Cello How To-Do guide

Notification Management



Notification Management

Contents

1.	Noti	fication Management	. 3
	1.1.	CelloSaaS Notification Architecture	. 3
	1.2.	Privileges required for Notification System	. 4
	1.3.	Default Notification Templates	. 5
	1.4.	How to Configure a Notification Details	. 6
	1.5.	Properties of Email Dispatch Details	. 9
	1.6.	Properties of FTP Notification	10
	1.7.	Properties of System Notification	10
	1.8.	Overriding Default Templates	11
	1.9.	Content Template	11
	1.10.	Types of Template	12
	1.11.	To add a template [Notification/Event]	13
	1.12.	To add Locale based template content	14
	1.13.	Introduction to Dynamic Place Holders	14
	1.14.	How to Add/Edit/Delete Locales	15
	1.15.	Consumption	16
	1.1	Scenario 1	18
	1.2	Scenario 2	20
	1.3	Scenario 3	21
	1.4	Scenario 4	22
	1.16.	Notification Audit	24
	1.17.	Batch Email Service	24
2.	Con	tact Information	26

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

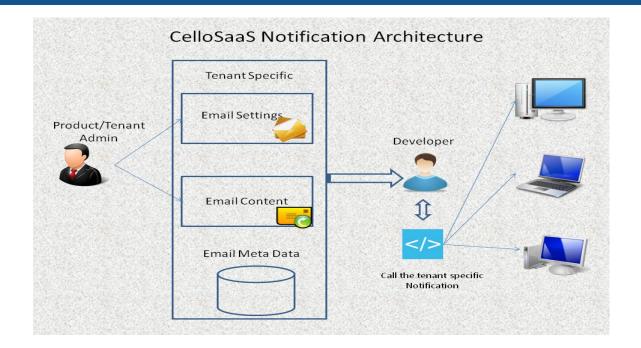
- 1. Dispatch details SMTP Details, Sender Details etc
- 2. Email Content Actual Content sent to Users

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.

CelloSaaS							admin@compa	ny.com ▼ ≓	Change Tenan
My Account Subscription	Tenant	Access Control	Configuration	Business Events	Notifications	Workflow	Reporting	Monitor	
Manage Privileges									Back
✓ 4 privilege(s) added!									
Role Name	Doctor			Module	No	tification	Ŧ		
Search Privileges	Enter keywo	rd to search		Features	Ма	nage Notification	•		
Available Privileges				Add Assigned Privileg	ges				-Remove
SelectAll				SelectAll	ation	1	Delete Notifica	tion	
				Description: Add	d Notification	D	escription: Delete	Notification	
				Edit Notific Description: Edit			View Notification escription: View N		

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, <u>refer</u>

	- See -				
My Account Subscription	Tenant Access Control Configuration Business Events	Notifications Workfl		or	
Manage Notification Details			Tenant: Company	•	Add
Search GO	l				
Notification Name	Description	Manage Dispatch	Manage Global Dispatch	Edit	Delete
Bill Generated Notification	Sent when bill is generated on the billing day.	ľ	F	2	Û
Forgot Password Mailer	Forgot Password Mailer	ľ	F	Ø	Û
Payment Failure Notification	Sent when the tenant has auto debit enabed and aut	ß	F	Ø	Û
Payment Success Notification	Sent when the tenant has auto debit enabed and gen	I	de la	3	Û
Pre-Subscription End Notification	Sent before Subscription ends.	đ	F	7	Û
Pre-Trail End Notification	Sent before trail ends.	đ	F	7	Û
Subscription Ended Notification	Sent when Subscription ends.	đ	JE.	7	Û
Tenant Activation Mail	Tenant Activation Mail	đ	JE.	3	Û
Tenant License Activation Mail	Sent when admin activates tenant license via UI.	ľ	al contract of the second s	1	Û
Tenant License Deactivation Mail	Sent when admin deactivates tenant license via UI	ľ	₽ ^E	đ	Û
Tenant Rejected Mail	Sent when parent tenant rejectes the skip level ch	ľ	al contract of the second s	Ø	Û
Tenant Subscription Changed Mail	Sent when tenant updates the subscription pacakge.	ľ	se.	Ø	Û
Tenant Welcome Mail	Sent when tenant clicks activates link or admin ap	ľ	de la construcción de la constru	8	Û
Trail Ended Notification	Sent when trail ends.	ľ	æ	Ø	Û
Un Paid Bills Alert Notification	Admin will sent this manually from Search bills UI	I	₽ ^E	Ø	Û
User Confirmation Mailer	User Confirmation Mailer	ľ	F	Ø	Û
User Password Mailer	User Password Mailer	ľ	J.	2	Û

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 Admin \rightarrow Notifications \rightarrow Manage Notification Details.

