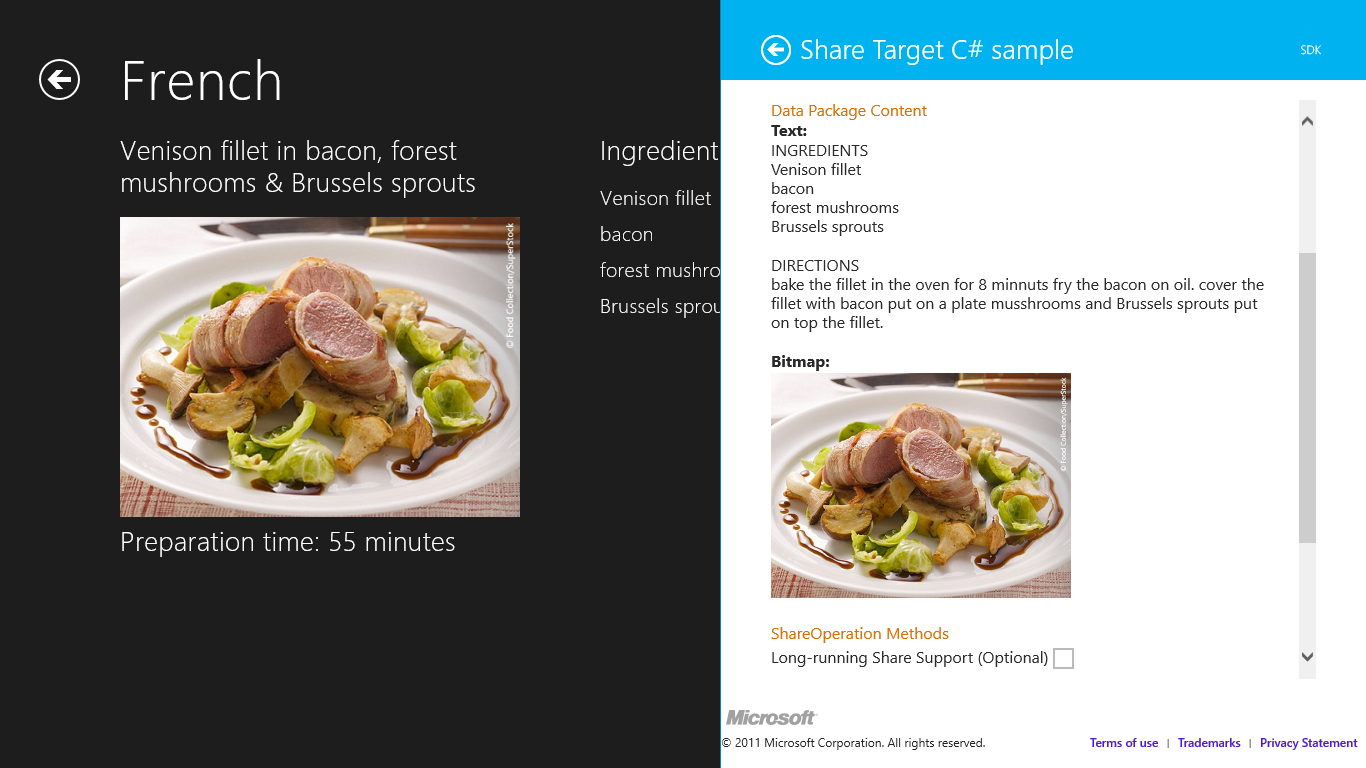
* 1. To get started, let’s see how the Share charm behaves when it is invoked in Contoso Cookbook before sharing support is added.
  2. Open the ContosoCookbook project you finished in Lab 2 in Visual Studio. If you did not complete Lab 2 or would like to start with a reference copy, you will find a completed version of the lab in the starting materials.
  3. Press F5 to launch Contoso Cookbook.
  4. Tap one of the recipes to show the item-detail page.
  5. Display the charms bar by swiping from right to left from the right edge of the screen if you are using a touch screen, or by pressing Win-C if you are not.
  6. Tap the Share charm to display the sharing pane.
  7. Since Contoso Cookbook currently does not implement a sharing contract, the sharing pane informs you “This app can’t share.”
  8. Return to Visual Studio and stop debugging.

