<http://blogs.techvedika.com/a-step-by-step-guide-for-using-azure-search-for-your-ecommerce-store/>

<https://docs.microsoft.com/en-us/azure/search/search-get-started-portal>

using Microsoft.Azure.Search;

using Microsoft.Azure.Search.Models;

using Microsoft.Spatial;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

public partial class Form1 : Form

{

SearchCredentials key;

SearchServiceClient serviceClient;

public Form1()

{

InitializeComponent();

////Initialize object for SearchServiceClient

key = new SearchCredentials("167BC71F4CA80289C807E111609B7838");

serviceClient = new SearchServiceClient("catalogs", key);

}

private void button1\_Click(object sender, EventArgs e)

{

//Create index using the serviceClient and field definition as in the given example

var definition = new Index()

{

Name = "hotels",

Fields = new[]

{

new Field("hotelId", DataType.String) { IsKey = true },

new Field("hotelName", DataType.String) { IsSearchable = true, IsFilterable = true },

new Field("baseRate", DataType.Double) { IsFilterable = true, IsSortable = true },

new Field("category", DataType.String) { IsSearchable = true, IsFilterable = true, IsSortable = true, IsFacetable = true },

new Field("city", DataType.String) { IsSearchable = true, IsFilterable = true, IsSortable = true, IsFacetable = true },

new Field("tags", DataType.Collection(DataType.String)) { IsSearchable = true, IsFilterable = true, IsFacetable = true },

new Field("parkingIncluded", DataType.Boolean) { IsFilterable = true, IsFacetable = true },

new Field("lastRenovationDate", DataType.DateTimeOffset) { IsFilterable = true, IsSortable = true, IsFacetable = true },

new Field("rating", DataType.Int32) { IsFilterable = true, IsSortable = true, IsFacetable = true },

new Field("location", DataType.GeographyPoint) { IsFilterable = true, IsSortable = true }

}

};

serviceClient.Indexes.Create(definition);

MessageBox.Show("yes Index created");

}

private void button2\_Click(object sender, EventArgs e)

{

//Create method to add the documents to the index

AddDocuments();

}

private void AddDocuments()

{

// SearchIndexClient indexClient = serviceClient.Indexes.GetClient("hotels");

var indexClient = serviceClient.Indexes.GetClient("hotels");

var documents =

new Hotel[]

{

new Hotel()

{

HotelId = "1058-441",

HotelName = "Fancy Stay",

City = "Hyderabad",

BaseRate = 199.0,

Category = "Luxury",

Tags = new[] { "pool", "view", "concierge" },

ParkingIncluded = false,

LastRenovationDate = new DateTimeOffset(2010, 6, 27, 0, 0, 0, TimeSpan.Zero),

Rating = 5,

Location = GeographyPoint.Create(47.678581, -122.131577)

},

new Hotel()

{

HotelId = "666-437",

HotelName = "Roach Motel",

City = "Mumbai",

BaseRate = 79.99,

Category = "Budget",

Tags = new[] { "motel", "budget" },

ParkingIncluded = true,

LastRenovationDate = new DateTimeOffset(1982, 4, 28, 0, 0, 0, TimeSpan.Zero),

Rating = 1,

Location = GeographyPoint.Create(49.678581, -122.131577)

},

new Hotel()

{

HotelId = "970-501",

HotelName = "Econo-Stay",

City = "Hyderabad",

BaseRate = 129.99,

Category = "Budget",

Tags = new[] { "pool", "budget" },

ParkingIncluded = true,

LastRenovationDate = new DateTimeOffset(1995, 7, 1, 0, 0, 0, TimeSpan.Zero),

Rating = 4,

Location = GeographyPoint.Create(46.678581, -122.131577)

},

new Hotel()

{

HotelId = "956-532",

HotelName = "Express Rooms",

City = "Delhi",

BaseRate = 129.99,

Category = "Budget",

Tags = new[] { "wifi", "budget" },

ParkingIncluded = true,

LastRenovationDate = new DateTimeOffset(1995, 7, 1, 0, 0, 0, TimeSpan.Zero),

Rating = 4,

Location = GeographyPoint.Create(48.678581, -122.131577)

},

new Hotel()

{

HotelId = "566-518",

HotelName = "Surprisingly Expensive Suites",

City = "Mumbai",

BaseRate = 279.99,

Category = "Luxury",

ParkingIncluded = false

}

};

try

{

var batch = IndexBatch.Upload(documents);

indexClient.Documents.Index(batch);

}

catch (IndexBatchException e)

{

// Sometimes when your Search service is under load, indexing will fail for some of the documents in

// the batch. Depending on your application, you can take compensating actions like delaying and

// retrying. For this simple demo, we just log the failed document keys and continue.

// Console.WriteLine(

// "Failed to index some of the documents: {0}",

// String.Join(", ", e.IndexingResults.Where(r = &amp; amp; amp; gt; !r.Succeeded).Select(r = &amp; amp; amp; gt; r.Key)));

}

// Wait a while for indexing to complete.

Thread.Sleep(2000);

MessageBox.Show("yes document added to the Index");

}

}

}