|  |  |
| --- | --- |
| **TABLE OF CONTENTS** | **PAGE NUMBER** |
| * Scheduler for archiving the SharePoint List item and sending multiple reminder before archiving | 2 |
|  |  |

**Scheduler for archiving the SharePoint List item and sending multiple reminder before archiving**

This is console application which is running on daily basis and check with current system date with expire date and archive SharePoint items to archive List, and it also check if any SharePoint items is going to expire in next 1 or 15 or 30 or 45 days it will send reminder email to business user if they wish to extend the expire date.

**Estimated Time Saved: 15 hrs/month** manual work (This may vary based on the action frequency)

**Implementation done :** June 2020

**code:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Microsoft.SharePoint.Client;

using SP = Microsoft.SharePoint.Client;

using System.Security;

using System.Net;

using System.Data;

using System.Configuration;

using System.Net.Mail;

using Microsoft.SharePoint.Client.Utilities;

namespace ItemArchiveUtilities

{

class Program

{

static void Main(string[] args)

{

SendEmail();

copydeletelistItem();

}

public static void copydeletelistItem()

{

try

{

string siteUrl = ConfigurationManager.AppSettings["siteurl"];

string SourceList = ConfigurationManager.AppSettings["sourcelist"];

string DestList = ConfigurationManager.AppSettings["Destinationlist"];

SP.ClientContext ctx = new SP.ClientContext(siteUrl);

ctx.Credentials = CredentialCache.DefaultCredentials;

SP.ClientContext context = ctx;

ctx.Load(ctx.Web);

ctx.ExecuteQuery();

List Sourcelist = ctx.Web.Lists.GetByTitle(SourceList);

ctx.Load(Sourcelist);

ctx.ExecuteQuery();

List DesinationList = ctx.Web.Lists.GetByTitle(DestList);

ctx.Load(DesinationList);

ctx.ExecuteQuery();

//caml query to fetch list item if expire date== current date

CamlQuery camqry = new CamlQuery();

camqry.ViewXml = camqry.ViewXml = @"<View>

<Query>

<Where><Eq><FieldRef Name='ExpDate'/><Value Type='DateTime'><Today/></Value></Eq></Where>

</Query>

</View>";

ListItemCollection listitems = Sourcelist.GetItems(camqry);

ctx.Load(listitems);

ctx.ExecuteQuery();

foreach (ListItem item in listitems)

{

Console.WriteLine("Item found for Archiveing");

ListItemCreationInformation newItemInfo = new ListItemCreationInformation();

ListItem newItem = DesinationList.AddItem(newItemInfo);

newItem["Title"] = item["Title"];

newItem["Description"] = item["Description"];

newItem["Date1"] = item["Date1"];

newItem.Update();

Console.WriteLine("------Item Archiving in Destination List-----");

ListItem oListItem = Sourcelist.GetItemById(item.Id);

oListItem.DeleteObject();

Console.WriteLine("------Item Deleting from Source List-----");

ctx.ExecuteQuery();

}

}

catch (Exception ex)

{

Errorlog(ex.Message, "CopyDeleteListItem");

}

}

//Sending reminder email

public static void SendEmail()

{

try

{

//int day = DateTime.Now.Day;

//int month = DateTime.Now.Month;

//int year = DateTime.Now.Year;

//string todaydt = month+"/"+day+"/"+year;

string siteUrl = ConfigurationManager.AppSettings["siteurl"];

SP.ClientContext ctx = new SP.ClientContext(siteUrl);

ctx.Credentials = CredentialCache.DefaultCredentials;

SP.ClientContext context = ctx;

ctx.ExecuteQuery();

string SourceList = ConfigurationManager.AppSettings["sourcelist"];

List Sourcelist = ctx.Web.Lists.GetByTitle(SourceList);

ctx.Load(Sourcelist);

ctx.ExecuteQuery();

CamlQuery camqry = new CamlQuery();

camqry.ViewXml = camqry.ViewXml = @"<View>

<Query>

<Where><Or><Or><Or>

<Eq><FieldRef Name='ExpDate' /><Value Type='DateTime'><Today OffsetDays='-4' /></Value></Eq>

<Eq><FieldRef Name='ExpDate' /><Value Type='DateTime'><Today OffsetDays='-3' /></Value></Eq>

</Or>

<Eq><FieldRef Name='ExpDate' /><Value Type='DateTime'><Today OffsetDays='-2' /></Value></Eq>

</Or>

<Eq><FieldRef Name='ExpDate' /><Value Type='DateTime'><Today OffsetDays='-1' /></Value></Eq>

</Or>

</Where>

</Query>

</View>";

ListItemCollection listitems = Sourcelist.GetItems(camqry);

ctx.Load(listitems);

ctx.ExecuteQuery();

foreach (ListItem item in listitems)

{

Console.WriteLine("Sending Reminder Emails");

int id = (int)item["ID"];

string title = item["Title"].ToString();

string expdate = item["ExpDate"].ToString();

var emailp = new EmailProperties();

emailp.BCC = new List<string> { "raghib.sarwer@Atlasair.com.com" };

emailp.To = new List<string> { "lakshmipriya.hari@Atlasair.com" };

emailp.Body = @"<html><body><h3>Hello Team</h3>" +

"<p><em>The Item Title is " + title + " is going to expire on " + expdate + " if you wish to extend the Expire please click on below link.</em></p>" +

"<p><a href='" + siteUrl + "/Lists/" + SourceList + "/EditForm.aspx?ID=" + id + "'>Click Here</a></p></body></html>";

emailp.Subject = "List item Expire reminder";

Utility.SendEmail(ctx, emailp);

ctx.ExecuteQuery();

}

}

catch (Exception ex)

{

Errorlog(ex.Message, "SendEmail");

}

}

public static void Errorlog(string errorMessage, string methodName)

{

string siteUrl = ConfigurationManager.AppSettings["siteurl"];

string ErrorlogList = ConfigurationManager.AppSettings["ErrorCapture"];

SP.ClientContext ctx = new SP.ClientContext(siteUrl);

ctx.Credentials = CredentialCache.DefaultCredentials;

SP.ClientContext context = ctx;

List errorLogList = ctx.Web.Lists.GetByTitle(ErrorlogList);

ListItemCreationInformation itemCreateInfo = new ListItemCreationInformation();

ListItem NewLogItem = errorLogList.AddItem(itemCreateInfo);

NewLogItem["Title"] = "ItemCopyDeleteScheduler";

NewLogItem["MethodName"] = methodName;

NewLogItem["ErrorMessage"] =errorMessage;

NewLogItem.Update();

ctx.ExecuteQuery();

}

}

}