Task 2 – Implement Recipe Sharing

* 1. Now that you have seen what the sharing pane looks like when an app does not support sharing, let’s add a sharing contract to Contoso Cookbook so it can share recipe data.
  2. Open itemDetail.js and add the following statements near the top of the file, after the statements that define variables named ui and utils:
     1. JavaScript
     2. var storage = Windows.Storage;
     3. var dtm = Windows.ApplicationModel.DataTransfer.DataTransferManager;
     4. var item;
  3. Remove “var” from the first line of the ready function so that the line looks like this:
     1. JavaScript
     2. item = options && options.item ? options.item : data.items.getAt(0);
  4. Add the following code to the bottom of the ready function:
     1. JavaScript
     2. // Register for datarequested events for sharing

dtm.getForCurrentView().addEventListener("datarequested", this.onDataRequested);

* 1. Finish up by adding the following functions after the ready function (note that you’ll need to add a comma after the closing curly brace of the ready function):
     1. JavaScript
     2. onDataRequested: function (e) {
     3. var request = e.request;
     4. request.data.properties.title = item.title;
     5. request.data.properties.description = "Recipe ingredients and directions";
     6. // Share recipe text
     7. var recipe = "\r\nINGREDIENTS\r\n" + item.ingredients.join("\r\n");
     8. recipe += ("\r\n\r\nDIRECTIONS\r\n" + item.directions);
     9. request.data.setText(recipe);
     10. },
     11. unload: function () {
     12. WinJS.Navigation.removeEventListener("datarequested", this.onDataRequested);
     13. }
     14. **Note:** Sharing is implementing by registering a handler for DataTransferManager’s DataRequested events, which fire when the user activates the Share charm. In this example, you are responding to that event by calling setText on the DataPackage object exposed through e.request.data to provide the recipe in the form of text. When the sharing pane appears with a list of share targets, the list will include only share targets that can consume text.
  2. Press F5 to launch the application.
  3. Tap one of the recipes to show the recipe-detail page.
  4. Display the charms bar and select the Share charm to display the sharing pane. The sharing pane now shows you a list of *share targets*– applications that can consume data shared by share sources- installed on your device
     1. **Note:** If you have not done so already, now would be a great time to install the share-target sample that comes with the Windows 8 SDK Samples pack. The Share Target Sample App demonstrates how to write share-target applications. More importantly, it provides a sharing target to test with as you develop applications that act as sharing sources, and it accepts images as well as text and other data types. To install the share-target sample in the system, open it in Visual Studio and run it one time. After that, it should appear in the list of share targets when you select the Share charm from any Metro style app.
  5. Select one of the share targets listed in the sharing pane and verify that it received the recipe text. Figure 1 shows how the Windows 8 SDK’s Share Target Sample App looks after accepting recipe text shared by Contoso Cookbook.
     1. 
     2. Figure 1
     3. Share Target Sample App showing a shared recipe
  6. Return to Visual Studio and stop debugging.

1. Task 3 – Implement Recipe Image Sharing
   1. Contoso Cookbook can now share textual recipe data, but since each recipe is accompanied by an image, it should share recipe images as well. That way, a share target that accepts images can show a photo of the recipe along with the recipe text (assuming the share target supports text as well as images). Let’s revise your sharing code to support bitmaps as well as text.
   2. In itemDetail.js, go to the onDataRequested function you added in Task 2.
   3. Add the following lines of code to the end of the function:
      1. JavaScript
      2. // Share recipe image
      3. var uri = item.backgroundImage;
      4. if (item.backgroundImage.indexOf("http://") != 0)
      5. uri = "ms-appx:///" + uri;
      6. uri = new Windows.Foundation.Uri(uri);
      7. var reference = storage.Streams.RandomAccessStreamReference.createFromUri(uri);
      8. request.data.properties.thumbnail = reference;
      9. request.data.setBitmap(reference);
   4. Press F5 to launch the application.
   5. Tap one of the recipes to show the recipe-detail page.
   6. Display the charms bar and select the Share charm to display the sharing pane.
   7. Select one of the share targets listed in the sharing pane and verify that it received the recipe image. Figure 2 shows how the Share Target Sample App looks after accepting a recipe image from Contoso Cookbook.
      1. 
      2. Figure 2
      3. Share Target Sample App showing a shared recipe image
   8. Return to Visual Studio and stop debugging.

