**Simple Push Notification Module**

* Get push notifications with simple configuration.
* User has the access to make the notification as read.
* A click on the notification will redirect to that particular context page, with a simple customization.
* Administration user can configure the notification Subject and Message.
* Module can be easily configurable to support role based notification or group based notification.
* It also support dynamic data in the notification using the MX Model Reflection.
* App store link : - <https://marketplace.mendix.com/link/component/112194>
* Git Hub Link:- <https://github.com/TechTammina/MendixSimplePushNotifications>
* See the Demo :- <https://pushnotifications113-sandbox.mxapps.io/login.html>

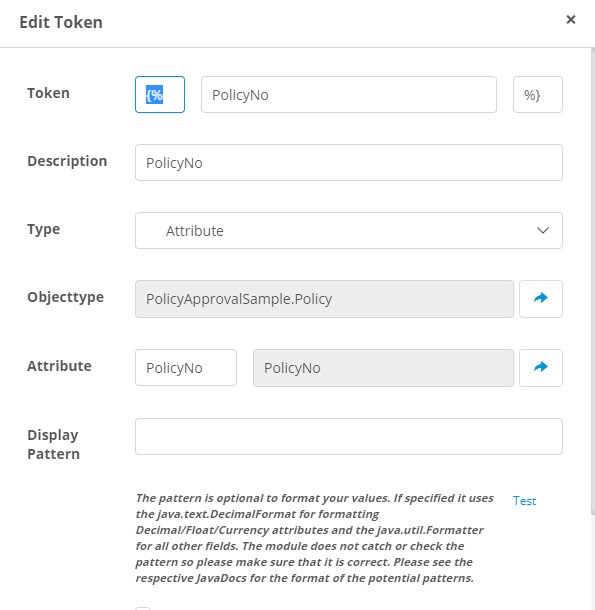
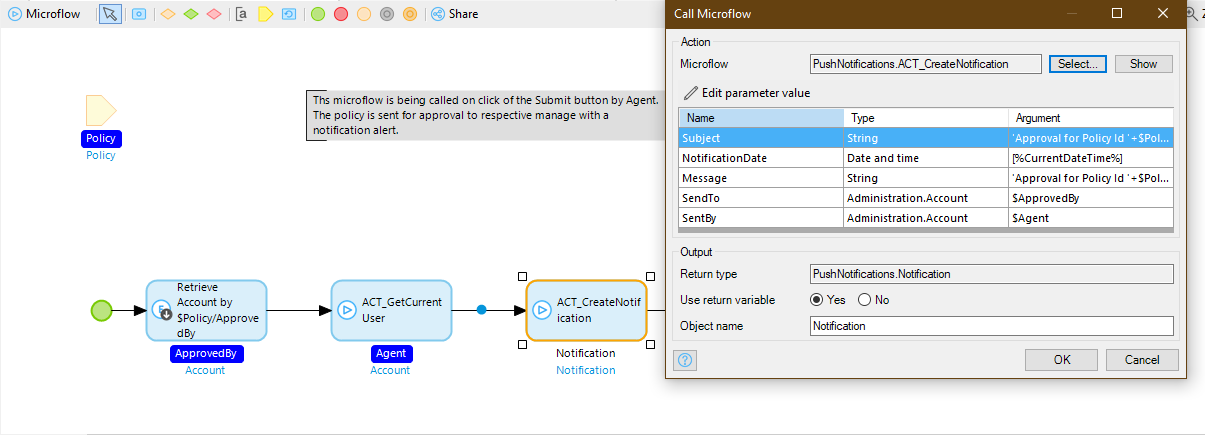
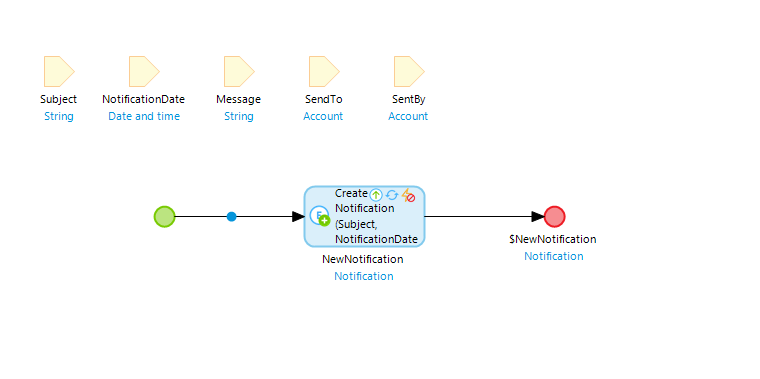
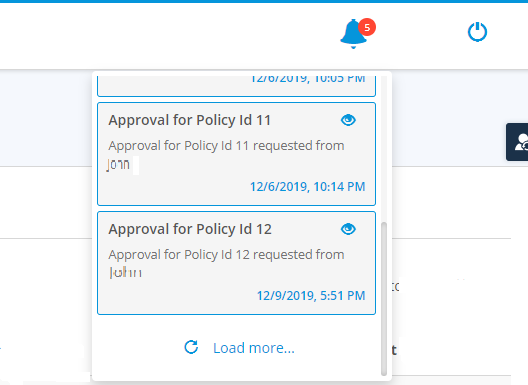
### Configuration for the module:-

