User Manual of Management Framework Architecture for 4.81 GA

Table of Contents

1. Document Versioning

Version	Date	Purpose	Author
MFA-4.81-v1	14-Aug-2020	First Draft	Saurabh Goyal
MFA-4.81-v2	08-Oct-2020	Added "Migration of Users and Tenant" Page	Saurabh Goyal

2. Quick Startup Guide for Management Framework Architecture

Perform the following steps to integrate and configure Management Framework Architecture with Ameyo Appserver

1. Integration of Management Framework Architecture
2. Integrate Management Server UI Application
3. Configuration on Application Server
4. Create Application Server from Management Framework Architecture
5. Create Existing Tenants, if already configuring on running Application Server
6. Create new Tenants
7. Create and Assign Users
8. Frequently Asked Questions

3. Working of MFA

3.1 Working of Management Framework Architecture (MFA)

When there are multiple Application Servers installed in a single organization, then all servers will get a different Domain Name, and hence the data of all servers will be stored as a different structure. There are chances when users may get confuse when to logon at which Application Server and how to identify which Application Server will serve a particular purpose. In such a case, the Management Framework Architecture helps the organization map all Domain Names with it and provides a single Domain Name for all users to logon to the different setups of Application Server.

When a user logon to the Management Framework Architecture at the location (Domain name or IP Address) given to them, the user credentials (username and password) are verified by Management Framework Architecture. After verification, it redirects them to their respective Application Servers in which they are assigned. The respective Application Server manages all data and work.

In a nutshell, we can say, a Management Framework Architecture allows the management of users of different Application Servers from a single platform.

3.1.1 Management Server Architecture Flow

The Architecture of the Management Framework with multiple Application Servers is given herein below.

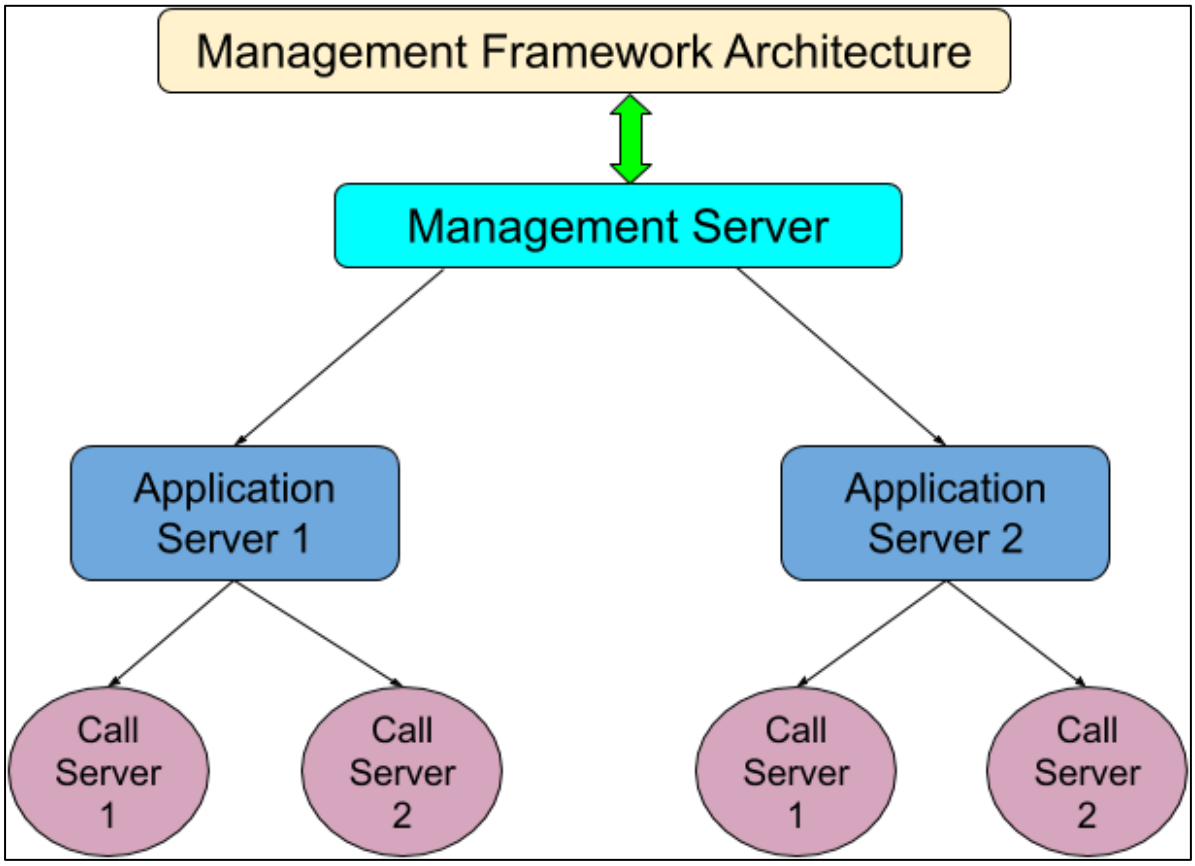


Figure: Management Framework Architecture

The architecture of the Management Framework comprises of the Management Server, Application Servers, Databases for both Management and Application Servers, and Call Servers. The architecture also includes the API invocation, direction of API's, and direction of data being exchanged with APIs.

3.2 Features and Capabilities of Management Framework Architecture

3.2.1 Features

- Migration of already existing users and Tenants from Application server to Management Server is possible.
- Single setup to manage multiple Application servers
- Google+ SSO
- Single Login URL for all User Roles of both Management Framework Architecture and Application Servers. After login, the user will be redirected automatically to the tenant where it is assigned. However, user will not observe any change. No need to maintain the different Login URLs
- Same and consistent User Experience for all Application Users
- User management for different Application servers from a single place
- Allow users to work from any location of the organizations
- Does not impact the working of users, if they try to work with different Application server other, then they are assigned.
- STD code management
- Tenant Management
- LDAP Authentication
- Duplicate users cannot be created, that is, if one user is created on Application Server one, then the same user cannot be created with the same username or user ID on Application Server two.

3.2.2 Capabilities

- Multiple Application servers can be added
- Users and Tenant can be managed from a single place
- Users for the specific server can be created.
- Single login URL is present for all the Application Servers

- Separate users for Management Framework Architecture can be created
- Users first authenticate from the Management Framework Architecture and then allowed to login to any application.

3.2.3 What if Management Framework Architecture stops working?

Following services will be impacted, if the Management Framework Architecture does not work correctly:

- New Users cannot be logged in to the system.
- User management at the application layer, that is, the administrator cannot create, modify, or delete any user.

Following features do not impact if the Management Framework Architecture is down.

- Already logged-in users, whether on the application or management, do not get impacted.
- The calling from any application would not get impacted.

Download PDF

3.3 Limitations of Management Framework Architecture

- If the user is not present at Management Framework Architecture, then the user is not able to login to the Application Server.
- SAML login authentication is not supported
- CRM login authentication is not supported
- Internationalization is not supported
- VAPT for Management Framework Architecture is not performed
- Call Manager needs to be managed by Application administrator separately.
- Unable to create the users with small alphabets or special characters in their names or IDs.

4. Integration of MFA

Perform the following steps to integrate the Management Framework Architecture with Application Server.

1. Execute the following commands to create the databases for Management Server and Management UI applications.

   A. Execute the following command to logon to PostgreSQL database System.

      psql -U postgres

   B. Run the following query to create a database for the Management Server and its Interface.

      create database <Name_of_Management_Server_DB>;
      create database <Name_of_Management_UI_DB>;

```
ameyodb=# create database management_server;
CREATE DATABASE
ameyodb=# create database management_ui;
CREATE DATABASE
ameyodb=#
```

Figure: Create Databases

   C. Run the following queries to create a "dacx" user and assign the privileges of the database to it.

      CREATE USER dacx SUPERUSER CREATEDB CREATEROLE INHERIT LOGIN;
      GRANT postgres to dacx WITH ADMIN OPTION;

   D. Run the following command to exit from the database console.

      \q

Click the following links to know more about them.

2. Perform the installation and integration of Management Server Application
3. Perform the installation and integration of Management Server UI Application
4. Perform the configuration in Ameyo Appserver Application

4.1 Installation and Integration of Management Server Application

Perform the following steps to install and integrate the Management Server with Application Server.

1. Execute the following command to install Management Server.

   rpm -ivh <Management_Server>.rpm

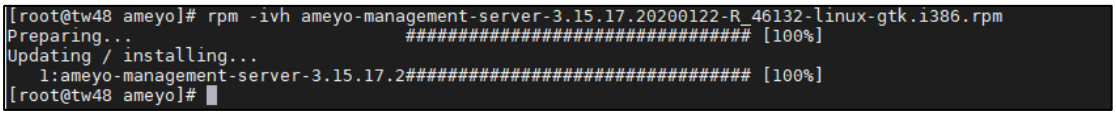


Figure: Install Management Server

You can contact Ameyo Support Team to get the Setup RPM files for Management Server.

2. Execute the following command to edit "Hibernate.properties" file of Management Server to provide the database connectivity into it.

   vim /ameyo_mnt/dacx/var/ameyo/dacxdata/ameyo.management.server.product/conf/hibernate.properties

   Provide the details of the database for Management Server

   hibernate.connection.url jdbc:postgresql://<IP_of_Database>/<Database_Name_of_Management_server>

   hibernate.connection.username <Database_User_Name>

   hibernate.connection.password <Password_of_Database_if_any>

```
## PostgreSQL
hibernate.dialect org.hibernate.dialect.PostgreSQLDialect
hibernate.connection.driver_class org.postgresql.Driver
hibernate.connection.url jdbc:postgresql://10.10.10.28/management_server
hibernate.connection.username postgres
hibernate.connection.password
hibernate.connection.socketTimeout 1200
hibernate.connection.loginTimeout 600
# For DB manager
hibernate.connection.profile_name postgres_81
```

Figure: "Hibernate.properties" file of Management Server

3. Execute the following command to edit "Tomcat.conf" file of Management Server to provide the SSL configuration for it.

   This file needs to e modified only if, SSL is enabled with Ameyo.

   ```
   vim /ameyo_mnt/dacx/var/ameyo/dacxdata/ameyo.management.server.product/conf/tomcat.conf
   ```

   Provide the following details of SSL in this file.

   ```
   connector.https.secure=true
   connector.https.httpsPort=<Management_Server_Port_Number>
   connector.https.https=https
   connector.https.SSLEnabled=true
   connector.https.clientAuth=false
   connector.https.keystoreFile=<Path_of_SSL_Keys><.jks_SSL_File>
   connector.https.keystorePass=<SSL_Certificates_Password>
   connector.https.keyAlias=<Alias_of_SSL_Certificates>
   connector.https.SSLHonorCipherOrder=on
   drishti.start.https.connector=true
   ```

   The changes in the file are highlighted with blue-colored underline.

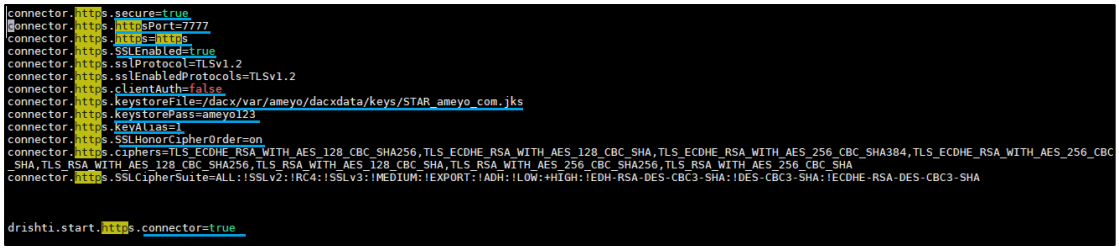


Figure: Tomcat.conf File of Management Server

4. Execute the following command to edit and provide the details of Management Server in "AmeyoManagementServer.ini" file.

   ```
   vim /ameyo_mnt/dacx/var/ameyo/dacxdata/ameyo.management.server.product/conf/AmeyoManagementServer.ini
   ```

Perform the following configurational changes in this file.

A. Change the following configurations in this file.

```
managementServerProtocol=<Protocol_of_Server>
SSOServerIp=<Domain_Name_of_server>
managementServerConfigured=true
SSOServerProtocol=<Protocol_of_Server>
defaultRedirectUriIp=<Domain_Name_of_server>
SSOServerIp=<Domain_Name_of_server>
```

B. Now, copy and delete the following log file entry.

```
Xloggc:/dacx/var/ameyo/dacxdata/ameyo.management.server.product/logs/gc-\$(date +%Y_%m_%d-%H_%M).log
```

The above log file entry is mandatory and needed. So before deleting this entry, keep it with you at safe place, as it has to be used later.

C. Cross-check for the following entry of "Connector.conf" file. If the entry is not available, then add this entry.

```
connectorConfPath=/dacx/var/ameyo/dacxdata/ameyo.management.server.product/conf/connector.conf
```

The changes in the file are highlighted with blue-colored underline.

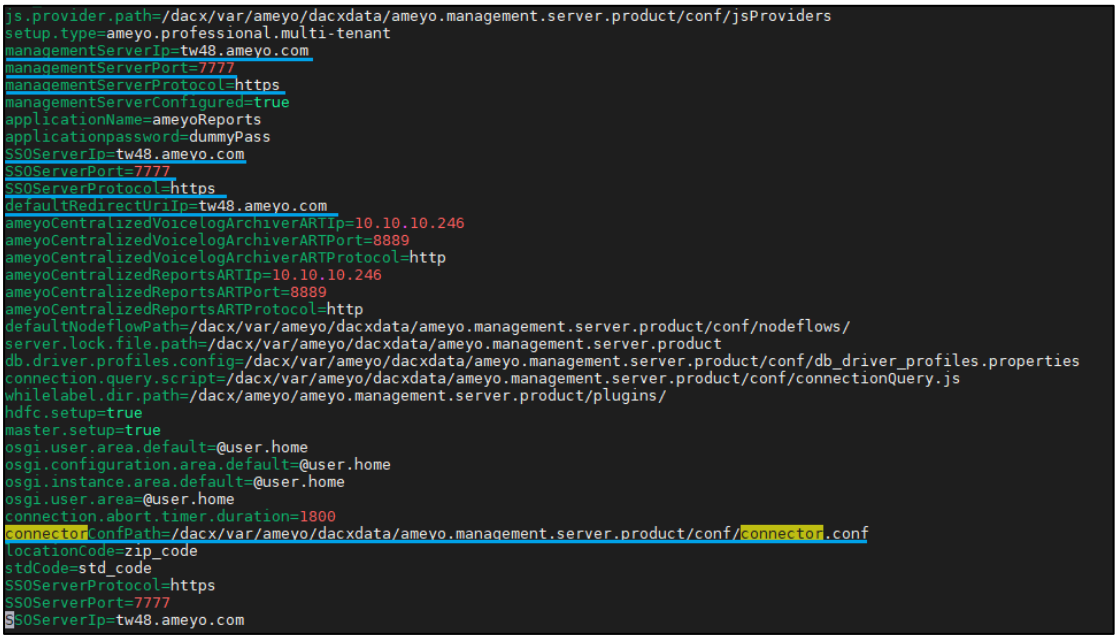


Figure: AmeyoMangementServer.ini File of Management Server

5. Execute the following command to edit "DACXAmeyoProServerDefault.ini" file to provide the SSO entries into it.

   ```
   vim
   /dacx/var/ameyo/dacxdata/com.drishti.dacx.server.product/conf/DAC
   XAmeyoProServerDefault.ini
   ```

   Add the following SSO entries in this file.

   ```
   SSOServerProtocol=<Server_Protocol>

   SSOServerPort=<Management_Server_Port_Number>

   SSOServerIp=<Domain_Name_of_Server>
   ```

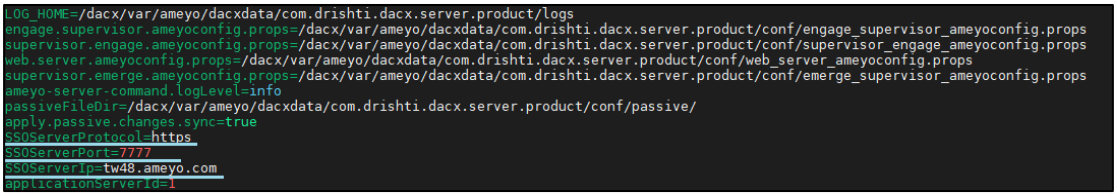


Figure: SSO entries in DACXAmeyoProServerDefault.ini File

6. Execute the following command to edit "Connector.conf" file to provide third-party database connectivity.

```
vim
/ameyo_mnt/dacx/var/ameyo/dacxdata/ameyo.management.server.product/conf/connector.conf
```

Add the following configurational change lines in this file.

```
connector.db.password=<Password_of_connector_Database>
connector.db.name=<Name_of_Connector_Database>
wc.crm.db.password=<database_Password_for_wc-crm>
connector.db.username=<UserName_of_connector_database>
wc.crm.db.name=<database_name_for_wc-crm>
```

The following table defines the sample codes used in the above lines.

Sample Value	**Definition**
<password_of_connector_Database>	It is the password of the connector's database. If there is no password, then provide any string form of password here.
<Name_of_Connector_Database>	It is the name of the connector's database.
<Password_for_wc-crm>	It is the password of "wc.crm" database. The wc_crm database is the database of the CRM used with Ameyo for integration of third-party CRM. If there is no password for it, then provide any string format password here.
<database_name_for_wc-crm>	It is the name of "wc_crm" database.
<UserName_of_connector_database>	It is the username of the connector database.

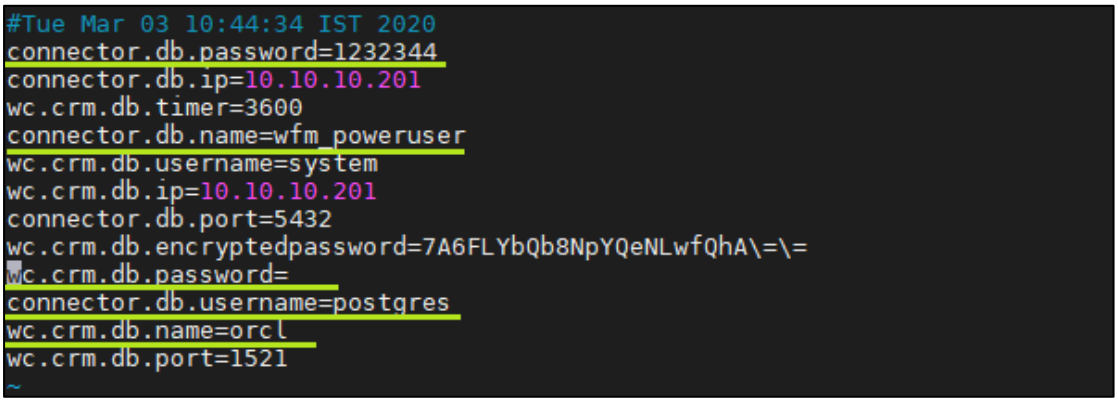


Figure: Connector.conf File of Management Server Application

After starting Management Server application, the password is deleted and an encrypted password is generated that is written in "wc.crm.db.encryptedpassword". Do not provide any string here manually.

7. Execute the following command to create the log files that will be used to save the errors for pull and push requests from Management Server. Push request is if something is inserted to the database and pull request is for the output of anything.

```
touch /ameyo_mnt/WCpull.log
touch /ameyo_mnt/WCpush.log
```

8. Execute the following command to provide the permission to the above created files

```
chown dacx.dacx /ameyo_mnt/WCpull.log
chown dacx.dacx /ameyo_mnt/WCpush.log
```

9. Execute the following commands to start the Management Server and Management Server UI applications

```
ameyoctl service ameyomanagementserver start
```

10. Once the services of Management Server is started, now, execute the following command to provide "Xlog" entry in "AmeyoManagementServer.ini" file again.

```
vim
/ameyo_mnt/dacx/var/ameyo/dacxdata/ameyo.management.server.product/conf/AmeyoManagementServer.ini
```

Now, add the following Xlog entry again which you have deleted while configuring "AmeyoManagementServer.ini" file.

```
.Xloggc:/dacx/var/ameyo/dacxdata/ameyo.management.server.product/
logs/gc-\$(date +%Y_%m_%d-%H_%M).log
```

This is the same entry of "Xlog", that you have deleted earlier.

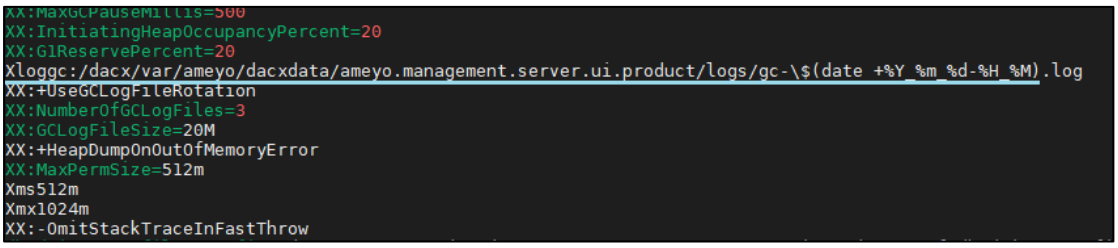


Figure: AmeyoMangementServer.ini File of Management Server

11. After adding the entry, now it is mandatory to restart Management Server and Management UI services again. Execute the following query to restart the services.

```
ameyoctl service ameyomanagementserver stop
ameyoctl service ameyomanagementserver start
```

4.2 Installation and Integration of Management Server User Interface

Perform the following steps to install and integrate the Management Server User Interface with Application server.

1. Execute the following command to install Management Server Interface. This application enables the Interface for Management Framework Architecture (MFA).

   `rpm -ivh <Management_UI_Server>.rpm`

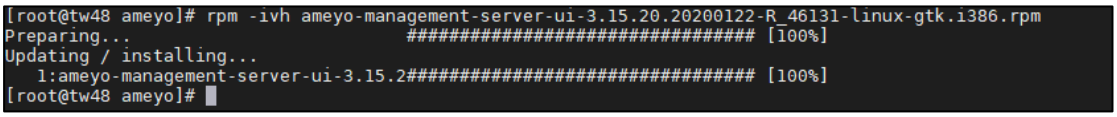


Figure: Install Management Server User Interface Application

You can contact Ameyo Support Team to get the above application RPM files.

2. Execute the following command to edit "Hibernate.properties" file of Management Server User Interface to provide the database connectivity into it.