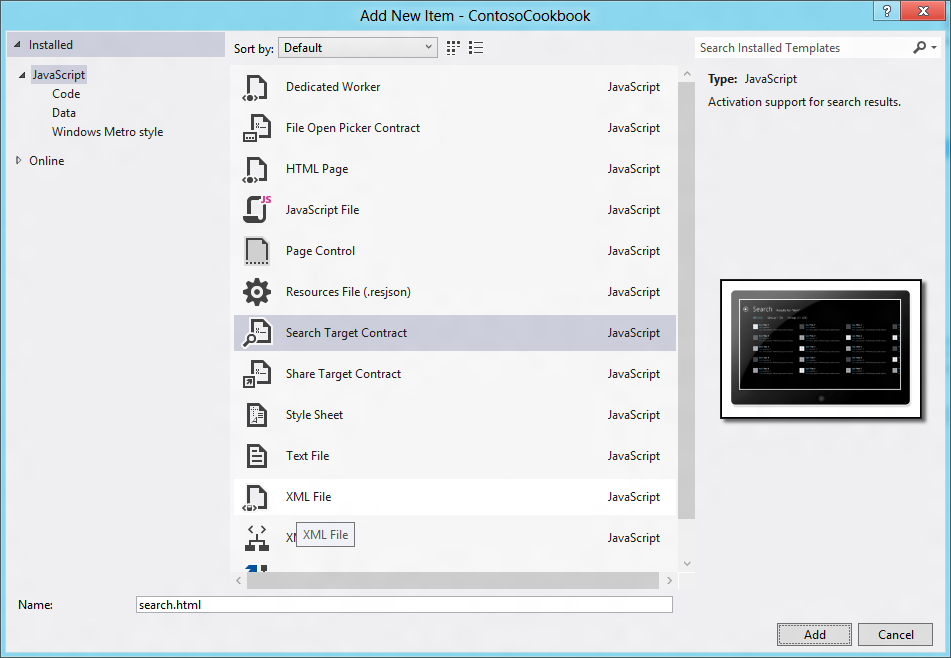
Exercise 2: Add Recipe Search

1. In Exercise 2, you will add search support to Contoso Cookbook so users can use the Search charm to search for recipe data. For example, a user who wants to find all recipes that contain sugar will invoke the Search charm, type “sugar” into the search box, and be presented with a list of recipes that include sugar.