1. Download **MxModelReflection**from app store, as it is dependency module to synchronize the entities.
2. Add **MxObjectOverview** page into the navigation with Administration rights.
3. Add **MessageTemplateConfig** page into the navigation giving only administration access to it.

Administrator can add multiple templates based on requirement and set Subject, Message of the notification. Dynamic values can also be set using tokens.

1. Add **CurrentUser\_Snippet**from **USE\_ME**folder in your layouts.
2. Call **SUB\_CreateNotification** microflow when needed to generate a notification in your microflow.
3. Retrieve MessageTemplate object with some valid name and add Subject, Message and CurrentDateTime to **SUB\_CreateNotificaiton** microflow parameters.
4. Also add SUB\_ReplaceNotificationToken microflow, and required parameters to it, to get the dynamic values from tokens.
5. Now retrieve the account from **Administrator.Account**entity to whom the notification has to be sent and map to **SentTo**parameter.
6. Similarly, retrieve the accountfrom whom the notification being sent and map to **SentFrom** parameter.
7. For redirecting to a page on click of notification, you can use the noitification category enumeration.
8. For example, if the user needs to send a notification for a policy approval, we can create a new enumeration value (like PolicyApproval). Create an association from notification entity to that particular entity (to which page it has to get redirected). Now use **ACT\_GoToSource** microflow for an onclick. You can use the exclusive split and divide the paths using enumeration values, retrieve the entity which is been associated while creating a notification object, now show page  pass the entity which is been retrieved.
9. After running the application, login with Admin, navigate to MxObject\_Overview page and sync all the necessary entities to select them in the MessageTemplate configuration as tokens.
10. Now navigate to Message template configuration, create new template with some valid name, given while retrieving  MessageTemplate object in step 6.
11. Select the Message template and create new tokens for the MessageTemplate, to get the dynamic values. Copy the token names and add them in the Subject or Message along with the string. Eg. Subject: Need approval for policy no {%PolicyNo%}, sent from {%Agent%}.

**Screenshots**



**Toastr Notifications Widget**

* Displays non-blocking notifications to the user, sourced from a microflow.
* Display notifications for 4 types: success, info, warning, error
* Hide notifications based on a configurable timeout (with or without a progress bar), when the user clicks the notification or when the user clicks the close button.
* Control whether notifications are displayed with a Boolean attribute in your context entity
* Call a microflow on any of the 4 available events: On Click, On Close Click, On Hidden and On Shown
* Define custom classes for use on the notifications
* Control configuration using attributes on the Notification entity or default properties within the widget.
* Shows notification to all the users not for the specific one.
* App store Link:- <https://marketplace.mendix.com/link/component/45837>

Screenshots:-