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.

My Account Subscription	Tenant Access Control Configuration Busi	ness Events	lotifications Workflo		or	
Manage Notification Details				Tenant: Company	•	Ado
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.	-	Ø	¥		Û
Forgot Password Mailer	Forgot Password Mailer	-	I	F	Ø	Û
Payment Failure Notification	Sent when the tenant has auto debit enabed and aut	-	ľ	Æ	7	Û
Payment Success Notification	Sent when the tenant has auto debit enabed and gen	-	ľ	F	Ø	Û
Pre-Subscription End Notification	Sent before Subscription ends.	-	ľ	F	7	Û
Pre-Trail End Notification	Sent before trail ends.	-	ľ	F	7	Û
Subscription Ended Notification	Sent when Subscription ends.	-	ľ	F	7	Û
Tenant Activation Mail	Tenant Activation Mail	-	ľ	F	7	Û
Tenant License Activation Mail	Sent when admin activates tenant license via UI.	-	ľ	F	Ø	Û
Tenant License Deactivation Mail	Sent when admin deactivates tenant license via UI	-	ľ	F	Ø	Û
Tenant Rejected Mail	Sent when parent tenant rejectes the skip level ch	-	ľ	p	8	Û
Tenant Subscription Changed Mail	Sent when tenant updates the subscription pacakge.	-	I	p	2	Û
Tenant Welcome Mail	Sent when tenant clicks activates link or admin ap	-	ľ	F	7	Û
Trail Ended Notification	Sent when trail ends.	-	ľ	F	Ø	Û
Un Paid Bills Alert Notification	Admin will sent this manually from Search bills UI	-	ľ	ş	Ø	Û
User Confirmation Mailer	User Confirmation Mailer	-	ľ	p	Ø	Û
User Password Mailer	User Password Mailer	-	ľ	F	Ø	Û
Show 25 - entries Showing 1 to 1	7 of 17 entries				K G	1 0
Manage Master Configurat						
Notification Name*						
Description						
Category						

