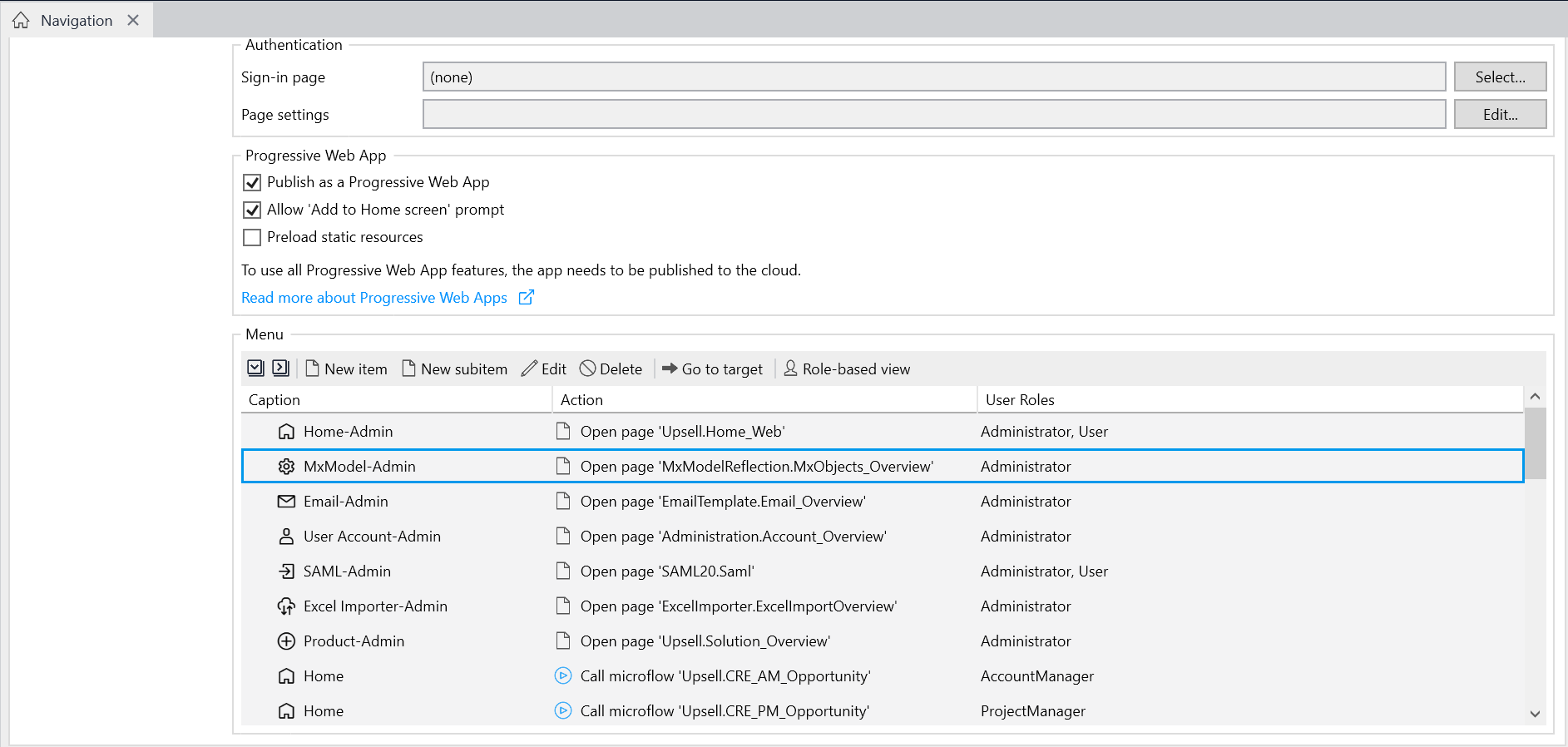
# SSO, email and Oauth Configuration Document

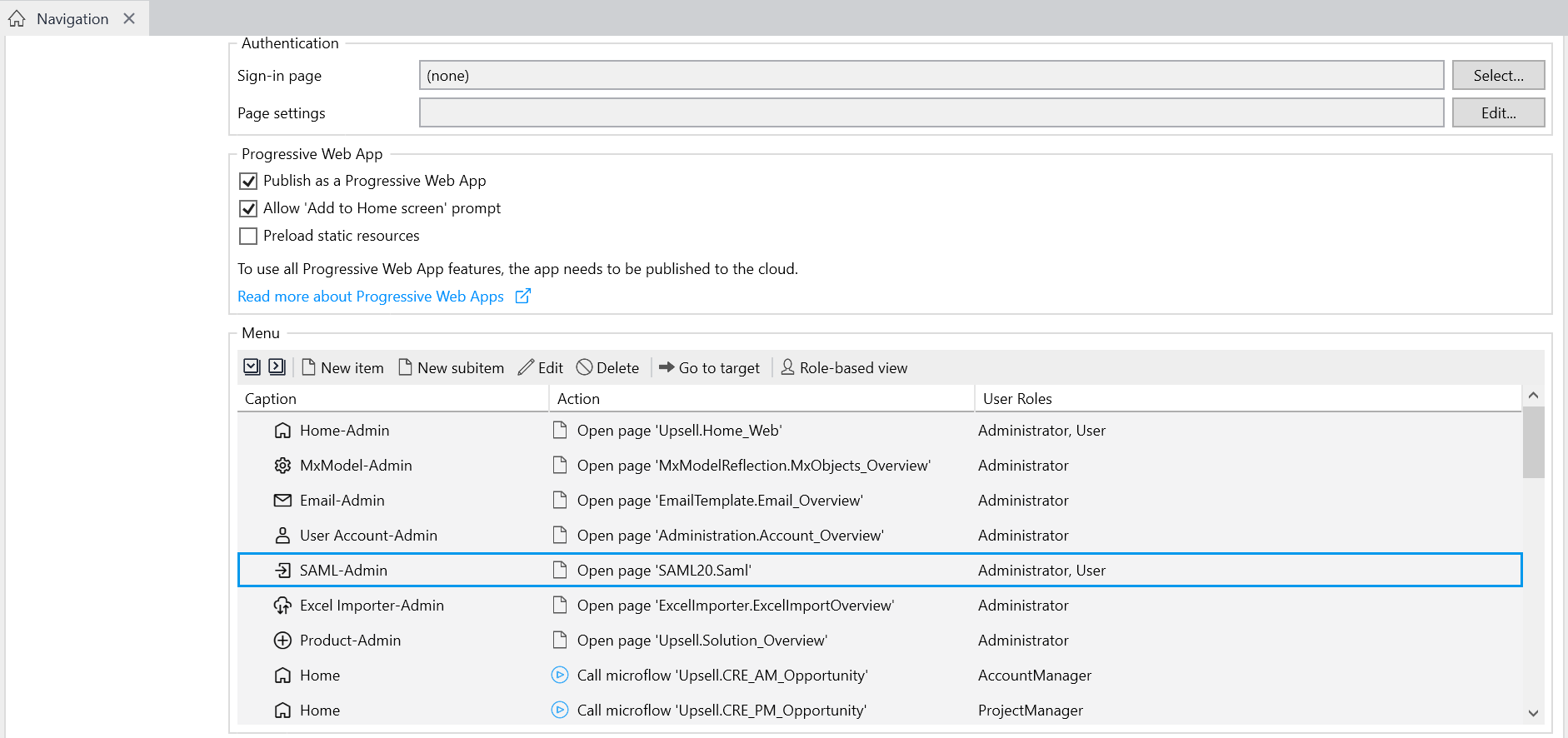
**SSO :**

Steps :

