Cello How To-Do guide

Notification Management

Contents

1. Notification Management

Every application would use some kind of notifications, to notify user's i.e. sending an email, uploading documents to remote ftp etc. For example: User password mail, Leave request email, financial report uploading etc. When it comes to notification, there are two parts attached to it, they are

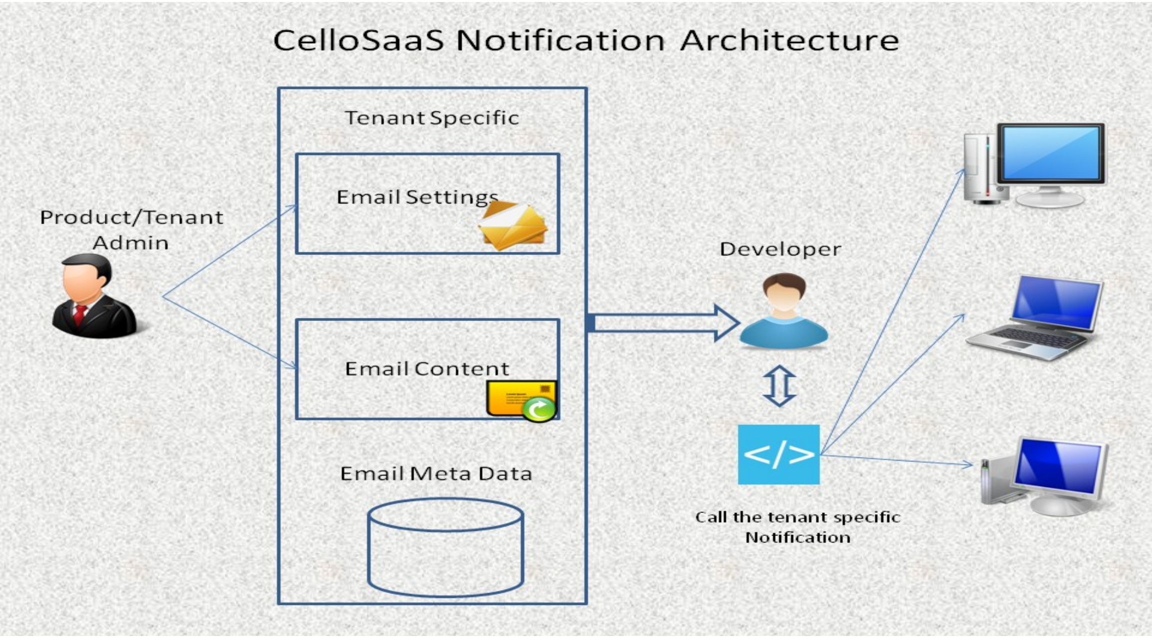
1. Dispatch details – SMTP Details, Sender Details etc
2. Email Content – Actual Content sent to Users

Developers might spend significant amount of time in taking care of Notifications part of the application. Cello Framework comes along with notification management capabilities. Developers can use these notification manager APIs to send email notification, uploading Files or doing system notifications.

CelloSaaS supports three types of notifications.

1. Email Notification- To Send email notifications to users
2. FTP Notification –To upload files to Secured and Unsecured FTP
3. System Notification. – To send internal notifications such as messages to users internally within the application.

1.1. CelloSaaS Notification Architecture



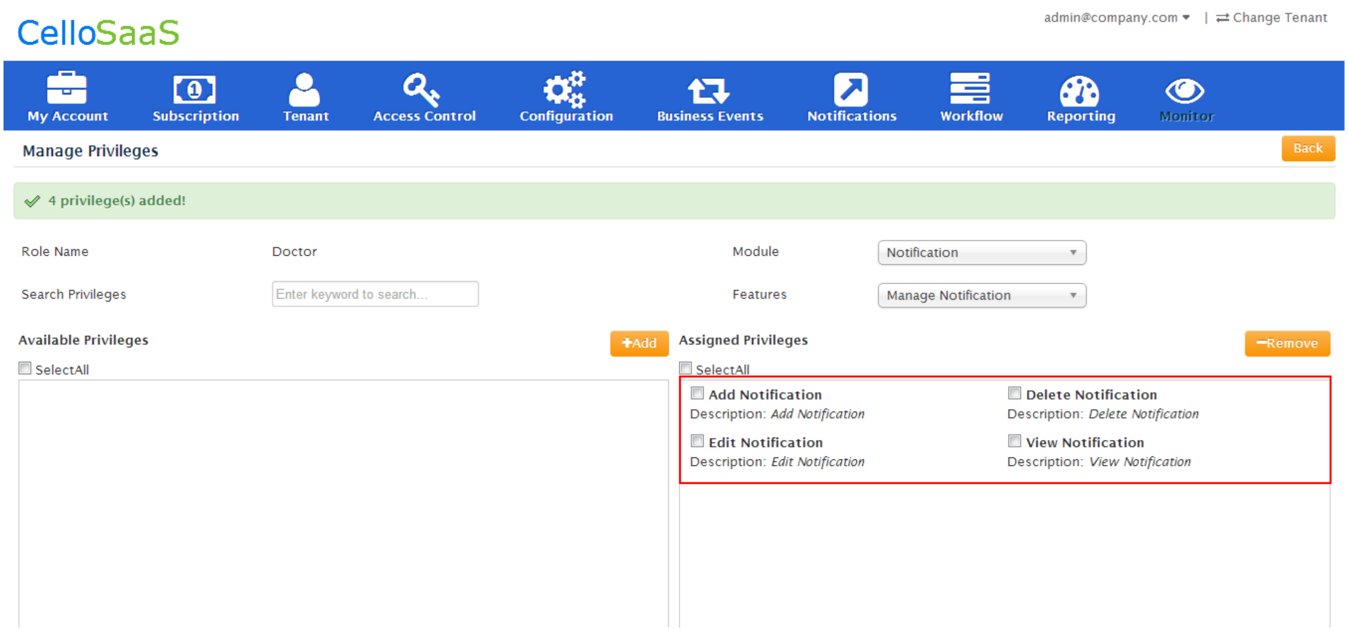
The Notifications created by the Tenant and Product Admin can be any of the below two types

- **Global Notifications** – Available for all the Sub Tenants
- **Tenant based notifications** – Available only for the Tenant who creates it.

1.2. Privileges required for Notification System

A Tenant must have subscribed to a package which has Notification module in order to take full advantage of the Notification System of CelloSaaS.

Below privileges are required to Add/Edit/Override Notification Details.



Manage Privileges

✓ 4 privilege(s) added!

Role Name: Doctor

Module: Notification

Search Privileges: Enter keyword to search...

Features: Manage Notification

Available Privileges

□ SelectAll

Assigned Privileges

□ SelectAll

□ Add Notification
Description: *Add Notification*

□ Delete Notification
Description: *Delete Notification*

□ Edit Notification
Description: *Edit Notification*

□ View Notification
Description: *View Notification*

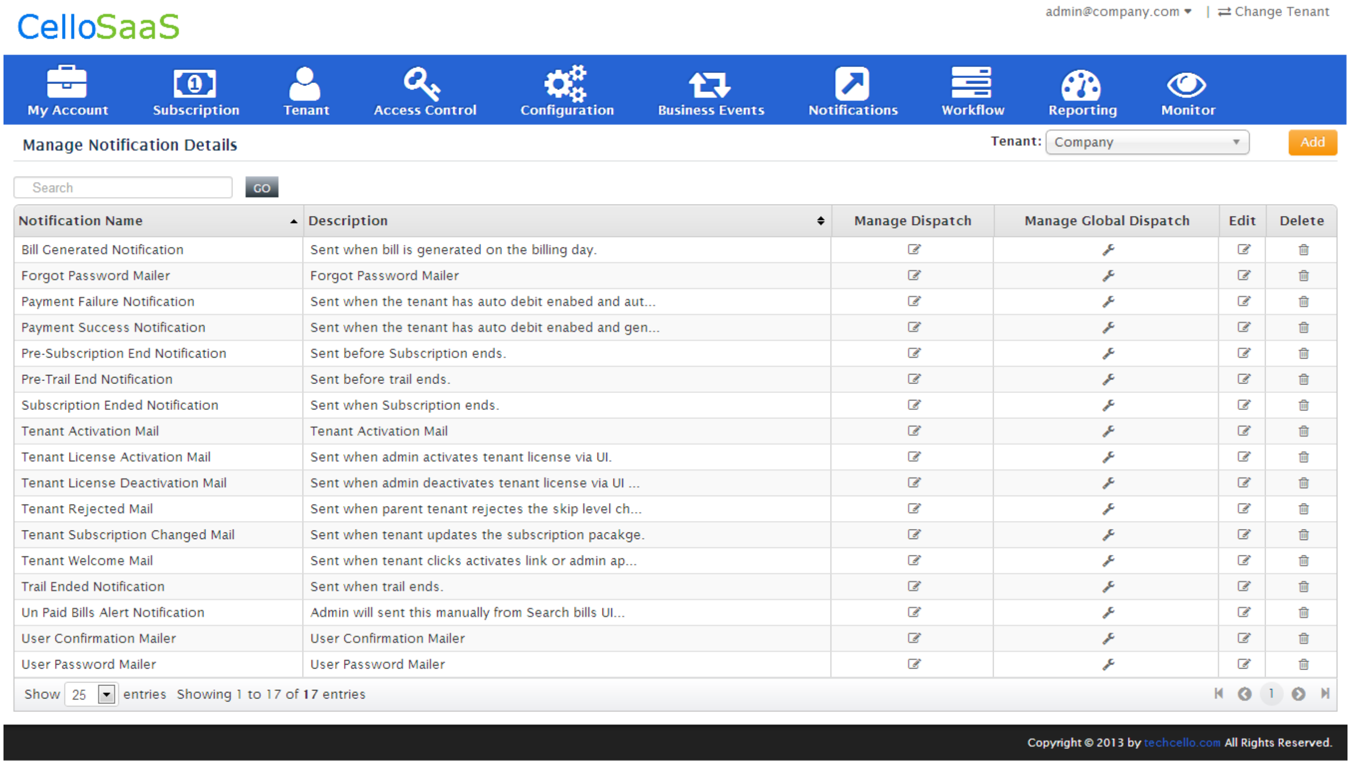
1.3. Default Notification Templates

Cello comes with a list of default email notifications [Refer below Screenshot]. Product owners can simply edit the dispatch details of all the default email notifications as well as the email contents (if required).