   `vim /ameyo_mnt/dacx/var/ameyo/dacxdata/ameyo.management.server.ui.product/conf/hibernate.properties`

   Provide the details of the database for Management Interface

   `hibernate.connection.url jdbc:postgresql://<IP_of_Database>/<Database_Name_of_Management_Server_UI>`

   `hibernate.connection.username <Database_User_Name>`

   `hibernate.connection.password <Password_of_Database_if_any>`

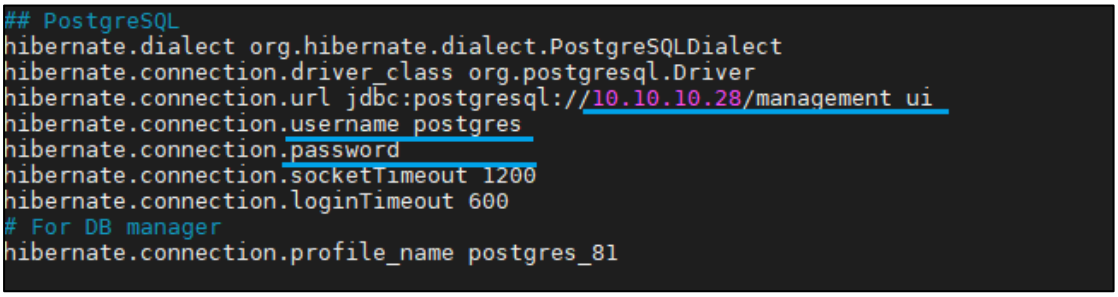


Figure: "Hibernate.properties" file of Management Interface

3. Execute the following command to edit "Tomcat.conf" file of Management Server UI to provide the SSL configuration for it.

   This file needs to be modified only if, SSL is enabled with Ameyo.

   ```
   vim
   /ameyo_mnt/dacx/var/ameyo/dacxdata/ameyo.management.server.ui.pro
   duct/conf/tomcat.conf
   ```

   Provide the following details of SSL in this file.

   ```
   connector.https.secure=true
   connector.https.httpsPort=<Management_UI_Port_Number>
   connector.https.https=https
   connector.https.SSLEnabled=true
   connector.https.clientAuth=false
   connector.https.keystoreFile=<Path_of_SSL_Keys><.jks_SSL_File>
   connector.https.keystorePass=<SSL_Certificates_Password>
   connector.https.keyAlias=<Alias_of_SSL_Certificates>
   connector.https.SSLHonorCipherOrder=on
   drishti.start.https.connector=true
   ```

   The changes in the file are highlighted with blue-colored underline.

Figure: Tomcat.conf File of Management Server UI

4. Execute the following command to edit and provide the details of Management Server UI in "AmeyoManagementServerUI.ini" file.

   ```
   vim
   /dacx/var/ameyo/dacxdata/ameyo.management.server.ui.product/conf/
   AmeyoManagementServerUI.ini
   ```

Copy and delete the following log file entry.

```
Xloggc:/dacx/var/ameyo/dacxdata/ameyo.management.server.ui.product/logs/gc-\$(date +%Y_%m_%d-%H_%M).log
```

The above log file entry is mandatory and needed. So before deleting this entry, keep it with you at safe place, as it has to be used later.

Figure: AmeyoMangementServerUI.ini File of Management Server

5. Execute the following command to provide the authentication to Management Server UI application in "ameyoconfig.props" file.

   ```
   vim /ameyo_mnt/dacx/var/ameyo/dacxdata/ameyo.management.server.ui.product/conf/ameyoconfig.props
   ```

   Provide the following Management Server information here.

   ```
   ameyoServerIP=<Domain_Name_of_Server>
   ameyoServerPort=<Management_Server_port_number>
   ameyoServerProtocol=<Ameyo_Server_Protocol>
   ameyoDatabaseIP=<Database_IP_or_Domain_Name>
   ameyoDatabaseName=<Management_Server_Database_Name>
   ameyoDatabaseUserName=<Database_User_Name>
   ameyoDatabasePassword=<Database_Password_if_any>
   ameyoDatabasePort=<Database_Port_Number>
   myDatabaseIP=<Domain_Name_of_Server>
   myDatabaseName=<Management_Server_Database_Name>
   myDatabaseUserName=<Database_User_Name>
   myDatabasePassword=<Database_Password_if_any>
   ```

Figure: Management Server UI AmeyoConfig.props File

6. Execute the following commands to start the Management Server and Management Server Interface

   ```
   ameyoctl service ameyomanagementserverui start
   ```

7. Once the services of Management Server UI is started, now, execute the following command to provide "Xlog" entry in "AmeyoManagementServerUI.ini" file again.

   ```
   vim
   /ameyo_mnt/dacx/var/ameyo/dacxdata/ameyo.management.server.ui.product/conf/AmeyoManagementServerUI.ini
   ```

   Now, add the following Xlog entry again which you have deleted while configuring "AmeyoManagementServer.ini" file.

   ```
   .Xloggc:/dacx/var/ameyo/dacxdata/ameyo.management.server.ui.product/logs/gc-\$(date +%Y_%m_%d-%H_%M).log
   ```

   This is the same entry of "Xlog", that you have deleted earlier.

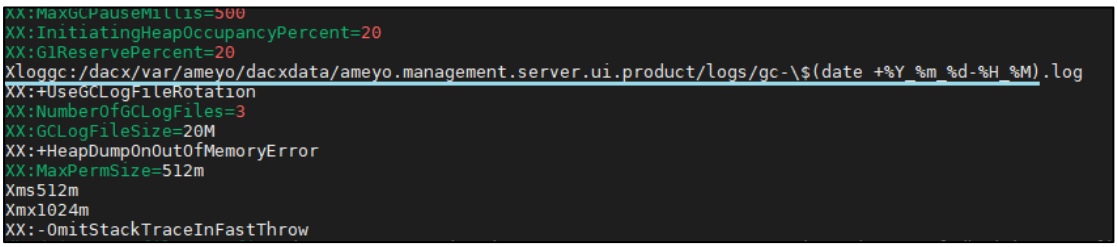


Figure: AmeyoMangementServerUI.ini File of Management Server

8. After adding the entry, now it is mandatory to restart Management Server and Management Server UI services again. Execute the following query to restart the services.

```
ameyoctl service ameyomanagementserverui stop
ameyoctl service ameyomanagementserverui start
```

4.3 Configuration in Application Server

Perform the following steps to integrate the Management Framework Architecture with Application Server.

1. Execute the following command to edit "DACXAmeyoProServerDefault.ini" file to provide the SSO entries into it.

   ```
   vim
   /dacx/var/ameyo/dacxdata/com.drishti.dacx.server.product/conf/DAC
   XAmeyoProServerDefault.ini
   ```

   Add the following SSO entries in this file:

   ```
   SSOServerProtocol=<Server_Protocol>

   SSOServerPort=7777

   SSOServerIp=<Domain_Name_of_Server>
   ```

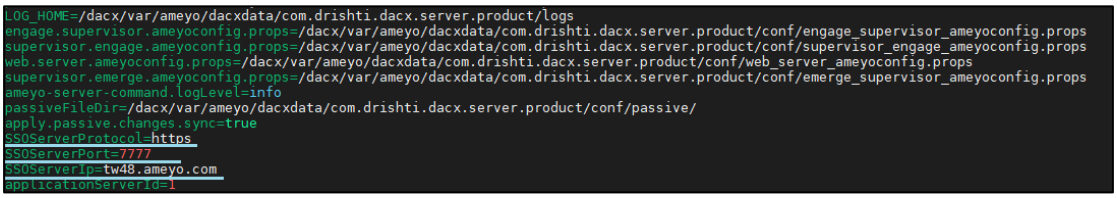


Figure: SSO entries in DACXAmeyoProServerDefault.ini File

2. Execute the following commands to allow Management Server to manage Application Servers.

   A. Execute the following command to logon to the Application Server's database.

      ```
      psql -U postgres <Application_Server_Database_Name>
      ```

   B. Run the following query to provide authentication of Application Server from Management Server.

      ```
      INSERT INTO system_configuration_parameter
      (name,type,value,default_value,date_added,date_modified)
      VALUES
      ('authentication.scheme','String','auth.type.ms','auth.type.
      passwd',now(),now());
      ```

C. Run the following query to provide the Management Server mode to Application Server.

```
INSERT INTO system_configuration_parameter (name,type,value,default_value,date_added,date_modified) VALUES ('systemMode','String','ms','ameyo',now(),now());
```

D. Run the following query to provide "root" user to logon to Application without authentication from Management Server.

```
INSERT INTO system_configuration_parameter (name,type,value,default_value,date_added,date_modified) VALUES ('override.authentication.scheme','String','{"authentication.scheme":"auth.type.passwd","applicable.user.roles":"root"}','{"authentication.scheme":"auth.type.passwd","applicable.user.roles":"root,PowerUser"}',now(),now());
```

The root user in Application Server is the main user who has the privileges to manage the complete Application Servers and Call Managers.

E. Execute the following commands to make Ameyo appserver from single tenant server to multi-tenant server.

- `select * from system_configuration_parameter where value ilike '%set%';`
- `UPDATE system_configuration_parameter SET value ='ameyo.professional.multi-tenant' where name='setup.type';`
- `DELETE FROM data_version;`
- `DELETE FROM schema_version;`

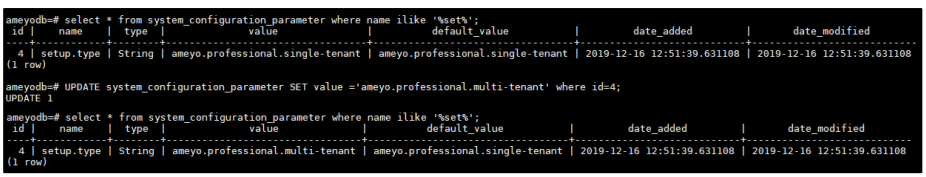


Figure: Updating Single tenant Server to Multi-Tenant Server

3. Execute the following commands to restart Ameyo Appserver application

```
ameyoctl service appserver stop
ameyoctl service appserver start
```

4.4 APIs to enable Business MetaData

Every organization has unique data that is assigned to every of its user, such as Employee code, Department Code, Email Id, and so on. It is required to invoke the following APIs to enable the Business MetaData tab.

Perform the following steps.

4.4.1 addUserBusinessMetadataColumnMapping API

Use this API to create database columns of Business MetaData.

4.4.1.1 Method

POST

4.4.2 Headers

The following are the header that has to be used in this API.

1. **sessionId:** Session Id of Administrator
2. Content-Type: Application/JSON

Ameyo will provide the header attributes. Contact Services team of Ameyo for the values of Header Attributes.

4.4.2.1 URL

```
<protocol>://<IP_Domain_Name>:<port>/dacx/jsonCommand?command=remote.processor.userBusinessDataConfigurationService.addUserBusinessMetadataColumnMapping&data=
```

4.4.2.2 Sample Request Data

```
{
  sessionId:<Session Id of Administrator>,
  dataTableId:1,
  name:FatherName,
  type:3,
  primaryKey:false,
  nullable:true,
  isUnique:false
}
```

4.4.2.3 Sample Response Data

```
{
  "id":1,
  "datatableId":1,
  "name":"FatherName",
  "type":3,
  "primaryKey":false,
  "nullable":true,
  "isUnique":false
}
```

4.4.2.4 Additional Cases

The following are the additional cases that may lead to the Error as a response.

1. While making the entry corresponding to the <data_table_id> does not exists in <user_business_metadata> then the response will be an error.
2. If there is an entry corresponding to the same <data_table_id> and <column_name> already exists, then the response will be an error.

4.4.3 addUserBusinessMetadataCcMapping API

Use this API to add Business MetaData Contact Center Mapping.

4.4.3.1 Method

POST

4.4.4 Headers

The following are the header that has to be used in this API.

1. **sessionId:** Session Id of Administrator
2. Content-Type: Application/JSON

Ameyo will provide the header attributes. Contact Services team of Ameyo for the values of Header Attributes.

4.4.4.1 URL

<protocol>://<IP_Domain_Name>:<port>/dacx/jsonCommand?command=remote.processor.userBusinessDataConfigurationService.addUserBusinessMetadataCcMapping&data=

4.4.4.2 Sample Request Data

```
1. {
2.   sessionId:<Session Id of Administrator>,
3.   dataTableId:1,
4.   ccId:<Contact_Center_Id>
5. }
```

4.4.4.3 Sample Response Data

```
1. {
2.  "mappingId":1,
3.  "dataTableId":1,
4.  "userBusinessMetadataTableName":"table1_user_business_metadata_1",
5.  "ccId":1
6. }
```

5. Migration of Tenants and Users to Management

It is possible to deploy the Management Server over the already running Application server(s). In such cases, it is also required to migrate all the existing tenants and users present at the application server to the management server. The tenants and users present at the application server contain information which thereby needs to be migrated. This migration process can be achieved with the "Migration Script". In this migration step, all the tenants (contact centers at the application server) and users will migrate to the management layer.

The following process is given for the migration of the tenants, and users will migrate all the tenants first, and then the users of all the migrated tenants will be migrated at last only.

It is mandatory to stop the management server and application server before running the migration script. It is also not suggested to update any tenant or user information while the migration script is in process, as it would create inconsistency to the data which has to be migrated.

Perform the following steps to migrate the tenants and users to the management server.

1. Run the following commands to stop the management servers.

   ```
   ameyoctl service ameyomanagementserver stop
   ameyoctl service ameyomanagementserverui stop
   ```

   It is also suggested to stop the application servers. If the application servers are in use, then make sure no tenant and user information will update.

2. Download the Migration script files and extract it from here. Copy this file under "dacx" location (location could be anything under dacx).

3. There is a special case before proceeding further that, if you are migrating more than 1 application server to the management layer, then all the application servers have a contact center named "DefaultCC". You have to change this name from all the application servers manually, as the management layer represents uniqueness and does not allow to create 2 tenants with the same name. Thus make sure that before running the below script, you have to change the "DefaultCC" contact center(CC) name.

4. Run the following command to edit "config.py" file of the migration script to provide the access of the application server into it.

```
vim <path_of_migration_file>/config.py
```

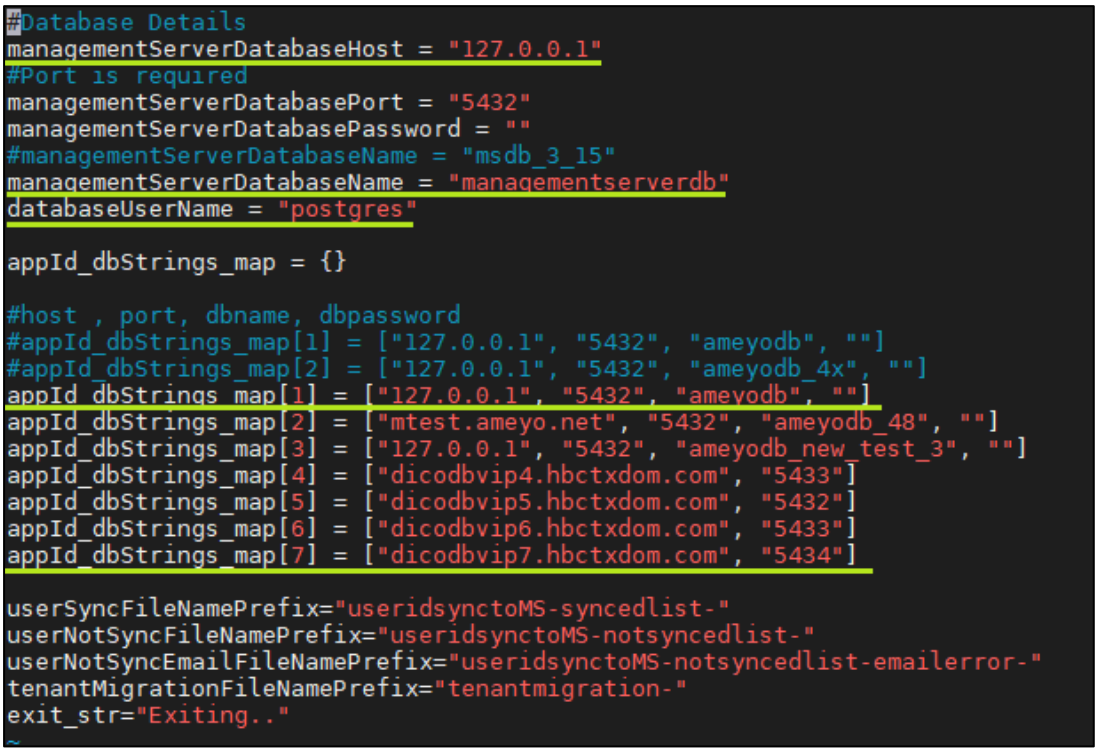


Figure: Edit Config.py File

Provide the following details in this file. The changes are marked with the green line in the above screenshot.

```
managementServerDatabaseHost = "<Management_DataBase_Server_IP>"
managementServerDatabasePort =
"<Management_Server_DataBase_PORT_Number>"
managementServerDatabasePassword =
"<Password_of_Management_Database>"
managementServerDatabaseName =
"<Management_Server_Database_Name>"
databaseUserName = "<Database_User_Name>"
appId_dbStrings_map[1] =
["<IP/Domain_of_First_Application_Server>",
"<First_Application_Server_DataBase_Port>",
"<First_Application_Server_DataBase_Name>",
```

```
"First_Application_Server_DataBase_Password"]
appId_dbStrings_map[2] =
["<IP/Domain_of_Second_Application_Server>",
"<Second_Application_Server_DataBase_Port>",
"<Second_Application_Server_DataBase_Name>",
"Second_Application_Server_DataBase_Password"]
```

Here, "appId_dbStrings_map[1/2]" represents the application server information that has to be bought under the Management Layer. You can add more than 2 application server by using the same nomenclature.

Save and exit from the console-based file editor.

5. Run the following command to start the migration script.

   python MigrationToMFA.py

6. After running the above command, the following information will be asked to insert:

   A. First, it will ask to provide the tenant Login URL for every migrated contact center. You can provide it for every migrated contact center or to any specific one. If now, then press enter to continue without it. This tenant login URL represents the

   B. Once, tenant migration step is done, it will ask to proceed to migrating the users of those tenants. Provide "Yes" to migrate users or "No" to disallow user migration.

   C. After the successful migration, a log is also generated, in which all the information about the migration of tenants and users is present.

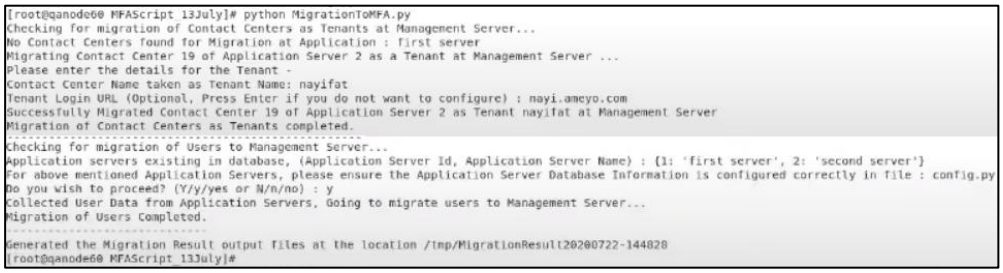


Figure: Migration of Users and Tenants

7. After migration, start the application server and management server and login to the user with Management layer architecture.

6. Configure Google SSO with MFA (Licensable)

Single Sign-On helps businesses to deliver a secure and compliant working environment. Ameyo has the support of the Single Sign-On for its Application Server with Microsoft, Google, and LDAP-based services. Ameyo supports Single Sign-On based on OAuth2 for Management Framework Architecture with Google.

Perform the following steps.

1. Login to the Google Developer Account with your Google ID. It shows the following page.

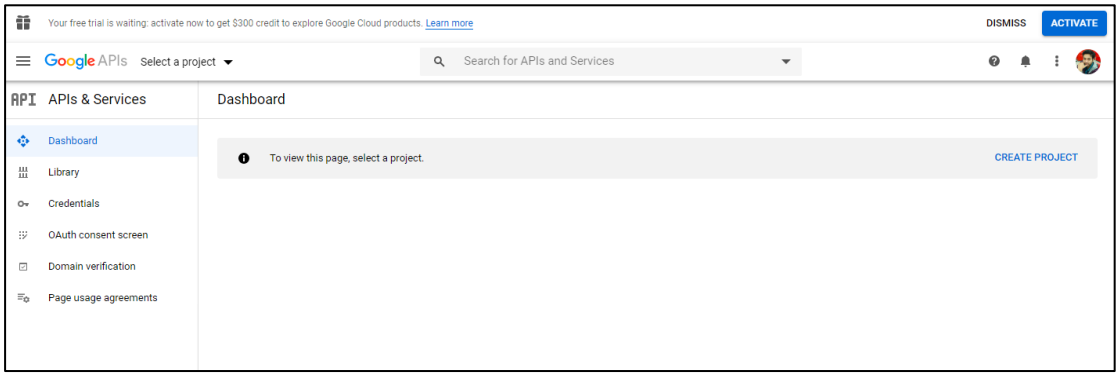


Figure: Google Developer Account

2. Click "Select a Project" option. A modal is opened.

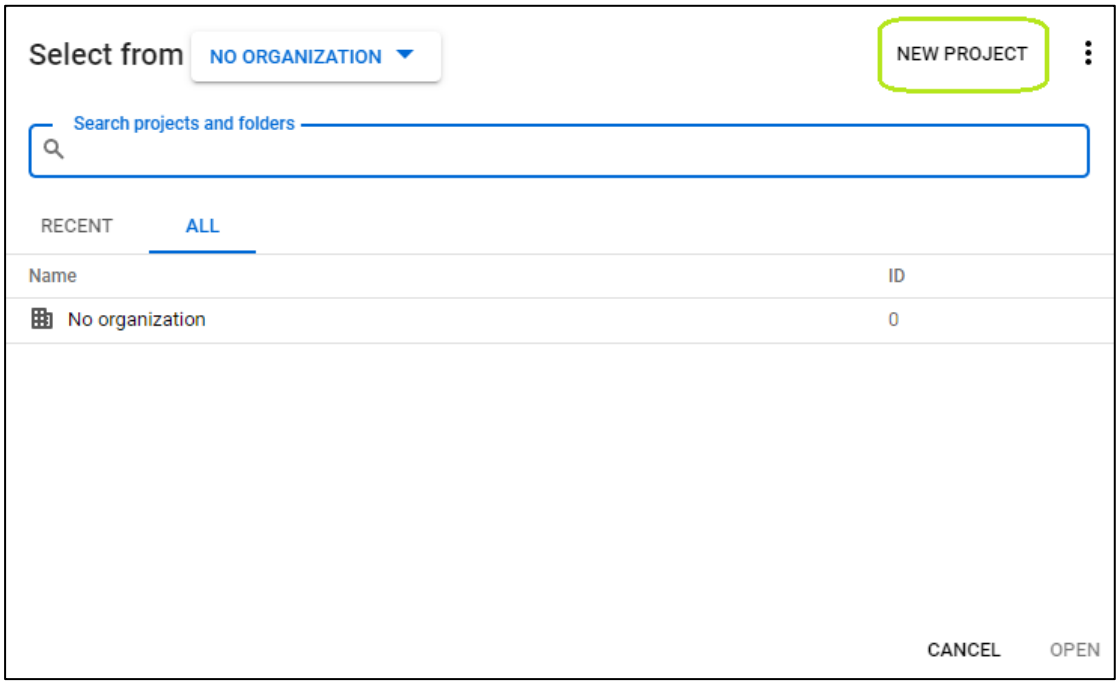


Figure: Project Selection Modal

3. Select the project (if already existed) or click "New Project" option (displayed with a green box in the above screenshot). The following modal is displayed.