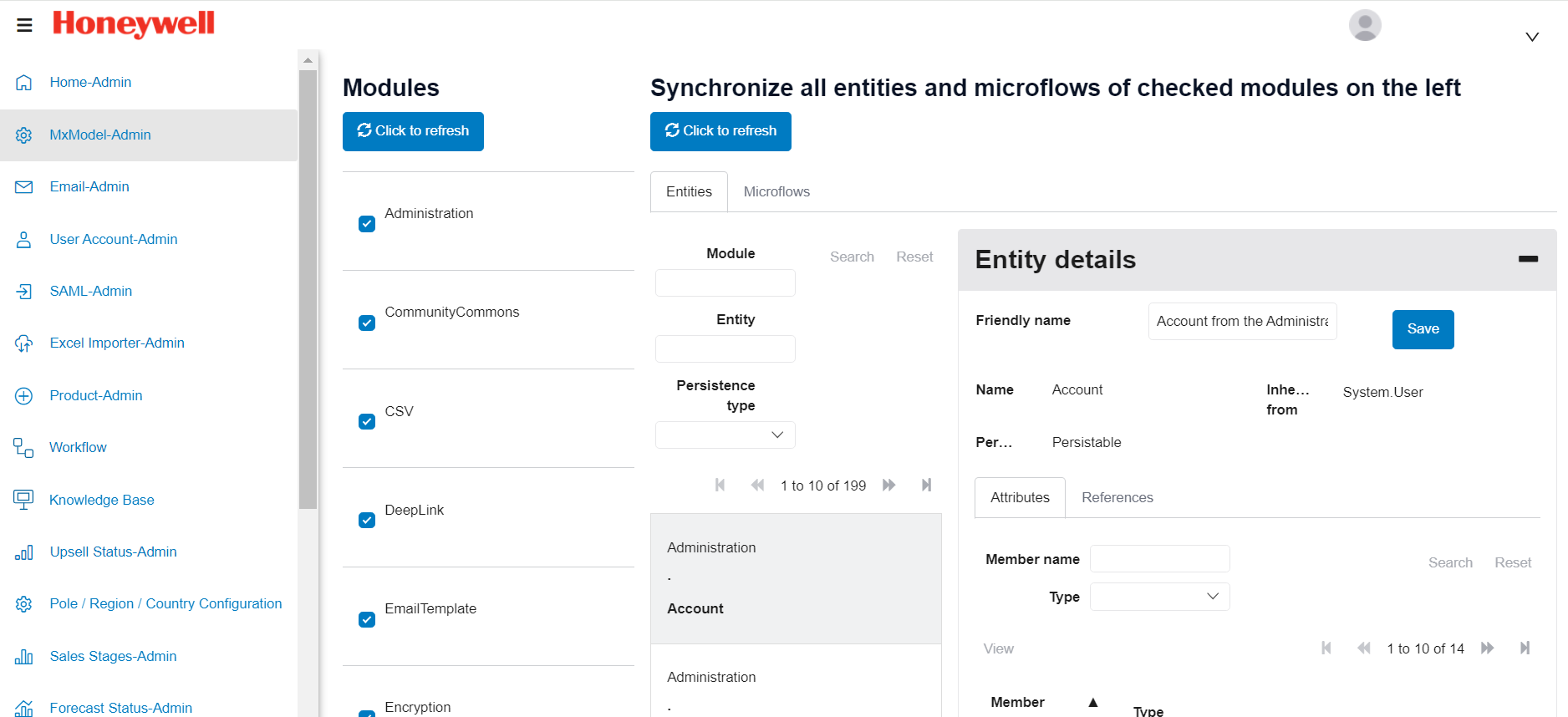
1. Download SAML 2.0 module from Mendix marketplace.
2. Download MX model reflection module from marketplace.
3. Get MxObjects\_overview page configured in navigation menu as highlighted below.



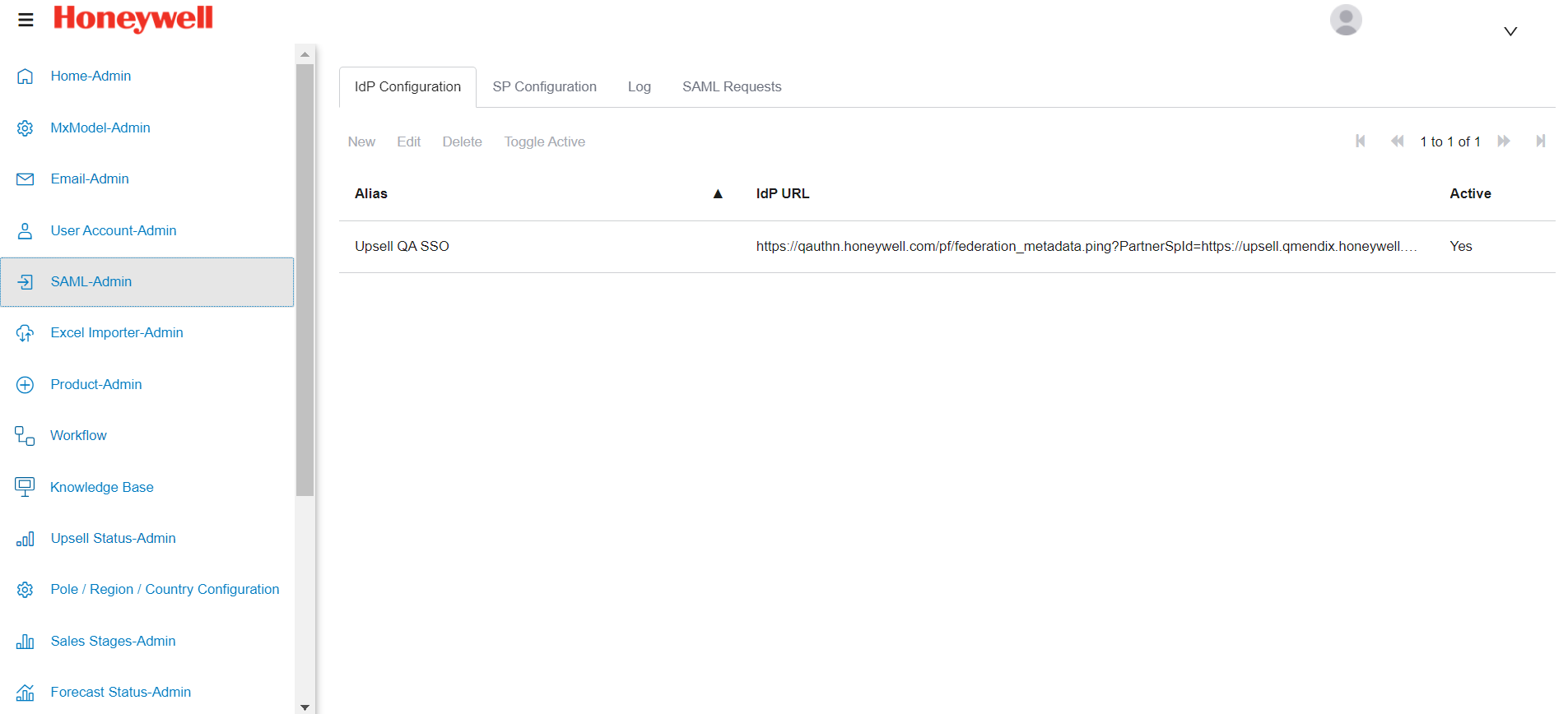
1. Also configure the SAML page configured in navigation menu as highlighted below.



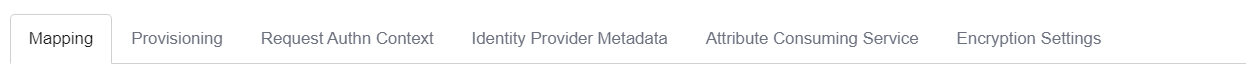
1. Now run the application and go to the web browser, click on navigation menu, then click on MxModel reflection page, post that hit refresh.