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

Search Secret of the secret of t	My Account Subscription	Tenant Access Control Configuration B	usiness Events Notificati	ions Workflow	Reporting Monitor		
Notification Description Category Manage Dispatch <th< th=""><th>Manage Notification Details</th><th></th><th></th><th>Tenar</th><th>nt: Company</th><th>•</th><th>Ad</th></th<>	Manage Notification Details			Tenar	nt: Company	•	Ad
all Cenerated Notification Sent when bill is generated on the billing day.	Search	GO					
Fengel Person Maker Forgel Person Or variating a new Employee Employee Management Image Person Contracting a new Employee Symem Staker Notification Sent when the ternant has auto obbit enabed and age Image Person Contracting a new Employee Symem Staker Notification Sent when the ternant has auto obbit enabed and age Image Person Contracting a new Employee Symem Staker Notification Sent when the ternant has auto obbit enabed and gen Image Person Contracting a new Employee Person Link Notification Sent when the ternant has auto obbit enabed and gen Image Person Contracting a new Employee Person Link Notification Sent when Subscription ends. Image Person Contracting a new Employee Ternant Activation Mail Ternant Activation Mail Image Person Contracting a new Employee Ternant Activation Nail Sent when parent ternant rejectes the site previous on the site of the subscription package. Image Person Contracting a new Employee Ternant Network Devides the subscription package. Image Person Contracting a new Employee Image Person Contracting a new Employee Ternant Network Devides the subscription package. Image Person Contracting a new Employee Image Person Contracting a new Employee Ternant Network Devides the subscription package. Image Person Contracting a new Employee Image Person Contracting a new Employee Ternant Network Devides the subscription package. Image Person Contracting a new Employee Image Person Contracting a new Employee Ternant Network Devides the new Scription Package. Image Person Contracting a new Employee <	Notification Name	Description	Category	Manage Dispatch	Manage Global Dispatch	Edit	Delet
On Employee Creation Email Email sent on creating a new Employee Employee Management I <t< td=""><td>Bill Generated Notification</td><td>Sent when bill is generated on the billing day.</td><td>-</td><td>ľ</td><td>F</td><td>Ø</td><td>Û</td></t<>	Bill Generated Notification	Sent when bill is generated on the billing day.	-	ľ	F	Ø	Û
Payment Falure Notification Sent when the tranant has auto debit enabed and gen 0 <t< td=""><td>Forgot Password Mailer</td><td>Forgot Password Mailer</td><td></td><td>ľ</td><td>F</td><td>Ø</td><td>â</td></t<>	Forgot Password Mailer	Forgot Password Mailer		ľ	F	Ø	â
Payment Success Notification Sent when the tenant has auto debit enabed and gen i G A A G A G A G A G A G A G A G A G A	On Employee Creation Email	Email sent on creating a new Employee	Employee Management	ľ	-	ß	Û
here Sablascription End Notification Sent before Subscription ends. Pre Tail Datorification Sent before Subscription ends. Pre Tail Datorification Sent when Subscription ends. Pre Tail Datorification Sent when Subscription ends. Pre Tail Check Softwards Sent when Subscription Pracks Pre Tail Check Softwards Sent when Tail ends. Pre Tail Subscription Pracks Pre Tail Check Softwards Sent when Tail ends. Pre Tail Subscription Pracks Pre Tail Check Softwards Sent when Tail ends. Pre Tail Subscription Pracks Pre Tail Check Softwards Sent when Tail ends. Pre Tail Subscription Pracks Pre Tail Check Softwards Sent when Tail ends. Pre Tail Subscription Pracks Pre Tail Check Softwards Sent when Tail ends. Pre Tail Subscription Pracks Pre Tail Pr	Payment Failure Notification	Sent when the tenant has auto debit enabed and aut		ľ	F	Ø	Ê
Pre-Trail End Notification Sent before trail ends 0 4 0 <	Payment Success Notification	Sent when the tenant has auto debit enabed and gen		ľ	F	Ø	Û
Subscription Ended Notification Sent when Subscription ends 2 4 2 1 Tenant Activation Mail Tenant Activation Mail Sent when admin excluses tenant leense via UL - 2 4 2 4 2 4 2 4 2 4	Pre-Subscription End Notification	Sent before Subscription ends.		ľ	J.	Ø	Û
Tenant Activation Mail Tenant Activation Mail Tenant Activation Mail Tenant Activation Mail Sent when admin activates tenant license via UI. Tenant Rejected Mail Sent when admin dectivates tenant license via UI. Tenant Represent Carbon Mail Sent when admin dectivates tenant license via UI. Tenant Represent Mail Sent when attain dectivates tenant license via UI. Tenant Represent Mail Sent when tenant clicks activates link or admin ap Tenant Machen Mail Sent when tenant clicks activates link or admin ap Tenant Machen Mail Sent when tenant clicks activates link or admin ap Tenant Machen Mail User Confirmation Sent when trail ends. User Confirmation User Confirmation Mailer User Password Mailer User Password Mailer User Password Mailer User Password Mailer Sent Password Sent	Pre-Trail End Notification	Sent before trail ends.	•	đ	J [€]	1	Û
Tenant License Activation Mall Sent when admin activates tenant license via UI 0 <td< td=""><td>Subscription Ended Notification</td><td>Sent when Subscription ends.</td><td>•</td><td>ľ</td><td>F</td><td>ß</td><td>Û</td></td<>	Subscription Ended Notification	Sent when Subscription ends.	•	ľ	F	ß	Û