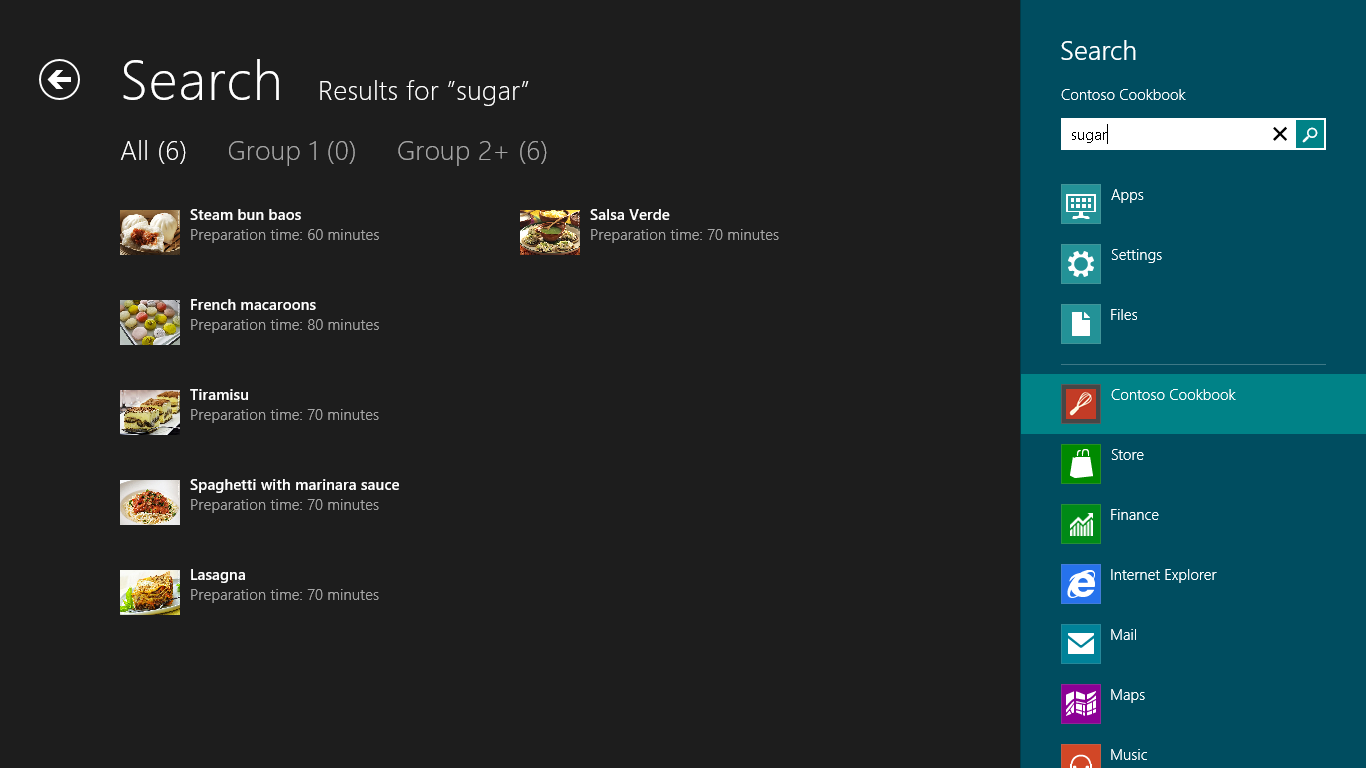
Task 1 – Invoke the Search Charm

* 1. Before adding search support to Contoso Cookbook, let’s see what the search UI looks like when it’s invoked while Contoso Cookbook is the foreground application.
  2. Press F5 to launch the application.
  3. Display the charms bar by swiping from right to left from the right edge of the screen if you are using a touch screen, or by pressing Win-C if you are not.
  4. Tap the Search charm to display the search pane.
  5. Type “sugar” (without the quotation marks) into the search box and press Enter or tap the magnifying-glass icon at the right end of the search box.
  6. Windows 8 tells you “This app can’t be searched.” That will change once you add search support to Contoso Cookbook.
  7. Return to Visual Studio and stop debugging.

