**Task:**

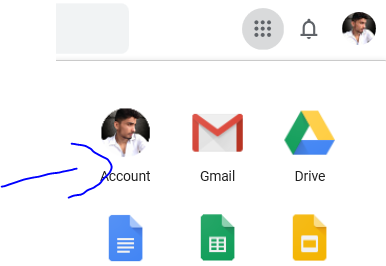
Task is to Select multiple customers form “CustInvoiceJournal” and Run Reports of customers and send invoices to their respective person via email template.

For this I made sub tasks.

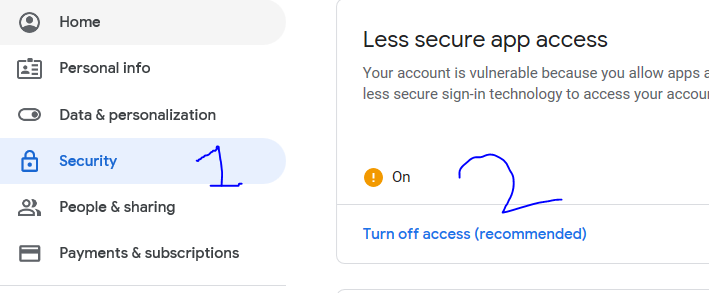
1)First I configured Gmail server with my gmail account for this.

Follow this:

First Click on the Menu button and left side with bell icon and then click on the Account, where is blue arrow .



Then click on Security and Mark “Yes” to “Less secure App”



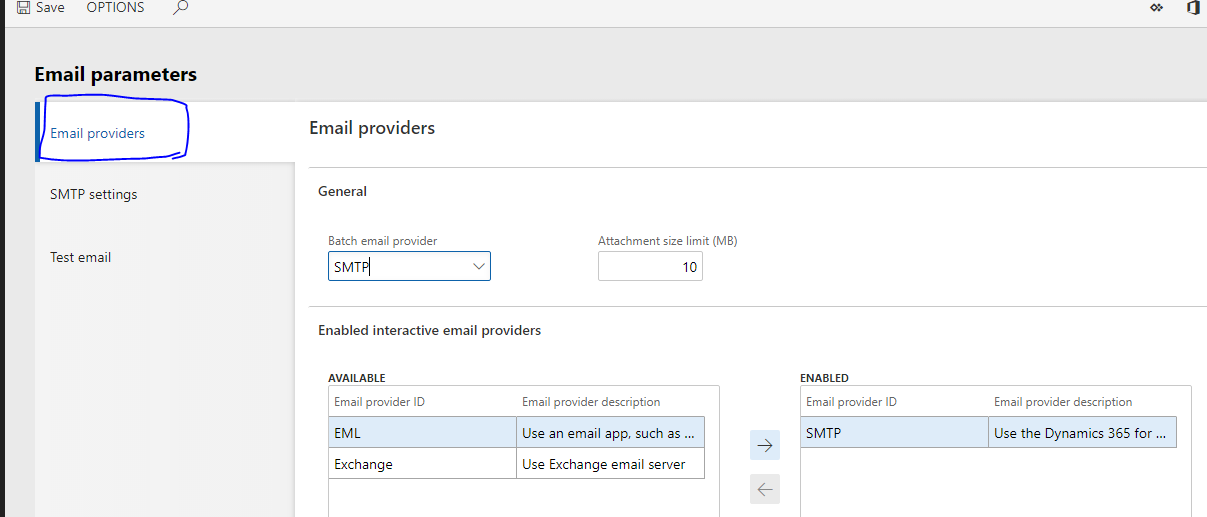
Then click Save. The Gmail id is now configured.

**Task II:**

Now Set the Email parameters.

Open D365 Dashboard Goto ------🡪 System Administration then Go to ---------🡪Email and the click Email Parameters.

Now set the parameters settings like show in screen short.

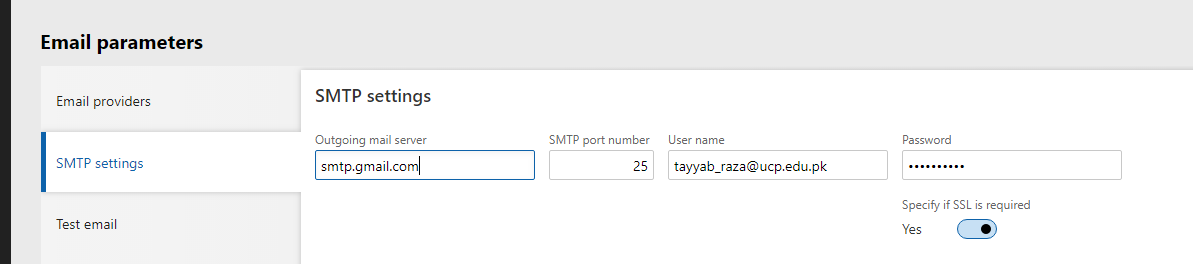


Now Set the SMTP Settings:

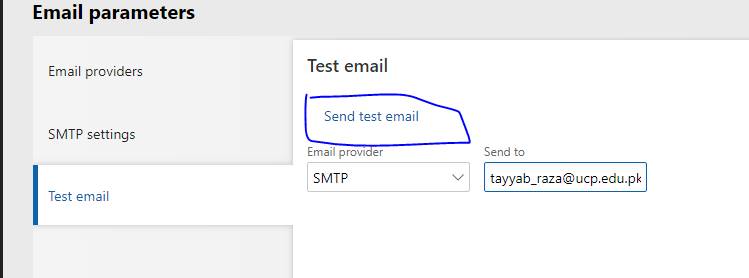
Outgoing gmail server = smtp.gmail.com

SMTP port number = 25 // you can set manual

User name will be your gmail id.



Now go to test mail and click the send Test Email

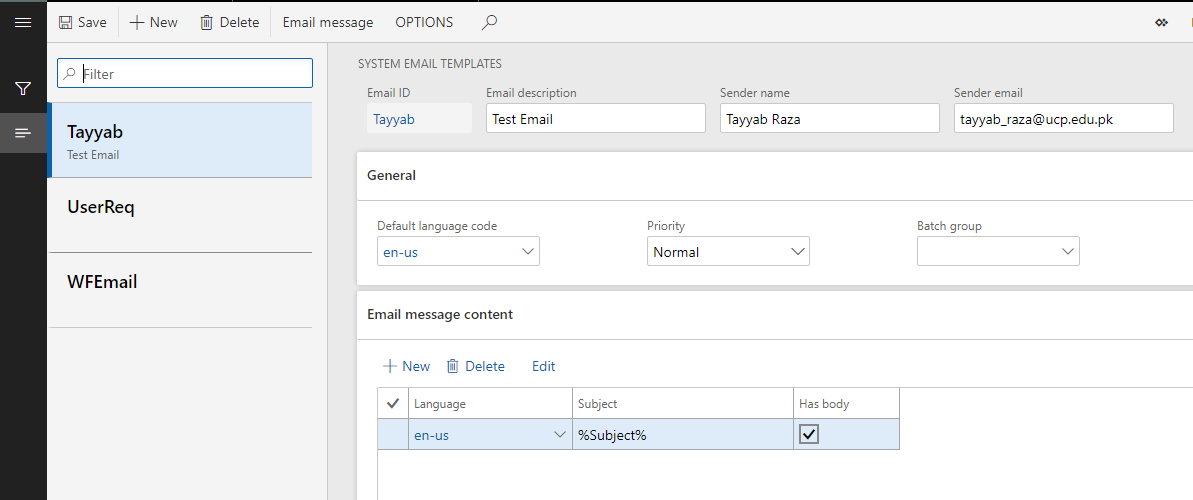


The email written in Send to is the email that we configured in first step.

If the email cannot be sent to your account you should check again Step 1.

**Now Make the Template:**

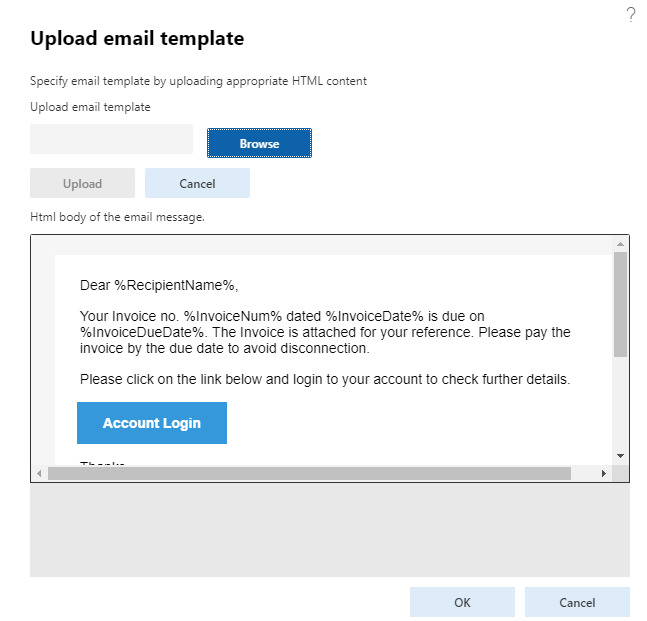
Goto ------🡪 System Administration then Go to ---------🡪Email and the click Email Parameters.