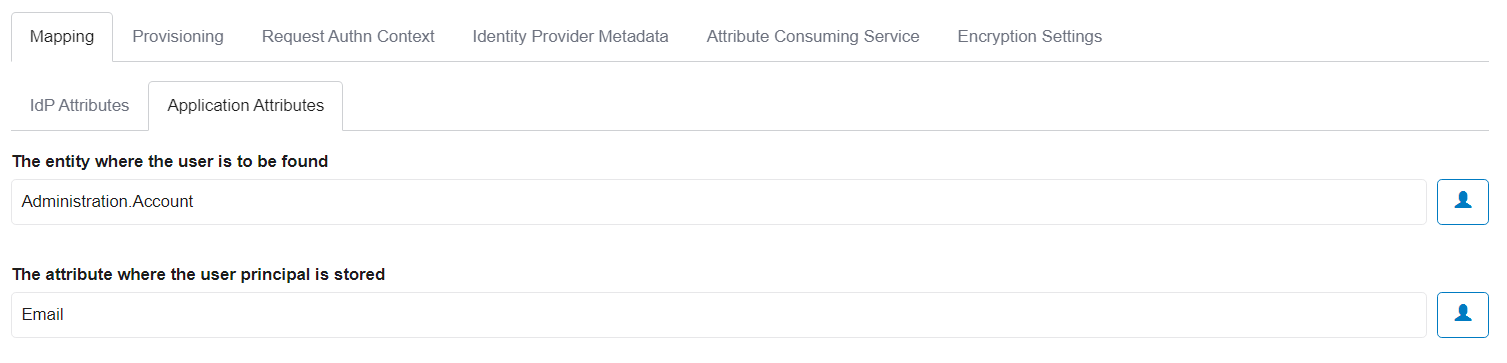
1. Once you are done with the refresh, click on saml page from the navigation menu.



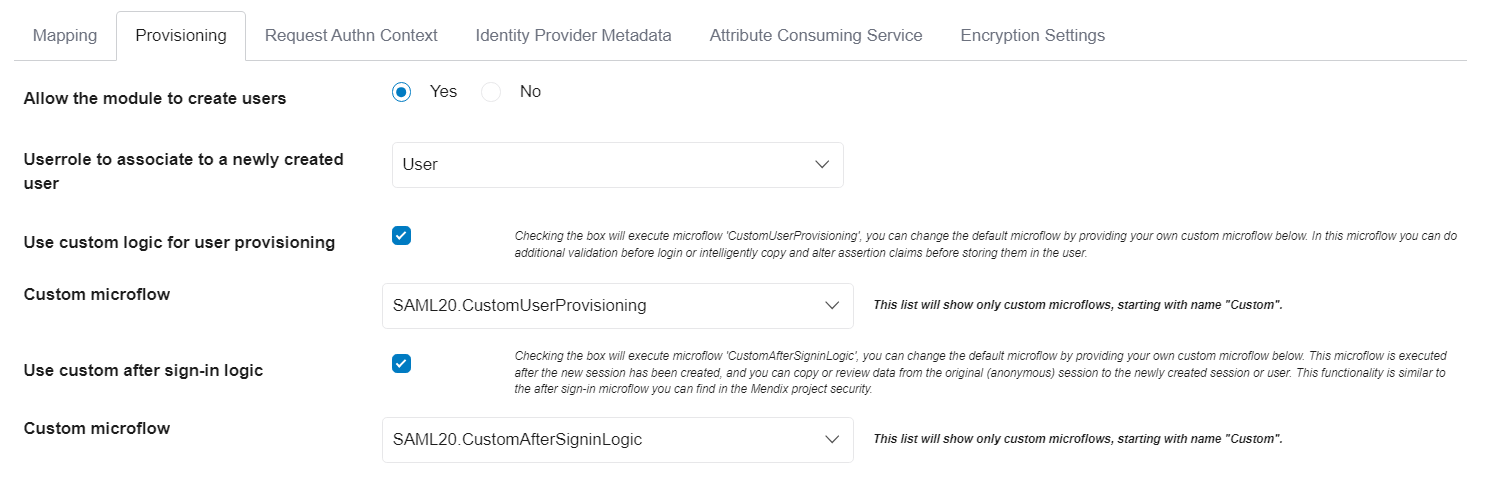
1. In the IDP configuration tab, click on new, you’ll see below tabs that you can use to configure the SSO.



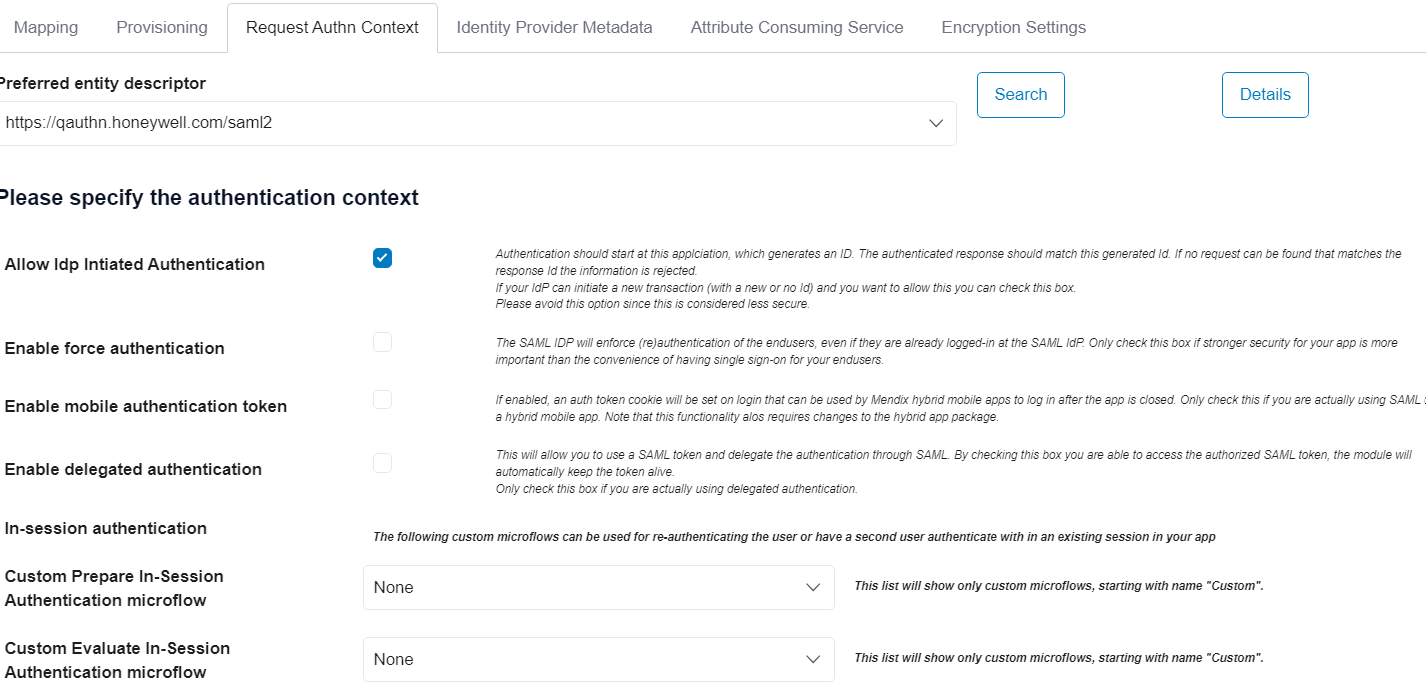
1. Click on mapping, and select the radio button idp provided.
2. Go to next sub tab “Application Attributes” >> Select Administrator.Account Entity, also mention the attribute where you wish to store the principal from SSO.