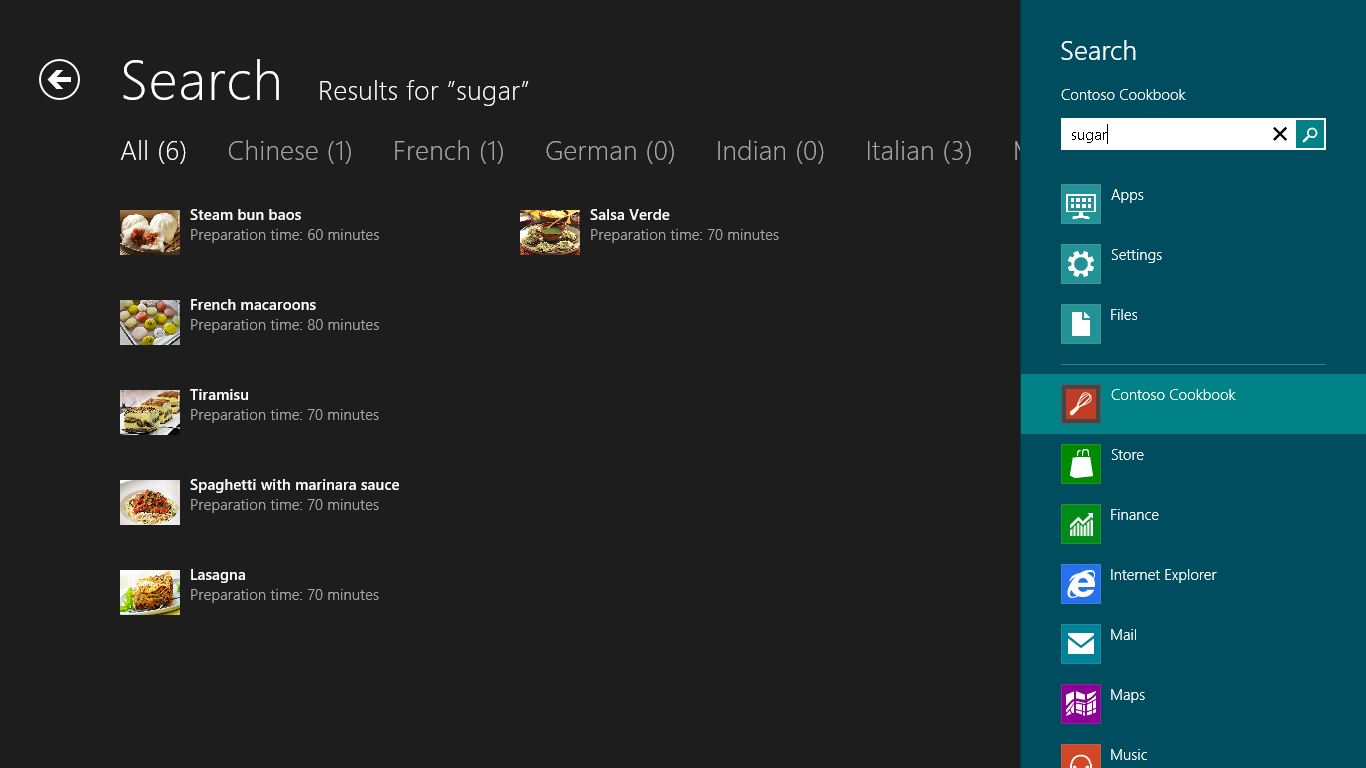
Task 2 – Add Search Support

* 1. To implement search, you code a search contract for your application. Visual Studio will do most of the work for you by inserting a JavaScript contract into your app. You will need to tweak the JavaScript to do domain-specific searches within your app’s data. It’s easy to do, as the next few steps will demonstrate.
  2. Right-click the pages folder in Solution Explorer and use the **Add - New Folder** command to add a folder named search to the pages folder.
  3. Right-click the search folder and use the **Add - New Item** command to add a Search Target Contract named search.html, as shown in Figure 3.
     1. 
     2. Figure 3
     3. Adding a search contract
  4. Add the following statement to default.html to ensure that search.js is loaded when the application is started:
     1. HTML
     2. <script src="/pages/search/search.js"></script>
  5. Open search.js and find the searchData function. This is the part of the search contract that is called when the user initiates a search from the Search charm. The code in this function needs to be modified to perform recipe searches. To that end, find the return statement that calls match three times with a regular expression and change it to this:
     1. JavaScript
     2. return (item.title.match(regex) || item.directions.match(regex));
     3. **Note:** The change you just made has Contoso Cookbook search the title and directions properties of each recipe for the text entered into the search box. That text is passed to the searchData function in the parameter named queryText. If you wanted to expand the search to include other recipe properties, this is where you would do it.
  6. While still in search.js, find the itemInvoked function. This function is called when the user selects an item in the ListView that displays search results. Uncomment the call to nav.navigate and modify it as shown:
     1. JavaScript
     2. nav.navigate("/pages/itemDetail/itemDetail.html ", { item: Data.getItemReference(item.data) });
  7. Open search.html, which contains an HTML fragment defining how search results are presented to the user. Find the DIV whose class is “item,” and inside it find the DIV whose class is “item-content.” Replace the contents of that DIV – one H3 element and two H4s -- with the statements below:
     1. HTML
     2. <h3 class="item-title win-type-ellipsis" data-win-bind="innerHTML: shortTitle search.markText"></h3>
     3. <h4 class="item-subtitle win-type-x-small win-type-ellipsis">