You can see in my insatance there are three templates “Tayyab” template is mine. You can click new button and make your own. When you click new a form open and enter the proper values.

Now click on the Email message button for uploading the template. You can google the template or make your own.

Click on browse button and add you template file. Then click upload.



**Task4**

Add button to the “Cust invoice Journal” form action pane and then override it’s click method.

[FormControlEventHandler(formControlStr(CustInvoiceJournal, EmailButton), FormControlEventType::Clicked)]

public static void EmailButton\_OnClicked(FormControl sender, FormControlEventArgs e)

{

#define.SubjectToken('Subject')

#define.CustNameToken('RecipientName')

#define.InvoiceNumberToken('InvoiceNum')

#define.InvoiceDateToken('InvoiceDate')

#define.InvoiceDueDateToken('InvoiceDueDate');

Map templateTokens;

container emailCon;

container pathCon;

templateTokens = new Map(Types::String, Types::String);

DirPartyContactInfoView dirPartyContactInfoView;

LogisticsElectronicAddress logisticsElectronicAddress;

DirPartyTable DirPartyTable;

CustTable Customer;

CustInvoiceJour custinvoicejour;

Filename path;

// This Code is for selected Record on the form.

FormDataSource custinvoicejour\_ds;

custinvoicejour\_ds = sender.formRun().dataSource();

*// recordscount = custinvoicejour\_ds.recordsMarked().lastIndex();*

custinvoicejour = custinvoicejour\_ds.getFirst(1);

while(custinvoicejour)

{

*//Varaiables for report run*

SrsReportRunController ssrsController = new SrsReportRunController();

SalesInvoiceContract salesInvoiceContract = new SalesInvoiceContract();

SRSPrintDestinationSettings printerSettings;

CustInvoiceJour custInvoiceReportDS;

//Map the values in templates we recently uploaded in email template.

Customer = CustTable::find(custinvoicejour.InvoiceAccount);

templateTokens.insert(#SubjectToken,"Invoice Details");

templateTokens.insert(#CustNameToken,Customer.name());

templateTokens.insert(#InvoiceNumberToken,custinvoicejour.InvoiceId);

info(strFmt("InvoiceID %1",custinvoicejour.InvoiceId));

templateTokens.insert(#InvoiceDateToken,date2Str(custinvoicejour.InvoiceDate,213,DateMonth::Digits2,DateSeparator::Hyphen,DateDay::Digits2,DateSeparator::Hyphen,DateYear::Digits4));

templateTokens.insert(#InvoiceDueDateToken,date2Str(custinvoicejour.DueDate,213,DateMonth::Digits2,DateSeparator::Hyphen,DateDay::Digits2,DateSeparator::Hyphen,DateYear::Digits4));

//this code is for picking up the email address where we send our emails.

select Locator from dirPartyContactInfoView

where dirPartyContactInfoView.Party == Customer.Party &&

dirPartyContactInfoView.Type == LogisticsElectronicAddressMethodType::Email &&

dirPartyContactInfoView.IsPrimary == NoYes::No;

if(dirPartyContactInfoView.Locator)

{

emailCon += dirPartyContactInfoView.Locator;

}

emailCon += Customer.email();

*//Report Logic*

//this code is for run the report through code.

select firstOnly custInvoiceReportDS

where custInvoiceReportDS.SalesId == custinvoicejour.SalesId;

*// Define which report design to print*

ssrsController.parmReportName(ssrsReportStr(SalesInvoice, Report));

*// Ddefine how we want to execute the report (right now or batch style)*

*// ssrsController.parmExecutionMode(SysOperationExecutionMode::Synchronous);*

*// Hide the report dialog*

ssrsController.parmShowDialog(false);

*// Need to populate the required parms for the current sales order controller*

salesInvoiceContract.parmRecordId(custInvoiceReportDS.RecId);

salesInvoiceContract.parmCountryRegionISOCode(SysCountryRegionCode::countryInfo());

salesInvoiceContract.parmDocumentTitle('Invoice');

*// Link the contract to the controller so we know how to run the dp*

ssrsController.parmReportContract().parmRdpContract(salesInvoiceContract);

*// Link the printer settings to the controller*

printerSettings = ssrsController.parmReportContract().parmPrintSettings();

*// Print to HTML and always overwrite if the file exists*

printerSettings.printMediumType(SRSPrintMediumType::File);

printerSettings.fileFormat(SRSReportFileFormat::PDF);

printerSettings.overwriteFile(true);

path = ("C:\\Temp\\Inv\\"+custinvoicejour.InvoiceId+".pdf");

*//info(strFmt("Path Is: %1",path));*

printerSettings.fileName(@path);

*// Run & save the report*

ssrsController.runReport();

pathCon += path;

*// SendEmail::Mail('Tayyab','en-us','tayyab\_raza@ucp.edu.pk',templateTokens,path,'',true,'',true);*

*//emailCon=conNull();*

custinvoicejour = custinvoicejour\_ds.getNext();

}

//Now call the email method. Send email is my class.

