

using Microsoft.SharePoint.Client;

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Linq;

using System.Data;

namespace Audit

{

public partial class Form1 : System.Windows.Forms.Form

{

System.Data.DataTable dt1;

FieldCollection fields;

public Form1()

{

InitializeComponent();

}

private string GetDate(DateTime dd)

{

string year1 = dd.Year.ToString();

string month1 = dd.Month.ToString();

string day1 = dd.Day.ToString();

return year1 + "-" + month1 + "-" + day1;

}

private string GetQuery()

{

DateTime dd = DateTime.Now.AddDays(-150);

string fecha = GetDate(dd);

string fecha2 = GetDate(date1.Value);

string fecha3 = GetDate(date2.Value);

string conditions = "<View><Query><Where><And><Gt><FieldRef Name='Created' /><Value IncludeTimeValue='TRUE' Type='DateTime'>DATE1</Value></Gt><Eq><FieldRef Name='Title' /><Value Type='Text'>xxxxx</Value></Eq></And></Where></Query></View>";

// RETRIEVE EMPL ID

if (t1.Text.Trim() != "")

{

conditions = conditions.Replace("xxxxx", t1.Text.Trim());

conditions = conditions.Replace("DATE1", fecha);

return conditions;

}

if (t2.Text.Trim() != "")

{

conditions = conditions.Replace("Title", "ATTUID");

conditions = conditions.Replace("xxxxx", t1.Text.Trim());

conditions = conditions.Replace("DATE1", fecha);

return conditions;

}

if (t1.Text.Trim() == "" && t2.Text.Trim() == "")

{

if (DateTime.Parse(fecha2) > DateTime.Parse(fecha3))

{

System.Windows.Forms.MessageBox.Show("Starting Date is After Until Date, Change Dates");

return "";

}

if (fecha2 == fecha3) fecha3 = fecha2 + "T12:00:00Z"; //2018-01-11T02:22:17Z

conditions = "<View><Query><Where><And><Geq><FieldRef Name='Created' /><Value IncludeTimeValue='TRUE' Type='DateTime'>" + fecha2 + " </Value></Geq><Leq><FieldRef Name='Created' /><Value IncludeTimeValue='TRUE' Type='DateTime'>" + fecha3 + "</Value></Leq></And></Where></Query></View>";

return conditions;

}

return "";

}

private void GetSharePointTitles()

{

string url = ConfigurationManager.AppSettings["SiteUrl"].ToString();

string listado = ConfigurationManager.AppSettings["ListTitle"].ToString();

ClientContext context = new ClientContext(url);

List list = context.Web.Lists.GetByTitle(listado);

context.Load(list);

context.ExecuteQuery();

fields = list.Fields;

context.Load(fields, collection => collection.Include(f => f.Title, f => f.InternalName));

context.ExecuteQuery();

}

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

{

string url = ConfigurationManager.AppSettings["SiteUrl"].ToString();

string listado = ConfigurationManager.AppSettings["ListTitle"].ToString();

button2.Visible = false;

gv.Visible = false;

label5.Visible = true;

ClientContext ctx = new ClientContext(url);

List list = ctx.Web.Lists.GetByTitle(listado);

CamlQuery query = new CamlQuery();

string condicion = GetQuery();

if (condicion == "") return;

// INFORMATION TO LOOKUP

var field2 = list.Fields.GetByInternalNameOrTitle("Automation Requestor");

var lookupField = ctx.CastTo<FieldLookup>(field2);

ctx.Load(lookupField);

ctx.ExecuteQuery();

var lookupListId = new Guid(lookupField.LookupList); //returns associated list id

var lookupList = ctx.Web.Lists.GetById(lookupListId);

ctx.Load(lookupList);

ctx.ExecuteQuery();

query.ViewXml = condicion;

ListItemCollection items = list.GetItems(query);

ctx.Load(items);

ctx.ExecuteQuery();

List<ListItem> itemList = items.ToList();

if (itemList.Count == 0)

{

System.Windows.Forms.MessageBox.Show("Change Starting Date");

return;

}

button2.Visible = true;

gv.Visible = true;

label5.Visible = false;

dt1 = new System.Data.DataTable();

dt1.Columns.Add("Employee ID", typeof(string));

dt1.Columns.Add("ATTUID", typeof(string));

dt1.Columns.Add("Employee Name", typeof(string));

dt1.Columns.Add("Attrition", typeof(string));

dt1.Columns.Add("Status", typeof(string));

dt1.Columns.Add("Comment", typeof(string));

dt1.Columns.Add("Created", typeof(string));

dt1.Columns.Add("Requestor", typeof(string));

dt1.Columns.Add("Modified", typeof(string));

dt1.Columns.Add("Work Order", typeof(string));

dt1.Columns.Add("Project", typeof(string));

dt1.Columns.Add("Location", typeof(string));

string name3 = t3.Text.Trim();

string choice = cb1.Text;