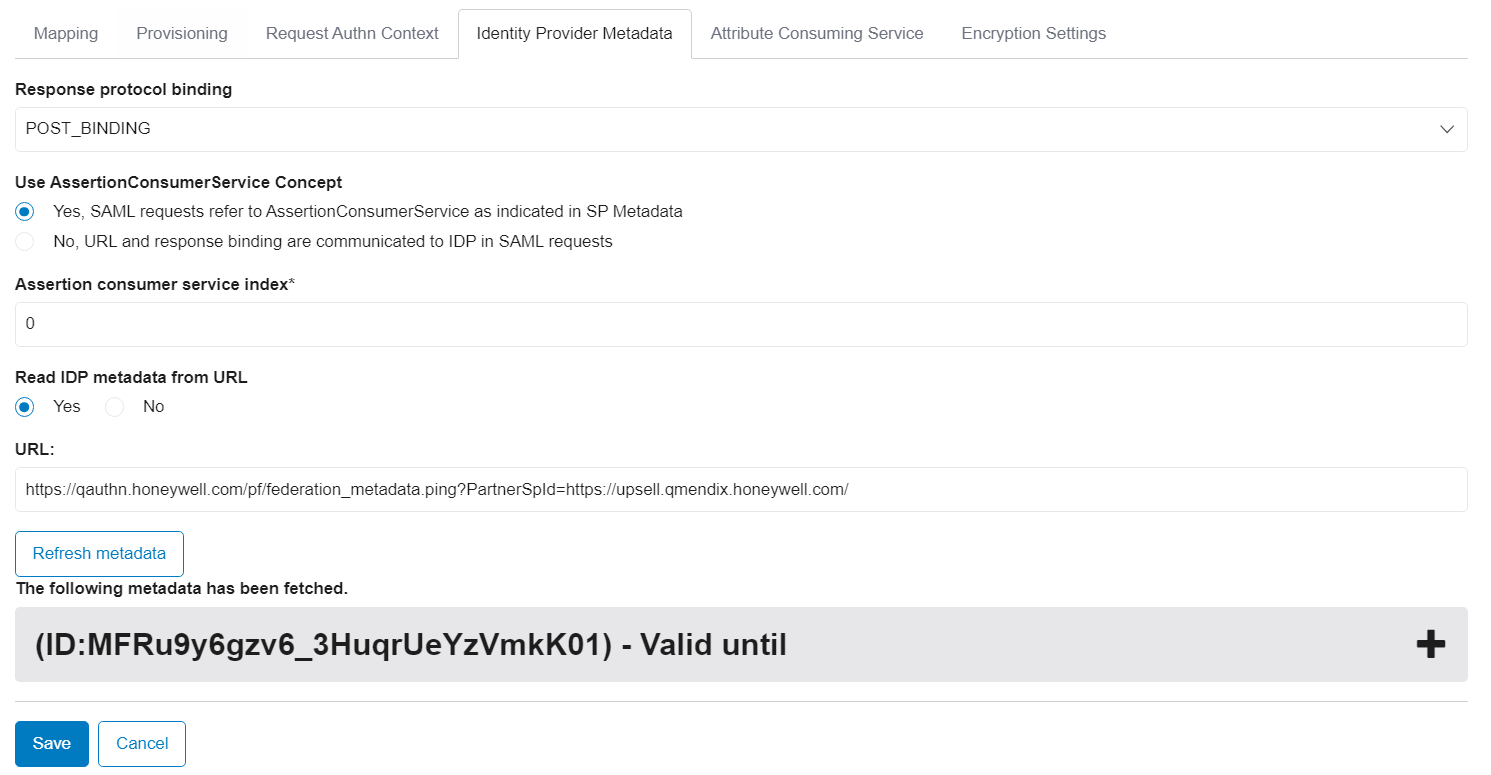
1. Click on provisioning tab >> select yes for allow the modules to create the users >> select anonymous user role from the dropdown >> click on use custom logic for user provisioning >> select the custom microflow where you have written the logic for user provisioning.



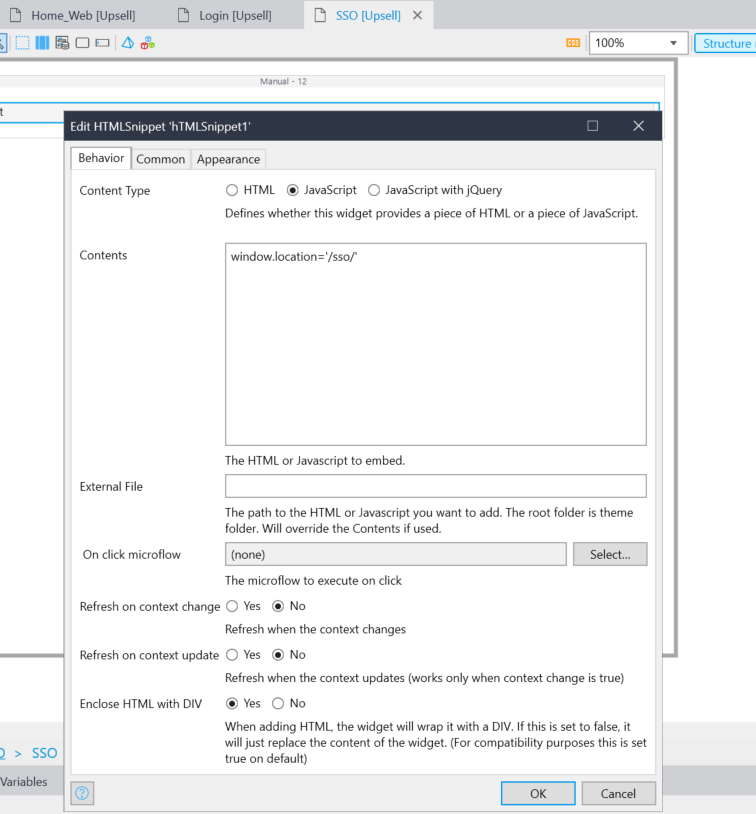
1. Click on Request Authn Context >> Enter the Preferred entity Descriptor >> click the checkbox of Allow idp Initiated Authentication.



1. Click on Identity Provider Metadata tab >> Select the required protocol binding method >> Select yes radio button for use AssertionConsumerService Concept. >> Select yes for Read IDP metadata from URL >> Paste the metadata URL, Click on refresh metadata and then save.