Note:

Each email notification requires separate dispatch information provided and there is no Common dispatch details apply for all the notifications.

Cello Utilizes Distributed Transactions, so it is recommended to enable MSDTC in the first hand before using the Notification system. To know more about MSDTC, refer



Manage Notification Details

Tenant: Company

Notification Name	Description	Manage Dispatch	Manage Global Dispatch	Edit	Delete
Bill Generated Notification	Sent when bill is generated on the billing day.	✎	🔧	✎	🗑
Forgot Password Mailer	Forgot Password Mailer	✎	🔧	✎	🗑
Payment Failure Notification	Sent when the tenant has auto debit enabed and aut...	✎	🔧	✎	🗑
Payment Success Notification	Sent when the tenant has auto debit enabed and gen...	✎	🔧	✎	🗑
Pre-Subscription End Notification	Sent before Subscription ends.	✎	🔧	✎	🗑
Pre-Trail End Notification	Sent before trail ends.	✎	🔧	✎	🗑
Subscription Ended Notification	Sent when Subscription ends.	✎	🔧	✎	🗑
Tenant Activation Mail	Tenant Activation Mail	✎	🔧	✎	🗑
Tenant License Activation Mail	Sent when admin activates tenant license via UI.	✎	🔧	✎	🗑
Tenant License Deactivation Mail	Sent when admin deactivates tenant license via UI ...	✎	🔧	✎	🗑
Tenant Rejected Mail	Sent when parent tenant rejectes the skip level ch...	✎	🔧	✎	🗑
Tenant Subscription Changed Mail	Sent when tenant updates the subscription pacakge.	✎	🔧	✎	🗑
Tenant Welcome Mail	Sent when tenant clicks activates link or admin ap...	✎	🔧	✎	🗑
Trail Ended Notification	Sent when trail ends.	✎	🔧	✎	🗑
Un Paid Bills Alert Notification	Admin will sent this manually from Search bills UI...	✎	🔧	✎	🗑
User Confirmation Mailer	User Confirmation Mailer	✎	🔧	✎	🗑
User Password Mailer	User Password Mailer	✎	🔧	✎	🗑

Show 25 entries Showing 1 to 17 of 17 entries

Copyright © 2013 by techcello.com All Rights Reserved.

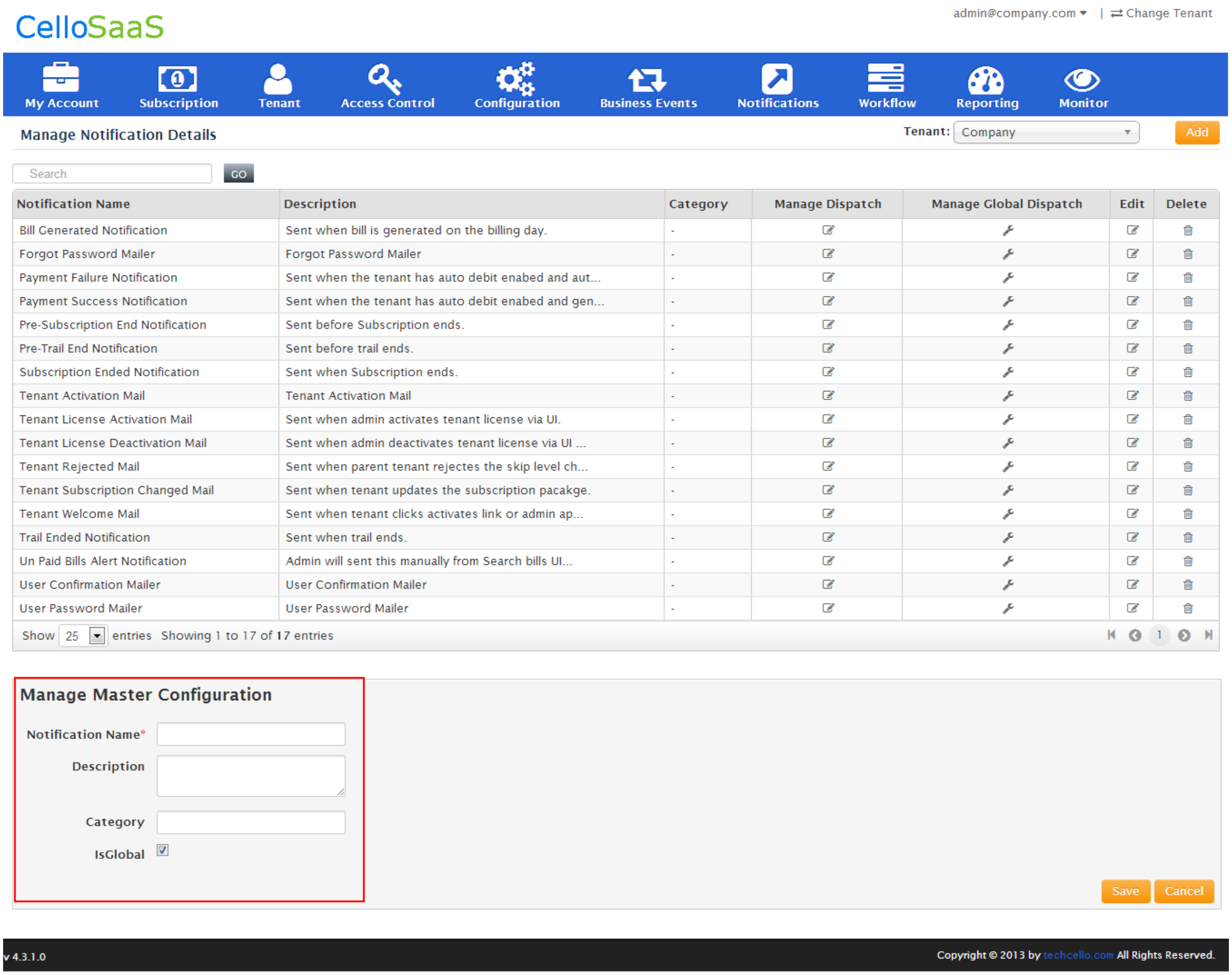
Manage Notification Details Screen

1.4. How to Configure a Notification Details

Notification Details can be configured using the Notification Manager using Dashboard. To configure a new Notification, Navigate to **Notifications→ Notification Management**.

Click on Add to configure a new Notification Details. Provide the Name, Description, Email Category and check "***IsGlobal***" property to make the Notification Global. These Global notifications will be available to all the tenants under the hierarchy, i.e. the content and dispatch details can be either overridden by the each tenant administrator or use the global details as such.

Notification Management



Manage Notification Details

Tenant: Company ▾ Add

Search GO

Notification Name	Description	Category	Manage Dispatch	Manage Global Dispatch	Edit	Delete
Bill Generated Notification	Sent when bill is generated on the billing day.	-	✎	🔧	✎	🗑
Forgot Password Mailer	Forgot Password Mailer	-	✎	🔧	✎	🗑
Payment Failure Notification	Sent when the tenant has auto debit enabed and aut...	-	✎	🔧	✎	🗑
Payment Success Notification	Sent when the tenant has auto debit enabed and gen...	-	✎	🔧	✎	🗑
Pre-Subscription End Notification	Sent before Subscription ends.	-	✎	🔧	✎	🗑
Pre-Trail End Notification	Sent before trail ends.	-	✎	🔧	✎	🗑
Subscription Ended Notification	Sent when Subscription ends.	-	✎	🔧	✎	🗑
Tenant Activation Mail	Tenant Activation Mail	-	✎	🔧	✎	🗑
Tenant License Activation Mail	Sent when admin activates tenant license via UI.	-	✎	🔧	✎	🗑
Tenant License Deactivation Mail	Sent when admin deactivates tenant license via UI ...	-	✎	🔧	✎	🗑
Tenant Rejected Mail	Sent when parent tenant rejectes the skip level ch...	-	✎	🔧	✎	🗑
Tenant Subscription Changed Mail	Sent when tenant updates the subscription pacakge.	-	✎	🔧	✎	🗑
Tenant Welcome Mail	Sent when tenant clicks activates link or admin ap...	-	✎	🔧	✎	🗑
Trail Ended Notification	Sent when trail ends.	-	✎	🔧	✎	🗑
Un Paid Bills Alert Notification	Admin will sent this manually from Search bills UI...	-	✎	🔧	✎	🗑
User Confirmation Mailer	User Confirmation Mailer	-	✎	🔧	✎	🗑
User Password Mailer	User Password Mailer	-	✎	🔧	✎	🗑