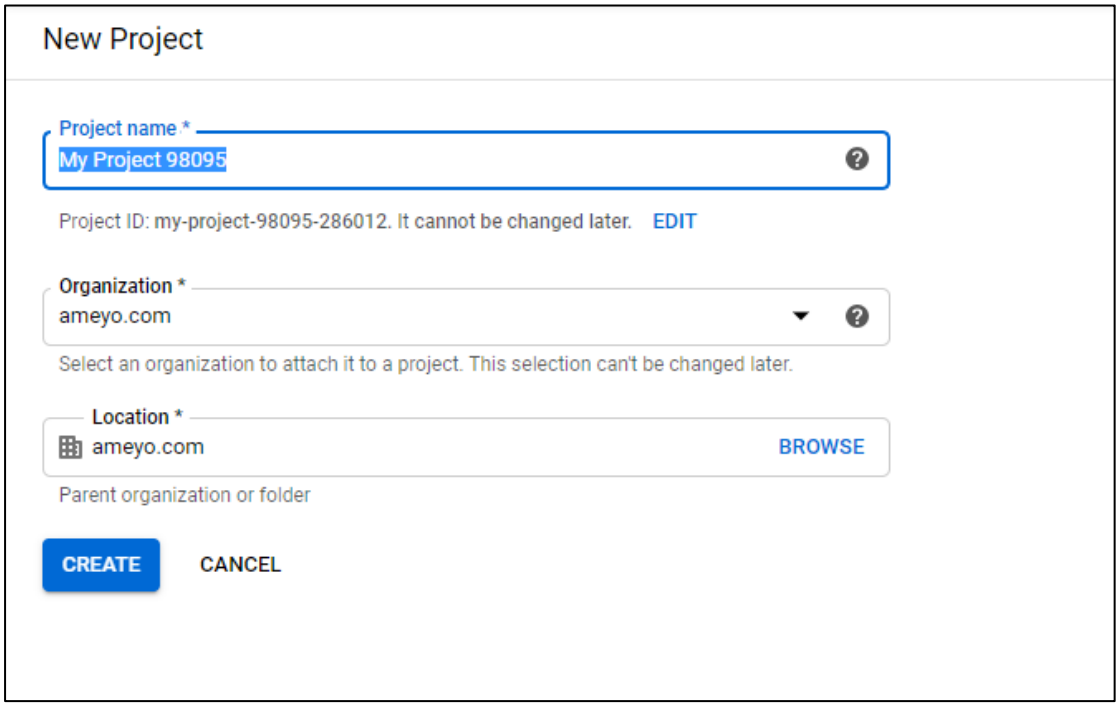


Figure: Project Creation Modal

4. Provide the following information in the opened modal.

   A. **Project Name:** Enter the name of the project in the project name data field.

   B. **Organization:** Select the name of the organization from the drop-down list.

   C. **Location:** Select the location where the project will be saved.

   D. **:**

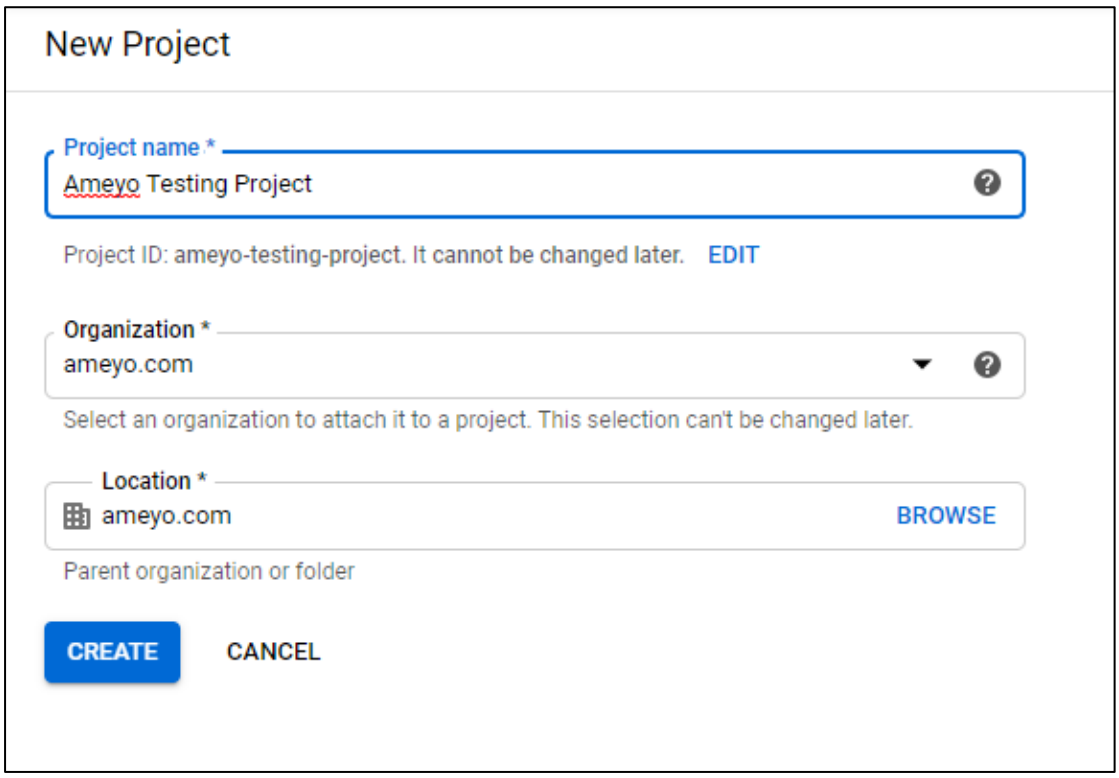


Figure: Sample Project

5. Click "Create" button. After creating the project, the page redirects to the previous page.
6. Now, click "Enable APIs and Services" button present at the top of the page to enable the APIs.
7. Search for the "Google+" API and enable it.

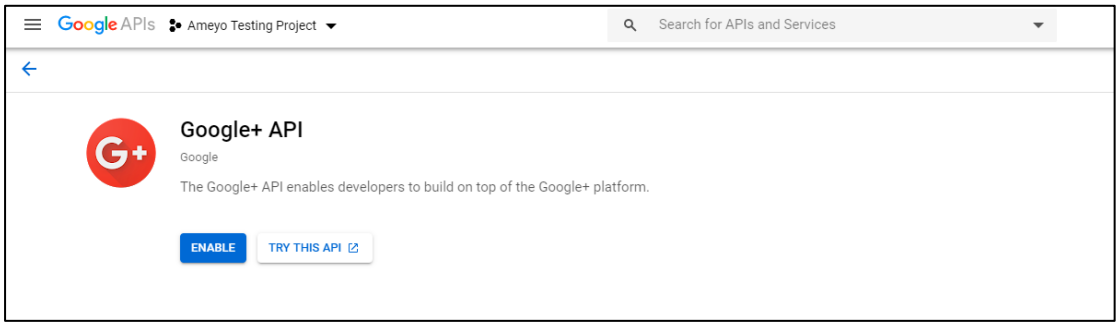


Figure: Enable Google+ API

8. After enabling the API, the following page is displayed.

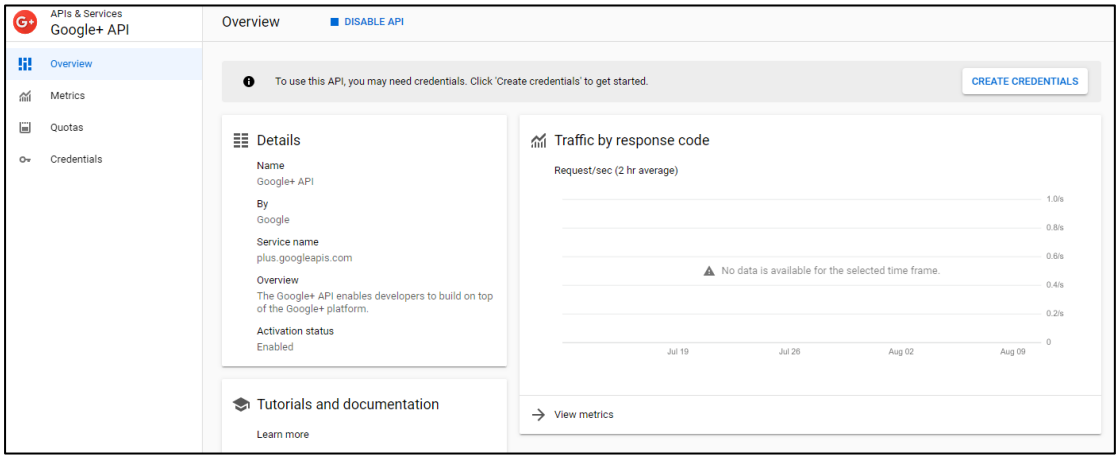


Figure: Overview Page of Google+ API

9. Now, go to the credentials tab present at the left sidebar. The following page is displayed.

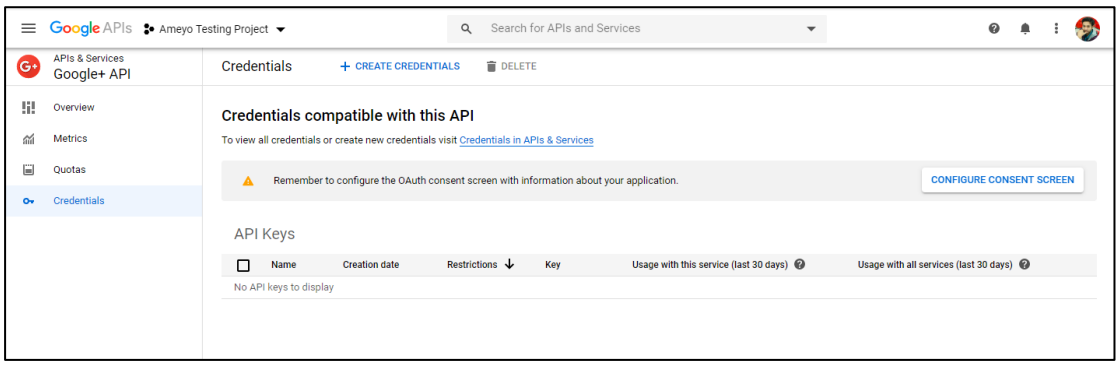


Figure: Credentials Tab of Google+ API

10. Click "Configure Consent Screen" button to configure the Email address. The following page is displayed.

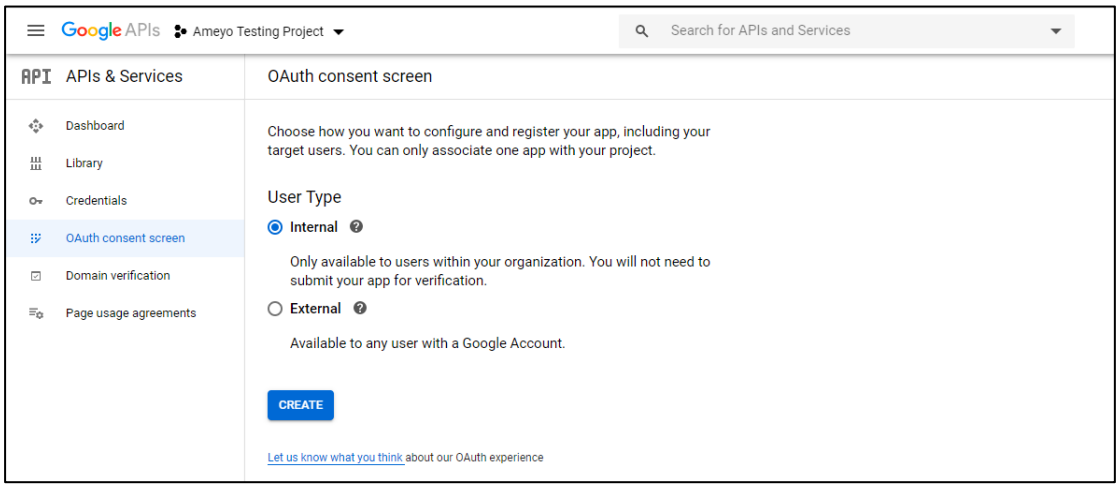


Figure: Consent Screen Configuration

11. Select the type of the user, which is allowed to use this app. The following two options are present here.

    A. **Internal:** Select it to allow users within the organization only.

    B. **External:** Select it to allow all the users with a Google Account.

12. Click "Create" button. The following OAuth form is displayed.

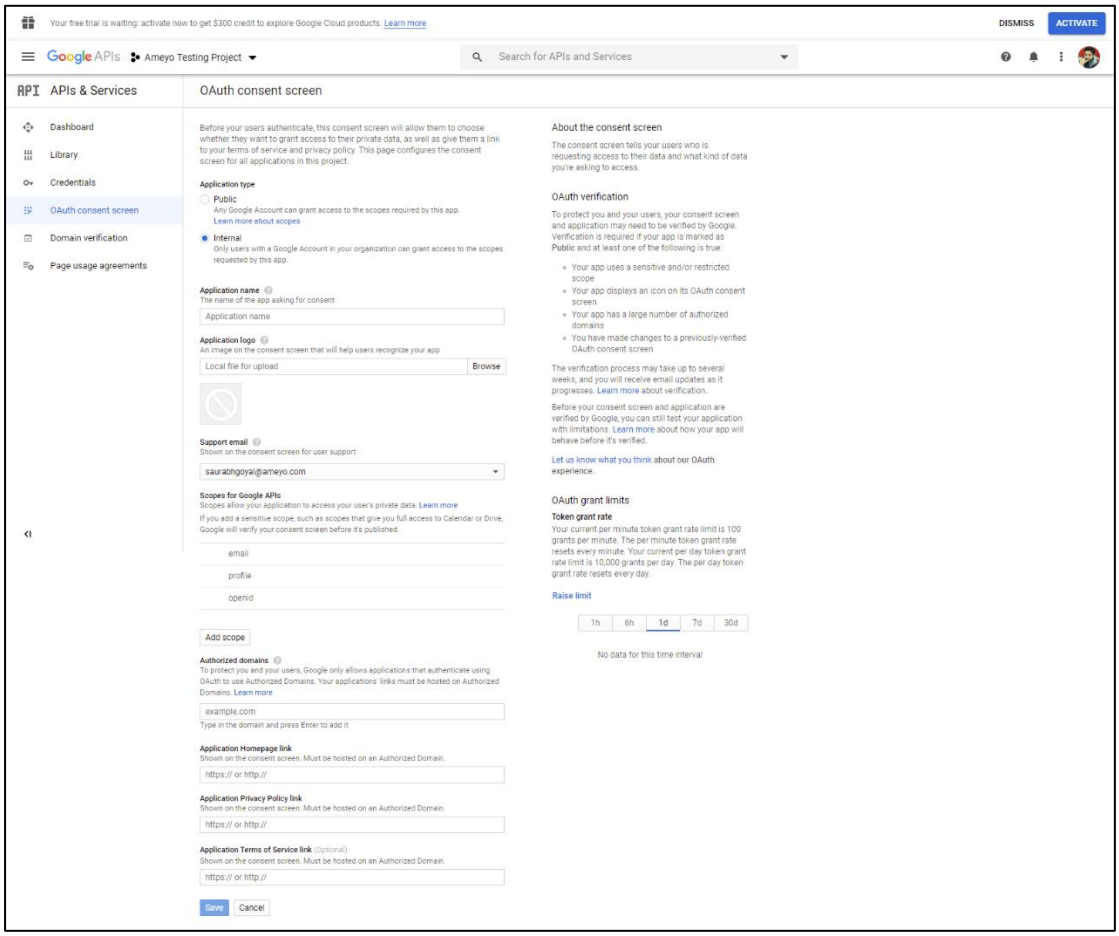


Figure: Consent Form after App creation

13. Fill the consent form that has been displayed, and click "Save" button.

14. Now, click "Create Credentials" button in the "Credentials" tab to create the credentials. Click "Create OAuth Client ID".

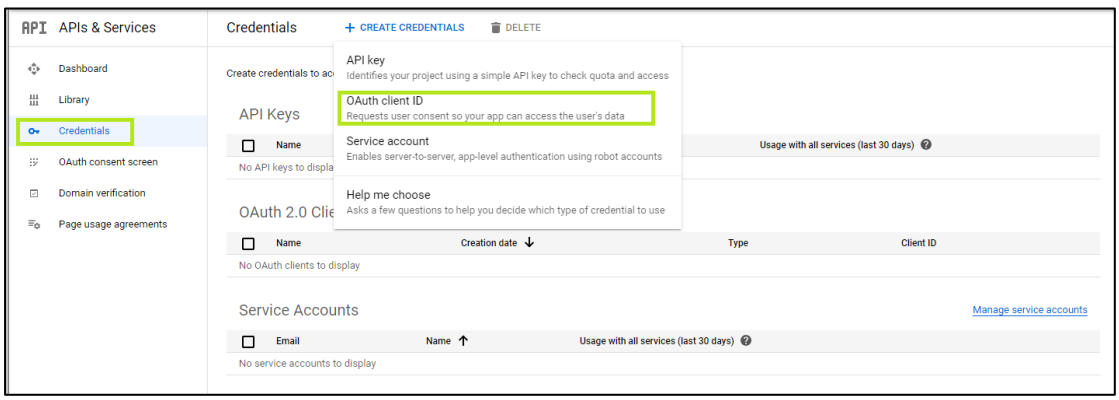


Figure: Creating Credentials

15. From the displayed page, select "Web Application" from the drop-down list of Application type.

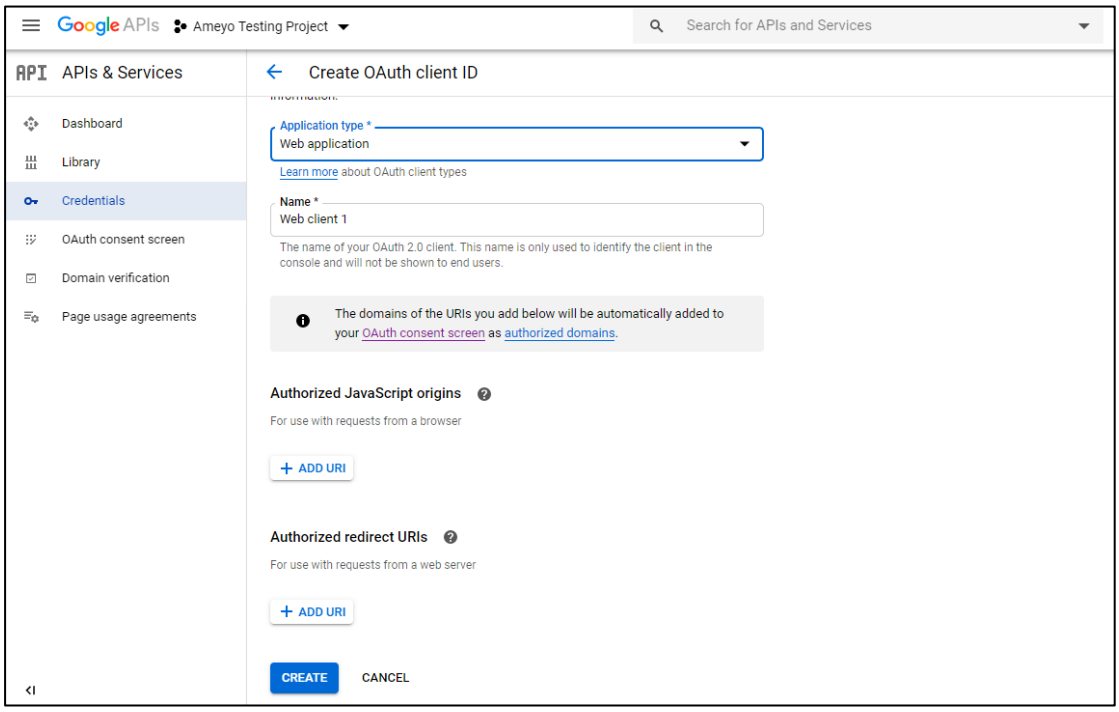


Figure: Creating OAuth Client ID

16. Enter the following URI in the Authorized JavaScript Origins.

```
https://ameyo.com:<PORT_Number_of_Application>
```

17. Enter the Authorized Redirect URI in the following format:

```
<Protocol>://<Domain_Name>:<PORT_Number>/ameyowebaccess/_callback?consumerId=<Consumer_ID>
```

18. Click "Create" button. A modal is displayed.

19. The displayed modal consists of the OAuth Client Id and Client Secret Keys generated after validation of all the above steps. Copy these keys and keep the pen down, as they have to be used later.