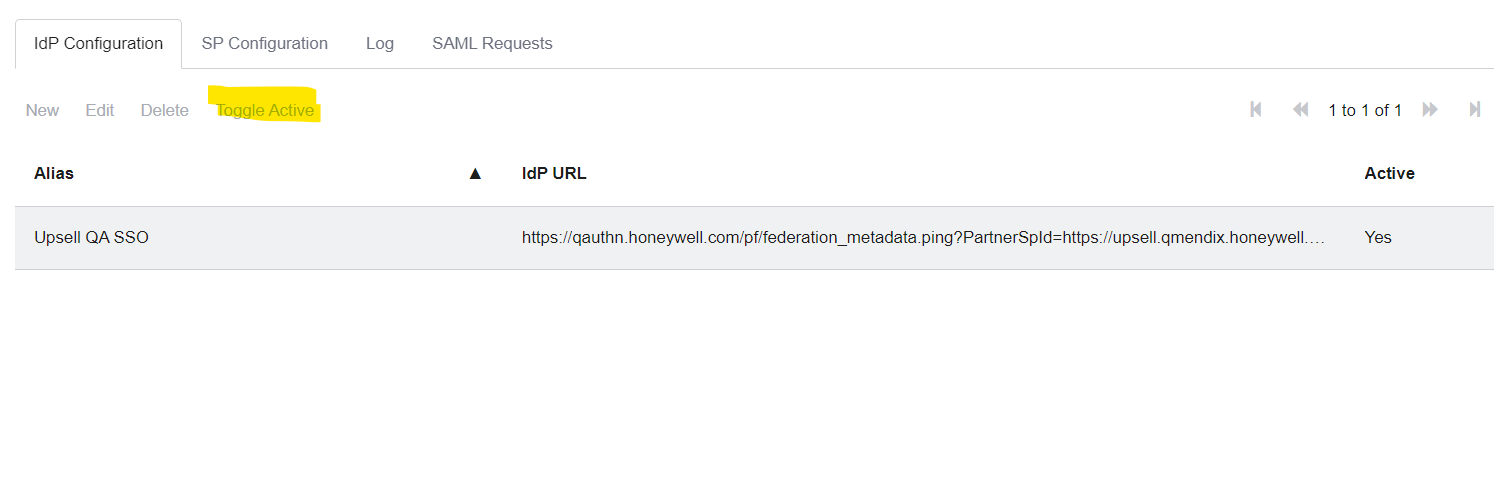


1. No need to change anything in Attribute consuming service and encryption settings.
2. Now go to Studio pro, and open the SSO page html snippet, configure the snippet like mentioned below



1. Now you have configure everything required for SSO, you just need to enable the SSO configuration so that it starts working for you.

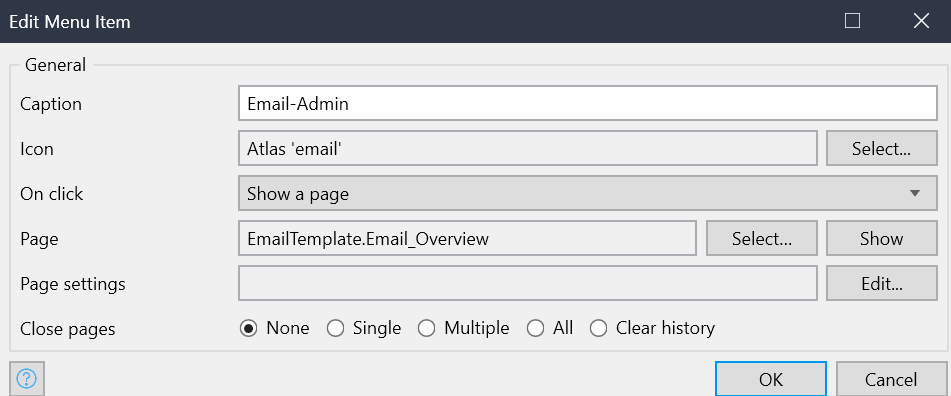
For this, go to Saml page from the navigation menu, Select the IDP configuration you just configured and then click on toggle active.



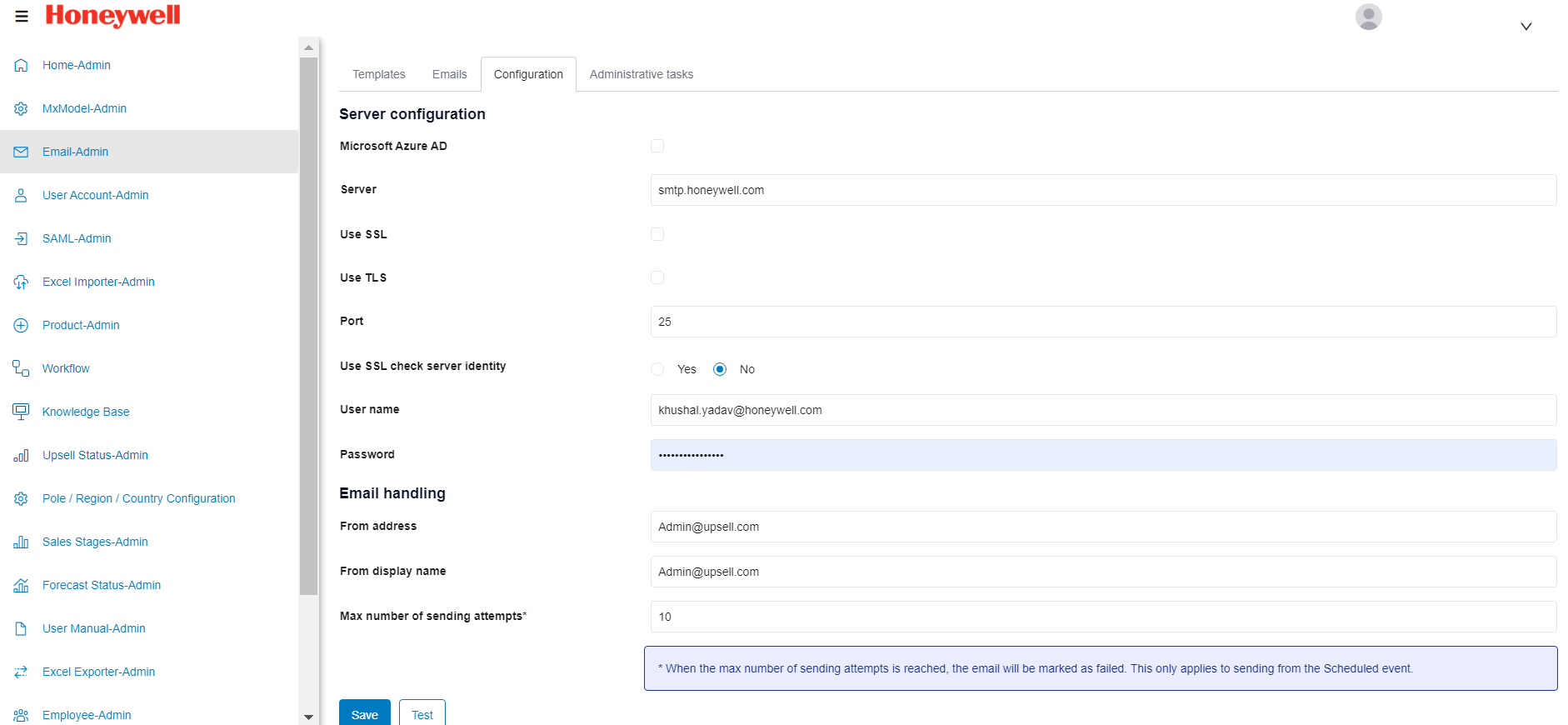
You are now all set to use the SSO!!