Show 25 ▾ entries Showing 1 to 17 of 17 entries

Manage Master Configuration

Notification Name*	
Description	
Category	
IsGlobal	☑

Save Cancel

v 4.3.1.0

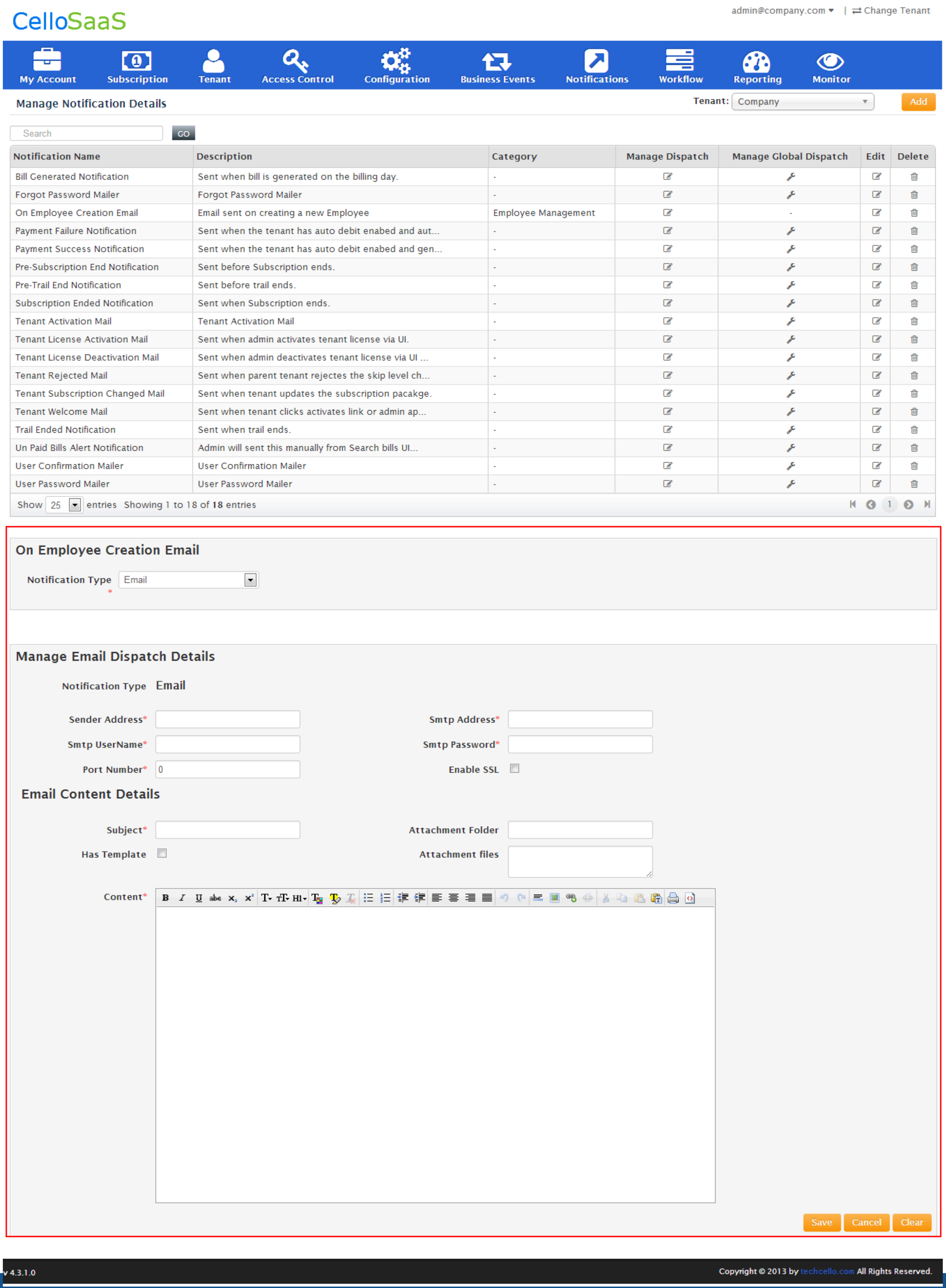
Copyright © 2013 by techcello.com All Rights Reserved.

Manage Notification Details Screen

Once the notification master is added, click on manage dispatch icon to configure the dispatch and content details.

The notification dispatch and content details are configured as follows

Notification Management



Manage Notification Details

Tenant: Company ▾ Add

Search GO

Notification Name	Description	Category	Manage Dispatch	Manage Global Dispatch	Edit	Delete
Bill Generated Notification	Sent when bill is generated on the billing day.	-	✎	🔧	✎	🗑
Forgot Password Mailer	Forgot Password Mailer	-	✎	🔧	✎	🗑
On Employee Creation Email	Email sent on creating a new Employee	Employee Management	✎	-	✎	🗑
Payment Failure Notification	Sent when the tenant has auto debit enabed and aut...	-	✎	🔧	✎	🗑
Payment Success Notification	Sent when the tenant has auto debit enabed and gen...	-	✎	🔧	✎	🗑
Pre-Subscription End Notification	Sent before Subscription ends.	-	✎	🔧	✎	🗑
Pre-Trail End Notification	Sent before trail ends.	-	✎	🔧	✎	🗑
Subscription Ended Notification	Sent when Subscription ends.	-	✎	🔧	✎	🗑
Tenant Activation Mail	Tenant Activation Mail	-	✎	🔧	✎	🗑
Tenant License Activation Mail	Sent when admin activates tenant license via UI.	-	✎	🔧	✎	🗑
Tenant License Deactivation Mail	Sent when admin deactivates tenant license via UI ...	-	✎	🔧	✎	🗑
Tenant Rejected Mail	Sent when parent tenant rejectes the skip level ch...	-	✎	🔧	✎	🗑
Tenant Subscription Changed Mail	Sent when tenant updates the subscription pacakge.	-	✎	🔧	✎	🗑
Tenant Welcome Mail	Sent when tenant clicks activates link or admin ap...	-	✎	🔧	✎	🗑
Trail Ended Notification	Sent when trail ends.	-	✎	🔧	✎	🗑
Un Paid Bills Alert Notification	Admin will sent this manually from Search bills UI...	-	✎	🔧	✎	🗑
User Confirmation Mailer	User Confirmation Mailer	-	✎	🔧	✎	🗑
User Password Mailer	User Password Mailer	-	✎	🔧	✎	🗑

Show 25 ▾ entries Showing 1 to 18 of **18** entries

On Employee Creation Email

Notification Type* Email ▾

Manage Email Dispatch Details

Notification Type Email

Sender Address*

Smtp Address*

Smtp UserName*

Smtp Password*

Port Number* 0

Enable SSL □

Email Content Details

Subject*

Attachment Folder

Has Template □

Attachment files

Content*

v 4.3.1.0

Copyright © 2013 by techcello.com All Rights Reserved.

1.5. Properties of Email Dispatch Details

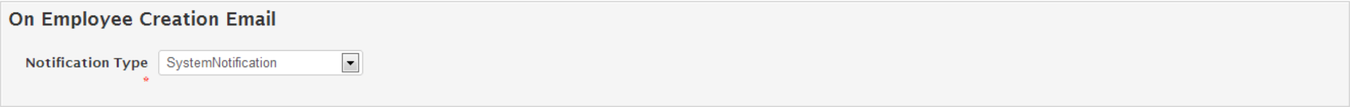
Based on the Notification type selected, such as Email or FTP, the appropriate email related properties such as dispatch, content details will be shown as follows.

Email Dispatch Details (Tenant/Email Specific)	
Notification Type	Email, FTP, System Notification
Sender Address	Sender Email Address
SMTP Address	SMTP Address details
SMTP Username	SMTP Username
SMTP Password	SMTP Password
Port Number	Port Number
Enable SSL	Check if SSL has to be enabled
Email/Batch Email Content Details	
Subject	Email Subject Content
Attachment Folder	Path of the Folder where the attachments are placed
Has Template	Check if the email has to be mapped to a Content Template
Attachment Files	List of Attachment (File Names Separated by Commas)
Template Rule Set	Providing a Ruleset Code, the notification manager will choose the template based on the outcome of the rule execution. (i.e.) The Notification manager executes the rule and the result of the rule will be the template name. The notification manager uses this template to send notification. If the rule set is empty, then the notification manager uses the default template.
Default Template	Select the Content Template (If the notification has the template, it can include a particular template by checking the template check box [refer the image below]. The notification template contains the body of the email with some placeholders.)