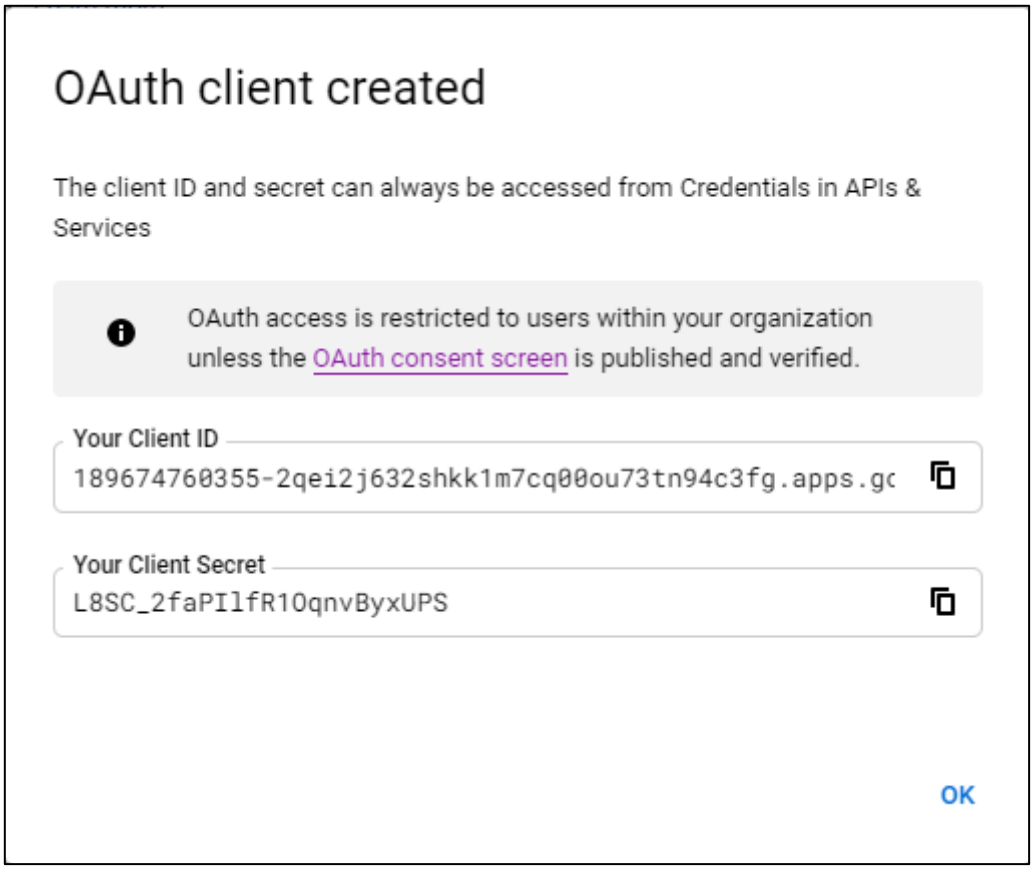


Figure: Generated Client Key and Client Secret Keys

6.1 Backend Configuration to enable Google SSO Configuration

Perform the following steps to configure the Google SSO configuration for Ameyo.

1. Run the following command to access the command line database.

```
psql -U postgres <Application_Database_Name>
```

2. Run the following query to insert the SSO OAuth policy in the database.

```
INSERT INTO open_auth_consumer_configuration_table
(consumer_id,service_provider,is_auto_create_user,retry_with_logi
n,ameyo_login_url,home_url) VALUES
(<consumer_id>,'googlePlus',false,false,'{"sso.oauth.policy" :
"/ameyowebaccess/login"}','{"sso.oauth.policy" : "/app"}');
```

3. Run the following query to insert the Domain Name with SSO

```
INSERT INTO open_auth_service_generic_configuration
(consumer_id,client_id,client_secret
,oauth_authorization_url,oauth_token_request_url,oauth_redirect_u
ri,oauth_scope_value ) VALUES ('<consumer-id>','<client-
id>','<client-
secret>','https://accounts.google.com/o/oauth2/auth',
'https://accounts.google.com/o/oauth2/token', '{"sso.oauth.policy"
: "https://<domain-configured-with-
sso>:8443/ameyowebaccess/_callback?consumerId=<Consumber_ID>"}',
'{"sso.oauth.policy" : "profile email"}');
```

4. Run the following query to insert the Instance URL in the database

```
INSERT INTO open_auth_response_additional_parameter_configuration
(consumer_id,parameter) VALUES (<consumer-id>,'{"sso.oauth.policy"
: ["instance_url"]}');
```

5. Run the following query to enable SSO with Application Domain Name

```
INSERT INTO
domain_auth_configuration(domain_name,consumer_id,service_provide
r) VALUES
('<Application_Domain_Name>','<Consumer_ID>','googlePlus');
```

You can contact the Support team of Ameyo for any reference or assistance on any of the above parameters.

7. STD Code Management Policy

In the Contact Centre Industry, the phone numbers are fetched from multiple sources. Each source provides phone numbers in different formats. For example, one source sends the phone number of 10 digits, whereas another source adds "0" (zero) before a 10-digit phone number. In some cases, a source sends the numbers with country code, whereas another source sends the numbers without country code. Phone numbers in different formats may be available in an Ameyo Setup. However, in Ameyo, a phone number with different formats will be counted as different phone numbers. For example, if John's number is stored as 123456789, then a call made to 0123456789 will be counted as a different number instead of John's number.

If STD Code Management is applied with other Phone Number Cleanup Policies, then it will be as the organizations collect phone numbers from the different sources that may have multiple number formats such as +91 123 456 7891, +91-1234567891, 12345647891, 01234567891, 001234567891. With the applicability of Phone Number Cleanup Policies, the Ameyo System can provide the same format for all phone numbers even if they are submitted in different formats.

Landline Number Policy maps the location code to the STD Codes, and the mapped STD code will be added as a prefix to all phone numbers. Its code name is "LandlineNumbersPolicy".

7.1.1 Architecture and Configuration of Number Cleanup Policies

The Number Cleanup Policies can be enabled at the System-level. After upgrading the Ameyo Server package to 4.7 GA, you have to run the following query to add the flag "shouldEnableAutoNumberCleanup" and provide its value as "true".

```
INSERT INTO server_preference_store(context_type, context_id, key, value) VALUES (<system/process>, <ID of cc/process>, 'shouldEnableAutoNumberCleanup', true );
```

If "shouldEnableAutoNumberCleanup" is not added in "server_preference_store" for system-level, then its value will be considered as "false," by default, which means the Number Cleanup Policies are disabled by default.

As soon as the value of the "shouldEnableAutoNumberCleanup" flag is provided as "true" at system-level, then all 5 Number Cleanup Policies will be applicable at the system-level and will be applicable to all processes in the system.

You can use the above query to disable the Number Cleanup Policy at the process-level. If this configuration is not available for a process, then the system-level configuration will be applicable to that process.

Number Cleanup Policies cannot be configured at the Campaign-level.

Now, you can define which Number Cleanup Policy out of total 5 policies (from the above list) will be applicable to a process. If this policy selection configuration is not available for a process, then all policies will be applicable in the process.

You have to provide the value of "allowedCleanupPolicies" flag in the server_preference_store to select which number cleanup policies will be enabled in a process. Run the following query.

7.1.2 Enable Number Cleanup Policy

INSERT INTO server_preference_store(context_type, context_id, key, value) VALUES ('process','<process_id>','allowedCleanupPolices','<comma_separated_list_of_policies>');

Replace <process_id> with the ID of the Process. Provide any of the following values in the comma separated format in the place of <comma_separated_list_of_policies>.

- RemoveLeadingZerosPolicy
- RemoveSpecialCharacterPolicy
- LastNDigitsPolicy
- AddCountryCodePolicy
- LandlineNumbersPolicy

Number Cleanup Policies cannot be configured at the Campaign-level.

7.1.3 RemoveSpecialCharacterPolicy

This Policy is used to remove any special character present within the phone number. For example, a phone number <+91-910 109 7656> is changed to <9101097656>.

There is no configuration for this; the only enablement of this policy is sufficient.

7.1.4 RemoveLeadingZerosPolicy

This policy is used to remove all the zeros before the phone numbers. For example, a phone number <09101097656> is changed to <9101097656>.

There is no configuration for this; the only enablement of this policy is sufficient.

7.1.5 AddCountryCodePolicy

This policy helps to add the prefix with the phone number. This policy reads another policy first and then apply its configuration.

For example, suppose the countryCodePrefixValue is +91, thus the phone number <9101097656> is changed to <+919101097656>.

Run the following query to insert the country code in the database.

```
INSERT INTO server_preference_store (context_type, context_id, key, value) VALUES ('process','<Process_Id>','AddCountryCodePolicy','+91');
```

Here, '+91' is the Country code of India. You can define another country code depend upon the organization's usage.

7.1.6 Configuration of Landline Number Policy for STD Code Management

After enabling this policy, you have to provide the mapping of a location code (that is a locality name, city name, or PIN Code) to the STD Code. The selected location code (that is locality name, city name, or PIN Code) should be a part of the Data Table Definition fields and should store the selected location code. Multiple queries can be run to store the mapping entries of a unique location code to a unique STD Code in "server_preference_store." **Make sure not to provide multiple mappings of an STD Code.**

- If the preferred location code that is locality name, city name, or PIN Code is not a part of the Data Table Definition Field to store the customer address, then "Landline Number Policy" will not work.
- Also, if the value of any location code (locality name, city name, or PIN Code) for any mapping entry is not available in the record of a customer, then Landline Number Policy will not work for that customer.

- Please do not use Country Code with STD Code in "Landline Number Policy" if you are also enabling "Country Code Policy" in the same process.

Run the following query to add a mapping entry to map a location code to an STD Code.

```
INSERT INTO std_code_location_code_mapping(location_code,std_code)
VALUES ('<location_code>','<std_code>');
```

Replace <location_code> with location name (such as Sohna Road), city name (such as Gurgaon), and PIN Code (such as 122003) and replace <STD_Code> with the STD Code (such as 124).

The "std_code_location_code_mapping" table will be available after upgrading Ameyo Server to 4.7 GA.

Consider the following examples of this query.

- **Using City Name as Location Code:** Run the following query.

  ```
  INSERT INTO std_code_location_code_mapping(location_code,std_code)
  VALUES('<city_name>','<std_code>');
  ```

  Replace <city_name> with city name (such as Delhi) and replace <STD_Code> with the STD Code (such as 11).

  Example:

  ```
  INSERT INTO std_code_location_code_mapping(location_code,std_code)
  VALUES('<Delhi>','<11>');
  ```

  After running this command in a process, the phone numbers of all customers having "Delhi" as City will be prefixed with STD Code = 11.

- **Using Location Name as Location Code:** Run the following query.

  ```
  INSERT INTO std_code_location_code_mapping(location_code,std_code)
  VALUES('<location_name>','<std_code>');
  ```

  Replace <location_name> with locality name (such as Sohna Road) and replace <STD_Code> with the STD Code (such as 124).

  Example:

```
INSERT INTO std_code_location_code_mapping(location_code,std_code) VALUES('<Sohna_Road>','<124>');
```

After running this command in a process, the phone numbers of all customers having "Sohna Road" as Locality Name will be prefixed with STD Code = 124.

- **Using PIN Code as Location Code:** Run the following query.

```
INSERT INTO std_code_location_code_mapping(location_code,std_code) VALUES('<PIN_Code>','<std_code>');
```

Replace <PIN_Code> with PIN Code (such as 122003) and replace <STD_Code> with the STD Code (such as 124).

Example:

```
INSERT INTO std_code_location_code_mapping(location_code,std_code) VALUES('<122003>','<124>');
```

After running this command in a process, the phone numbers of all customers having "122003" as PIN Code will be prefixed with STD Code = 124.

You can configure these policies for different processes. For example, "RemoveSpecialCharacterPolicy" is for process 1, and "LastNDigitsPolicy" is for process 2.

7.1.7 Disable any Particular Policy

Run the following query to disable any particular number policy.

```
UPDATE number_correction_policies SET enabled = <false> where policy_name = '<Policy_Name>' ;
```

Provide the name of the policy that you want to disable.

7.1.8 API for STD Code Management

The following two APIs are available for STD Code Management Policy.

1. **Add STD Code Management API:** This API is used to add the STD code management in Management Framework Architecture. Know more...

2. **Get STD Code Management API:** This API is used to fetch the list of the STD codes that have been uploaded in Management Framework Architecture. Know more...

Provide the credentials of your Application server and other necessary details, as mentioned in the APIs itself.

7.2 Add STD Code Management API

This API is used to add the STD code management in Management Framework Architecture.

7.2.1 Method

POST

7.2.2 EndPoint URL

```
<Protocol>://<Domain_Name>:<Port_Number>/ameyorestapi/managementServer
/addStdCodeToLocation
```

Here, Domain_Name and Port_Number are the details of the management server.

7.3 Header parameters

```
sessionId : <Session_Id_of_Administrator>
```

7.3.1 Request Parameters

For the STD code management, the user has to upload the CSV file with the STD codes implemented in it.

```
form-data:
        fileName : <Upload *.CSV file of STD code management>
```

7.4 Sample URL-based Command for Add STD Code Management API

```
<protocol>://<IP_Domain_Name>:<port>/ameyorestapi/managementServer/add
StdCodeToLocation
```

7.5 Get STD Code Management API

This API is used to fetch the list of the STD codes that have been uploaded in Management Framework Architecture.

7.5.1 Method

GET

7.5.2 EndPoint URL

```
<Protocol>://<Domain_Name>:<Port_Number>/ameyorestapi/customerManager/getStdCodeCSV
```

Here, Domain_Name and Port_Number are the details of the management server.

7.6 Header parameters

```
sessionId : <Session_Id_of_Administrator>
```

7.6.1 Response Output Parameters

This API will return the uploaded STD codes in the management layer.

7.7 Sample URL-based Command for Add STD Code Management API

```
<protocol>://<IP_Domain_Name>:<port>/ameyorestapi/customerManager/getStdCodeCSV
```

8. Logon to Management Framework Architecture

After the configuration, logon to the Management Framework Architecture. Use the following URL to logon to the Management Framework Architecture Interface.

`https://<Domain_Name>:<Management_UI_Port_Number>/managementserverui/NoPopUpIndex.html`

The default port for Management UI is 8887, until and unless it hasn't been changed.

The following screenshot shows the User Logon Page of the Management Framework Architecture.

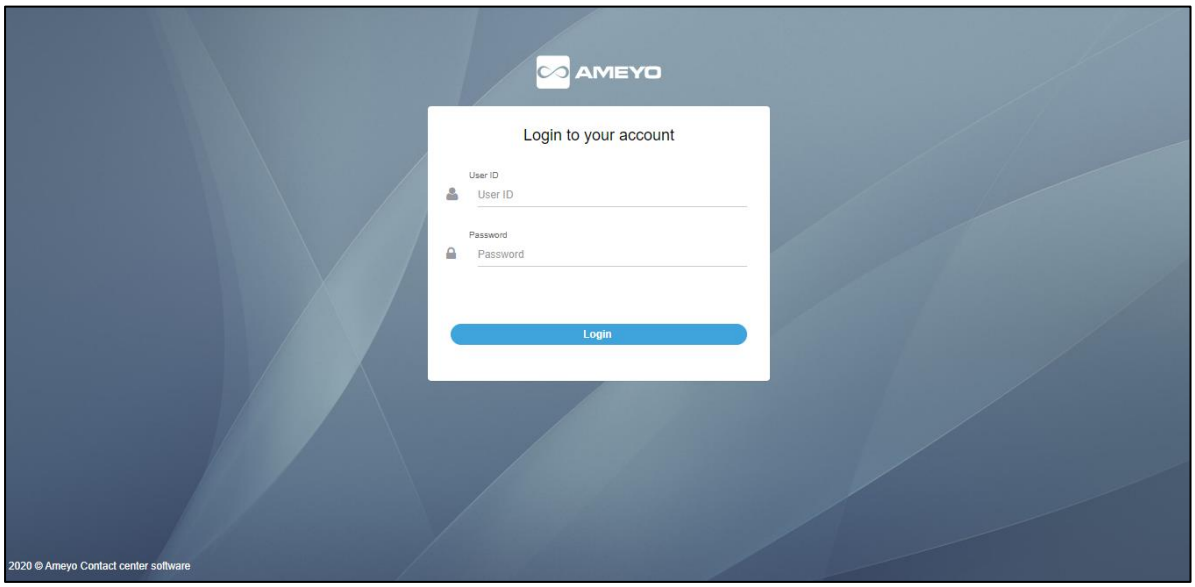


Figure: Logon Page of Management Framework Architecture

9. MAmdinistrator

The MAdministrator, administrator of Management Framework Architecture, has the privileges to manage the multiple tenants, users, and Application Servers. The MAdministrator has access to the Web-based interface of the Management Framework Architecture Interface. The MAdministrator Use the following URL to logon to the Management Framework Architecture Interface.

https://<Domain_Name>:<Management_UI_Port_Number>//managementserverui/NoPopUpIndex.html

The default port for Management UI is 8887, until and unless it hasn't been changed.

The following screenshot shows the logon page of the Management Framework Architecture for The MAdministrator.

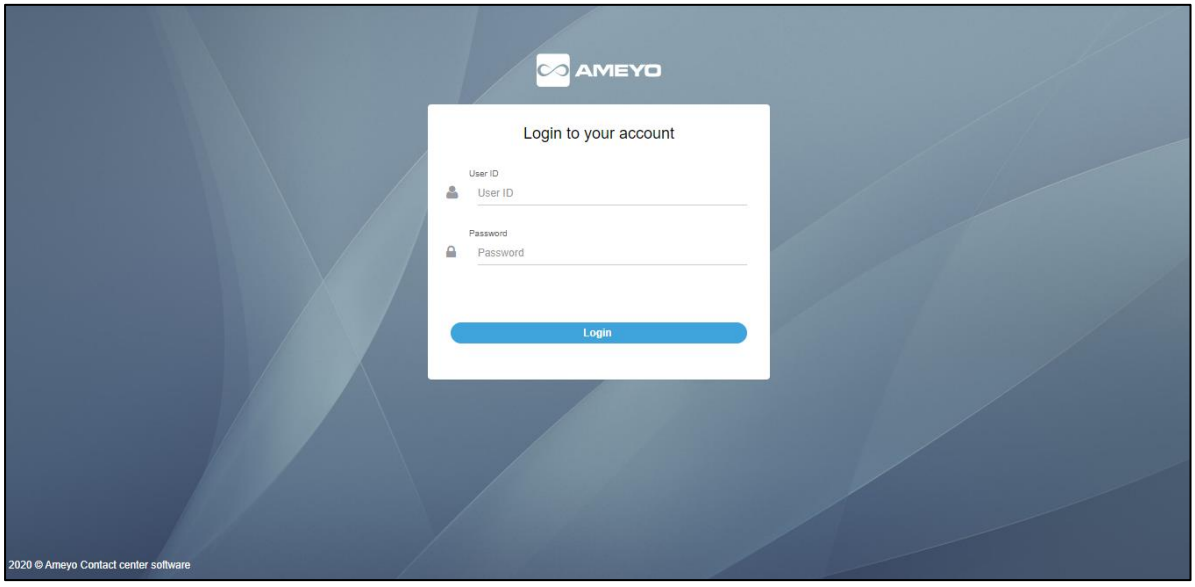


Figure: Login page of Management Framework Architecture for MAdministrator

Enter User ID and Password and click "Login" button. The following page is displayed after the logon.

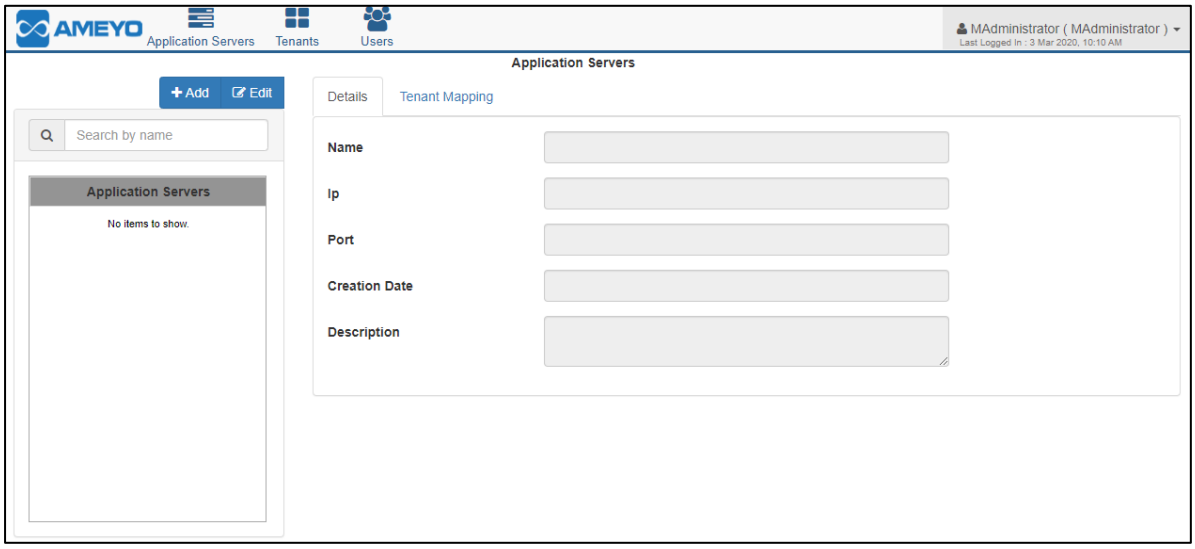


Figure: First Screen after Login

The top bar on the above page contains the following tabs. no servers to show

1. **Application Servers:** It allows The MAdministrator to map multiple Ameyo Application Server. Know more…
2. **Tenants:** It provides the feature to The MAdministrator to add multiple tenants and assign them to their respective Application Servers. Know more…
3. **Users:** It allows The MAdministrator to add and assign the users to their campaigns. Know more…

9.1 Application Server Tab

"Application Servers" Tab of Management Framework Architecture allows the MAdministrator to manage Ameyo Application Servers.

The MAdministrator can perform the following operations here.

9.1.1 Add Application Server

Perform the following steps to add the Application Server.