SendEmail::EmailConfiguration('Tayyab',templateTokens,emailCon,pathCon);

//now I am adding it’s emaiconfiguration method.

public static void EmailConfiguration(SysEmailId emailTempID, Map map, container \_tos,container \_attachmentPaths = conNull())

{

str to;

str path;

int tocount = conLen(\_tos);

*//int pathscount = conLen(\_attachmentPaths);*

int i=1;

while(i<=tocount)

{

to = conPeek(\_tos,i);

path = conPeek(\_attachmentPaths,i);

SendEmail ::Email(emailTempID,map,to,path);

}

}

//from upper method I am configuring the email parameters like sender email address and attachment paths.Now I am adding the Email method where we can actually send the Mail and attachment to the respective customers.

Public static void Email(SysEmailId emailTempID, Map map, str \_tos,str \_attachmentPaths = '')

{

System.Net.Mail.MailMessage mailMessage;

System.Net.Mail.SmtpClient myMail;

System.Net.Mail.MailAddressCollection mailcoll;

System.Net.Mail.MailAddress mailFrom;

System.Net.Mail.MailAddress mailTo;

System.Net.Mail.MailAddress mailBCC;

System.Net.Mail.Attachment attachment;

System.Net.Mail.AttachmentCollection attachementCollection;

SysEmailTable table = SysEmailTable::find(emailTempID);

SysEmailMessageTable message =SysEmailMessageTable::find(emailTempID, table.DefaultLanguage);

str receiverMailAddress;

SysEmailContents mailBody;

str smtpServer;

str mailSubject;

str CcMailAddress;

int SMTPPort;

System.Exception excObj;

FileName filename;

#File

str mail;

userinfo userInfo;

str pwd;

SysEmailParameters parameters = SysEmailParameters::find();

new InteropPermission(InteropKind::ClrInterop).assert();

filename = \_attachmentPaths;

receiverMailAddress = \_tos;

CcMailAddress = \_tos;

mailFrom = new System.Net.Mail.MailAddress(parameters.SMTPUserName,"Name");

mailTo = new System.Net.Mail.MailAddress(receiverMailAddress);

mailBCC = new System.Net.Mail.MailAddress(CcMailAddress,"");

mailcoll = new System.Net.Mail.MailAddressCollection();

mailBody = SysEmailMessage::stringExpand(Message.Mail,map);

*//mailBody = WebLet::weblets2Html4Help(mailBody, '');*

*//mailMessage.set\_IsBodyHtml(True);*

mailSubject = SysEmailMessage::stringExpand(Message.Subject,map);

try

{

smtpServer = SysEmaiLParameters::find(false).SMTPRelayServerName;*// using the SMTP server ip //setup in email Parameters*

mailMessage = new System.Net.Mail.MailMessage(mailFrom,mailTo);

mailmessage.set\_Subject(mailSubject);

mailmessage.set\_Body(mailBody);

mailMessage.set\_IsBodyHtml(true);

attachementCollection = mailMessage.get\_Attachments();

attachment = new System.Net.Mail.Attachment(filename);

attachementCollection.Add(attachment);

SMTPPort = SysEmaiLParameters::find(false).SMTPPortNumber;

myMail = new System.Net.Mail.SmtpClient(smtpServer, SMTPPort);

myMail.set\_EnableSsl(true); *// For SSL enabled mail servers. Ex: gmail, smtp.gmail.com, port 465 or 587*

pwd = SysEmaiLParameters::password();

mymail.set\_Credentials(New System.Net.NetworkCredential(parameters.SMTPUserName, pwd));

mymail.Send(mailmessage);

info("sent");

}

catch

{

excObj = CLRInterop::getLastException();

if(excObj)

{

error(excObj.ToString());

}

}

mailMessage.Dispose();

CodeAccessPermission::revertAssert();

}

//Now we need to delete the report from local system.

public static void deleteReportFile(str \_ReportPath)

{

str ReportPath = \_ReportPath;

info("Now in delete Method");

if(!ReportPath)

{

info("No file in local to remove");

}

else

{

WinAPI::deleteFile(@ReportPath);

info("Success");

}

}