1.6. *Properties of FTP Notification*

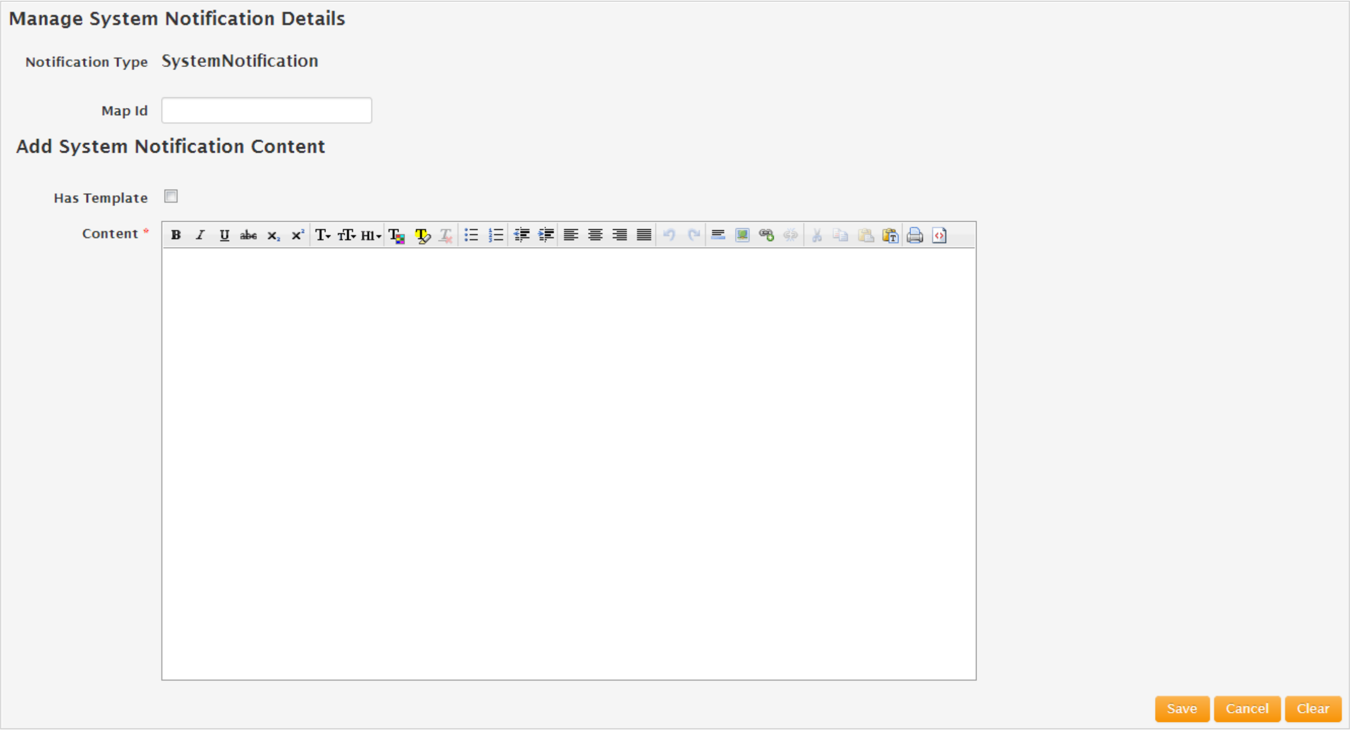
FTP/Batch FTP/SFTP/Batch SFTP - Dispatch Details (Tenant/Email Specific)	
Notification Type	FTP
FTP Username	Username of FTP Location
FTP Password	Password of FTP Location
FTP Content Details	
FTP File Path	Path of the Folder where the files are placed
File Name	List of File Names (Separated by Commas)

1.7. *Properties of System Notification*



On Employee Creation Email

Notification Type * SystemNotification



Manage System Notification Details

Notification Type SystemNotification

Map Id

Add System Notification Content

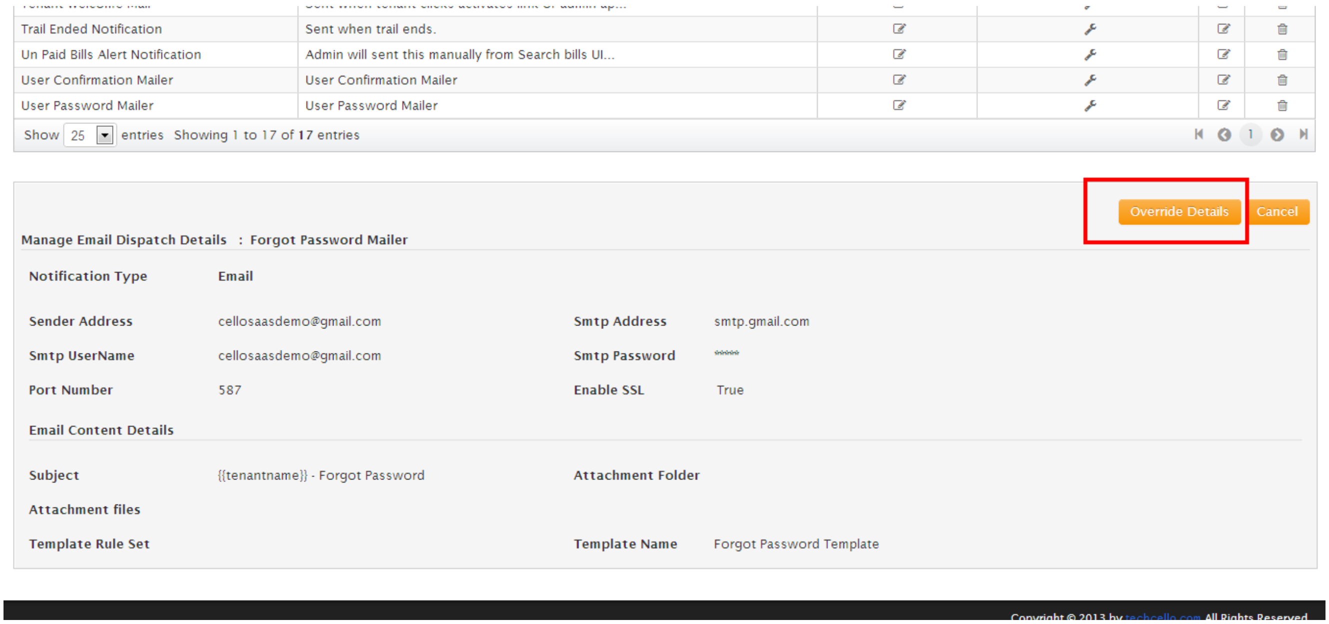
Has Template □

Content *

System Notification - Dispatch Details (Tenant Specific)	
Notification Type	System Notification
Has Template	Check if the Notification has to be mapped with Template
Content	System Notification Content

1.8. *Overriding Default Templates*

If the Notification is Global, the tenant admin can override the dispatch and content [there by the content and dispatch will belong to this tenant admin for the global template] details as follows



Trail Ended Notification	Sent when trail ends.
Un Paid Bills Alert Notification	Admin will sent this manually from Search bills UI...
User Confirmation Mailer	User Confirmation Mailer
User Password Mailer	User Password Mailer

Show 25 entries Showing 1 to 17 of 17 entries

Manage Email Dispatch Details : Forgot Password Mailer

Notification Type	Email		
Sender Address	cellosaasdemo@gmail.com	Smtp Address	smtp.gmail.com
Smtp UserName	cellosaasdemo@gmail.com	Smtp Password	*****
Port Number	587	Enable SSL	True

Email Content Details

Subject	{{tenantname}} - Forgot Password	Attachment Folder	
Attachment files			
Template Rule Set		Template Name	Forgot Password Template

Copyright © 2013 by techcello.com All Rights Reserved

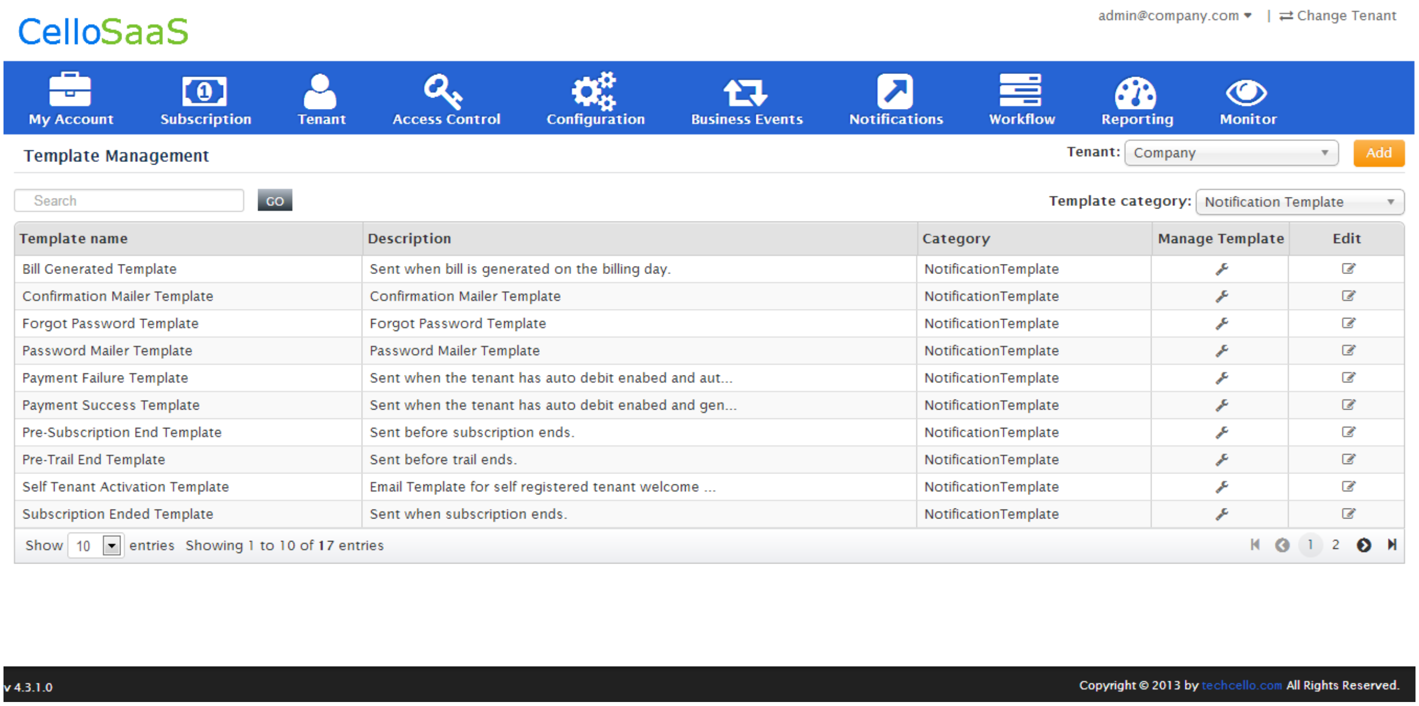
By this way, each tenant can update their own dispatch, content details for the global notification.