**Email Configuration :**

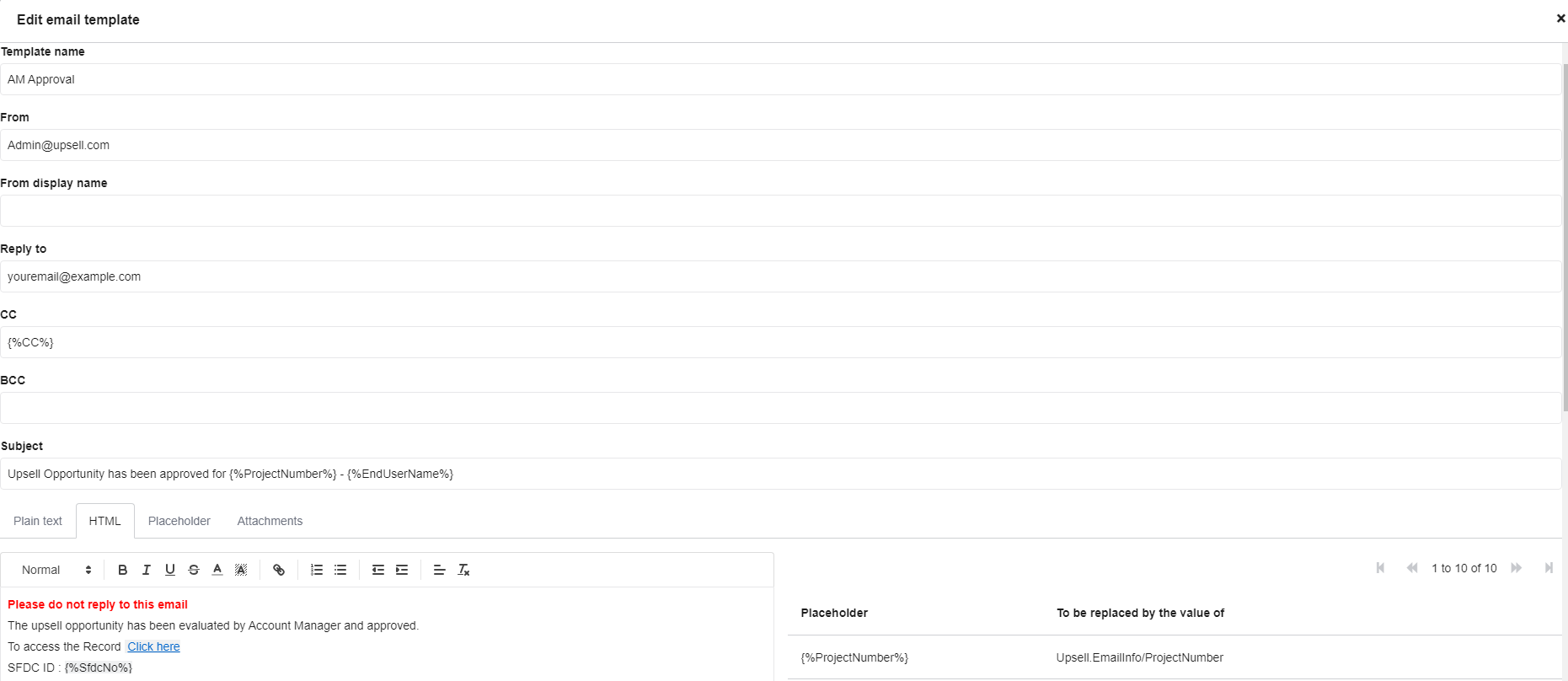
1. Import “EmailTemplate” marketplace module in your Mendix app.
2. Configure Email\_Overview Page on the navigation menu like mentioned below.



1. Now run your Mendix application and click on Email page you configured in the navigation menu >> after that click on sub navigation “Configuration”. >> configure the email configuration like mentioned below.



1. The next step is to configure the email templates in the app which can be used to send the emails to user. You can configure the email templates like mentioned below

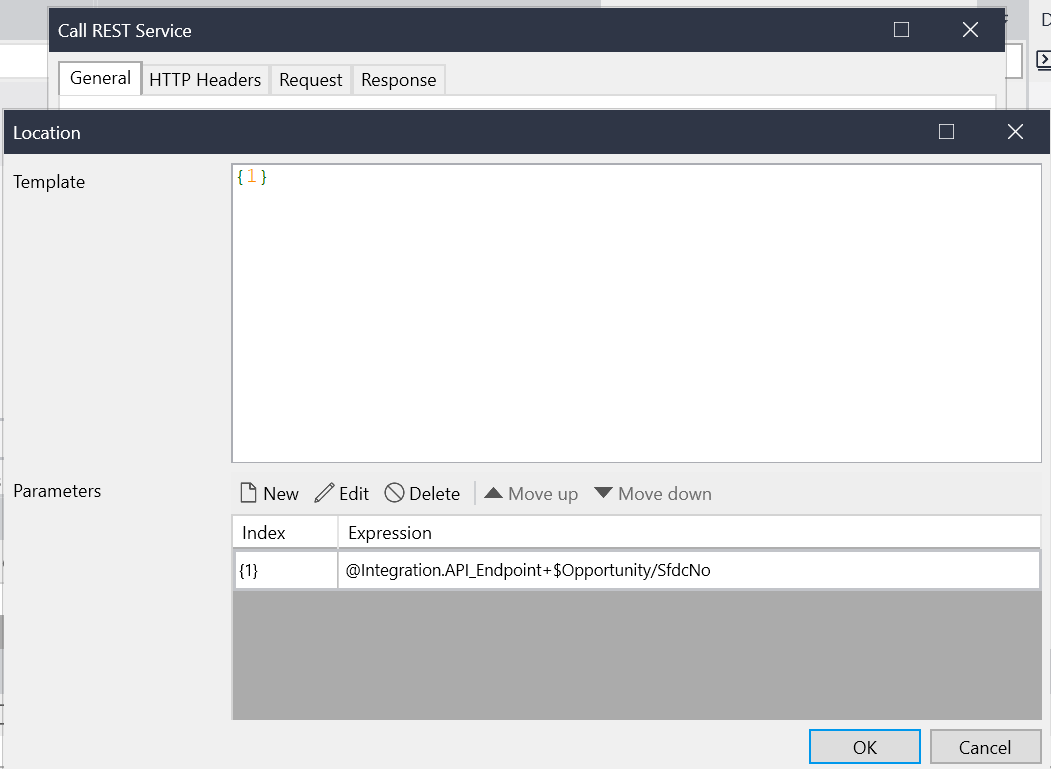
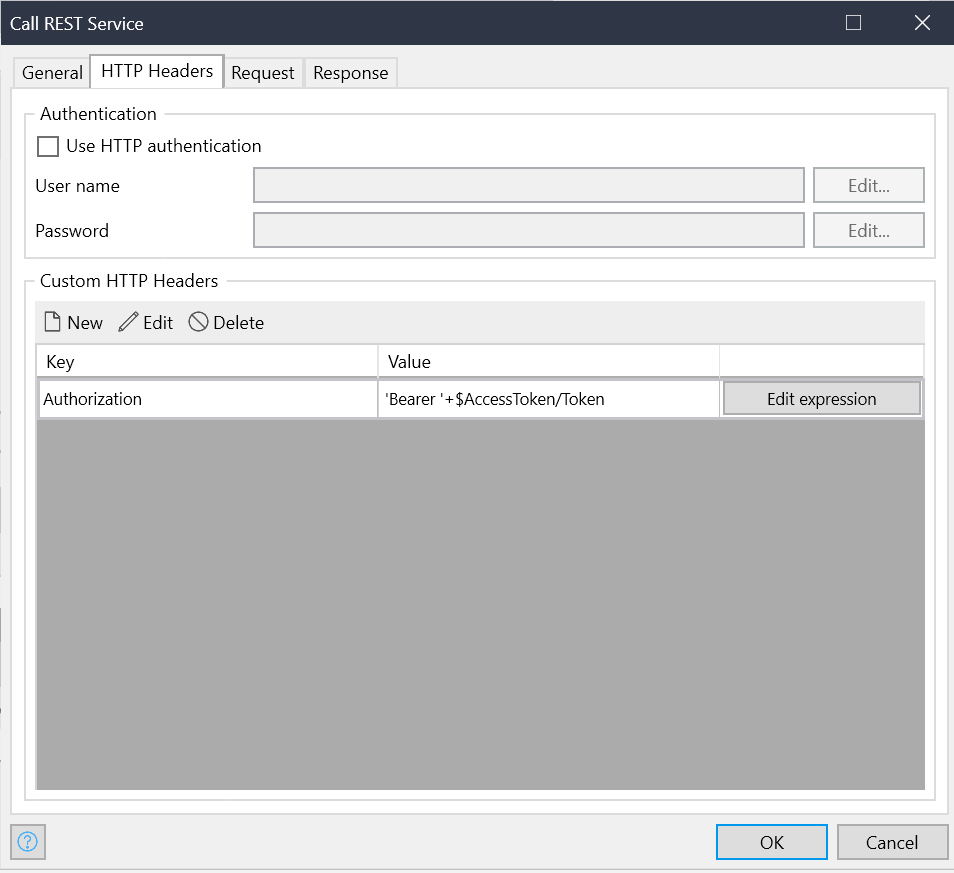


