

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Templated EmailMessaging through apex code – things to remember

We all might have come across situations where emails need to be fired from apex code. In this blog, I would like to highlight some areas which sometimes create confusion for developers.

As you all know, Messaging.SingleEmailMessage is the apex class that we use for sending emails via apex code. You can send plain text emails as well as template emails using this apex class.

A straight forward implementation of plain text email sending using Messaging.SingleEmailMessaging would look like:

|  |  |
| --- | --- |
|  | Messaging.SingleEmailMessage mail =new Messaging.SingleEmailMessage(); |
|  | mail.setToAddresses(To); |
|  | mail.setPlainTextBody(Body); |
|  | mail.setReplyTo(FromEmail); |
|  | mail.setSubject(Subject); |
|  | Messaging.sendEmail(new Messaging.SingleEmailMessage[]{mail}); |

[**view raw**](https://gist.github.com/abhisheksubbu/6cf675967a282ec94dfd/raw/d1d05b48c0b81c18aef553e82d9217fbac82f68d/plain.java)[**plain.java**](https://gist.github.com/abhisheksubbu/6cf675967a282ec94dfd#file-plain-java) hosted with  by [**GitHub**](https://github.com/)

A typical implementation of templated email sending using Messaging.SingleEmailMessaging would look like:

|  |  |
| --- | --- |
|  | Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage(); |
|  | mail.setTargetObjectId(contactId); |
|  | mail.setTemplateId(templateId); |
|  | mail.setSenderDisplayName(senderDisplayName); |
|  | mail.setReplyTo(fromEmail); |
|  | mail.setWhatId(contextId); |
|  | Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail }); |

[**view raw**](https://gist.github.com/abhisheksubbu/343e112030561fff2112/raw/e476163de830f20b079ca45f6ab423ee98443d80/template.java)[**template.java**](https://gist.github.com/abhisheksubbu/343e112030561fff2112#file-template-java) hosted with  by [**GitHub**](https://github.com/)

On the template email code, there are two methods which are referred on line no.2 and line no.6 that set the targetObjectId and whatId of the email object. TargetObjectId refers to the person’s Id to whom you are going to send email. This would probably be a contact or lead’s Id. WhatId refers to the context on which the template is loaded. This means that if we are using merge fields on the email template then whatId refers to the base object Id used in the merge fields.

As an example, lets think of a use-case where you send an email containing the recent product related information based on his/her purchase activities to a contact in your salesforce org. In this case, the contactId becomes the targetObjectId and the productId becomes the whatId. In this way, you could set an email template that contains the merge fields for product name, product price etc and fire that email to the concerned contact.

In Salesforce apex documentation, they say that whatId is an optional field. The reason why they say is because if your email template doesn’t contain any merge fields, then whatId need not be set from apex code. But if you email template contains merge fields, it is recommended that whatId is set in the apex code. You need not worry about targetObjectId and whatId when you deal with plain text emails. These properties are important when email is fired using an email template in salesforce.