1.9. *Content Template*

Cello provides an elegant WYSIWYG Content Management System to capture the Email content along with the dynamic place holders. Below are the properties of Content Templates, They are

- Supports Dynamic Place Holders
- Supports Multi Language
- Tenant Specific

To Configure a Template, Navigate to **Notifications→ Content Template**



Template Management

Tenant: Company ▾ Add

Search GO

Template category: Notification Template ▾

Template name	Description	Category	Manage Template	Edit
Bill Generated Template	Sent when bill is generated on the billing day.	NotificationTemplate	🔧	✎
Confirmation Mailer Template	Confirmation Mailer Template	NotificationTemplate	🔧	✎
Forgot Password Template	Forgot Password Template	NotificationTemplate	🔧	✎
Password Mailer Template	Password Mailer Template	NotificationTemplate	🔧	✎
Payment Failure Template	Sent when the tenant has auto debit enabed and aut...	NotificationTemplate	🔧	✎
Payment Success Template	Sent when the tenant has auto debit enabed and gen...	NotificationTemplate	🔧	✎
Pre-Subscription End Template	Sent before subscription ends.	NotificationTemplate	🔧	✎
Pre-Trail End Template	Sent before trail ends.	NotificationTemplate	🔧	✎
Self Tenant Activation Template	Email Template for self registered tenant welcome ...	NotificationTemplate	🔧	✎
Subscription Ended Template	Sent when subscription ends.	NotificationTemplate	🔧	✎

Show 10 ▾ entries Showing 1 to 10 of 17 entries 1 2

v 4.3.1.0

Copyright © 2013 by techcello.com All Rights Reserved.

1.10. Types of Template

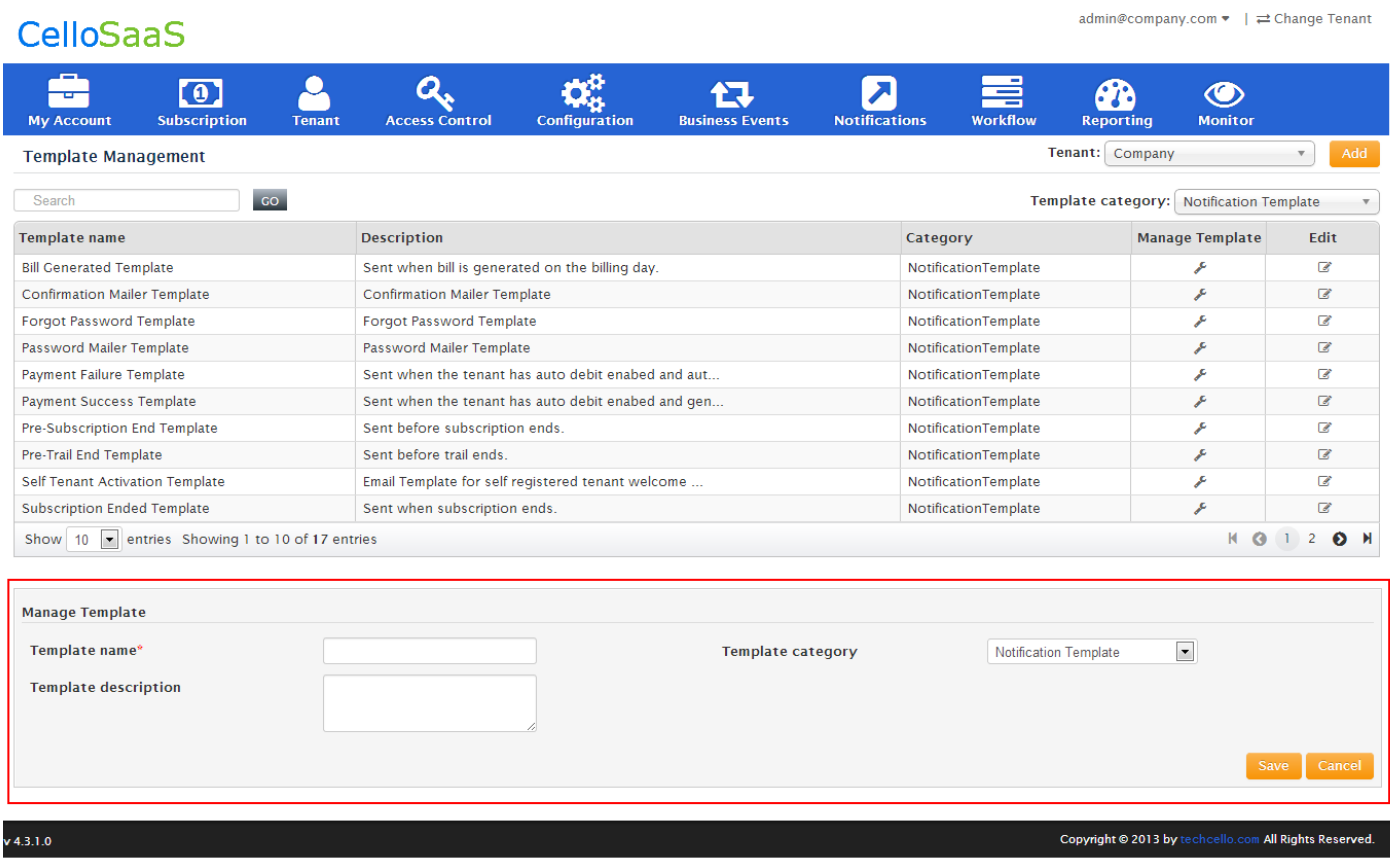
Users can create two types of Template, They are

1. Notification Template – used for Notifications Purpose
2. Event Template– used for Events Purpose i.e. based on an event triggered in the system, content can be mapped to the event.

   Example: Displaying Dynamic Offers based on the Tenant who login.

1.11. To add a template [Notification/Event]

Click on add button on top of the Screen to add a content Template



Template Management

Tenant: Company

Template category: Notification Template

Template name	Description	Category	Manage Template	Edit
Bill Generated Template	Sent when bill is generated on the billing day.	NotificationTemplate		
Confirmation Mailer Template	Confirmation Mailer Template	NotificationTemplate		
Forgot Password Template	Forgot Password Template	NotificationTemplate		
Password Mailer Template	Password Mailer Template	NotificationTemplate		
Payment Failure Template	Sent when the tenant has auto debit enabed and aut...	NotificationTemplate		
Payment Success Template	Sent when the tenant has auto debit enabed and gen...	NotificationTemplate		
Pre-Subscription End Template	Sent before subscription ends.	NotificationTemplate		
Pre-Trail End Template	Sent before trail ends.	NotificationTemplate		
Self Tenant Activation Template	Email Template for self registered tenant welcome ...	NotificationTemplate		
Subscription Ended Template	Sent when subscription ends.	NotificationTemplate		

Show 10 entries Showing 1 to 10 of 17 entries 1 2

Manage Template

Template name*

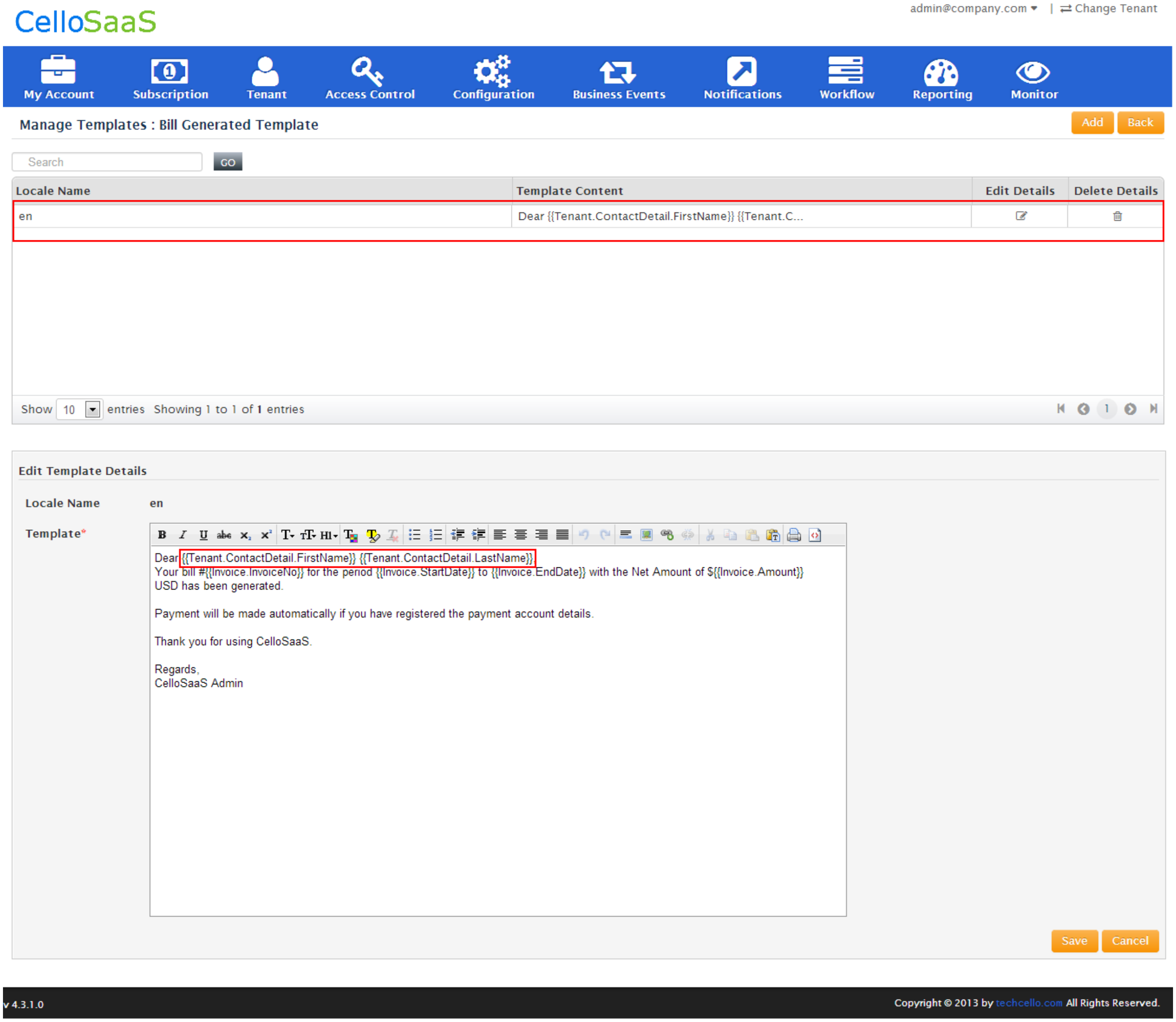
Template category: Notification Template

Template description

v 4.3.1.0

Copyright © 2013 by techcello.com All Rights Reserved.

