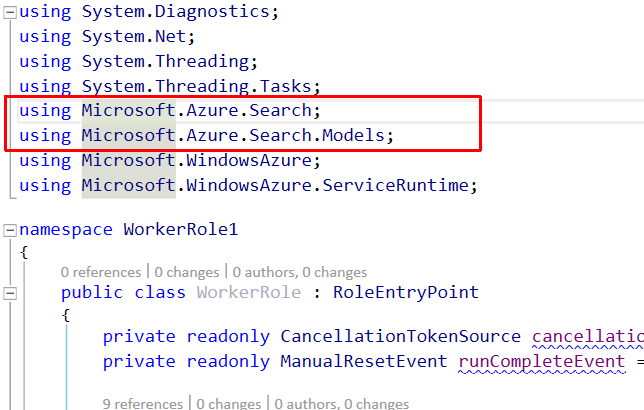
Azure Search

1. Lag en ny søkeindex
2. Hente tweets fra meldingskøen
3. Last opp tweets til indeksen

# Del 1: Lage ny søkeindeks

1. Legg til nyt prosjekt -> Cloud -> Azure Cloud Serivce -> Worker Role
2. Legg til NuGet pakke “Microsoft.Azure.Search” -> 0.13.0-preview



1. Legg til konfigurasjon for Search API I kloudservisen: (“SearchServiceName”) og (“SearchApiKey”)
2. Hent CloudConfiguration seetings i OnStart()

var searchServiceName = CloudConfigurationManager.GetSetting("SearchServiceName");

var apiKey = CloudConfigurationManager.GetSetting("SearchApiKey");

1. Opprett søkeklienten

var searchServiceClient = new SearchServiceClient(searchServiceName, new SearchCredentials(apiKey));

1. Definer en index for dataen din. I vårt tilfelle er dataen tweets. Datamodellene for Tweets er definert i Entities prosjeket (Entities.Twitter.Tweet). For indeksen så bruker vi en utflatet, forenklet modell “FlattendTweet” (sic) som ligger under Entities.Twitter.SearchIndex. For å enkelt kunne mappe til / fra JSON er det bra å bruke identiske navn på feltene som du definert i modellen din (FlattendTweet).

var index = new Index()

{

    Name = "tuttweets",

    Fields = new[]

    {

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

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

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

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

    }

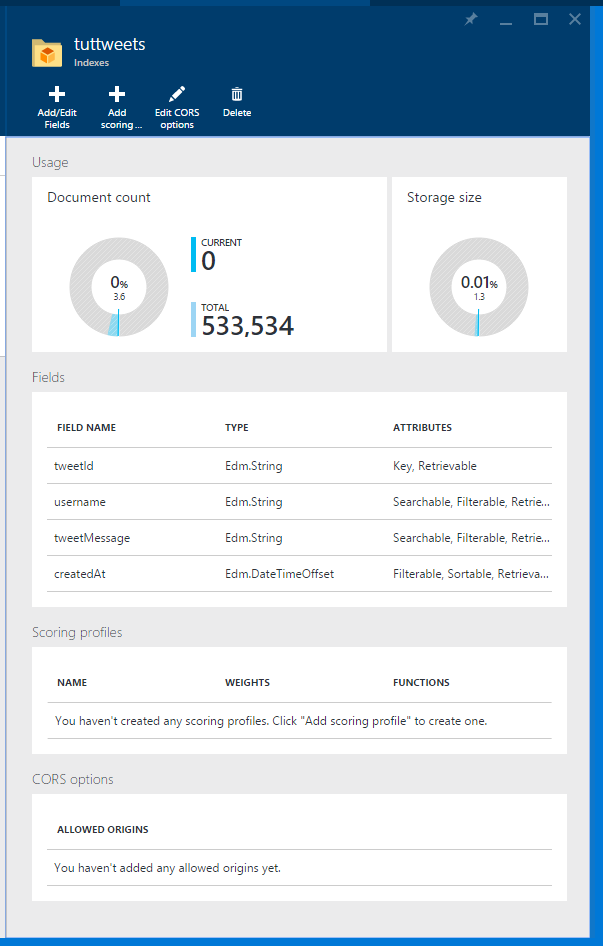
};

1. Man kan også definere hvilken text analyser som skal brukes for hvert felt. En text analyser gjør typiske behandlinger av teksten for å få best mulige søkeresultat. Dette er f.eks. stemming, stop word removal, lemmitization m.m. Azure Search tilbyr to versjoner: Lucene analyzer (35 stk), som gjør stop word removal og Porter Stemming, og Microsoft sin egen analyzer (50 stk), som gjør er litt mer avansert men litt tregere. For å legge til, definer den etter datatype-definisjonen:

new Field("TweetMessage", DataType.String, AnalyzerName.NbMicrosoft)

1. Last opp indexen til Azure Search:

searchServiceClient.Indexes.CreateOrUpdate(index);

1. Start opp cloud service
2. Logg deg inn I Azure Portal og inspiser indeksen din:

Completed code:



# Del 2: Hent tweets fra køen

1. Lag et CloudQueue felt

private CloudQueue \_searchIndexQueue;

1. Koble til en storage account med køen som dere pusher til fra TweetPublisher, I OnStart():

var storageAccount = CloudStorageAccount.Parse(CloudConfigurationManager.GetSetting("StorageConnectionString"));

var queueClient = storageAccount.CreateCloudQueueClient();

\_searchIndexQueue = queueClient.GetQueueReference("searchindexqueue");

\_searchIndexQueue.CreateIfNotExists();

1. Hent meldinger fra køen, i RunAsync():

var tweets = new List<Tweet>();

foreach (var msg in \_searchIndexQueue.GetMessages(10).Where(msg => msg != null))

{

    var messageString = msg.AsString;

    tweets.Add(JsonConvert.DeserializeObject<Tweet>(messageString));

    \_searchIndexQueue.DeleteMessage(msg);

}

1. Pass på at du har satt tweet publisheren og tweet even handleren som startup prosjekt
2. Start workerolen din og se om du finner noen tweets ☺

Trace.TraceInformation("WorkerRole1 fetched {0} tweets", tweets.Count);

# Del 3: Last opp tweets til søkeindksen

1. Lag en SearchIndexClient for den navngitte indeksen du opprette tidligere

var client = \_searchServiceClient.Indexes.GetClient("tuttweets");

1. Lag et flatt objekt med tweet data som representerer det du ønsker å indeksere fra dataen

var flatTweets = tweets.Select(tweet => new FlattendTweet

{

    TweetId = tweet.IdString,

    Username = tweet.User.ScreenName,

    CreatedAt = tweet.CreatedAtUtc,

    TweetMessage = tweet.Text,

}).ToList();

1. Definer en liste av Index Action for søkeobjektet ditt var actions = new List<IndexAction<FlattendTweet>>();

foreach (var t in flatTweeets)

{

    var action = new IndexAction<FlattendTweet>(IndexActionType.**Upload**, t);

    actions.Add(action);

}

1. Legg listen med actions til en batch jobb og send den til indeksen

var batch = new IndexBatch<FlattendTweet>(actions);

client.Documents.Index(batch);

1. 3) og 4) kan forenkles om du bare vil laste opp, siden Upload er default IndexActionType

client.Documents.Index(IndexBatch.Create(flatTweeets.Select(IndexAction.Create)));