Field field;

List<Field> fieldList = fields.ToList();

foreach (ListItem item2 in itemList)

{

field = fieldList.Where(f => f.Title == "Automation Requestor").FirstOrDefault();

string requestor2 = ((FieldLookupValue)item2[field.InternalName]).LookupValue;

if (choice == "From Automation" && requestor2 != "ATT UID Deactivation") continue;

if (choice == "From a User" && requestor2 == "ATT UID Deactivation") continue;

field = fieldList.Where(f => f.Title == "Employee ID").FirstOrDefault();

string EmplID = item2[field.InternalName].ToString();

field = fieldList.Where(f => f.Title == "Emp First Name").FirstOrDefault();

string name1 = item2[field.InternalName].ToString();

field = fieldList.Where(f => f.Title == "Emp Last Name").FirstOrDefault();

string name2 = item2[field.InternalName].ToString();

name1 = name1 + " " + name2;

if (name3 != "" && name1.IndexOf(name3) < 0) continue;

string Status = item2["Status"].ToString();

string Created = item2["Created"].ToString();

string Modified = item2["Modified"].ToString();

string ATTUID = (item2["ATTUID"] == null) ? "" : item2["ATTUID"].ToString();

string Comments = (item2["Comments"] == null) ? "" : item2["Comments"].ToString();

string location = (item2["Location"] == null) ? "" : item2["Location"].ToString();

string project = (item2["Project"] == null) ? "" : item2["Project"].ToString();

field = fieldList.Where(f => f.Title == "Effective Date of Attrition").FirstOrDefault();

string DateAttr = item2[field.InternalName].ToString();

int pos1 = DateAttr.IndexOf(" ");

if (pos1 > 1) DateAttr = DateAttr.Substring(0, pos1);

field = fieldList.Where(f => f.Title == "WorkOrder Id").FirstOrDefault();

string wkId = (item2[field.InternalName] == null) ? "" : item2[field.InternalName].ToString();

dt1.Rows.Add(EmplID, ATTUID, name1, DateAttr, Status, Comments, Created, requestor2, Modified, wkId, project, location);

}

if (dt1 != null)

{

gv.DataSource = dt1;

gv.Columns[0].Width = 75;

gv.Columns[1].Width = 65;

gv.Columns[2].Width = 150;

gv.Columns[3].Width = 80;

gv.Columns[4].Width = 80;

}

else

System.Windows.Forms.MessageBox.Show("There is no data, change conditions");

}

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

{

Microsoft.Office.Interop.Excel.\_Application excel = new Microsoft.Office.Interop.Excel.Application();

Microsoft.Office.Interop.Excel.\_Workbook workbook = excel.Workbooks.Add(Type.Missing);

Microsoft.Office.Interop.Excel.\_Worksheet worksheet = null;

worksheet = workbook.ActiveSheet;

worksheet.Name = "Exported";

// headers

int ColumnIndex = 0;

foreach (System.Data.DataColumn col in dt1.Columns)

{

ColumnIndex++;

worksheet.Cells[1, ColumnIndex] = col.ColumnName;

}

// detail

int rowIndex = 0;

foreach (System.Data.DataRow row in dt1.Rows)

{

rowIndex++;

ColumnIndex = 0;

foreach (System.Data.DataColumn col in dt1.Columns)

{

ColumnIndex++;

worksheet.Cells[rowIndex + 1, ColumnIndex] = row[col.ColumnName];

}

}

excel.Visible = true;

workbook.Activate();

excel.ActiveWindow.Activate();

//excel.Application.WindowState = true;

//excel.Application.WindowState = -4137;

//workbook.SaveAs("C:\\temp\\sss3.xls");

//excel.Quit();

//workbook = null;

//excel = null;

}

private void Form1\_Load(object sender, EventArgs e)

{

date1.Text = DateTime.Parse(date1.Text).AddDays(-10).ToString();

GetSharePointTitles();

cb1.Items.Add("All");

cb1.Items.Add("From Automation");

cb1.Items.Add("From a User");

cb1.SelectedIndex = 0;

}

}

}

BEST OPTION

using Microsoft.SharePoint.Client;

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Linq;

using System.Data;

namespace Audit

{

public partial class AuditForm : System.Windows.Forms.Form

{

FieldCollection fields;

Lookup<string, string> fieldmap;

string url = ConfigurationManager.AppSettings["SiteUrl"].ToString();

string listTitle = ConfigurationManager.AppSettings["ListTitle"].ToString();

public AuditForm()