1.12. *To add Locale based template content*



Manage Templates : Bill Generated Template

Locale Name	Template Content	Edit Details	Delete Details
en	Dear {{Tenant.ContactDetail.FirstName}} {{Tenant.C...	✎	🗑

Show 10 entries Showing 1 to 1 of 1 entries

Edit Template Details

Locale Name en

Template*

Dear {{Tenant.ContactDetail.FirstName}} {{Tenant.ContactDetail.LastName}}
Your bill #{{Invoice.InvoiceNo}} for the period {{Invoice.StartDate}} to {{Invoice.EndDate}} with the Net Amount of ${{Invoice.Amount}} USD has been generated.

Payment will be made automatically if you have registered the payment account details.

Thank you for using CelloSaaS.

Regards,
CelloSaaS Admin

1.13. *Introduction to Dynamic Place Holders*

Users can utilize the dynamic place holders in places where the dynamic content has to be updated with the data that comes from the data store, refer the above Screen. Based on the access model, there are two types of placeholders are supported

- Entity type Placeholders

- Xml type Placeholder

Entity Type Place Holders

it requires the objects to replace the Placeholders. The placeholder should be as follows.

Example

{{UserDetails.MembershipDetails.UserName}}

Here, the placeholder requires the ***UserDetails*** object to replace the placeholder.

Xml type Placeholder

it requires xml to replace the Placeholder. The placeholder should be as follows.

Example

{{Element(“/UserDetails/MembershipDetails/UserName”)}}

Here the placeholder requires the xml to replace the placeholder. The xml can be set by the following way.

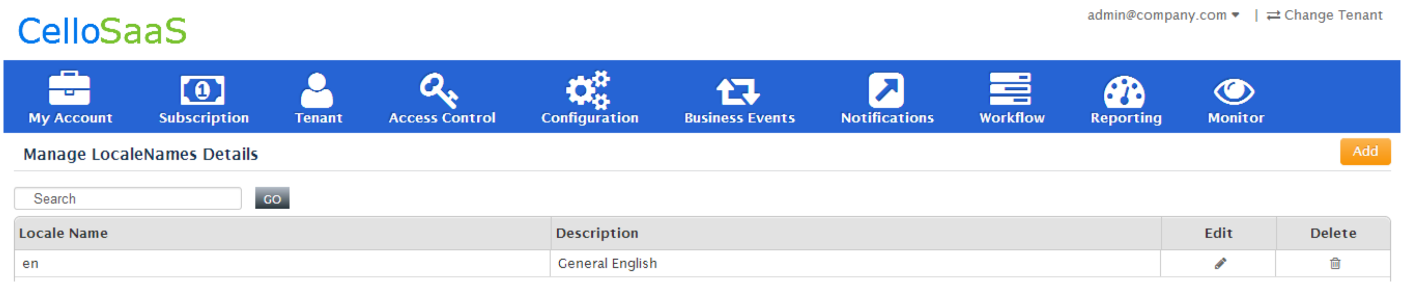
Consumption

```
String xml =
”<Root><UserDetails><MembershipDetails><UserName>John></UserName></MembershipDetails></UserDetails></Root>”;

PlaceholderXml placeholderXml = new PlaceholderXml{ Xml = xml};
```

Then this ***placeholderXml*** object will be passed through the notification placeholder while sending the notification.

1.14. How to Add/Edit/Delete Locales

By default en-EN Locale comes with the package, but administrators can add other locales using the Master Data Management.Navigate to Configuration Management → Manage Master Data→ Manage Locale



Manage LocaleNames Details

Locale Name	Description	Edit	Delete
en	General English	✎	🗑

1.15. Consumption

The Notification Manager API is available in the following Namespace.

Namespace: CelloSaaS.Notification.ServiceProxies
Class: NotificationServiceProxy

Description: The following method is used to send an email or system notification or upload a file through ftp. This method will not process the template or the email content. It requires the processed content, to send the email or system notification.

```
/// This method is used to send a email or upload a file.
public static void SendNotification(String notificationId, String tenantId, INotificationContent
notificationContent, INotificationDestination notificationDestination)
```

Description: The following method is used to process the template content or the direct content, and send that through the email or system notification. For that template processing, it requires the placeholder objects to replace the placeholders inside the template. For file uploading also can be done through this method.

```
/// This reads the content from the configuration using the notificationId, replaces the placeholders and
delivers the notification via notification dispatch manager
/// The kind of content here can only be text content
public static void ProcessAndSendNotification(String notificationId, String tenantId, NotificationPlaceholder
placeholders, INotificationDestination notificationDestination)
```

Description: The following method is used to process the template content or the direct content, and send that through the email or system notification. Here the template content will be based on the given localeId. Each template can have various locale based contents. For that template processing, it requires the placeholder objects to replace the placeholders inside the template. For file uploading also can be done through this method.

```
///This reads the content from the configuration using the notificationId, replaces the placeholders
///based on the locale id and delivers the notification via notification dispatch manager
/// The kind of content here can only be text content
public static void ProcessAndSendNotification(String notificationId, String tenantId, String localeId,
NotificationPlaceholder placeholders, INotificationDestination notificationDestination)
```

Description: The following method is used to process and send that content through email or system notification. We can also use this method to upload a file through ftp.

```
/// <summary>
/// This method is used to Processes and send the notification. That is process the content and it is sent by
using the notification dispatch manager.
public static void ProcessAndSendNotification(String notificationId, String tenantId, INotificationContent
notificationContent, INotificationDestination notificationDestination)
```

Description: The following method is used to process the content and send it through the batch email or batch system notification. It requires list of placeholder objects for replace the placeholders. By using this method different content can be send to different destinations.

```
/// This reads the content from the configuration using the notificationId, replaces the placeholders and
delivers the notification via notification dispatch manager
/// The kind of content here can only be batch text content
public static void ProcessAndSendBatchNotification(String notificationId, String tenantId,
List<NotificationPlaceholder> placeholders, IBatchNotificationDestination notificationDestination)
```

Description: The following method is used to process the locale based content and send it through the batch email or batch system notification. It requires list of placeholder objects for replace the placeholders. By using this method different content can be send to different destinations for a particular locale.

```
/// This method is used to Processes and send batch notification with locale id
public static void ProcessAndSendBatchNotification(String notificationId, String tenantId, String localeId,
List<NotificationPlaceholder> placeholders, IBatchNotificationDestination notificationDestination)
```

Description: The following method is used to process the list of content and send this content to the given list of destinations.

```
/// This method is used to processes the batch content and send the batch notification.
public static void ProcessAndSendBatchNotification(String notificationId, String tenantId,
IBatchNotificationContent content, IBatchNotificationDestination notificationDestination)
```

Description: The following method is used to gets the notification details by using the notification id and the tenant id. By using this method we can get the content and destination details for the given notification id.

```
/// This method is used to gets the the notification details by using the notification id and the tenant id.
public static NotificationDetails GetNotificationDetails(String notificationId, String tenantId)
```

Description: The following method is used to gets the notification details by using the notification id and the tenant id based on the locale. By using this method we can get the locale based content and destination details for the given notification id.

```
/// This method is used to gets the notification details by using the notification id and the tenant id based on the locale id.
public static NotificationDetails GetNotificationDetails(string notificationId, string tenantId, string localeId)
```

Description: The following method is used to gets the notification details by using the notification name and the tenant id. By using this method we can get content and destination details for the given notification name for the given tenantId.

```
/// This method is used to gets the notification details by using the notification name and the tenant id.
public static NotificationDetails GetNotificationDetailsByName(string notificationName, string tenantId)
```

Description: This above method is used to gets the notification details by using the notification name and the tenant id based on the locale. By using this method we can get the locale based content and destination details for the given notification name and the tenantId.

```
/// This method is used to gets the notification details by using the notification name with the given locale id.
public static NotificationDetails GetNotificationDetailsByName(string notificationName, string tenantId,
string localeId)
```

Example

In an employee management system, the notifications can be sent for the following scenarios.

1. When an employee/user profile was created or updated.
2. When the employee request / approve the leave.
3. When the user's password was changed in the system.
4. When the user was activated/deactivated.
5. When the employee get confirmed.
6. When an employee was enrolled, a batch notification was send to all the employees.

Let us take the following scenario as example for sending email notification.

1.1 Scenario 1

When an employee profile is created, the username and password will be sent to that employee through email.

To send the notifications, the following steps need to be followed.

1. Configure the notification details
2. Call the notification service to send the notification.

After the configuring the notification details, the following services will be used to send the notifications.