1. Click "Add" button to add a new Application Server using the following modal.

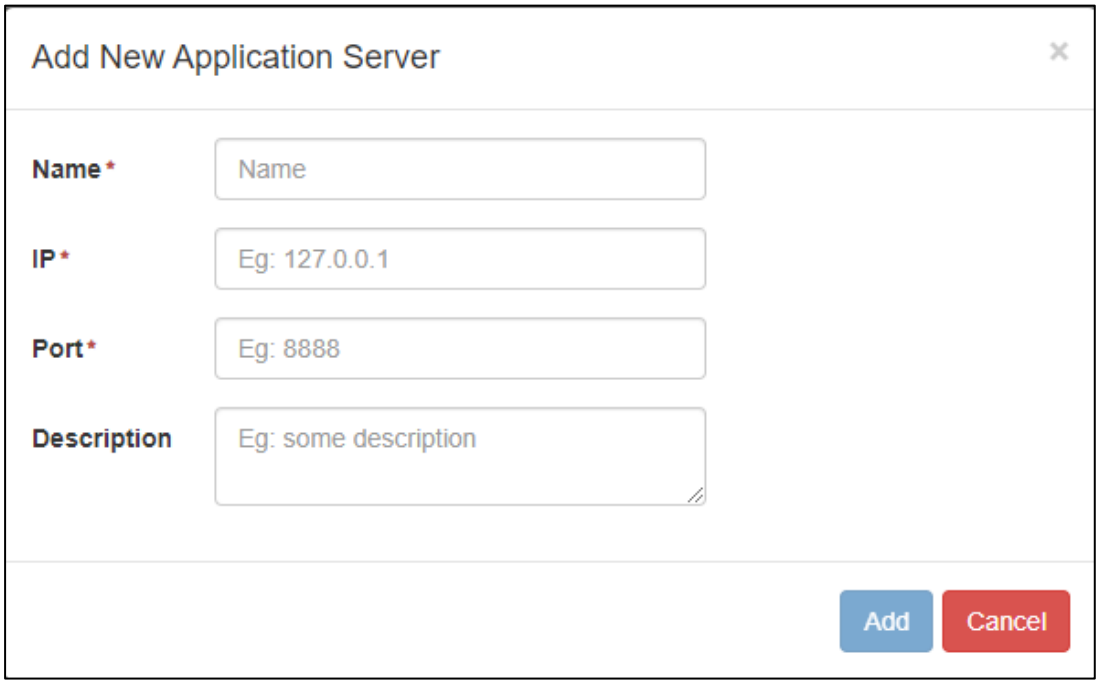


Figure: Add new Application Server

2. Provide the following details in this modal.
    A. **Name:** Enter the name of the Application Server.
    B. **IP:** Provide the IP address of the Application Server. If Ameyo is enabled with Secure mode, then provide your domain name instead of your IP.
    C. **Port:** Enter the Port Number of the Application Server at which it is configured.
    D. **Description (Optional):** Provide the description of Application Server, if required.

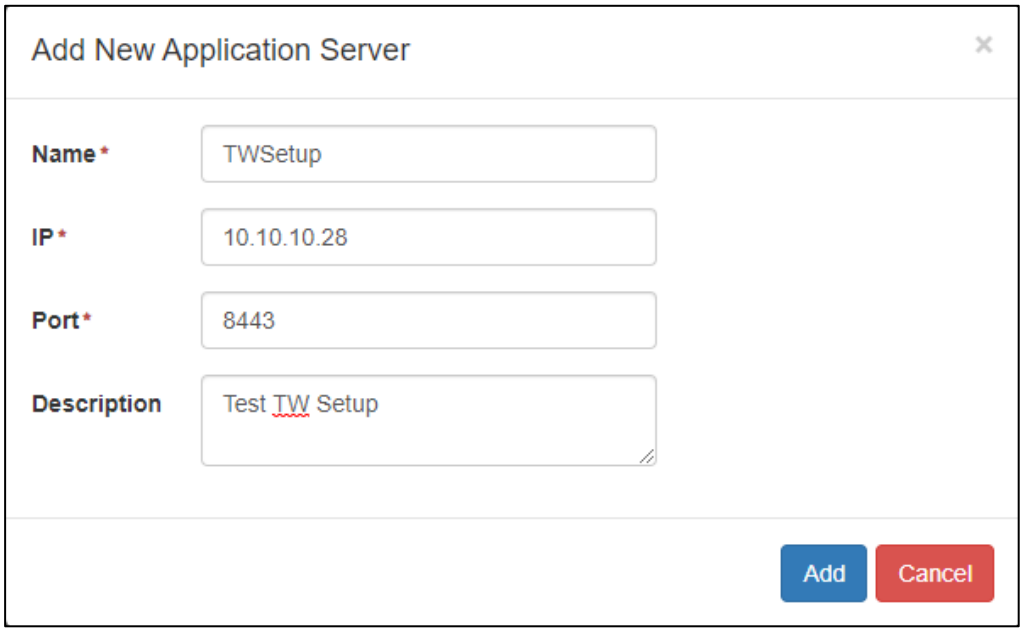


Figure: Sample Application Server Details

3. After providing all details, click "Add" button. The following page shows the added Application Server.

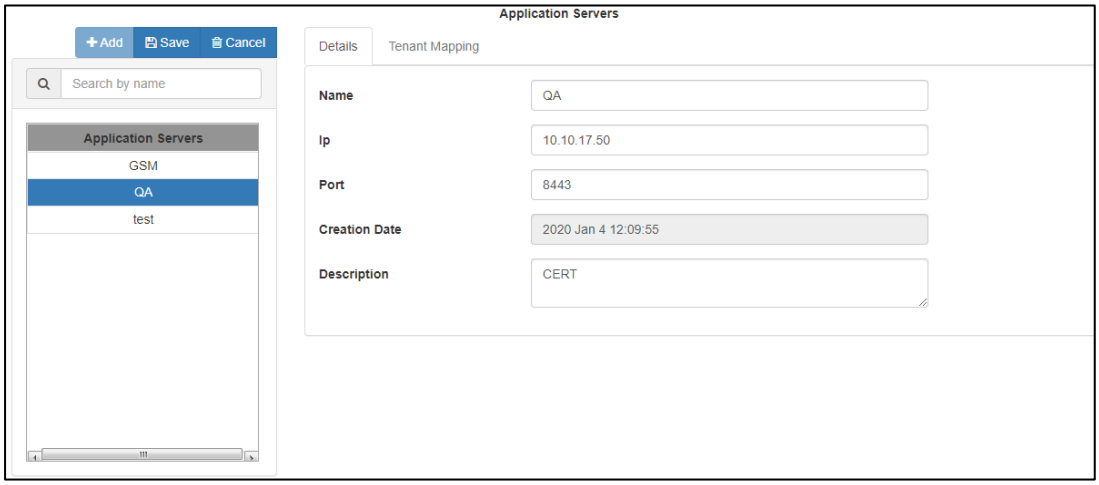


Figure: Added Application Server

9.1.2 Edit Application Server

Perform the following steps to edit an Application Server.

1. Select the Application server present at the left panel and click "Edit" button.

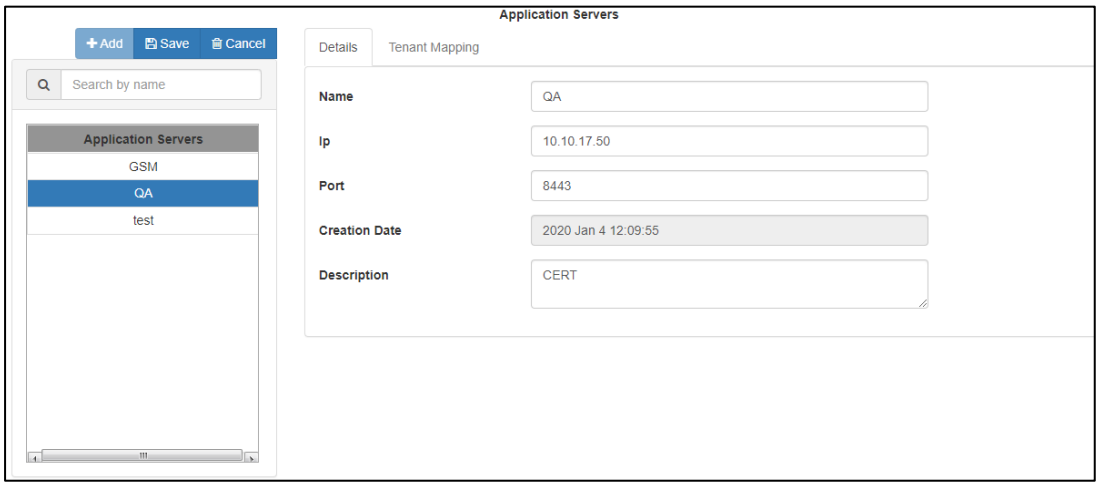


Figure: Edit Details of Application Server

2. All the details of the Application Server are now editable in the right panel.
3. Edit the details and click "Save" button.

9.1.3 View the list of Application Servers

The left panel of the Application Servers tab contains the list of all servers that are configured on the Management Framework Architecture. You can search for the Application Server from the list of the Application Servers through "Search" option present in the left panel.

Application Servers Tab has the following two different tabs, which help to create an Application Server and assign tenants to its respective Application Server.

1. **Details Tab:** It helps to create the Application Servers. Know more...
2. **Tenant Mapping:** It helps to map the Application Server with its respective Tenants. Know more...

9.1.4 Details Tab

It shows the details of the added Application Servers. Perform the following steps to view the details of any Application Server.

1. Select the Application Server from the left panel.
2. The details of the selected Application Server are now visible on Details tab, but in a non-editable format, means, it cannot be edited. Following details are visible.
3. Name of the Application server.
4. IP Address of that Application Server.
5. Port Number of the server, on which it is configured.
6. Creation (Addition) Date of the Application Server on Management Framework Architecture.
7. Description of that Application Server.

All the above details are non-editable until you click "Edit" button present in the left panel of the page.

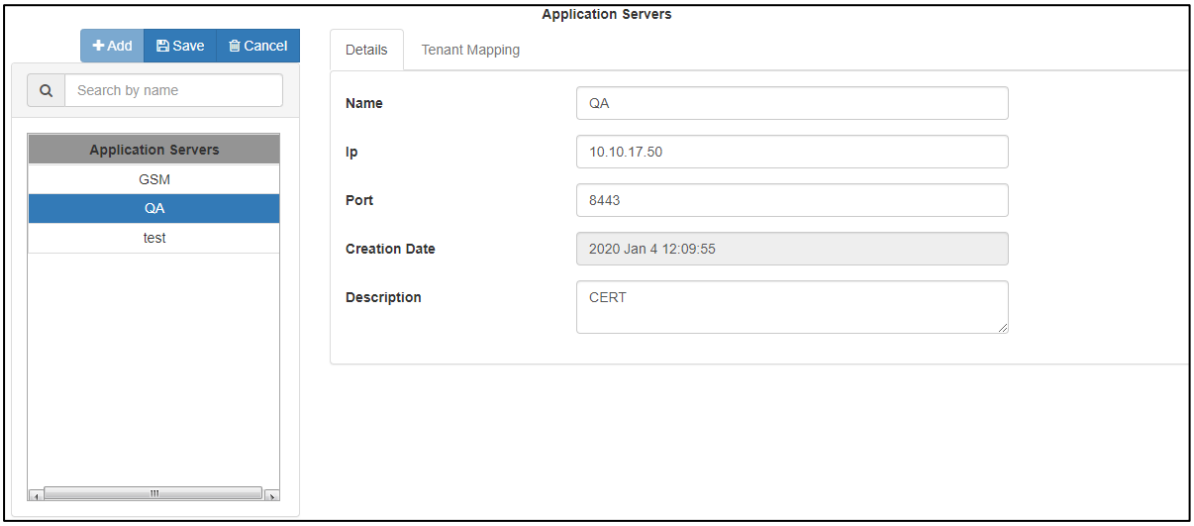


Figure: Details tab of Application Servers Menu

Use the above steps to create multiple Application Servers.

It is recommended not to enter the same name for multiple Application Servers.

9.1.5 Tenant Mapping Tab in Application Server

Before performing this step, tenants should be created. know more...

The Tenant Mapping tab allows the user to assign the tenants into the Application Server. Perform the following steps.

1. Select the Application Server from the left panel. All the tenants are shown in "Available Tenants" section.
2. Select the Tenant that you want to assign to the Application Server.
3. Click 

button to assign the tenants. All the tenants that are assigned are listed into "Assigned Tenants" section.
4. You can search for any specific Tenants by typing its name in the search option in any section.

Once the Tenant is assigned to the Application Server, then there is no option to unassign it.

The following screenshot shows the User Logon Page of the Management Framework Architecture.

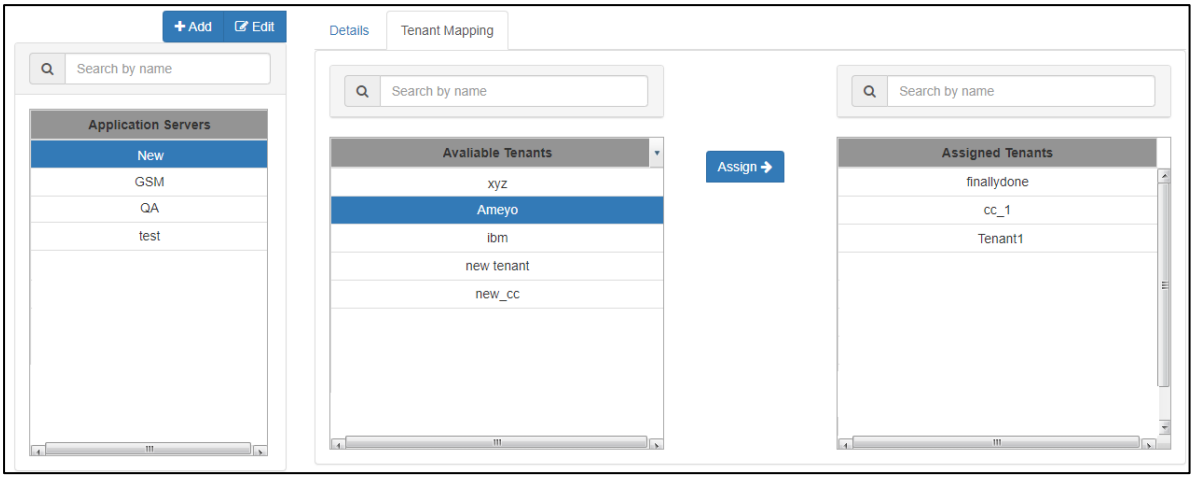


Figure: Tenant Mapping

9.2 Creation and Management of Tenants

"Tenants" Tab helps the MAdministrator to add and map the existing Application Servers. The MAdministrator can also create Tenants, that is Application Servers, from the Management Framework Architecture. These tenants can also be edited and deleted here also.

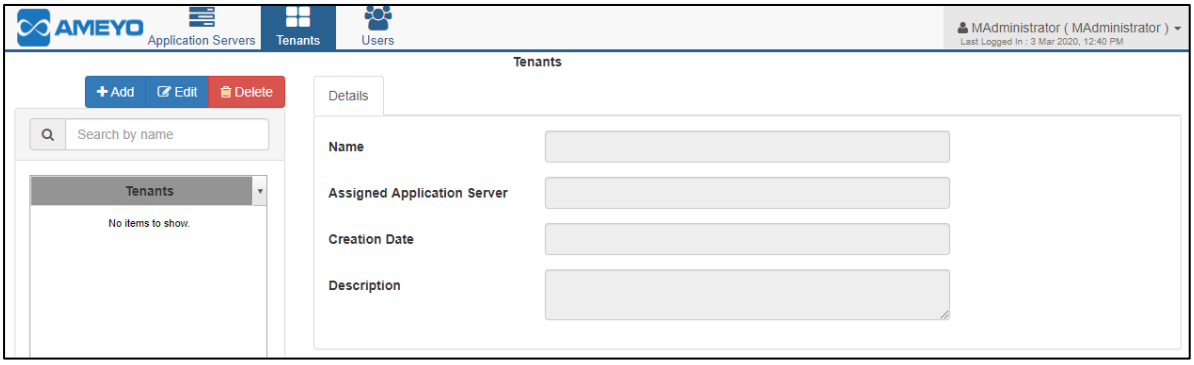


Figure: Tenants Tab

You can perform the following steps in Tenants Tab.

- Add and Map Existing Application Servers as Tenants
- Create New Tenants, that is, Application Servers
- Edit and Delete the Tenants, that is, Application Servers

9.2.1 Add and Map Existing Application Servers as Tenants

If you are integrating Management Framework Architecture on presently running server, then it is important to sync all the existing Tenants and map them with their respective Application Servers manually. If you do not create and map them, then there are chances that creation of new Tenants can be failed.

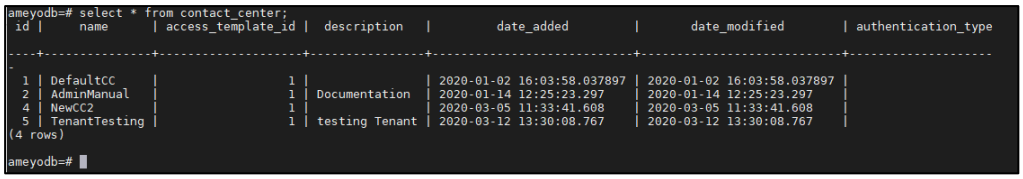
There is a possibility that Management Framework Architecture is integrating on already existing Application Server. In such a case, there may be tenants that are already created and mapped with their respective application servers. Thus, you have to map those tenants manually. Perform the following steps to create and map already existing Tenants:

1. Execute the following commands to create the databases for Management Server and management UI applications.
   A. Execute the following command to logon to PostgreSQL database System.

      psql -U postgres <Ameyo_Appserver_Database>

   B. Run the following query to check the available tenants on any application server.

      SELECT * from contact_center;

2. 

3. **Figure:** Checking the Existing Tenants

4. Copy the names of the Tenants from here with their respective "ID". It has to be kept in mind that the ID's of Tenants from here has to be used later. Thus, the ID of Tenants should be correct.

   A. Execute the following command to change and select Management Server application's database:

      \c <Management_Server_Database_Name>

   B. Run the following query to add all the tenants one by one:

      INSERT into tenant (tenant_id,tenant_name,description,date_added,date_modified,contact_center_id)

VALUES

('<'Tenant_ID>','&;t'Tenant_Name>','<Tenant_Description>',now(),now(),'1');

You have to add all the tenants with the help of above query one by one. Adding tenants with this query is irrespective of the application server. In above query, <Tenant_ID> should be in increasing and sequential order.

There is a possibility that name of tenants are same on multiple applications server, that case, you can provide different names to the tenants at Management Server. There is no need to change the name at application server. You can use the following query to check whether the tenants are added or not.

SELECT * from tenant:

5.

```
management_server1=# select * from tenant;
 tenant_id |       tenant_name       |       description       |         date_added         |       date_modified        | contact_center_id
-----------+-------------------------+-------------------------+----------------------------+----------------------------+-------------------
         1 | Saurabh_Default_CC      | Saurabh ka default CC   | 2020-03-12 12:25:31.367106 | 2020-03-12 12:25:31.367106 |                 1
         2 | Saurabh_AdminManual_CC  | Saurabh ka second CC    | 2020-03-12 12:27:38.993668 | 2020-03-12 12:27:38.993668 |                 1
         3 | Saurabh_NewCC2_CC       | Saurabh ka fourth CC    | 2020-03-12 12:28:15.166831 | 2020-03-12 12:28:15.166831 |                 1
         5 | TenantTesting           | testing Tenant          | 2020-03-12 13:29:48.458    | 2020-03-12 13:29:48.458    |                 1
(4 rows)

management_server1=#
```

6. **Figure:** List of Added Tenants

A. Execute the following query to map the created tenants with application server which are created in above step. You have to run this query for all the above created tenants.

INSERT into application_tenant_mapping (tenant_id,application_id,mapped_cc_id,date_added) VALUES ('<Tenant_Id>','<Application_Server_Id>','<Mapped_Contact_Center_Id>',now());

Following table defines the sample values used in the aboev query.

Sample Value	**Definition**
<Tenant_ID>	It is the ID of the Tenants that are added in previous step. This ID should be correct.
<Application_Server_ID>	It is the ID of the Application server that are created in the above steps using Management Server Interface.

Sample Value	Definition
<Mapped_Contact_Center_ID>	It is the ID of the contact_centers which are retrieved from above database query. It is recommended to cross check this ID once before inserting.

7. Execute the following command to exit from the database console

   \q

9.2.2 Create New Tenants from Tenants Tab

Perform the following steps to add new Tenant.

1. Click "Add" button to add the tenant using the following modal.

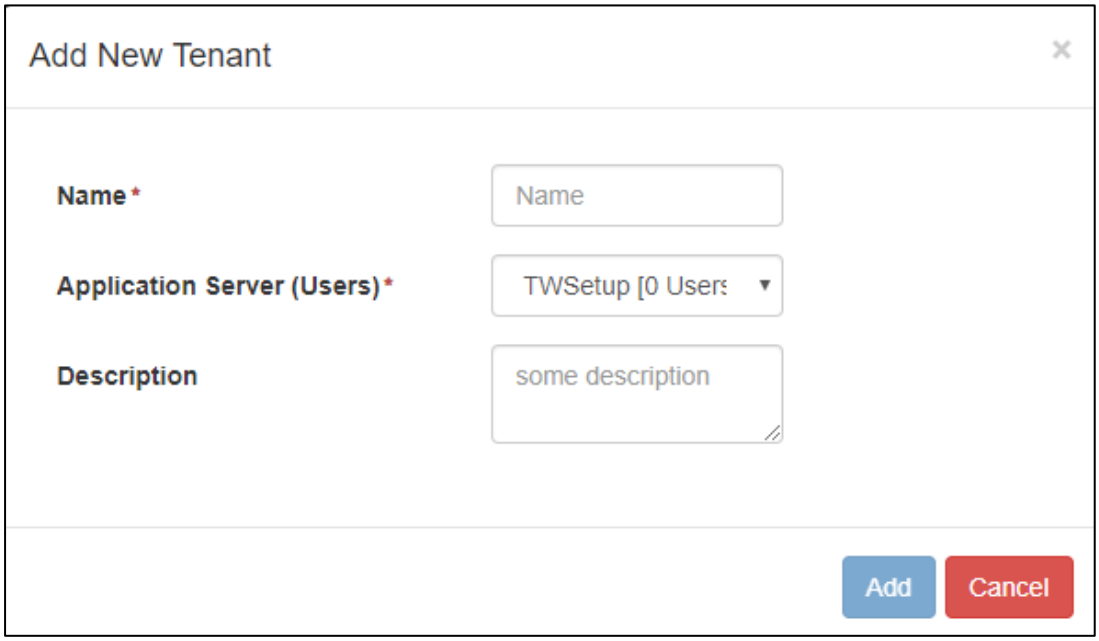


Figure: Add new Tenant

2. Provide the name of Tenant in "Name" text field.
3. Select the Application Server from "Application Servers (Users)" drop-down menu.
4. Add the description of the Tenant in the description textbox.
5. Click "Add" button. A new tenant is being created and added to the Application Server.