{

InitializeComponent();

}

private string GetDate(DateTime dd)

{

string year1 = dd.Year.ToString();

string month1 = dd.Month.ToString();

string day1 = dd.Day.ToString();

return year1 + "-" + month1 + "-" + day1;

}

private string GetQuery()

{

DateTime dd = DateTime.Now.AddDays(-150);

string threeMonthsBeforeTodaysDate = GetDate(dd);

string startDate = GetDate(txtStartDate.Value);

string endDate = GetDate(txtEndDate.Value);

string conditions = "<View>" +

"<Query>" +

"<Where>" +

"<And>" +

"<Gt>" +

"<FieldRef Name='Created' />" +

"<Value IncludeTimeValue='TRUE' Type='DateTime'>DATE1</Value>" +

"</Gt>" +

"<Eq>" +

"<FieldRef Name='Title' /><Value Type='Text'>xxxxx</Value>" +

"</Eq>" +

"</And>" +

"</Where>" +

"</Query>" +

"</View>";

// RETRIEVE EMPL ID

if (txtEmployeeID.Text.Trim() != "")

{

conditions = conditions.Replace("xxxxx", txtEmployeeID.Text.Trim());

conditions = conditions.Replace("DATE1", threeMonthsBeforeTodaysDate);

return conditions;

}

// RETRIEVE ATTUID

if (txtATTUID.Text.Trim() != "")

{

conditions = conditions.Replace("Title", "ATTUID");

conditions = conditions.Replace("xxxxx", txtEmployeeID.Text.Trim());

conditions = conditions.Replace("DATE1", threeMonthsBeforeTodaysDate);

return conditions;

}

if (txtEmployeeID.Text.Trim() == "" && txtATTUID.Text.Trim() == "")

{

if (DateTime.Parse(startDate) > DateTime.Parse(endDate))

{

System.Windows.Forms.MessageBox.Show("Starting Date is After Until Date, Change Dates");

return "";

}

if (startDate == endDate) endDate = startDate + "T12:00:00Z"; //2018-01-11T02:22:17Z

string \_requestor = GetRequestorType();

string \_requestorAnd = (\_requestor =="") ? "" : "<And";

string \_requestorAnd2 = (\_requestor == "") ? "" : "</And";

string \_name = GetNameLikeCondition();

conditions = "<View>" +

"<Query>" +

"<Where>" +

"<And>" +

"<Geq>" +

"<FieldRef Name='Created' />" +

"<Value IncludeTimeValue='TRUE' Type='DateTime'>" + startDate + " </Value>" +

"</Geq>" +

\_requestorAnd +

"<Leq>" +

"<FieldRef Name='Created' />" +

"<Value IncludeTimeValue='TRUE' Type='DateTime'>" + endDate + "</Value>" +

"</Leq>" +

\_requestor+

"</And>"+ \_requestorAnd2+

"</Where>"+

"</Query>"+

"</View>";

return conditions;

}

return "";

}

private string GetNameLikeCondition()

{

string \_condition = "";

string \_partialName = txtPartialName.Text.Trim();

if (\_partialName != "")

{

\_condition = " <Or>" +

"<Contains>" +

"<FieldRef Name = 'Emp\_x0020\_First\_x0020\_Name'/>" +

"<Value Type = 'Text' >" + \_partialName + "</Value>" +

"</Contains>" +

"<Contains>" +

"<FieldRef Name = 'Emp\_x0020\_Last\_x0020\_Name'/>" +

"<Value Type = 'Text' >" + \_partialName + "</Value>" +

"</Contains>" +

"</Or>" +

"</And>";

}

return \_condition;

}

private string GetRequestorType()

{

if (!cbRequestor.Visible) return "";

string \_type = cbRequestor.Text;

string \_condition = "";

if (\_type == "All") return "";

if (\_type == "From Automation")

{

\_condition = "<And>" +

"<Eq>" +

"<FieldRef Name = 'Automation\_x0020\_Requestor'/>" +

"<Value Type = 'User' > ATT UID Deactivation</Value>" +

"</Eq>" +

"</And>";

}

else

{

\_condition = "<And>" +

"<Neq>" +

"<FieldRef Name = 'Automation\_x0020\_Requestor'/>" +

"<Value Type = 'User' > ATT UID Deactivation</Value>" +

"</Neq>" +

"</And>";

}

return \_condition;

}

private void GetSharePointTitles()

{

ClientContext context = new ClientContext(url);

List list = context.Web.Lists.GetByTitle(listTitle);

context.Load(list);

context.ExecuteQuery();

fields = list.Fields;

context.Load(fields, collection => collection.Include(f => f.Title, f => f.InternalName));

context.ExecuteQuery();

fieldmap = (Lookup<string, string>)fields.ToLookup(x => x.Title, x => x.InternalName);