Namespace: CelloSaaS.Notification.ServiceProxies
Class: NotificationServiceProxy
Method : public static void ProcessAndSendNotification(String notificationId, String tenantId, NotificationPlaceholder placeholders, INotificationDestination notificationDestination)

Example:

```
NotificationServiceProxy.ProcessAndSendNotification(NotificationConstant.AddEmployeeNotificationId,
UserIdentity.TenantID, notificationPlaceholder, emailDestination);
```

Sample Code:

```
public string AddEmployeeDetails(EmployeeDetails employeeDetails)
{

        /* Write the Logic to adding employee details*/

        //To sends the email notification
        SendEmailNotification(employeeDetails);

        return employeeDetails.Identifier;
}

private static void SendEmailNotification(EmployeeDetails employeeDetails)
{
        NotificationDetails notificationDetails = NotificationServiceProxy.
        GetNotificationDetails(NotificationConstant.AddEmployeeNotificationId, UserIdentity.TenantID);

        EmailDestination emailDestination = (EmailDestination) notificationDetails.NotificationDestination;
        emailDestination.RecipientAddress = employeeDetails.EmailID;

        NotificationPlaceholder notificationPlaceholder = new NotificationPlaceholder();
        notificationPlaceholder.Keys = new Dictionary<string, string>();
        notificationPlaceholder.Keys.Add("password", "company");
        notificationPlaceholder.ObjectPlaceholders = new object[] { employeeDetails };

        NotificationServiceProxy.ProcessAndSendNotification(NotificationConstant.AddEmployeeNotificationId, UserIdentity.TenantID, notificationPlaceholder, emailDestination);
}
```

This code is available in the sample application. Kindly go through it in EmployeeManagement. Services. EmployeeModule Namespace.

1.2 Scenario 2

When an employee profile is updated, the updated employee details will be sent to that employee through system notification within the application.

Here, the employee's userId will be the MapId for the system notification.

After the notification details configuration, the following services will be used to send the notifications.

Namespace: CelloSaaS.Notification.ServiceProxies
Class: NotificationServiceProxy
Method : public static void ProcessAndSendNotification(String notificationId, String tenantId, NotificationPlaceholder placeholders, INotificationDestination notificationDestination)

For various scenarios we can use the overloaded methods instead of the given one.

Example:

```
NotificationServiceProxy.ProcessAndSendNotification(NotificationConstant.AddEmployeeNotificationId,
UserIdentity.TenantID, notificationPlaceholder, systemNotificationDestination);
```

Sample Code:

```
public  void UpdateEmployeeDetails(EmployeeDetails employeeDetails)
{
        /* Write the Logic to updating the employee details*/

         SendSystemNotification(employeeDetails);
}


private static void SendSystemNotification(EmployeeDetails employeeDetails)
{
        NotificationDetails notificationDetails  = NotificationServiceProxy. GetNotificationDetails
        (NotificationConstant.RemoveEmployeeNotificationId, UserIdentity.TenantID);

        SystemNotificationDestination systemNotificationDestination = (SystemNotificationDestination)
        notificationDetails.NotificationDestination;

        NotificationPlaceholder notificationPlaceholder = new NotificationPlaceholder();
        notificationPlaceholder.ObjectPlaceholders = new object[] { employeeDetails };
        NotificationServiceProxy.ProcessAndSendNotification (NotificationConstant.
        UpdateEmployeeNotificationId,  UserIdentity.TenantID,  notificationPlaceholder,
        systemNotificationDestination);
}
```

1.3 Scenario 3

When an employee is created, the employee's photo will be uploaded to the server through Ftp notification.

After the notification details configuration, the following services will be used to send the notifications.

Namespace: CelloSaaS.Notification.ServiceProxies
Class: NotificationServiceProxy
Method : public static void SendNotification(String notificationId, String tenantId, INotificationContent notificationContent, INotificationDestination notificationDestination)

For various scenarios we can use the overload methods instead of the given method.

Example:

```
NotificationServiceProxy.SendNotification(NotificationConstant. UploadEmployeeDocumentNotificationId,
UserIdentity.TenantID, fileContent, ftpDestination);
```

Sample Code:

```
public  void AddEmployeeDetails(EmployeeDetails employeeDetails)
{
        /* Write the Logic to add the employee details*/
         SendFtpNotification(employeeDetails);
}


private static void SendFtpNotification(EmployeeDetails employeeDetails)
{
        NotificationDetails notificationDetails  = NotificationServiceProxy.GetNotificationDetails
        (NotificationConstant.UploadEmployeeDocumentNotificationId, UserIdentity.TenantID);

        FtpDestination ftpDestination = (FtpDestination)notificationDetails.NotificationDestination;
        FileContent fileContent = (FileContent)notificationDetails.NotificationDestination;

        NotificationServiceProxy.SendNotification(NotificationConstant.UploadEmployeeDocumentNotificati
        onId, UserIdentity.TenantID, fileContent, ftpDestination);
}
```

1.4 Scenario 4

When a new employee is enrolled into the system, a batch email notification will be sent to all the employees, for introducing the new employee.

After the notification details configuration, the following services will be used to send the batch notifications.

Namespace: CelloSaaS.Notification.ServiceProxies
Class: NotificationServiceProxy
Method : public static void ProcessAndSendBatchNotification(String notificationId, String tenantId, NotificationPlaceholder placeholders, INotificationDestination notificationDestination)

For various scenarios we can use the overload methods instead of the given method.

Example:

NotificationServiceProxy.ProcessAndSendBatchNotification(NotificationConstant.BatchEmployeeNotificationId, UserIdentity.TenantID, placeHoldersList, notificationDestinationList)

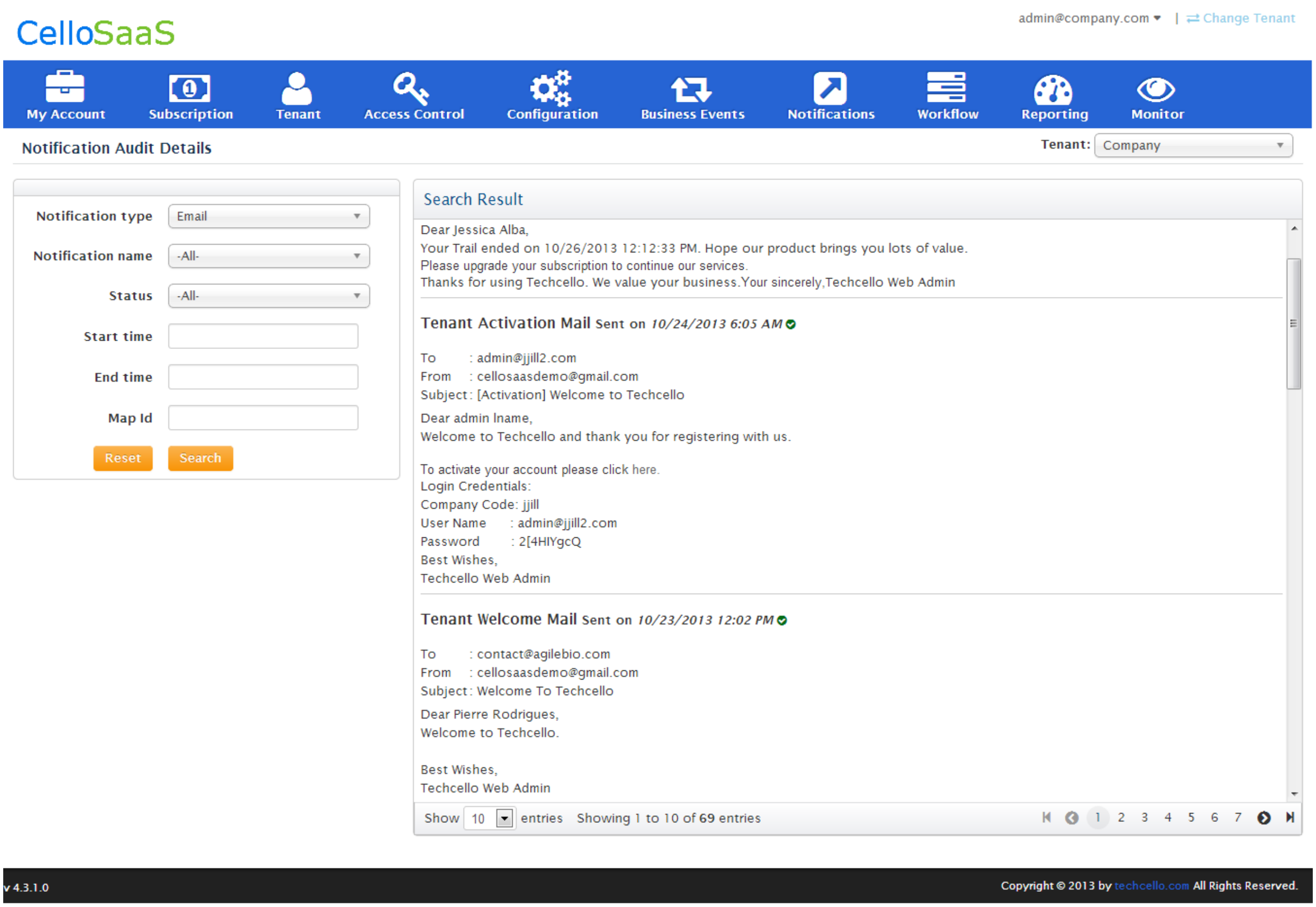
Sample Code:

```
public string AddEmployeeDetails(EmployeeDetails employeeDetails)
{

        /* Write the Logic to adding employee details*/

        //To sends the email notification
        SendGroupEmailNotification (employeeDetails);

        return employeeDetails.Identifier;
}


private void SendGroupEmailNotification(EmployeeDetails employeeDetails)
{
        NotificationDetails notificationDetails = NotificationServiceProxy. GetNotificationDetails
        (NotificationConstant. BatchEmployeeNotificationId, UserIdentity.TenantID);

        BatchEmailDestination batchEmailDestination = (BatchEmailDestination)
        notificationDetails.NotificationDestination;

        EmailDestination emailDestination = (EmailDestination)
        batchEmailDestination.CommonNotificationDestination;

         Dictionary<string, EmployeeDetails> employees = this. GetEmployeeDetailssByTenantId
        (UserIdentity.TenantID);
        if (employees != null && employees.Count > 1)
        {
            IBatchNotificationDestination notificationDestinationList = new BatchEmailDestination();
            notificationDestinationList.NotificationDestinations = new List<INotificationDestination>();
            notificationDestinationList.CreatedBy = notificationDestinationList.UpdatedBy =
            UserIdentity.UserId;

            List<NotificationPlaceholder> placeHoldersList = new List<NotificationPlaceholder>();
            NotificationPlaceholder placeHolder;
            foreach (EmployeeDetails employee in employees.Values)
            {
                emailDestination.RecipientAddress =employee.EmailID;

                notificationDestinationList.NotificationDestinations.Add(GetClonedEmailDestination
                (emailDestination));

                placeHolder = new NotificationPlaceholder();
                placeHolder.Keys = new Dictionary<string, string>();
                placeHolder.Keys.Add("name", employee.FirstName);
                 placeHolder.ObjectPlaceholders = new object[] { employeeDetails };
                placeHoldersList.Add(placeHolder);
            }

        NotificationServiceProxy.ProcessAndSendBatchNotification(NotificationConstant.BatchEmployeeNot
        ificationId, UserIdentity.TenantID, placeHoldersList, notificationDestinationList);

       }
    }
```