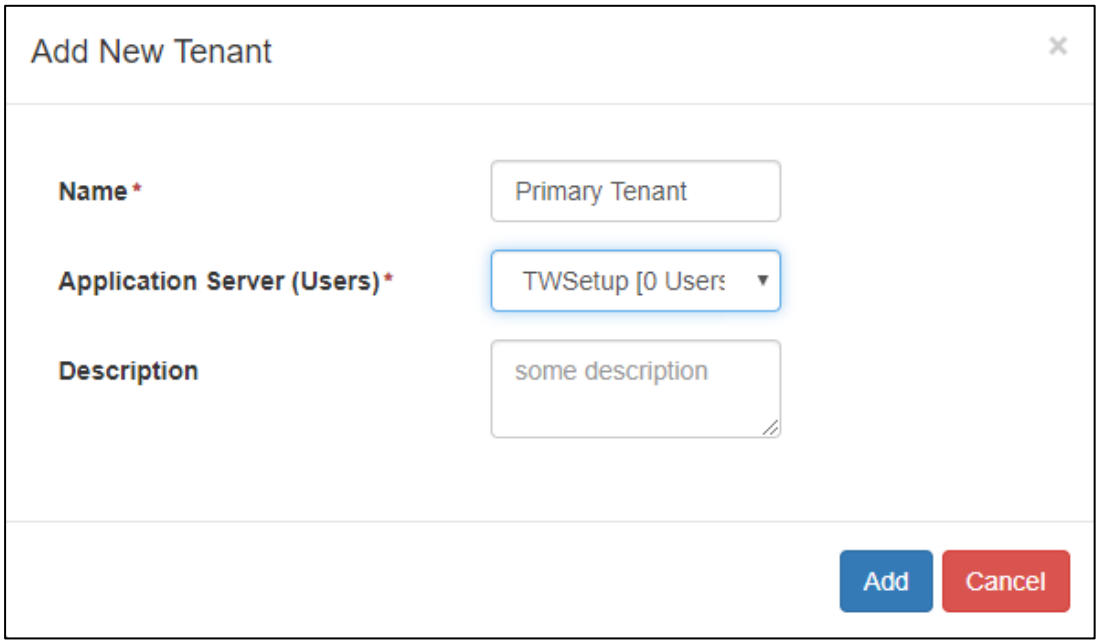


Figure: Tenant Sample Details

9.2.3 Edit and Delete Tenants from Tenants Tab

9.2.3.1 Edit the Tenant Details

Perform the following steps to edit the existing Tenant.

1. Select the Tenant from the left panel and click "Edit" button. It shows the details of the tenants.

   You can search for the tenants as well.

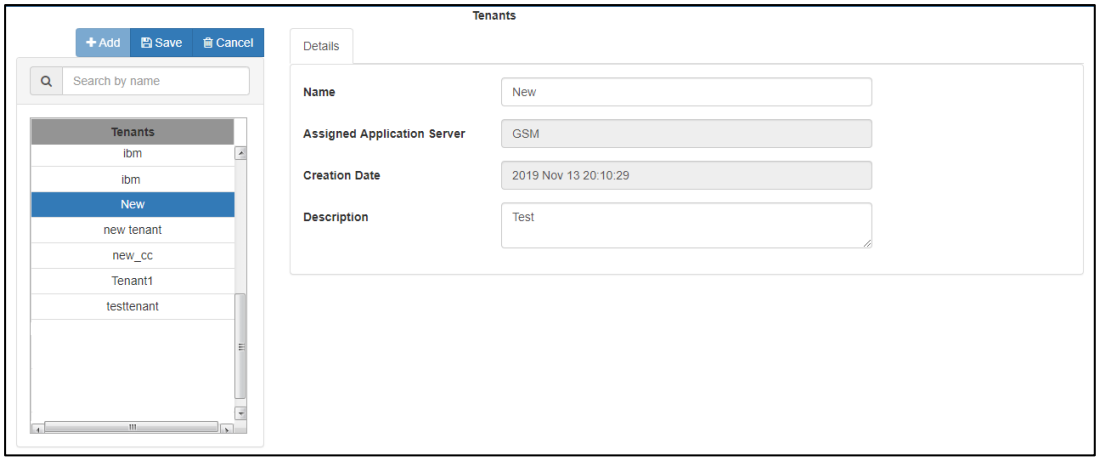


Figure: Editing the Tenant

2. You can change the name and description. However, you cannot change "Assigned Application Server" and "Creation Date".
3. Click "Save" button.

9.2.3.2 Delete the Tenant

Perform the following steps to delete any tenant.

1. Select the Tenant which you want to delete from the left panel.
2. Click "Delete" button. It shows the confirmation modal.

Figure: Confirmation Modal to Delete a Tenant

3. Click "OK" button to delete the Tenant. Else, click "Cancel" button to cancel the action.

9.3 Users Tab in MAdministrator

"Users" Tab allows The MAdministrator to view the details of all users. Perform the following steps to view the information of any user.

1. Select the user from the left panel of the users.

You can search for the users also.

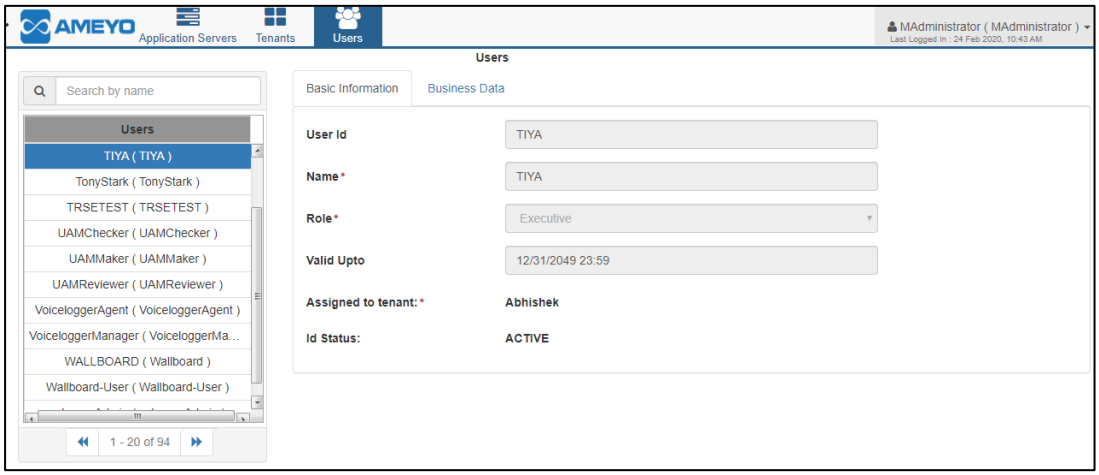


Figure: Basic Information tab of Users

2. Following details of the user are displayed in "Basic Details" tab.
    - UserId
    - Name
    - Role of the User
    - Date and Time up to which the user is active
    - Status of the user such as Active or Inactive.
3. Following details of the user are displayed in the Business Data tab.
    - Employee Code of that user.
    - Branch code of the user, if any.
    - Department code of the user.
    - Email Id of that user, if registered in the system.

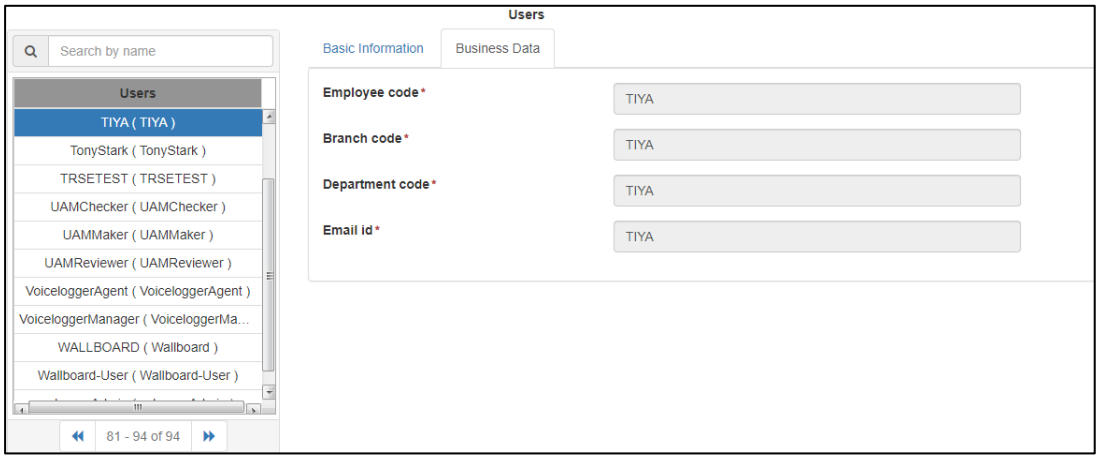


Figure: Business Data Information of User

9.4 MAdministrator Logout from Management Framework Architecture

Click the user account menu located on the top right corner.

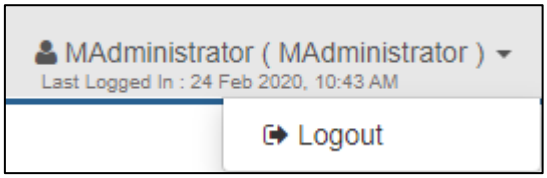


Figure: Logout from The MAdministrator Console

Click "Logout" to logout from The MAdministrator Console.

10. UAMMaker

The UAMMaker has the access to the Web-based Management Framework Architecture Console to manage and approve the users of all the Application Servers. Use the following URL to logon to the Management Framework Architecture Interface.

https://<Domain_Name>:<Management_UI_Port_Number>//managementserverui/NoPopUpIndex.html

The default port for Management UI is 8887, until and unless it hasn't been changed.

The following screenshot shows the logon page of the Management Framework Architecture for the UAMMaker.

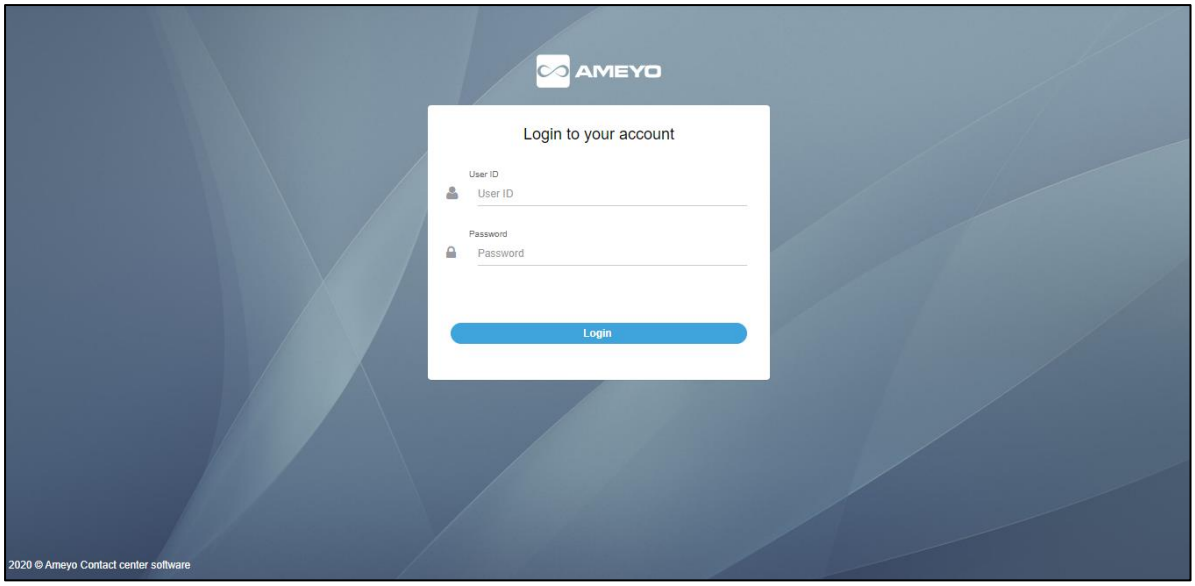


Figure: Logon page of Management Framework Architecture for UAMMaker

Enter User ID and Password and click "Login" button. The following page is displayed after the logon.

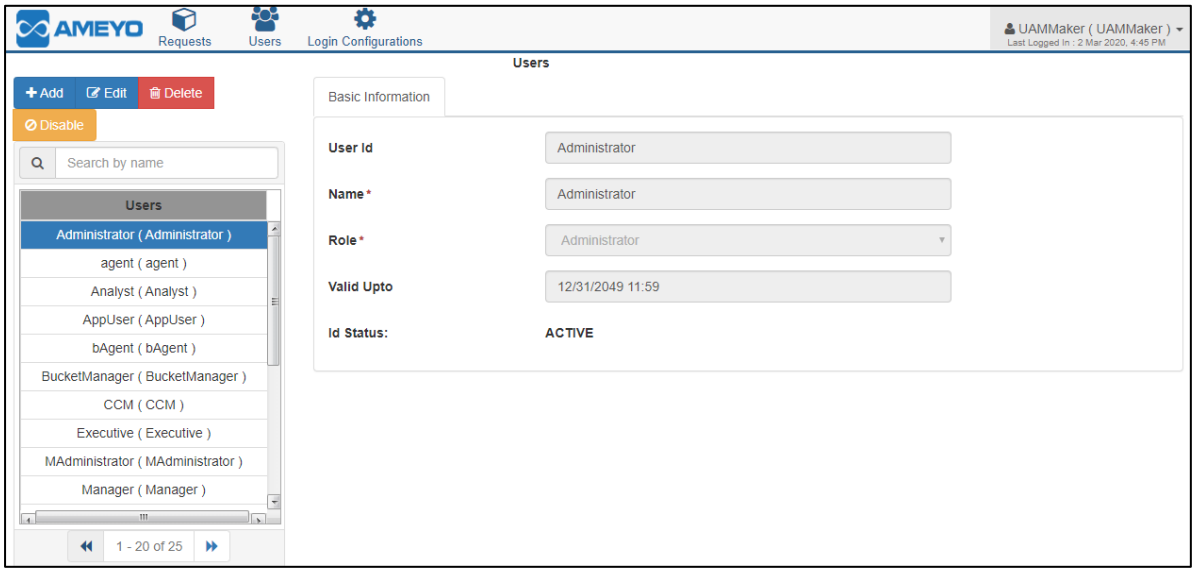


Figure: First Screen of The UAMMaker

The working of The UAMMaker can be described among the following menu.

- **Requests:** The UAMMaker can view all requests of the users from various Application Servers. From here, the UAMMaker can approve or reject the users. Know more...
- **Users:** The UAMMaker can manage the users here. Know more...
- **Login Configurations:** The UAMMaker can configure a few configurations of login and provide restrictions on them to the users. Know more...

10.1 Users Tab in UAMMaker

The default screen after the first login of The UAMMaker is of Users. The Users menu provides the privileges to The UAMMaker to manage users, which includes creation, edition, and deletion of users.

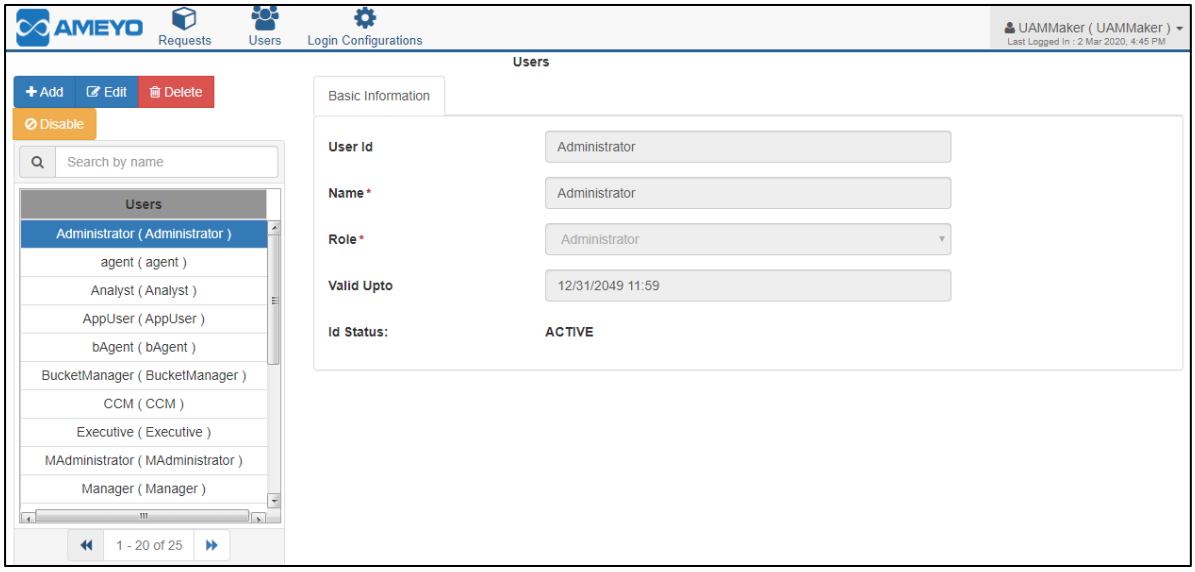


Figure: Users Menu of The UAMMaker

The left panel contains the list of all users. You can search for any user by typing its name in the search field. Select the user from the left panel to view the details of the user.

10.1.1 Add new User

Perform the following steps to add a new user:

1. Click "Add" button located at the top of the left panel. It shows the following modal.

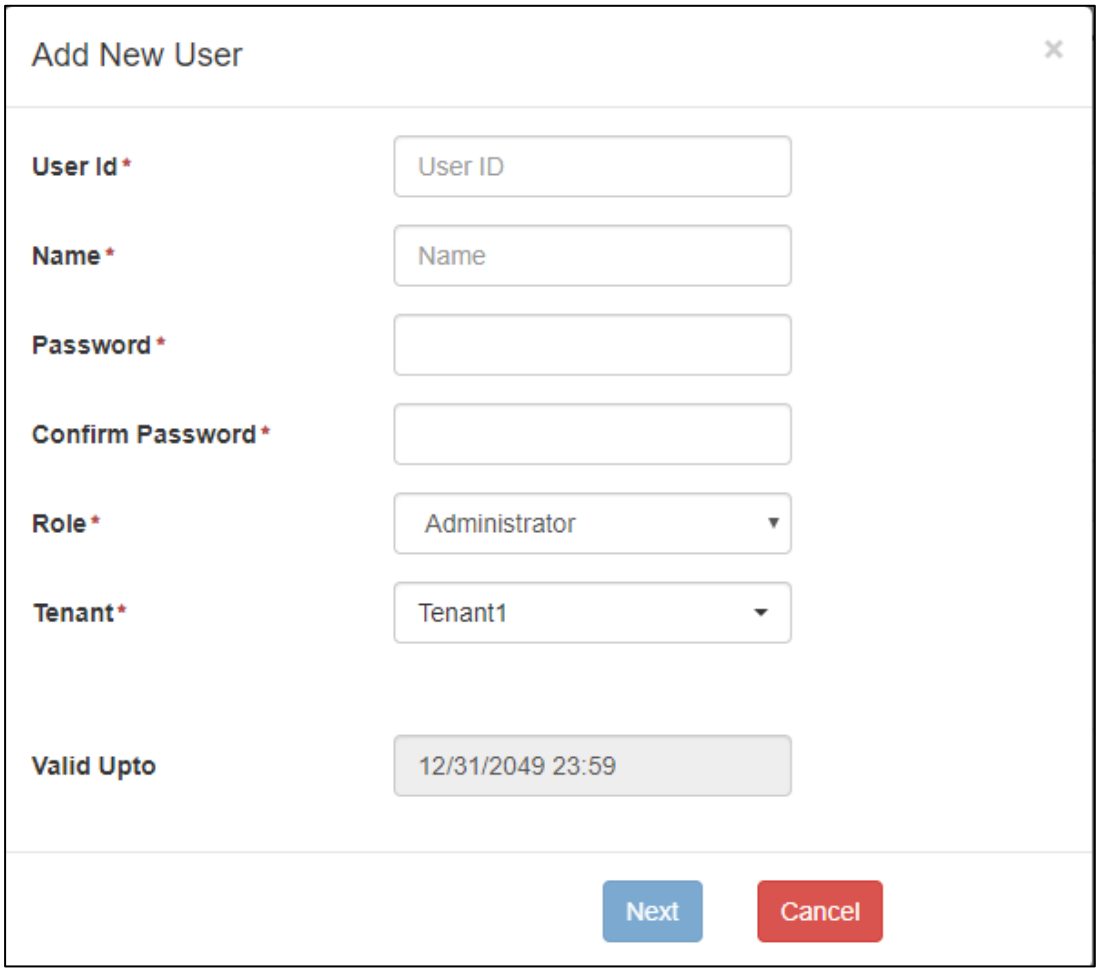


Figure: Modal to provide User's Information

2. Enter the User ID of the user in the User ID text field.
3. Provide the name of the user in Name text field.
4. Provide the password for that user. The user can change this password later.
5. Retype the same password to confirm in "Confirm Password" text field.
6. Select the role of the user from the drop-list of "Role".
7. Select the tenant name in which you want to assign the user from the drop-down list of Tenant.
8. Click "Next" button to proceed further. It shows the following page.

If "Business Fields" are not enabled from backend, then no Employee fields will be visible and hence, the user will not be able to provide the following details. To know more about, how to integrate "Business Fields",Click here...