Tenant License Deactivation Mail Sent when parent tenant rejectse the skip level ch · <td>Fenant Activation Mail</td> <td>Tenant Activation Mail</td> <td>•</td> <td>đ</td> <td>₽^E</td> <td>1</td> <td>Û</td>	Fenant Activation Mail	Tenant Activation Mail	•	đ	₽ ^E	1	Û
Tenant Rejected Mail Sent when parent tenant rejectes the skip level ch Image Comparison Changed Mail Sent when tenant updates the subscription pacakage. Image Comparison Changed Mail Sent when tenant updates the subscription pacakage. Image Comparison Changed Mail Sent when tenant updates the subscription pacakage. Image Comparison Changed Mail Imag	Fenant License Activation Mail	Sent when admin activates tenant license via UI.		I	F	Ø	Û
Tenant Subscription Changed Mall Sent when tenant updates the subscription package. Image: Control Contro	Fenant License Deactivation Mail	Sent when admin deactivates tenant license via UI		đ	₽ ^E	1	Û
Terlant Welcome Mall Sent when tenant clicks activates link or admin ap ·	Tenant Rejected Mail	Sent when parent tenant rejectes the skip level ch		I	F	Ø	Û
Tail Ended Notification Sent when trail ends. in Paid Bills Alert Notification Jus Paid Bills Alert Notification User Confirmation Mailer User Confirmation Mailer User Password Mailer User Password Mailer Show 25 entries W 25 entries Showing 1 to 18 of 18 entries Wardification Type Email Sender Address* Subject* Attachment Fielder Has Template	enant Subscription Changed Mail	Sent when tenant updates the subscription pacakge.		đ	p	ľ	Û
In Paid Bils Alert Notification Admin will sent this manually from Search bills U If I and I an	Fenant Welcome Mail	Sent when tenant clicks activates link or admin ap		ľ	F	Ø	Û
Jser Confirmation Mailer User Confirmation Mailer User Password Password Smtp Password Enable SSL Email Content Details Subject* Attachment Folder Has Template Attachment files Induced Password Pass	Frail Ended Notification	Sent when trail ends.		ľ	F	Ø	Û
beer Password Mailer User Password Mailer 25 entries Showing 1 to 18 of 18 entries	In Paid Bills Alert Notification	Admin will sent this manually from Search bills UI	-	I	F	Ø	Û
Show 25 entries Showing 1 to 18 of 18 entries H	Jser Confirmation Mailer	User Confirmation Mailer		C	Æ	6	Û
Date Employee Creation Email Notification Type Email Manage Email Dispatch Details Notification Type Email Sender Address* Smtp UserName* Port Number* Port Number* Enable SSL Email Content Details Subject* Has Template	Jser Password Mailer	User Password Mailer	-	ľ	JE.	Ø	Û
Da Employee Creation Email Notification Type Email Manage Email Dispatch Details Motification Type Email Sender Address* Smtp UserName* Port Number* Port Number* Subject* Has Template	Show 25 💌 entries Showing 1 to	0 18 of 18 entries			k	0	0
Smtp UserName* Smtp Password* Port Number* 0 Email Content Details Subject* Has Template Attachment files	• Manage Email Dispatch D	retails					
Smtp UserName* Port Number* 0 Email Content Details Subject* Has Template Attachment files							
Port Number* 0 Enable SSL Email Content Details Subject* Attachment Folder Has Template Attachment files	Notification Type Ema	11					
Email Content Details Subject* Has Template Attachment Folder	Notification Type Ema	11	Idress*				
Subject* Attachment Folder Has Template Attachment files	Notification Type Ema	il Smtp Ac					
Has Template Attachment files	Notification Type Ema Sender Address*	il Smtp Ac	sword*				
Has Template	Notification Type Ema Sender Address ^e Smtp UserName ^e Port Number ^e	il Smtp Ac	sword*				
	Notification Type Ema Sender Address* Smtp UserName* Port Number* 0 Email Content Details	il Smtp Ac Smtp Pas Enal	sword*				
Content* B I ∐ de x, x' T. nF. H. T _a T _b ⊥ ⊟ 詳 詳 ≣ ≣ ≡ ≡ ≡ □ ○ ○ ≡ ■ % ↔ ½ ≥ 2. C C C C C C C C C C C C C C C C C C	Notification Type Ema Sender Address* Smtp UserName* Port Number* 0 Email Content Details Subject*	il Smtp Ac Smtp Pas Enal	sword*				
	Notification Type Ema Sender Address* Smtp UserName* Port Number* 0 Email Content Details Subject*	il Smtp Ac Smtp Pas Enal	sword*				
	Notification Type Ema Sender Address* . Smtp UserName* . Port Number* 0 Email Content Details Subject* . Has Template .	il Smtp Ac Smtp Pas Enal Attachment Attachme	sword* ble SSL Folder nt files				
	Notification Type Ema Sender Address* . Smtp UserName* . Port Number* 0 Email Content Details Subject* . Has Template .	il Smtp Ac Smtp Pas Enal Attachment Attachme	sword* ble SSL Folder nt files				
	Notification Type Ema Sender Address* . Smtp UserName* . Port Number* 0 Email Content Details Subject* . Has Template .	il Smtp Ac Smtp Pas Enal Attachment Attachme	sword* ble SSL Folder nt files				
	Notification Type Email Sender Address* Smtp UserName* Port Number* Email Content Details Subject* Has Template	il Smtp Ac Smtp Pas Enal Attachment Attachme	sword* ble SSL Folder nt files				
	Notification Type Ema Sender Address* Smtp UserName* Port Number* Email Content Details Subject* Has Template	il Smtp Ac Smtp Pas Enal Attachment Attachme	sword* ble SSL Folder nt files				
	Notification Type Email Sender Address* Smtp UserName* Port Number* Email Content Details Subject* Has Template	il Smtp Ac Smtp Pas Enal Attachment Attachme	sword* ble SSL Folder nt files				
	Notification Type Ema Sender Address* . Smtp UserName* . Port Number* 0 Email Content Details Subject* . Has Template .	il Smtp Ac Smtp Pas Enal Attachment Attachme	sword* ble SSL Folder nt files				