}

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

{

ExcelBtn.Visible = false;

DataGrid.Visible = false;

label5.Visible = true;

ClientContext ctx = new ClientContext(url);

List list = ctx.Web.Lists.GetByTitle(listTitle);

CamlQuery query = new CamlQuery();

string condicion = GetQuery();

if (condicion == "") return;

query.ViewXml = condicion;

ListItemCollection items = list.GetItems(query);

ctx.Load(items);

ctx.ExecuteQuery();

List<ListItem> itemList = items.ToList();

if (itemList.Count == 0)

{

System.Windows.Forms.MessageBox.Show("Change Starting Date");

return;

}

ExcelBtn.Visible = true;

DataGrid.Visible = true;

label5.Visible = false;

List<DataResultItem> dataresult = new List<DataResultItem>();

dataresult = itemList.Select(x => {

string \_ATTUID = x.FieldValues[fieldmap["ATTUID"].FirstOrDefault()] == null ? "" : x.FieldValues[fieldmap["ATTUID"].FirstOrDefault()].ToString();

string \_Comments = x.FieldValues[fieldmap["Comments"].FirstOrDefault()] == null ? "" : x.FieldValues[fieldmap["Comments"].FirstOrDefault()].ToString();

string \_project = x.FieldValues[fieldmap["Project"].FirstOrDefault()] == null ? "" : x.FieldValues[fieldmap["Project"].FirstOrDefault()].ToString();

string \_location = x.FieldValues[fieldmap["Location"].FirstOrDefault()] == null ? "" : x.FieldValues[fieldmap["Location"].FirstOrDefault()].ToString();

string \_workOrder = x.FieldValues[fieldmap["WorkOrder Id"].FirstOrDefault()] == null ? "" : x.FieldValues[fieldmap["WorkOrder Id"].FirstOrDefault()].ToString();

string \_requestor = ((FieldUserValue)x.FieldValues[fieldmap["Automation Requestor"].FirstOrDefault()]).LookupValue;

return new DataResultItem

{

EmployeeID = x.FieldValues[fieldmap["Employee ID"].FirstOrDefault()].ToString(),

EmployeeName = x.FieldValues[fieldmap["Emp First Name"].FirstOrDefault()].ToString() + " " + x.FieldValues[fieldmap["Emp Last Name"].FirstOrDefault()].ToString(),

Status = x.FieldValues[fieldmap["Status"].FirstOrDefault()].ToString(),

WorkOrder = \_workOrder,

Created = x.FieldValues[fieldmap["Created"].FirstOrDefault()].ToString(),

Modified = x.FieldValues[fieldmap["Modified"].FirstOrDefault()].ToString(),

ATTUID = \_ATTUID,

Comments = \_Comments,

Location = \_location,

Project = \_project,

Requestor = \_requestor

};

}).ToList();

DataGrid.DataSource = dataresult;

}

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

{

Microsoft.Office.Interop.Excel.\_Application excel = new Microsoft.Office.Interop.Excel.Application();

Microsoft.Office.Interop.Excel.\_Workbook workbook = excel.Workbooks.Add(Type.Missing);

Microsoft.Office.Interop.Excel.\_Worksheet worksheet = null;

worksheet = workbook.ActiveSheet;

worksheet.Name = "Exported";

int \_NumCols = DataGrid.Columns.Count;

int \_NumRows = DataGrid.Rows.Count;

for (int \_col = 0; \_col < \_NumCols; \_col++)

{

worksheet.Cells[1, \_col + 1] = DataGrid.Columns[\_col].HeaderText.ToString();

for (int \_row = 0; \_row < \_NumRows; \_row++)

{

var \_value = DataGrid.Rows[\_row].Cells[\_col].Value;

if (\_value != null)

worksheet.Cells[\_row + 1, \_col + 1] = \_value.ToString();

}

}

excel.Visible = true;

workbook.Activate();

excel.ActiveWindow.Activate();

//excel.Application.WindowState = true;

//excel.Application.WindowState = -4137;

//workbook.SaveAs("C:\\temp\\sss3.xls");

//excel.Quit();

//workbook = null;

//excel = null;

}

private void AuditForm\_Load(object sender, EventArgs e)

{

txtStartDate.Text = DateTime.Parse(txtStartDate.Text).AddDays(-10).ToString();

GetSharePointTitles();

cbRequestor.Items.Add("All");

cbRequestor.Items.Add("From Automation");

cbRequestor.Items.Add("From a User");

cbRequestor.SelectedIndex = 0;

}

}

}