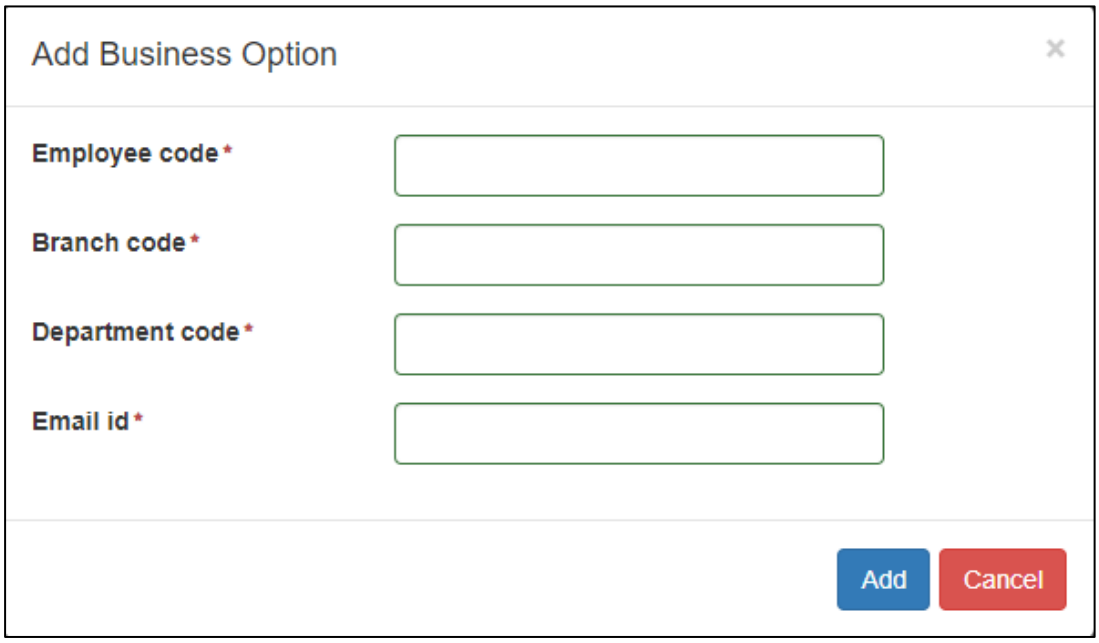


Figure: More information of Users

9. Here, enter "Employee code" in "Employee Code" text field.
10. Enter the branch code of the user in which the user is supposed to work in "Branch code" text field.
11. Enter the code of department of the user in "Department Code".
12. You can also provide the official Email ID of the user in "Email ID" text field.
13. Click "Add" button to add the user.

After the creation of the user, the approval request is sent to UAMChecker. After the approval of UAMChecker, the user can logon to Ameyo and can work.

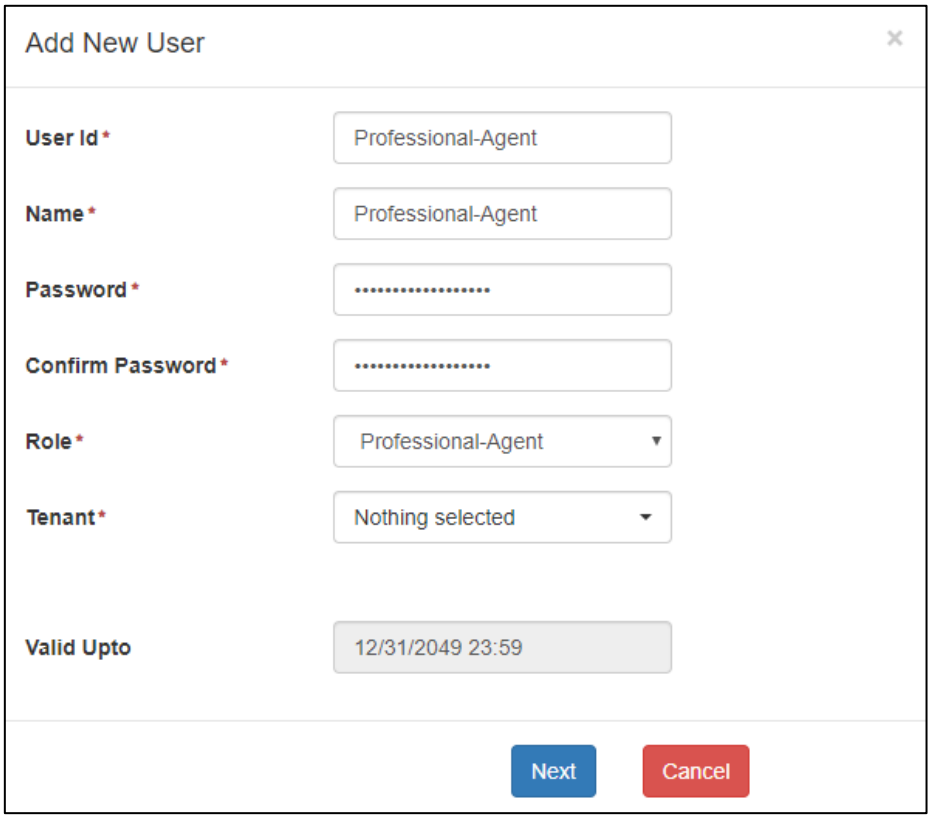


Figure: Sample Data for creating Users

10.1.2 Edit the User

Perform the following steps to edit an already existing user.

1. Select the user from the left panel, which you want to edit. All of the details of the user are now displayed in "Basic Information" tab of the users.

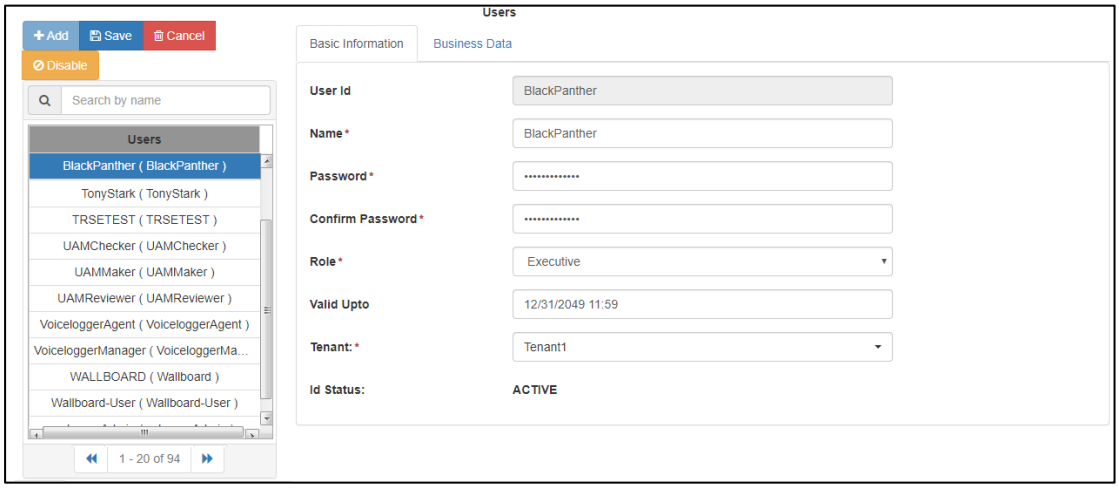


Figure: Edit Basic Information of User

2. Click "Edit" button present at the top of the left panel. Once you click the edit button, the details of the user are in editing mode.
3. You can change the following information of the user in the basic information tab.
    - User Id
    - Name
    - Password
    - Role
    - Tenant Assignment
4. Now, you can change the tab to Business Data to change the business data information of the user. You can edit the following details.
    - Employee Code
    - Branch Code
    - Department Code
    - Email Id

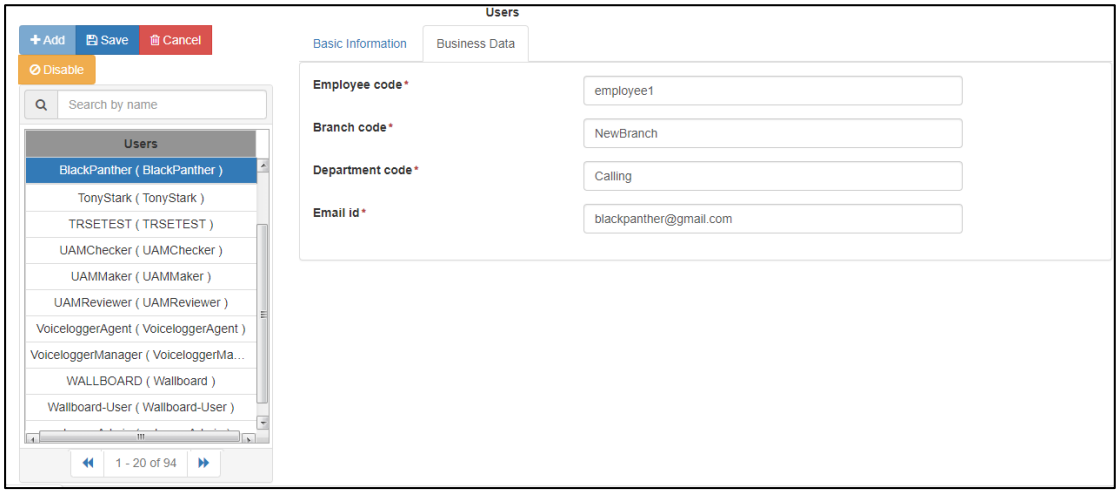


Figure: Edit Business Data Information of the user

5. Click "Save" button to save the changes.

10.1.3 Delete the User

Perform the following steps to delete the user.

1. Select the user which you want to delete.
2. Click "Delete" button present at the top of the left panel. It displays a confirmation modal.

Figure: Delete User Confirmation Modal

3. Click "OK" button to delete the user. Else click "Cancel" button.

10.1.4 Enable / Disable the user

The UAMMaker can enable or disable the user. Perform the following steps:

1. Select the user.
2. If the user is already enabled, then "Disable" is displayed on the top of the left panel. Else if the user is disabled, then "Enable" button is displayed on the top of the left panel.
3. Click "Enable" or "Disable" button according to the use case.
4. After clicking the enable or disable button, the request is sent for approval.

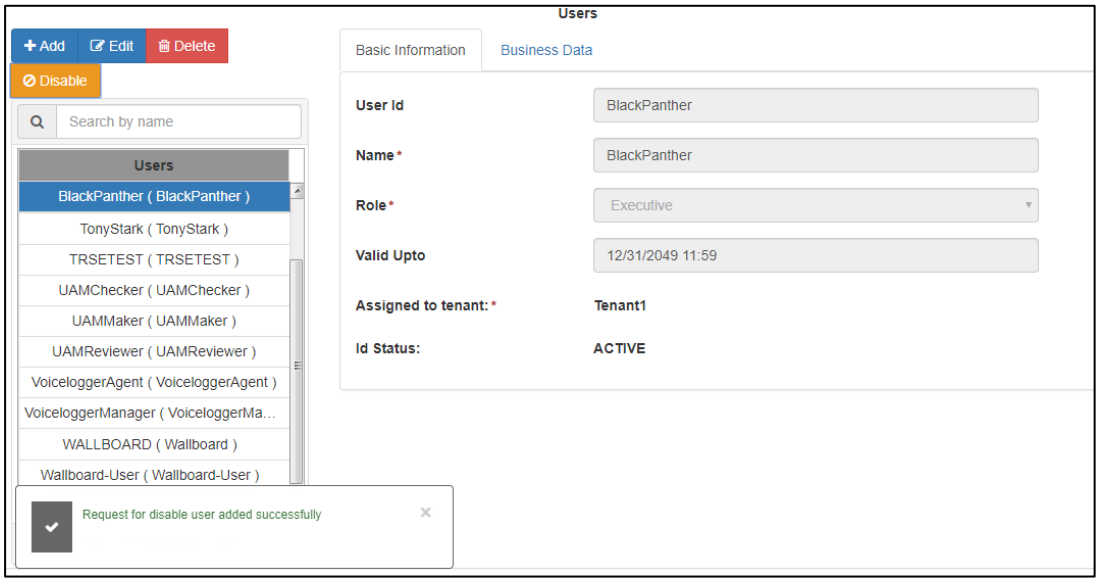


Figure: Disable the User

10.2 Requests Tab in UAMMaker

"Requests" menu shows the total number of Pending Requests raised by the UAMMaker. It contains the following tabs.

- **Pending Requests:** It shows the requests that are pending for approval. Know more...
- **History Queue:** It shows the status of the previously raised requests. Know more...

10.2.1 Pending Requests Tab in UAMMaker Interface

The UAMMaker can view all requests, made by the UAMMaker, that are pending for approval. All these changes require approval from the approver.

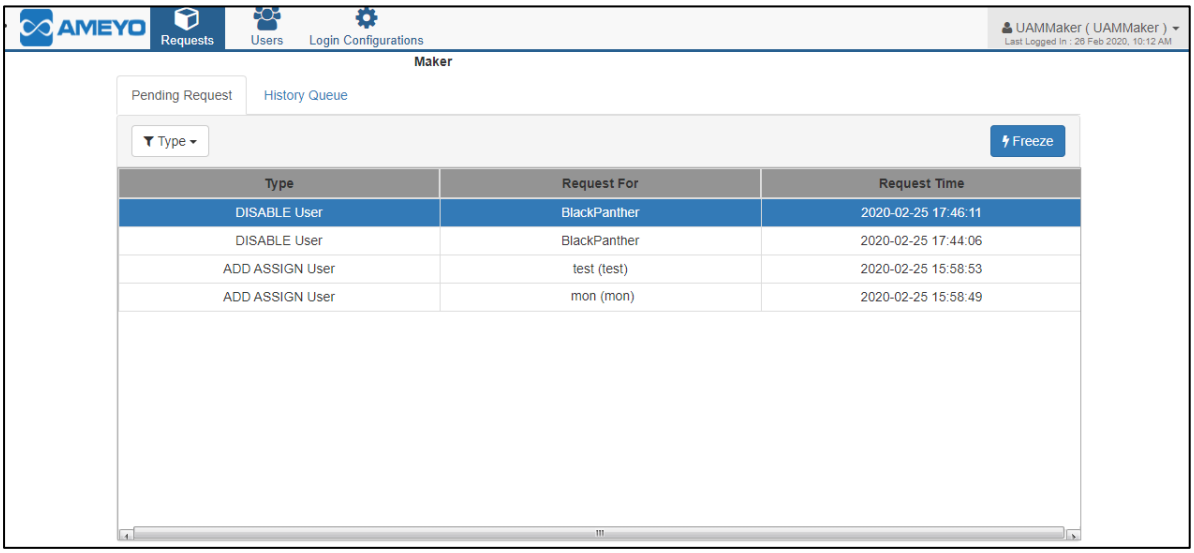


Figure: Pending Requests Tab

10.2.1.1 Filter

The UAMMaker can filter the requests. Select the filter type from "Type" drop-down menu.

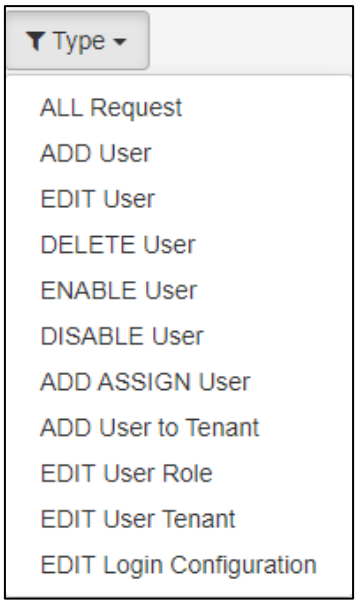


Figure: Filters

Following filters are available to filter the pending requests:

1. **All Request:** Click it to view all the pending requests.
2. **Add User:** Click it to view all the pending requests to add the users.
3. **Edit User:** Click it to view all the pending requests to edit the users.
4. **Delete User:** Select it to view the pending requests to delete the users.
5. **Enable User:** Select it to view the pending requests to enable the users.
6. **Disable User:** Select it to view the pending requests to disable the users.
7. **ADD ASSIGN User:** Select to view the pending requests to assign the users.
8. **Add user to Tenant:** Select it to view the pending requests to add the users to the tenants.
9. **Edit User Role:** Select it to view the pending requests to edit the user roles.
10. **Edit User Tenant:** Select it to view the pending requests to edit the user tenants.
11. **Edit Login Configuration:** Select it to view the pending requests to edit the login configuration.

10.2.1.2 Freeze

Click "Freeze" button to freeze the pending requests on the screen. It means that no more requests will be shown after freezing the screen.

10.2.1.3 Columns

"Pending Requests" tab contains the following columns.

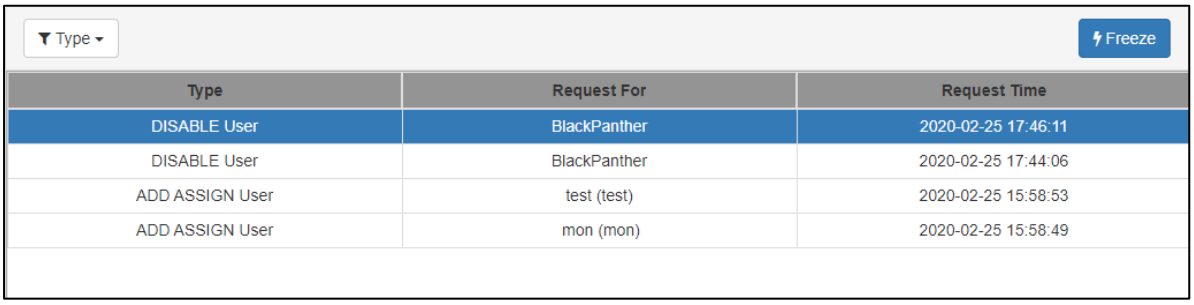


Figure: Columns of Pending Request Tab

1. **Type:** It shows the type of the request asked for the approval.
2. **Requested For:** It contains the name of the user for whom the request is made.
3. **Request Time:** It contains the date and time when The UAMMaker has requested.

10.2.2 History Queue Tab in UAMMaker Interface

All the previously raised requests by all users, which either are accepted or rejected, are displayed in "History Queue" tab.

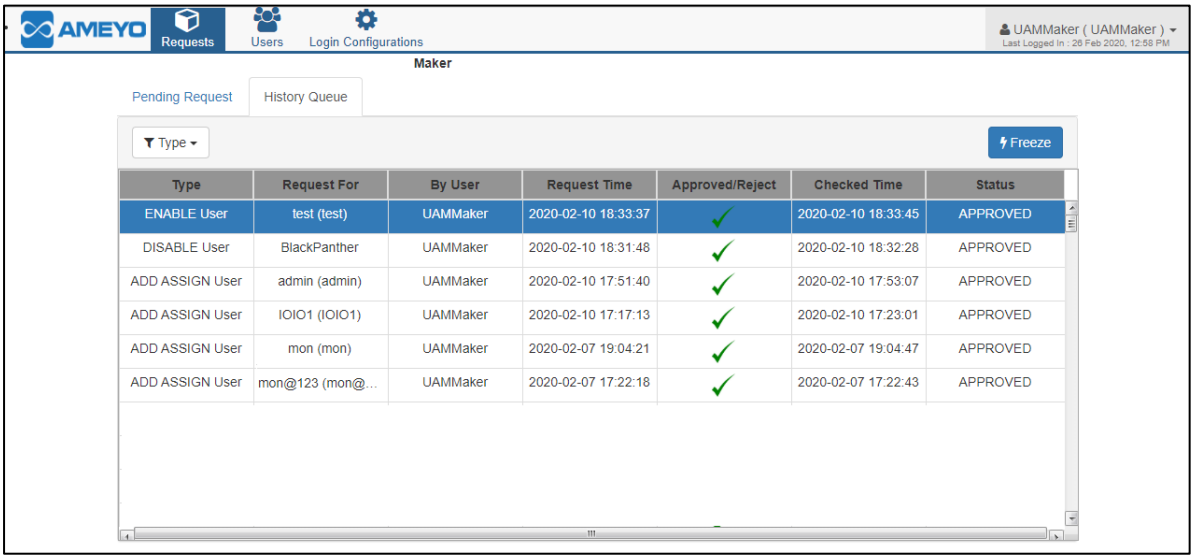


Figure: History Queue Tab of Requests

10.2.2.1 Filter

The UAMMaker can filter the history queue. Select the filter type from "Type" drop-down menu.

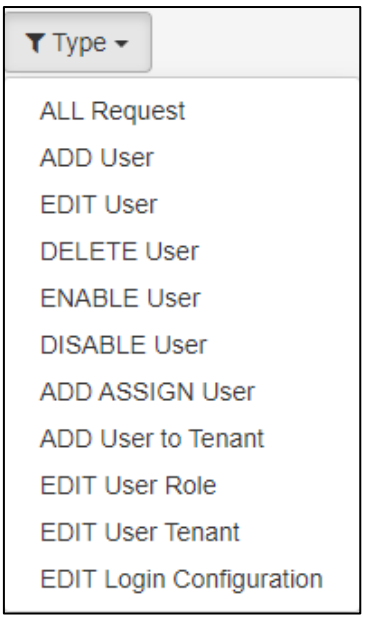


Figure: Filters

Following filters are available to filter the pending requests:

1. **All Request:** Click it to view all the pending requests.
2. **Add User:** Click it to view all the pending requests to add the users.
3. **Edit User:** Click it to view all the pending requests to edit the users.
4. **Delete User:** Select it to view the pending requests to delete the users.
5. **Enable User:** Select it to view the pending requests to enable the users.
6. **Disable User:** Select it to view the pending requests to disable the users.
7. **ADD ASSIGN User:** Select to view the pending requests to assign the users.
8. **Add user to Tenant:** Select it to view the pending requests to add the users to the tenants.
9. **Edit User Role:** Select it to view the pending requests to edit the user roles.
10. **Edit User Tenant:** Select it to view the pending requests to edit the user tenants.
11. **Edit Login Configuration:** Select it to view the pending requests to edit the login configuration.

10.2.2.2 Freeze

Click "Freeze" button to freeze the pending requests on the screen. It means that no more requests will be shown after freezing the screen.

10.2.2.3 Columns

History Queue contains the following columns.

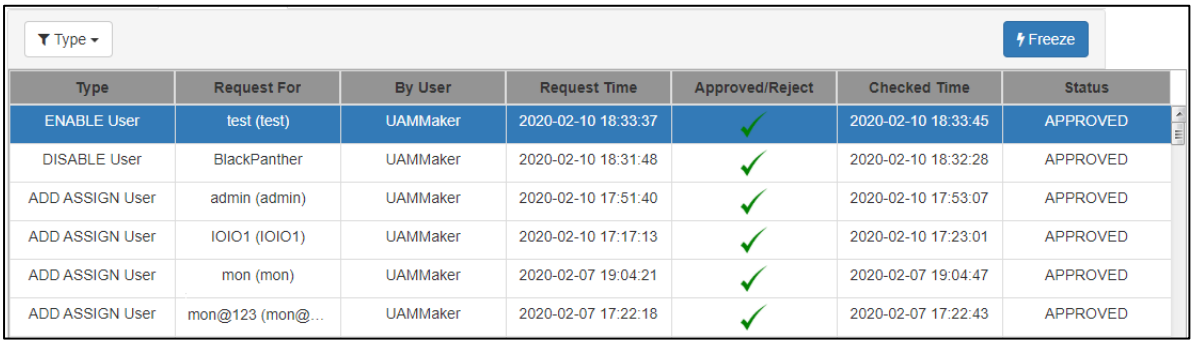


Figure: Columns of Pending Request Tab

1. **Type:** It shows the type of the request asked for the approval.