Copyright © 2013 by techcello.com All Rights Reserved.

v 4.3.1.0

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 Detai	ls (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 Co	ontent 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 System Notification Map Id		
мар Id Add System Notification Content Has Template Content* В Г ц нь х, х Тъ Пънн-Та Тъ ПЕ Р Р Р Р Р Р Р Р Р Р Р Р Р Р Р Р Р Р	Manage System Notification Details	
Add System Notification Content Has Template Content B / II de x, x' T- fT- HI- Ta Ta Ta FIF # FF	Notification Type SystemNotification	
Has Template Content* B I I I de x, x' T- fT- H- Ta	Map Id	
	Add System Notification Content	
	Has Template 🔲	
Save Cancel Clear	Content * B / U 地 x, x* T- T- H- Tg 野 足 汪 汪 譯 譯 臣 吾 君 〓 ッ や 三 圖 � ↔ ※ 心 & 隆 읍 @	
Save Cancel Clear		
		Save Cancel Clear

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.			C.	F	Ø	Ē
Un Paid Bills Alert Notification		Admin will sent this manually from Search	bills UI		2	F	Ø	Û
User Confirmation Mailer		User Confirmation Mailer			ľ	JF.	Ø	Ē
User Password Mailer		User Password Mailer			Z	F	Ø	Û
Show 25 💌 entries Show	ving 1 to 17 of	17 entries					KØ	1 Ø И
Manage Email Dispatch Det	ails : Forgot	Password Mailer				Override	Details	Cancel
Notification Type	Email							
Sender Address	cellosaasden	no@gmail.com	Smtp Address	smtp.gmail.com				
Smtp UserName	cellosaasden	no@gmail.com	Smtp Password	sinininini				
Port Number	587		Enable SSL	True				
Email Content Details								
Subject	{{tenantname	2}} - Forgot Password	Attachment Folder					
Attachment files								
Template Rule Set			Template Name	Forgot Password	d Template			

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 Admin \rightarrow Notifications \rightarrow Content Template

CelloSaaS		admir	n@company.com ▼ ≓ C	hange Tenar
My Account Subscription Tena	nt Access Control Configuration Business Events		orting Monitor	
Template Management		Tenant:	Company	Add
Search		Template ca	ategory: Notification Ter	nplate
Template name	Description	Category	Manage Template	Edit
Bill Generated Template	Sent when bill is generated on the billing day.	NotificationTemplate	ş	Ø
Confirmation Mailer Template	Confirmation Mailer Template	NotificationTemplate	F	ľ
Forgot Password Template	Forgot Password Template	NotificationTemplate	JE.	ľ
Password Mailer Template	Password Mailer Template	NotificationTemplate	J.E.	
Payment Failure Template	Sent when the tenant has auto debit enabed and aut	NotificationTemplate	Æ	ľ
Payment Success Template	Sent when the tenant has auto debit enabed and gen	NotificationTemplate	æ	
Pre-Subscription End Template Sent before subscription ends.		NotificationTemplate	JE.	ľ
Pre-Trail End Template	Sent before trail ends.	NotificationTemplate	æ	ľ
Self Tenant Activation Template	Email Template for self registered tenant welcome	NotificationTemplate	F	ľ
Subscription Ended Template	Sent when subscription ends.	NotificationTemplate	F	6

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 a event triggered in the system, a 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