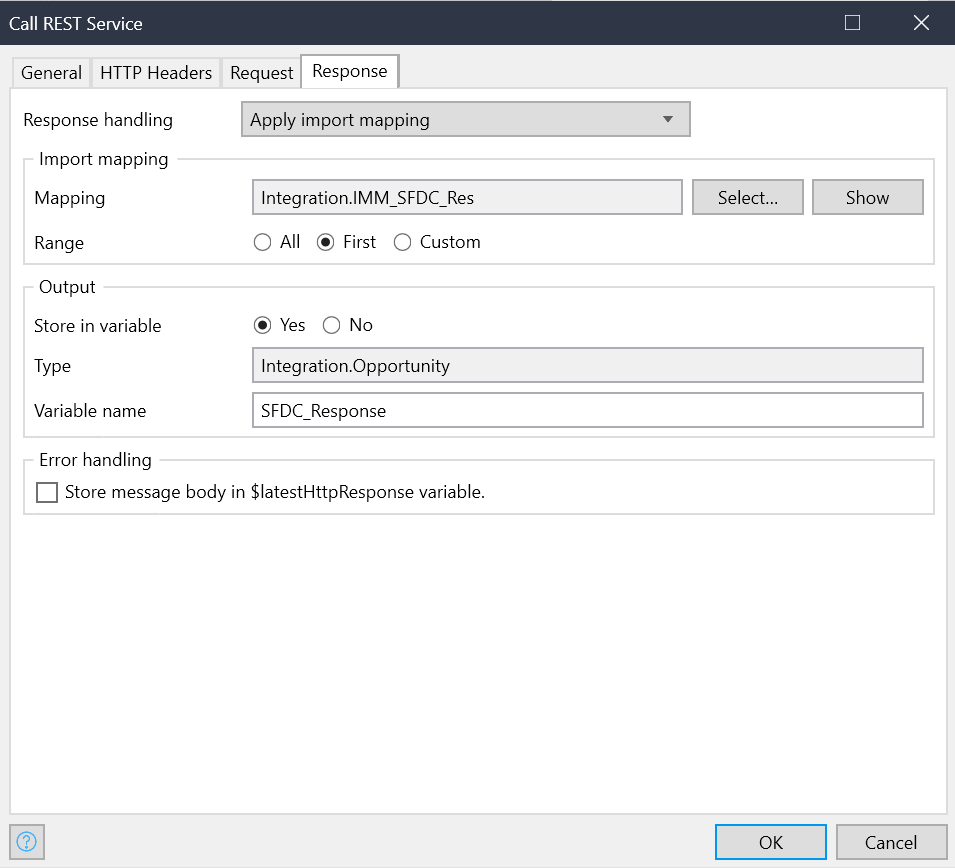
1. After setting up the templates, you can retrieve these in your microflow which is supposed to be sending an email.

You’ll now be able to send emails through the Mendix app.

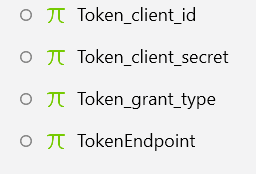
**API Details :**

* In upsell application we have 2 API services
  + - Upsell consuming SFDC service
    - Upsell exposing service to other app.
  + Upsell app is consuming one service on Account manager screen to get some values associated with that opportunity.
  + We are making a service call in AM approval microflow which is “ACT\_AM\_Save\_Opportunity”
  + You can find below screenshot to see how we are calling the service.

1. In general tab we are appending SFDC number with the API endpoint  
   
2. In header we are passing the bearer token we received from APIGEE,  
   
3. Request will be empty and response we are storing in upsell application using import mapping. You can find this in below screenshot.



1. We are using some constants to define the REST call values, you can find below screenshot for this.



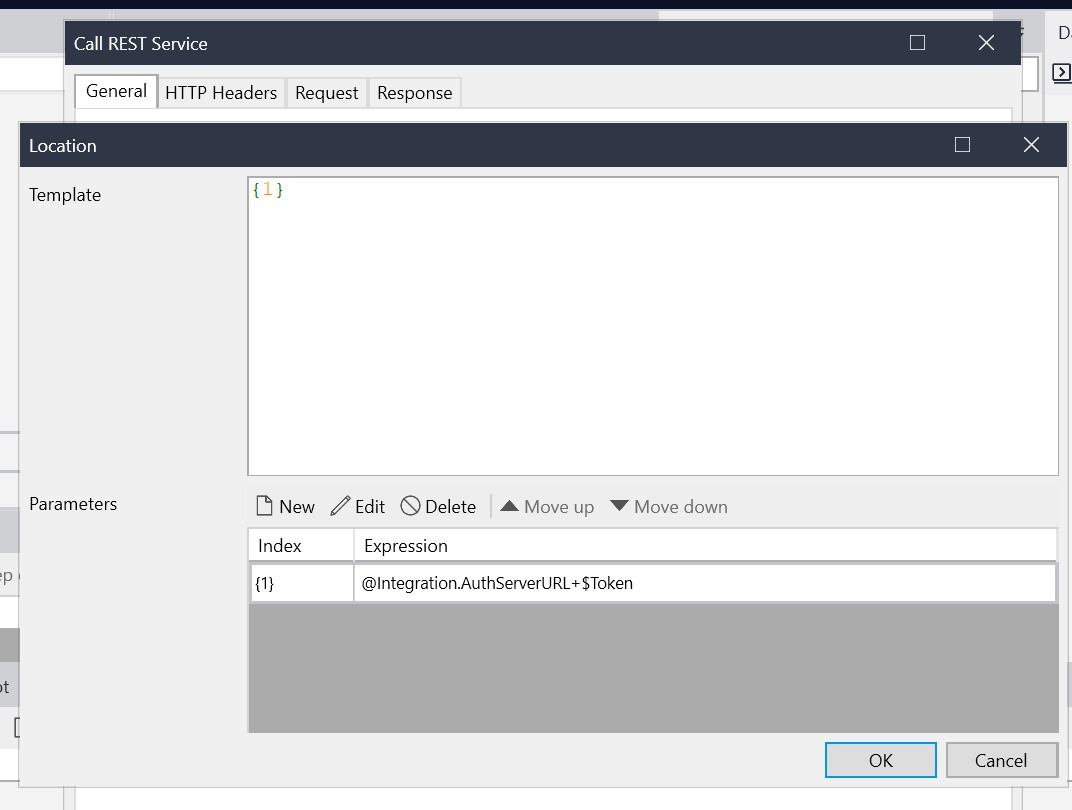
Similarly we are also exposing one service, the published rest service name is “MxUpsell”

As other applications would be calling these service we have protected this service with HW resource server using Oauth 2.0

