

EMAIL PROGRAMMING

By using Email Programming, we can send the Email Notifications to the required users with the required subject, content and attachments based on the need.

Email Programming supports both "Outbound Email Services" and "Inbound Email Services".

Apex provides "Messaging" namespace which contains a group of readymade classes and interfaces to communicate with the people through Email Notifications.

1. Outbound Email Services:

To send the Email Notifications to the required users.

i. Messaging.SingleEmailMessage Class

ii. Messaging.MassEmailMessage Class

2. Inbound Email Services:

To receive the Email Notifications from the External Email Id's.

Messaging.InboundEmailHandler Interface.

|

----> Method: HandleInboundEmail()

SingleEmailMessage Class: By using this class, we can send the Email Notifications to one or more users with the required subject, content and attachments.

While Sending the Email, we can specify the To Addresses, BCC Addresses, CC Addresses, Subject, Content, Attachments along with the email.

We can add both "Text Attachments and Binary Attachments" along with the email.

We can Track the Email Delivery Status by using "Email Logs".

It provides a Collection of Instance Methods, used to supply the required input.

Syntax: Messaging.SingleEmailMessage <objectName> = new
Messaging.SingleEmailMessage();

Methods:

Ex: Messaging.SingleEmailMessage sEmail = new Messaging.SingleEmailMessage();

1. SetToAddresses(<Array / Collection>)

Ex:String[] toAddress = new String[]{'ram.kumar@gmail.com','shyam@gmail.com'};
sEmail.SetToAddresses(toAddress);

2. SetCCAddresses(<Array / Collection>)

Ex: List<String> ccAddress = new <String>{'suresh@gmail.com','Santhosh@tcs.com'};
sEmail.SetCCAddresses(ccAddress);

3. SetBCCAddresses(<Array / Collection>)

Ex: String[] bccAddress = new String[]{'ram.k@gmail.com','shyam@gmail.com'};
sEmail.SetBCCAddresses(bccAddress);

4. SetSenderDisplayName(String displayName)

Ex: sEmail.SetSenderDisplayName('ICICI Bank Credit Card Department');

5. SetReplyTo(String emailAddress)

Ex: sEmail.SetReplyTo('support@icicibank.com');

6. SetSubject(String emailSubject)

Ex: sEmail.SetSubject('Email Subject to be Delivered');

7. SetPlainRequestBody(String plainTextContent)

Ex: sEmail.SetPlainRequestBody('Plain Text Email Content');

8. SetHTMLBody(String htmlContent)

Ex: sEmail.SetHTMLBody('HTML Formatted Email Content');

9. SetFileAttachments(List<Messaging.EmailFileAttachment> <refName>):

Ex: sEmail.SetFileAttachments(<refName>);

Messaging.EmailFileAttachment Class: By using this Class, we can prepare the required attachments at runtime through programming which can be shared to the people based on the need.

We can add one/more attachments along with the Email.

It provides a collection of instance methods to prepare the attachment file.

Syntax

Messaging.EmailFileAttachment <refName> = new Messaging.EmailFileAttachment();

Methods

Ex: Messaging.EmailFileAttachment attach = new Messaging.EmailFileAttachment();

1. SetFileName(String fileName)

Ex: attach.SetFileName('Ram Kumar.pdf');

2. SetContentType(EmailContentType):

Ex: attach.SetContentType('Application/pdf');

3. Body

Ex: attach.body = Blob.topdf(<EmailContent>);

Sending Email: We can send the Email Notifications to the specified users by using "SendEmail()" method. We can track the result by using "SendEmailResult" class.

Syntax:

List<Messaging.SendEmailResult> <refName> =

Messaging.SendEmail(List<Messaging.SingleEmailMessage> <refName>);

(OR)

```
List<Messaging.SendEmailResult> <refName> = Messaging.SendEmail(new  
Messaging.SingleEmailMessage[]{{<refName>}});
```

Use Case: Write an apex program, to create a lead record in the object and send the Email notification to lead person with the required Subject and Content.

Write the Email Body/Content inside a PDF File and add the PDF File as an attachment along with the Email.