CelloSaaS		admin@	©company.com ▼ ≓ Change Tenant				
My Account Subscription Tenant	Access Control Configuration Business Events Not	tifications Workflow Repor	ting Monitor				
Template Management		Tenant: C	ompany 🔹 Add				
Search		Template cat	egory: Notification Template 🔹				
Template name	Description	Category	Manage Template Edit				
Bill Generated Template	Sent when bill is generated on the billing day.	NotificationTemplate	₽ B				
Confirmation Mailer Template	Confirmation Mailer Template	NotificationTemplate	₽ I				
Forgot Password Template	Forgot Password Template	NotificationTemplate	₽ I				
Password Mailer Template	Password Mailer Template	NotificationTemplate	₽ B				
Payment Failure Template	Sent when the tenant has auto debit enabed and aut	NotificationTemplate	₽ B				
Payment Success Template	Sent when the tenant has auto debit enabed and gen	NotificationTemplate	₽ I				
Pre-Subscription End Template	Sent before subscription ends.	NotificationTemplate	₽ I				
Pre-Trail End Template	Sent before trail ends.	NotificationTemplate	₽ C				
Self Tenant Activation Template	Email Template for self registered tenant welcome	NotificationTemplate	₽ C				
Subscription Ended Template	Sent when subscription ends.	NotificationTemplate	₽ I				
Show 10 💌 entries Showing 1 to 10 of 17 entries 🛛 🖌 🖉 🕨							
Manage Template Template name® Template description	Template category		Save Cancel				

To add Locale based template content *1.12.*

admin@company.com ▼ | ≓ Change Tenant

CelloSaa	S							admin@compa	any.com 🔻 🕴 i	⇒ Change Tenan	it
My Account	1 Subscription	Tenant		Configuration	E Business Events	Notifications	Workflow	Reporting	Monitor		
Manage Templat	es : Bill Gener	ated Templat	e							Add Back	¢
Search		GO									
Locale Name					late Content	11/m			Edit Details	Delete Detail	is
en				Dear	{{Tenant.ContactDetail.Fi	rstName}} {{Tenant.C			ľ	â	_
Show 10 entre Edit Template Deta Locale Name Template*	nils en B <u>Z</u> Za	ње х. х. Т. 1	С- нı- Т<u>а</u> Ђ ⊥ ∺ Ξ ∦ stName)} {(Tenant. Conta		■ *) (*) ■ ● %	@ ¥ % & % @			Ņ	010)	М
	Your bill #{{In USD has bee Payment will	voice.InvoiceNo}} 1 in generated. be made automat r using CelloSaaS	or the period {{Invoice.Sta	artDate}} to {{Invoice.En	dDate}} with the Net Amou	nt of \${[Invoice.Amount]	9			Save Cancel	
								Conversite @ 2012.1			
v 4.3.1.0								Copyright © 2013 l	by techceno.com	An Rights Reserved	e.

Introduction to Dynamic Place Holders *1.13.*

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></UserD etails></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 Admin \rightarrow Configuration \rightarrow Master Data \rightarrow Locale Names

CelloSa	aS							admin@compa	uny.com ▼ ≓	Change Tenant
My Account	Subscription	Tenant	Access Control	Configuration	Business Events	Notifications	Workflow	Reporting	O Monitor	
Manage Local	eNames Details									Add
Search	G	0								
Locale Name				Description					Edit	Delete
en				General English					ø	Û

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 localeld. 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 notificationDestination 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 notification details by using the notification id and the tenant id. public static NotificationDetails GetNotificationDetails(String notificationId, String tenantId)

Notification Management

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 <u>sample application</u>. 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.BatchEmployeeNotification Id, 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 Admin \rightarrow Monitor \rightarrow Notification to see the audited information. All the Failure and Succesfully sent emails along with the outgoing content can be seen in the Email audit Screen.