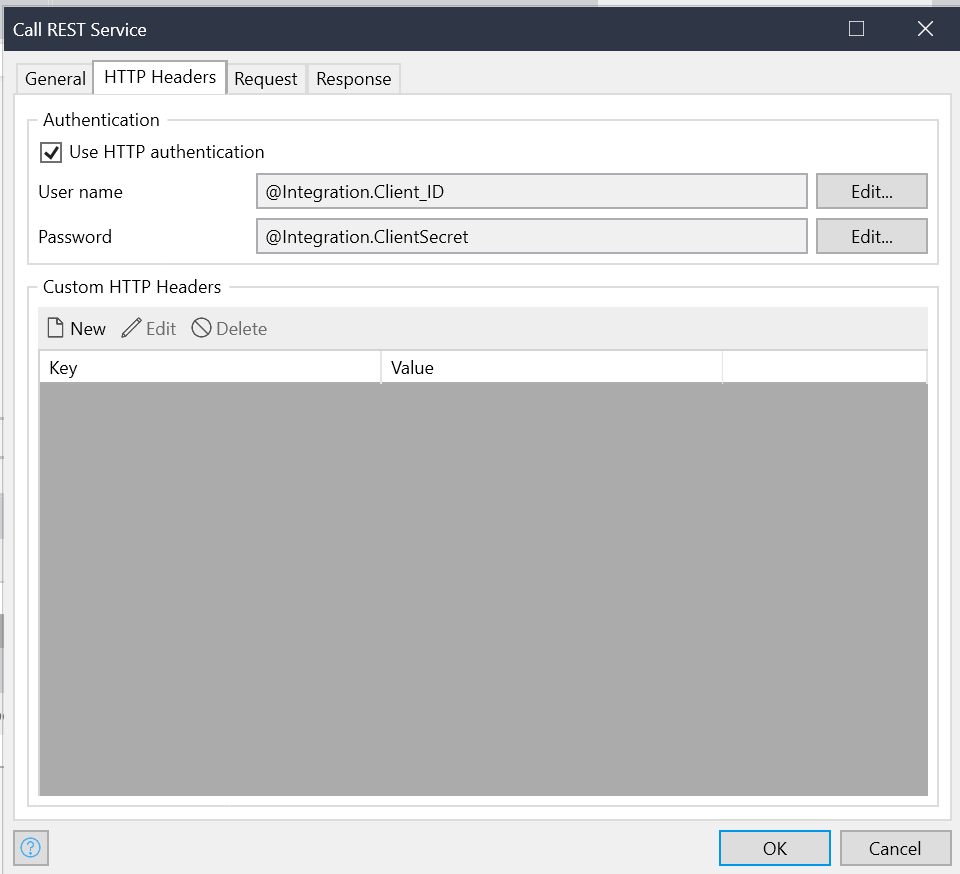
We are receiving one token in the request which we need to validate first before processing any request.

To validate the token we are using one service from resource server which is called under one microflow called “VAL\_SFDC\_Token”.

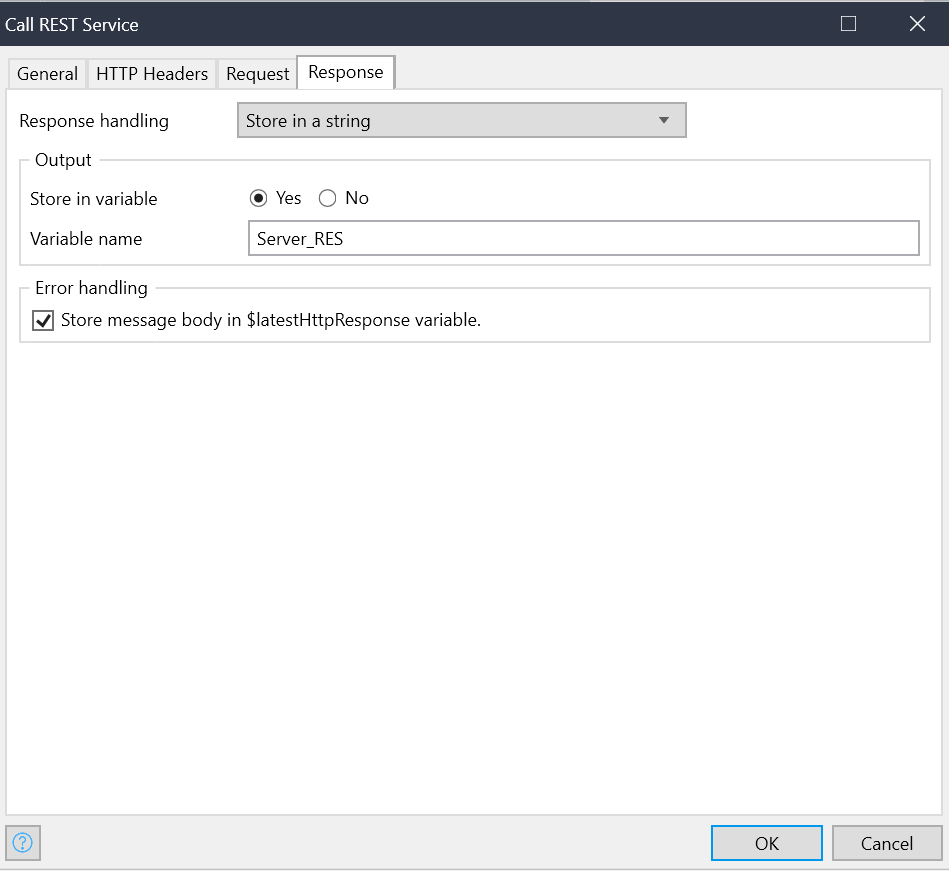
1. In general tab we are appending the token with endpoint URL, like mentioned below



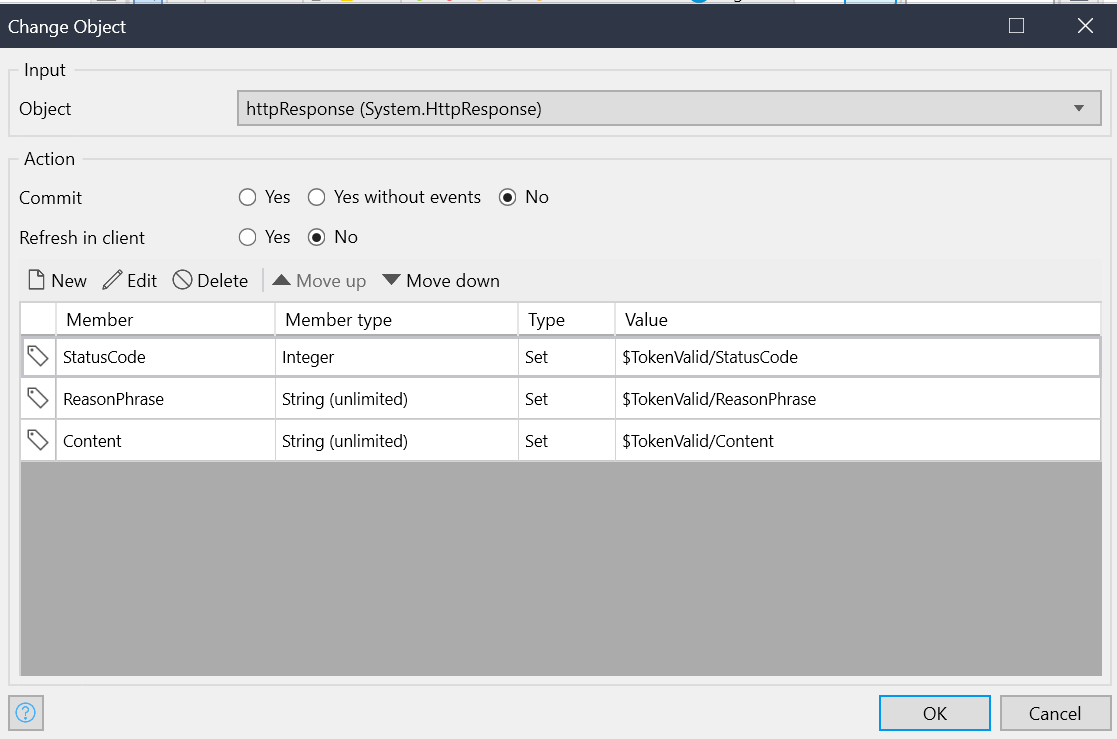
1. HTTP headers we are passing the username and password.



1. Request will be empty as we are just validating the token.
2. Response we are storing in a String to see if this was a valid token.



If the token is valid, we are processing the request and if the token is not valid, we are updating the http response and sending back to the requester.



1. We are also using some of the constants to expose the service that you can find in below screenshot.