Class Code:

```
public class MessagingHelper  
{  
    Public static void CreateLeadRecord()  
{  
    Lead ldRecord = new Lead();  
  
    ldRecord.FirstName = 'Sanjeev';  
    ldRecord.LastName = 'Kumar';  
    ldRecord.Email = 'sanjeev.yr2010@gmail.com';  
    ldRecord.Title = 'Technical Lead';  
    ldRecord.Phone = '9900445599';  
    ldRecord.Fax = '9900334466';  
    ldRecord.Rating = 'Hot';  
    ldRecord.Industry = 'Technology';  
    ldRecord.AnnualRevenue = 2300000;  
    ldRecord.Company = 'Infosys Inc.';  
    ldrecord.Status = 'Working - Contacted';  
    ldRecord.City = 'Pune';  
    ldRecord.State = 'Maharashtra';  
    ldRecord.Country = 'India';  
    ldRecord.Website = 'www.gmail.com';  
    ldRecord.LeadSource = 'Web';  
  
    Insert ldrecord;  
  
    if(ldRecord.Id != Null)  
{
```

```

        System.debug('Lead Record Inserted Successfully, with Id .....: '+ ldRecord.id);

        // Call the Method to Send Email Alert
        SendNotificationEmailToLead(ldrecord);
    }
}

Public static void SendNotificationEmailToLead(Lead ld)
{
    if(ld.Id != Null && ld.Email != Null)
    {
        // Write the Code to send Email Alert
        Messaging.SingleEmailMessage sEmail = new Messaging.SingleEmailMessage();

        String[] toAddress = new String[]{ld.Email,ksr.1986@gmail.com};
        sEmail.setToAddresses(toAddress);

        List<String> ccAddress = new List<String>{'venkatnaidu2014@gmail.com'};
        //sEmail.setCcAddresses(ccAddress);

        String[] bccAddress = new String[]{satishbk007@gmail.com};
        //sEmail.setBccAddresses(bccAddress);

        sEmail.setSenderDisplayName('ICICI Bank Personal Loans.');
```

sEmail.setReplyTo('loans@icicibank.com');

```

        String emailSubject = 'Congratulations '+ ld.FirstName + ' '+ ld.LastName +
                                '...!! Your Loan Applicaiton has been Submitted Successfully.';
        sEmail.setSubject(emailSubject);

        String emailContent = 'Dear '+ ld.FirstName + ' '+ ld.LastName + ',<br/><br/>' +
                                'Thanks for Showing your Interest in our Banking Services. <br/><br/>' + 'We are Pleased to
                                inform you that your Personal Loan Application has been Submitted Successfully.
```


'+ 'Your Loan Application is under Process, one of our Relationship Manager will contact you shortly.

'+ 'Here are your Loan Application Details.....:

'+ 'Loan Application Number: '+ Id.Id+ '
Applicant Name is.....: '+ Id.FirstName + ' '+ Id.LastName+ '
Current Employer.....: '+ Id.Company + '
Current Designation.....: '+ Id.Title+ '
Organization Type.....: '+ Id.Industry+ '
Contact Number: '+ Id.Phone+

'
Email Address: '+ Id.Email+ '
Fax Number is.....: '+ Id.Fax+

'
Current CTC: '+ Id.AnnualRevenue+ '
Lead Source.....: '+ Id.LeadSource+ '
Rating Value is.....: '+ Id.Rating+ '
Loan Application Status.....: '+ Id.Status+ '
Communication Address.....: '+ Id.City + ', '+ Id.State + ', '+ Id.Country+

'

 Please Contact on the below address, if any queries.

'+

'Please Click on the below link to view the record. Click Here

'+ '***<i>This is a System-Generated Email. Please Do Not Reply.</i>

'+ 'Thanks & Regards,
 ICICI Bank Personal Loans,
 Contact us: +1 (800)787-8899,
 Email us: loans@icicibank.com';

sEmail.setHtmlBody(emailContent);

// Write the Code to Prepare an Attachments

```
Messaging.EmailFileAttachment attachment = new
    Messaging.EmailFileAttachment();
```

```
attachment.setFileName(Id.FirstName + ' '+ Id.LastName+'.pdf');
```

```
attachment.setContentType('Application/pdf');
```

```
attachment.body = Blob.toPdf(emailContent);
```

```
// Prepare a list collection of Attachment and add the file to Collection
```

```
List<Messaging.EmailFileAttachment> lstAttachments = new
    List<Messaging.EmailFileAttachment>();
```

```
lstAttachments.Add(attachment);
```

```
// Add the PDF File as an Attachment with Email
```

```
sEmail.setFileAttachments(lstAttachments);
```

```
// Send the Email Alert
```

```
Messaging.SendEmailResult[] results = Messaging.sendEmail(new
    Messaging.SingleEmailMessage[] {sEmail});
```

```
}
```

```
}
```

```
}
```

Execution: MessagingHelper.CreateLeadRecord();