Image: My Account Image: Subscription Image: Subscription <thimage: subscription<="" th=""> Image: Subscription Image: Subscri Subscription Image: Subscrip</thimage:>	CelloSaa	S							admin@compa	ny.com 🔻	≓ Cha	ange T	ena	at
Notification type Email Notification name -All- Status -All- Tenant Activation Mail Sent on 10/24/2013 6:05 AM ©	My Account S		Tenant A		19 <u>8</u>		Notifications	Workflow	Reporting	\sim				
Notification type Email 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 O	Notification Audit	Details							Tenant:	Company			Ŧ)
End time Map Id Reset Search To : :doins@gmail.com Subjet: [Activation] Welcome to Techcello Dear admin hame, Welcome to Techcello and thank you for registering with us. To activate your account please click here. Login Credentials: Company Code: jill User MaininBjill2.com Password :: :[4HingcQ Best: Welcome to not/23/2013 12:02 PM © To :: contact@agilebio.com From :: cellosaasdemo@gmail.com Subject: Welcome To Techcello Dear MaininBjill2.com Password :: :[4HingcQ Best: Welcome To Techcello Do :: contact@agilebio.com From :: cellosaasdemo@gmail.com Subject: Welcome To Techcello Best: Welcome To Techcello Best: Welcome To Techcello Best: Welcome to Techcello Best: Wishes, Techcello Web Admin Million: the million of to the field of the million of the field of the million of the field of the million of the million of the field of the million of	Notification name Status Start time End time Map Id	-All- -All-		Dear Jessi Your Trail - Please upg Thanks for Tenant A To : a From : c Subject: [Dear admi Welcome t To activate Login Cree Company User Name Password Best Wishe Techcello Tenant V To : c Subject: V Dear Plerrr Welcome t Best Wishe	ca Alba, ended on 10/26/2013 rade your subscription t rusing Techcello. We Activation Mail Ser dmin@jjill2.com ellosaasdemo@gmail.d Activation] Welcome tr n Iname, to Techcello and thani your account please cli fentials: Code: jjill e : admin@jjill2.com : 2[4HIYgcQ es, Web Admin Velcome Mail Sent ontact@agilebio.com ellosaasdemo@gmail.ov (elcome To Techcello e Rodrigues, to Techcello. es, web Admin	o continue our services. value your business. You nt on 10/24/2013 6:05 / com o Techcello k you for registering with ck here. n on 10/23/2013 12:02 P com	r sincerely,Techcello V AM ⊘ h us.		И () 1	2 3	4 5 6	7	8	

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.

Notification Management

Organize 🔻 📑 Open	Print Burn New folder					800 -	
🔶 Favorites	Name	Date modified	Туре	lize			
Desktop	🚳 install	24-09-2013 19:18	Windows Batch File	1 KB			
Downloads	Ins C:\windows\System32\cmd.exe	A	And in case of the local				
🔢 Recent Places	 Mic Rolling back assembly 'D:\ Mic nstallers Notification Cel Affected parameters are: 	cellosaas_v4.3 loSaaS.Notific	.1_developer_pack ationService.exe'	age\CelloSaaS S(•	ervice I		
 Libraries Documents Music Pictures Videos 	Min logtoconsole = logfile = D:xcellosaas. Min Notification:XcelloSaaS.Not Min logstath = D:xcello Min lers:Notification:XcelloSaa Min Restoring event log to pre Min The Rollback phase complet Min The transacted install has	ificationServ: saas_v4.3.1_de S.Notification vious state fo ed successful:	ice.InstallLog veloper_package\C Service.exe or source BatchNot	elloSaaS Service	e Instal		
Computer S (C:) New Volume (D:)	 Mit The installation failed, a Mit D:\cellosaas_v4.3.1_develo Mit D:\cellosaas_v6.3.1_develo net start "BatchNotificati Mit The requested service has 	nd the rollbad per_package\Cd onService" already been s	elloSaaS Service I started.		ication>		
 New Volume (E:) Internal (\\aspire13) projects (\\aspire13) 	Mic More help is available by Mic Mic Mic D:\cellosaas_v4.3.1_develo pause	per_package\Ce		nstallers\Notif:	ication>		
🖵 QA (\\aspire13) (Q:)	Mic Press any key to continue 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			
Network	Microsoft.Practices.Unity.Configuration.dll	19-06-2013 19:53	Application extens	50 KB			
- Herron	Microsoft.Practices.Unity.dll	19-06-2013 19:53	Application extens	70 KB			
- Helitone	MySql.Data.dll	19-06-2013 19:52	Application extens	346 KB			
		19-06-2013 19:58	Application extens	2.636 KB			
	NHibernate.dll						
	NHibernate.dll	19-06-2013 19:58	Application extens	168 KB			
	Sorg.Mentalis.Security.dll		Application extens Application extens	168 KB 644 KB			
	Security.dll	19-06-2013 19:53 19-06-2013 19:53	Application extens				
	Sorg.Mentalis.Security.dll	19-06-2013 19:53		644 KB			

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