Preparation time: <span data-win-bind="textContent: preptime"></span> minutes

* + 1. </h4>
  1. Open search.css. Find “.search section[role=main] .resultslist .item” and modify the -ms-grid-columns line as shown below:
     1. CSS
     2. .search section[role=main] .resultslist .item {
     3. /\* Define a grid with columns for an icon, spacing and item details \*/
     4. -ms-grid-columns: 62px 8px 1fr;
     5. -ms-grid-rows: 1fr;
     6. display: -ms-grid;
     7. height: 64px;
     8. padding-left: 7px;
     9. padding-top: 1px;
     10. padding-right: 7px;
     11. width: 340px;
     12. }
  2. Find “.search section[role=main] .resultslist .item .item-image” and change the width to 60px and height to 45px to allow recipe images to retain their original aspect ratios:
     1. CSS
     2. .search section[role=main] .resultslist .item .item-image {
     3. -ms-grid-column: 1;
     4. -ms-grid-row: 1;
     5. height: 45px;
     6. margin-top: 5px;
     7. width: 60px;
     8. }
  3. Press F5 to start the application.
  4. Display the charms bar.
  5. Tap the Search charm to display Metro’s search pane.
  6. Type “sugar” (without the quotation marks) into the search box at the top of the search pane and press Enter or tap the magnifying-glass icon at the right end of the search box.
  7. Verify that six recipes appear in the search results (see Figure 4).
     1. 
     2. Figure 4
     3. Search results for “rice”
  8. Select one of the recipes and verify that the recipe detail appears.
  9. Return to Visual Studio and stop debugging.

Task 3 – Refine the Search Results Page

* 1. So far, so good! You got a lot free functionality when Visual Studio added search contract for you. After some minor modifications, you have search working in Contoso Cookbook. Did you notice the odd nomenclature at the top of the search results page? Instead of displaying group names such as “Chinese” and “Italian,” it shows “Group 1,” “Group 2+,” and so on. Let’s fix that by making a simple modification to search.js.
  2. Open search.js and find the comment near the top of the file that reads “TODO: Replace or remove example filters.”
  3. Replace the two lines of code below the comment (the two calls to this.filters.push) with these statements:
     1. JavaScript
     2. Data.groups.forEach(function (group) {
     3. this.filters.push({ results: null, text: group.title, predicate: function (item) { return item.group.key === group.key; } });
     4. }, this);
     5. **Note:** These statements replace the hardcoded group names generated by Visual Studio with the names of the recipe groups in your application.
  4. Start the application again and repeat the search for “sugar.” Verify that the generic group names at the top of the page are replaced by recipe group names, as shown in Figure 5.
     1. 
     2. Figure 5
     3. Search results with recipe group titles
  5. Return to Visual Studio and stop debugging.

Task 4 – Add Search Suggestions

* 1. A final enhancement that you can add to the search experience is to provide suggestions as the user types a search term into the search box. It’s easy to do; all you have to do is add a handler for the SuggestionsRequested event. Here’s how.
  2. Open search.js and add the following function near the bottom of the file (after the onquerysubmitted event handler):
     1. JavaScript
     2. appModel.Search.SearchPane.getForCurrentView().onsuggestionsrequested = function (eventObject) {
     3. var text = eventObject.queryText.toLowerCase();
     4. var terms = ["salt", "pepper", "water", "egg", "vinegar", "flour", "rice", "oil"];
     5. terms.forEach(function(term) {
     6. if (term.indexOf(text) == 0) {
     7. eventObject.request.searchSuggestionCollection.appendQuerySuggestion(term); }
     8. });
     9. };
     10. **Note:** The code you just added provides search suggestions for words that pattern-match salt, pepper, water, egg, vinegar, flour, rice, sugar, and oil. If the user types “sa,” the word “salt” will appear in the search pane as a suggested completion term. Of course, you can add as many suggestions as you’d like. If you want “ketchup” to appear when the user types “ke,” simply add that term to the list.
  3. Start the application again and type “su” into the search box. Verify that “sugar” appears in the suggestion list underneath the search box, as shown in Figure 6.
     1. 
     2. Figure 6
     3. Search suggestions in action
  4. Return to Visual Studio and stop debugging.

Summary

* 1. Contracts are an important part of Windows 8 because they allow apps to integrate with the shell and provide a user experience that is both consistent and predictable. This loosely coupled collaboration is very synergistic and extensible, allowing you to share anything with any app, search within any app, and so on.
  2. In this lab, you learned two types of contracts first-hand: sharing contracts and search contracts. In a later lab, you will use another type of contract to integrate with the Settings charm. But first there is something else we need to tackle: media capture using the cameras built into most PCs and mobile devices today. That is the subject of the next lab, so let’s keep moving!