Similarly, the batch ftp, batch system notification also handled.

1.16. Notification Audit

Cello Audits the entire Notification engine and captures all the logs/details and Stores in the metadata. Navigate to **Monitor→ Notification** to see the audited information. All the Failure and Successfully sent emails along with the outgoing content can be seen in the Email audit Screen.



Notification Audit Details

Tenant: Company

Notification type	Email
Notification name	-All-
Status	-All-
Start time	
End time	
Map Id	

Search Result

Dear Jessica Alba,
Your Trail ended on 10/26/2013 12:12:33 PM. Hope our product brings you lots of value.
Please upgrade your subscription to continue our services.
Thanks for using Techcello. We value your business.Your sincerely,Techcello Web Admin

Tenant Activation Mail Sent on *10/24/2013 6:05 AM* ✓

To : admin@jjill2.com
From : cellosaasdemo@gmail.com
Subject : [Activation] Welcome to Techcello

Dear admin lname,
Welcome to Techcello and thank you for registering with us.

To activate your account please click here.
Login Credentials:
Company Code: jjill
User Name : admin@jjill2.com
Password : 2[4HIYgcQ
Best Wishes,
Techcello Web Admin

Tenant Welcome Mail Sent on *10/23/2013 12:02 PM* ✓

To : contact@agilebio.com
From : cellosaasdemo@gmail.com
Subject : Welcome To Techcello

Dear Pierre Rodrigues,
Welcome to Techcello.

Best Wishes,
Techcello Web Admin

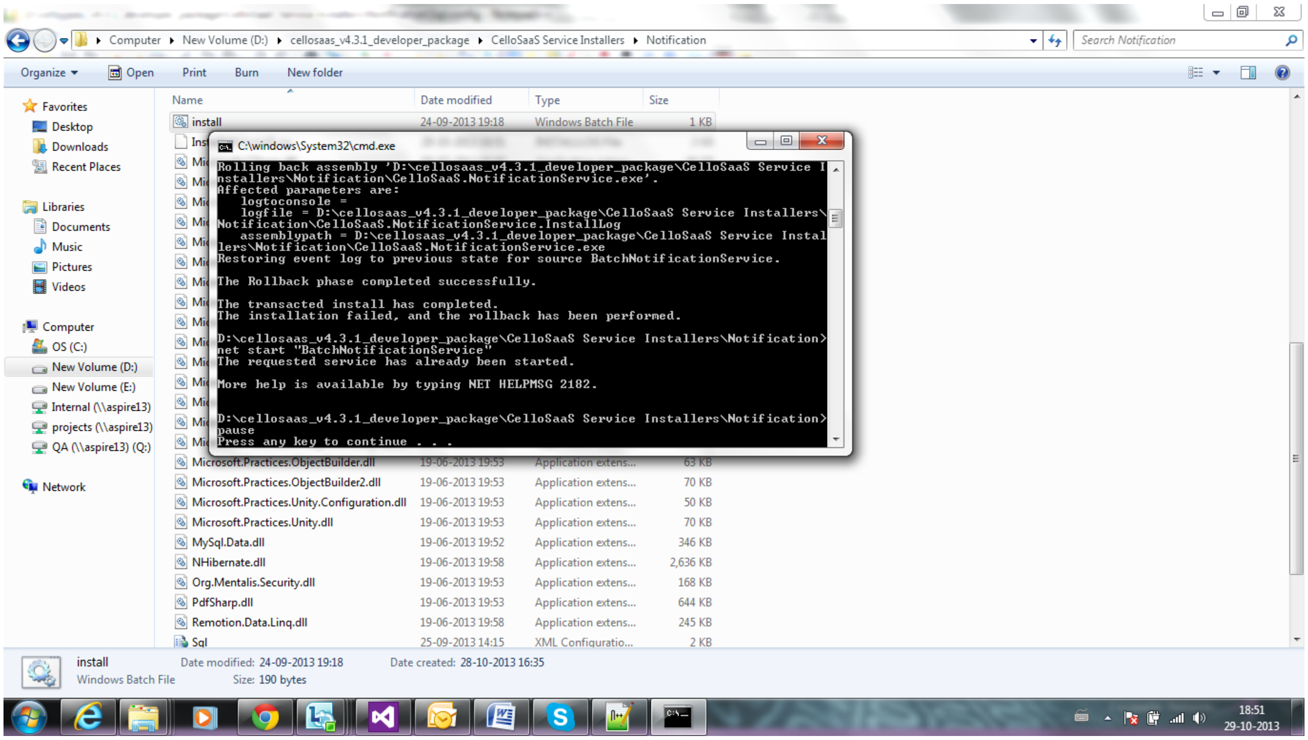
Show 10 entries Showing 1 to 10 of 69 entries 1 2 3 4 5 6 7

Copyright © 2013 by techcello.com All Rights Reserved.

1.17. Batch Email Service

Cello Provides Notification Windows Service to send bulk emails using the Notification engine. This Windows Service will be running in the Hosted server and continuously poll for the list if there is any bulk email scheduled.

The window Service installer is placed in the CelloSaaS Package with the name “Notification”, look for “install.bat” inside the folder and right clicks and run with Administrator Permission.



```
Rolling back assembly 'D:\cellosaas_v4.3.1_developer_package\CelloSaaS Service I
nstallers\Notification\CelloSaaS.NotificationService.exe'.
Affected parameters are:
   logtoconsole =
   logfile = D:\cellosaas_v4.3.1_developer_package\CelloSaaS Service Installers\
Notification\CelloSaaS.NotificationService.InstallLog
   assemblypath = D:\cellosaas_v4.3.1_developer_package\CelloSaaS Service Instal
lers\Notification\CelloSaaS.NotificationService.exe
Restoring event log to previous state for source BatchNotificationService.

The Rollback phase completed successfully.

The transacted install has completed.
The installation failed, and the rollback has been performed.

D:\cellosaas_v4.3.1_developer_package\CelloSaaS Service Installers\Notification>
net start "BatchNotificationService"
The requested service has already been started.

More help is available by typing NET HELPMSG 2182.


D:\cellosaas_v4.3.1_developer_package\CelloSaaS Service Installers\Notification>
pause
Press any key to continue . . .
```

Name	Date modified	Type	Size
install	24-09-2013 19:18	Windows Batch File	1 KB
Microsoft.Practices.ObjectBuilder.dll	19-06-2013 19:53	Application extens...	63 KB
Microsoft.Practices.ObjectBuilder2.dll	19-06-2013 19:53	Application extens...	70 KB
Microsoft.Practices.Unity.Configuration.dll	19-06-2013 19:53	Application extens...	50 KB
Microsoft.Practices.Unity.dll	19-06-2013 19:53	Application extens...	70 KB
MySql.Data.dll	19-06-2013 19:52	Application extens...	346 KB
NHibernate.dll	19-06-2013 19:58	Application extens...	2,636 KB
Org.Mentalis.Security.dll	19-06-2013 19:53	Application extens...	168 KB
PdfSharp.dll	19-06-2013 19:53	Application extens...	644 KB
Remotion.Data.Linq.dll	19-06-2013 19:58	Application extens...	245 KB
Sql	25-09-2013 14:15	XML Configuratio...	2 KB

install
Windows Batch File
Date modified: 24-09-2013 19:18
Size: 190 bytes
Date created: 28-10-2013 16:35

Once the Service is installed successfully, update the connection strings in the ***SQL.Config*** file placed inside the same folder to point to the CelloSaaS Metadata.

2. Contact Information

Any problem using this guide (or) using Cello Framework. Please feel free to contact us, we will be happy to assist you in getting started with Cello.

Email: support@techcello.com

Phone: +1(609)503-7163

Skype: techcello