2. **Requested For:** It contains the name of the user for whom the request is made.
3. **By User:** It shows the name of the user who raised the request for approval.
4. **Request Time:** It contains the date and time when The UAMMaker has requested.
5. **Approved/Reject:** If the request is approved, then ✓ is displayed, and if the request is rejected, then ✗ icon is displayed. icon to approve the request for that user.

For the rejected requests, you can check the comments which are provided by the approver. Click ✗ icon, and the following modal with the comments is opened.

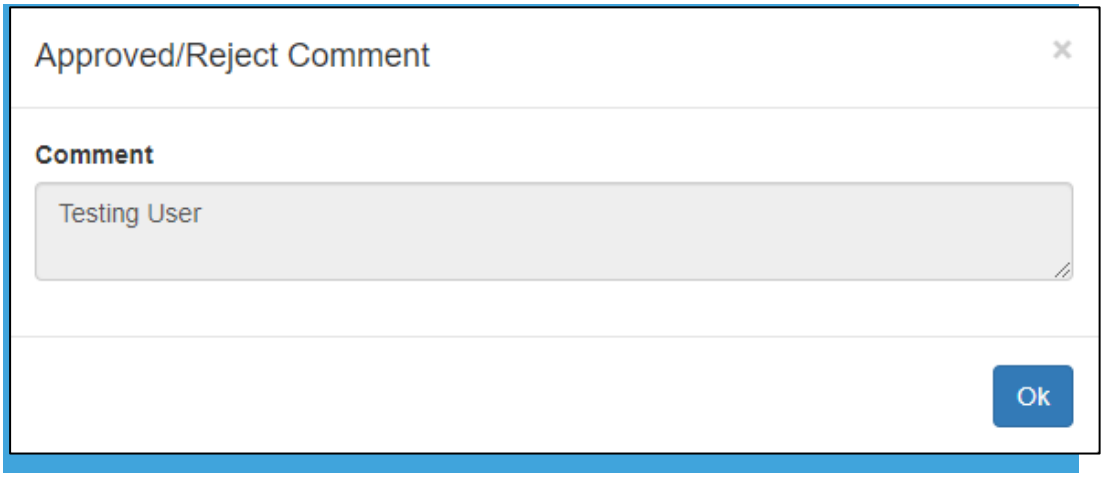


Figure: Rejected Comments

6. **Checked Time:** It contains the time at which the approver saw the request.
7. **Status:** It shows the status of the request, whether the request is approved or rejected.

10.3 UAMMaker Login Configurations

The Login Configurations tab provides the privileges to The UAMMaker to configure the login related settings.

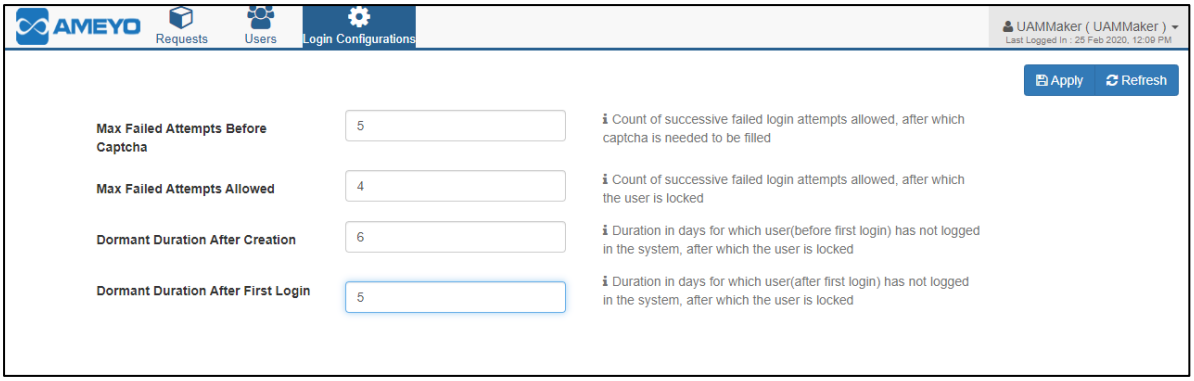


Figure: Login Configurations Menu

The UAMMaker can do the following configurations:

1. **Max Failed Attempts Before Captcha:** Enter the maximum number of attempts that the user can use to login before the captcha to authorize. It only works, if the captcha-based authentication is enabled in Ameyo.
2. **Max Failed Attempts Allowed:** It represents the total number of failed attempts that a user can try login. If the failed attempts exceeded this limit, then user account gets locked out and the user will not be able to login further.
3. **Dormant Duration After Creation:** Provide the duration after which the user accounts that are dormant since account creation will be locked. For example, if the provided value is 15 then a user that is dormant or inactive for 15 days after creation will be locked out.
4. **Dormant Duration After First Login:** Provide the duration after which the users accounts that are dormant since first logon will be locked. For example, if the provided value is 15 then a user that is dormant or inactive for 15 days after first logon will be locked out.

10.4 UAMMaker Logout from Management Framework Architecture

Click the user account menu located on the top right corner.

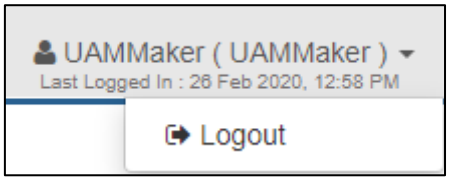


Figure: Logout from the UAMMaker Console

Click "Logout" to logout from the UAMMaker Console.

11. UAMChecker

The UAMChecker has the access to the Web-based Management Framework Architecture Console to manage and approve the users of all the Application Servers.

Use the following URL to logon to the Management Framework Architecture Interface.

https://<Domain_Name>:<Management_UI_Port_Number>//managementserverui/NoPopUpIndex.html

The default port for Management UI is 8887, until and unless it hasn't been changed.

The following screenshot shows the logon page of the Management Framework Architecture for The UAMChecker.

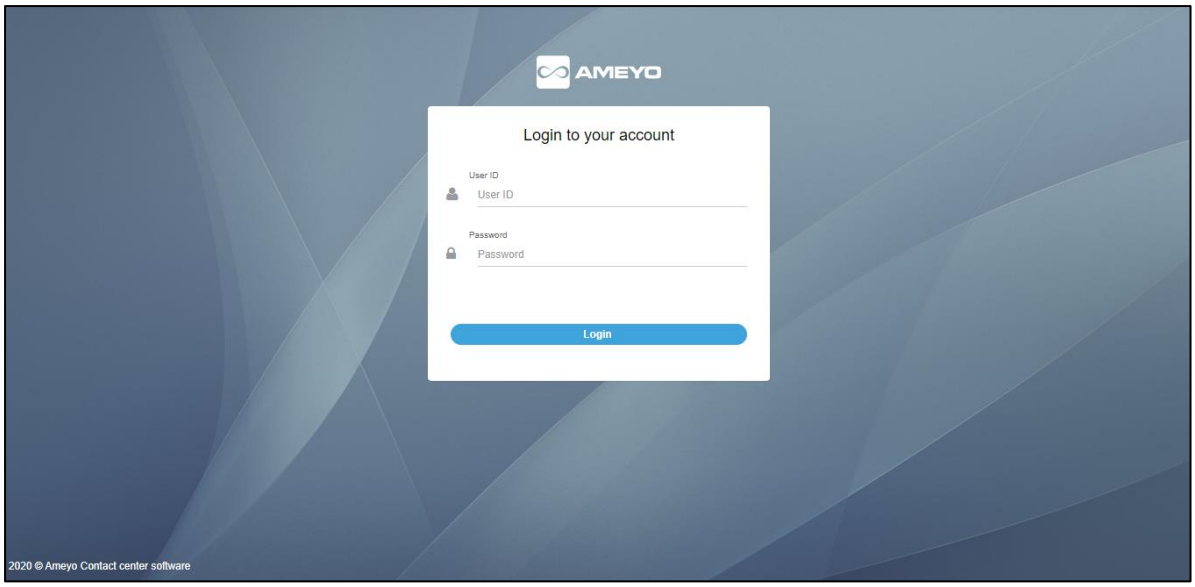


Figure: Login page of Management Framework Architecture for UAMChecker

Enter User ID and Password and click "Login" button. The following page is displayed after the logon.

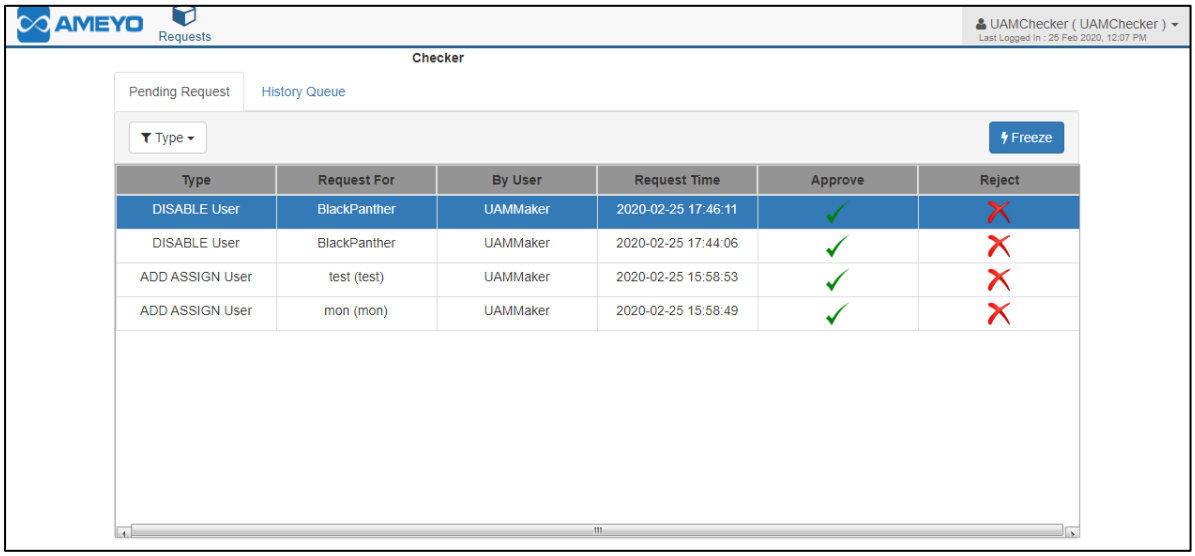


Figure: The UAMChecker Screen on Management Framework Architecture UI

The UAMChecker has the authority to approve or reject the request raised by other users. Any request that has been raised either to create, edit, or delete the user first comes to The UAMChecker. The UAMChecker can approve or reject the request. If the request is approved, then the changes requested are applied. However if the request is rejected, then the changes will not be applied, and hence these changes would not be reflected on the server.

The UAMChcker is able to perform the following operations through Management Framework Architecture.

- **Pending Requests:** The Pending Requests tab allows the UAMChecker to approve or reject the requests raised by other Ameyo users. Know more...
- **History Requests:** The UAMChecker is able to view all the previous requests on which the action has been taken.Know more...

11.1 Pending Requests Tab in UAMChecker Interface

The UAMChecker can view all the requests that are pending for approval.

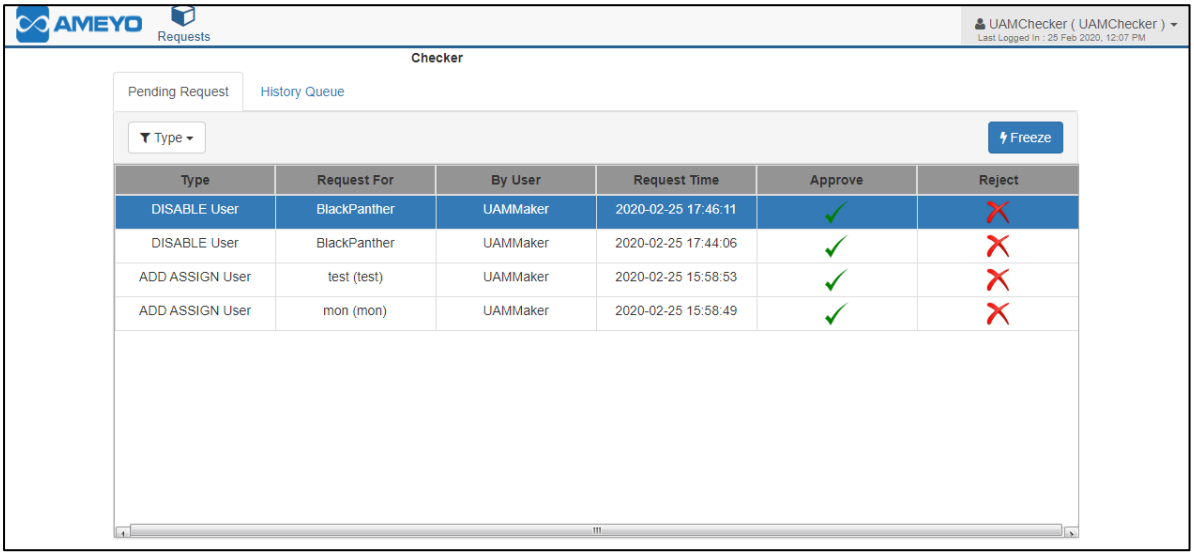


Figure: Pending Requests Tab

11.1.1 Filter

The UAMChecker can filter the requests. Select the filter type from "Type" drop-down menu.

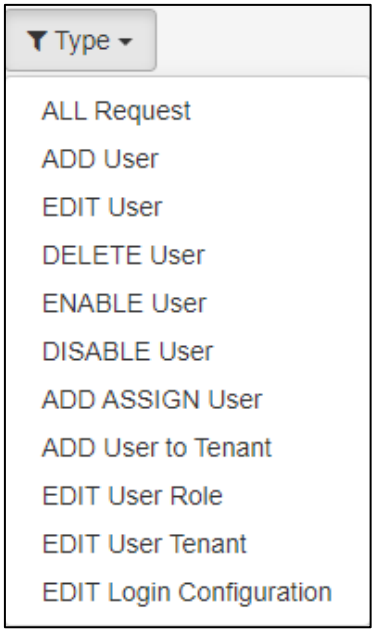


Figure: Filters

Following filters are available to filter the pending requests:

1. **All Request:** Click it to view all the pending requests.
2. **Add User:** Click it to view all the pending requests to add the users.
3. **Edit User:** Click it to view all the pending requests to edit the users.
4. **Delete User:** Select it to view the pending requests to delete the users.
5. **Enable User:** Select it to view the pending requests to enable the users.
6. **Disable User:** Select it to view the pending requests to disable the users.
7. **ADD ASSIGN User:** Select to view the pending requests to assign the users.
8. **Add user to Tenant:** Select it to view the pending requests to add the users to the tenants.
9. **Edit User Role:** Select it to view the pending requests to edit the user roles.
10. **Edit User Tenant:** Select it to view the pending requests to edit the user tenants.
11. **Edit Login Configuration:** Select it to view the pending requests to edit the login configuration.

11.1.2 Freeze

Click "Freeze" button to freeze the pending requests on the screen. It means that no more requests will be shown after freezing the screen.

11.1.3 Approve the Request

Perform the following steps to approve any raised request.

1. Select the row of the user for whom you want to approve the request.
2. Click  icon to approve the requests. he following notification is displayed at the bottom of the page to confirm the same.

Figure: Notification after Approving the Request

11.1.4 Reject the Request

Perform the following steps to reject any raised request.

1. Select the row of the user for whom you want to reject the request.
2. Click  icon. A confirmation modal is displayed.

Figure: Comments Modal

3. Provide the reason due to which you have rejected the request.
4. Click "Reject" button to reject the request. The following notification is displayed at the bottom of the page to confirm the same.

Figure: Notification after Rejecting the Request

Once the request is approved or rejected, then that request will not be displayed in the Pending Requests tab.

11.1.5 Columns

"Pending Requests" contains the following columns.

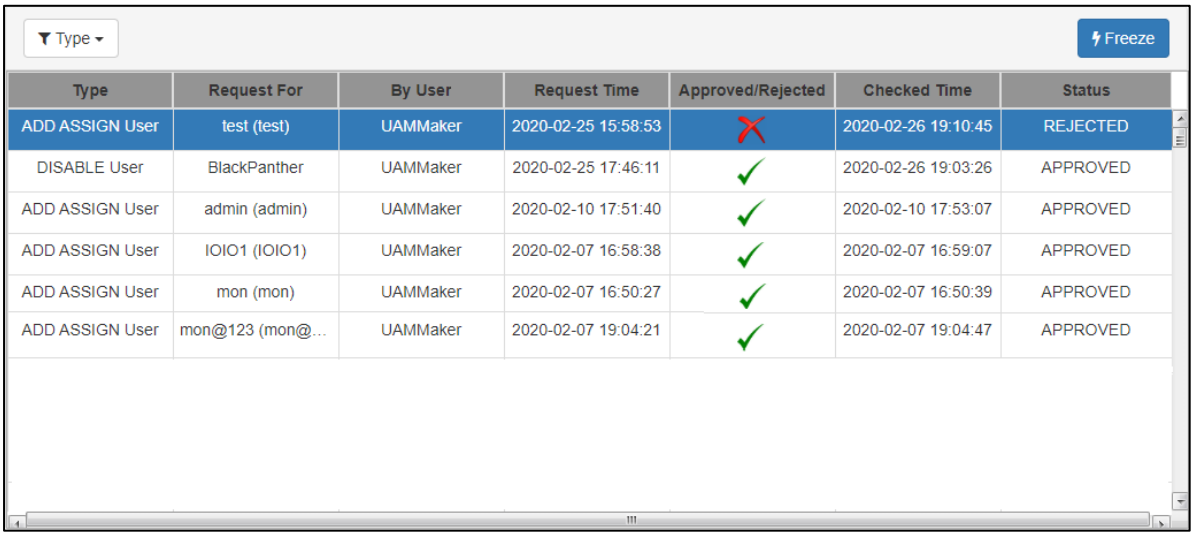


Figure: Columns of Pending Request Tab

1. **Type:** It shows the type of the request asked for the approval.
2. **Requested For:** It contains the name of the user for whom the request is made.
3. **By User:** It shows the name of the user who raised the request for approval.
4. **Request Time:** It contains the date and time when The UAMMaker has requested.
5. **Approve:** Click  icon to approve the request for that user.
6. **Reject:** Click  icon to reject the request for that user.

11.2 History Queue Tab in UAMChecker Interface

All the previously raised requests by other users, which either are accepted or rejected, are displayed in "History Queue".

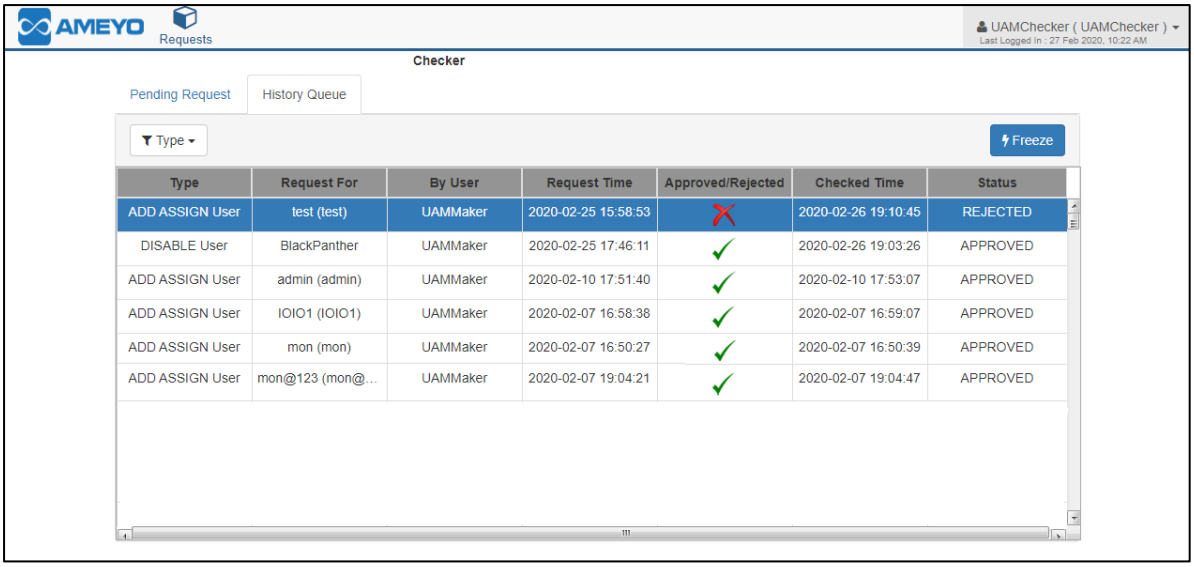


Figure: History Queue Tab of Requests

11.2.1 Filter

The UAMChecker can filter the history queue. Select the filter type from "Type" drop-down menu.

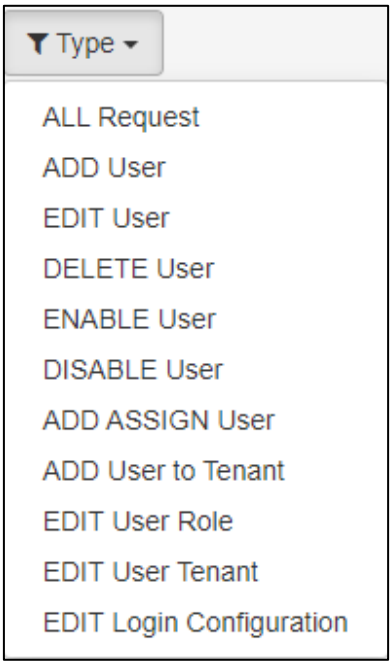


Figure: Filters

Following filters are available to filter the pending requests:

1. **All Request:** Click it to view all the pending requests.
2. **Add User:** Click it to view all the pending requests to add the users.
3. **Edit User:** Click it to view all the pending requests to edit the users.
4. **Delete User:** Select it to view the pending requests to delete the users.
5. **Enable User:** Select it to view the pending requests to enable the users.
6. **Disable User:** Select it to view the pending requests to disable the users.
7. **ADD ASSIGN User:** Select to view the pending requests to assign the users.
8. **Add user to Tenant:** Select it to view the pending requests to add the users to the tenants.
9. **Edit User Role:** Select it to view the pending requests to edit the user roles.
10. **Edit User Tenant:** Select it to view the pending requests to edit the user tenants.
11. **Edit Login Configuration:** Select it to view the pending requests to edit the login configuration.

11.2.2 Freeze

Click "Freeze" button to freeze the pending requests on the screen. It means that no more requests will be shown after freezing the screen.

11.2.3 Columns

History Queue contains the following columns.

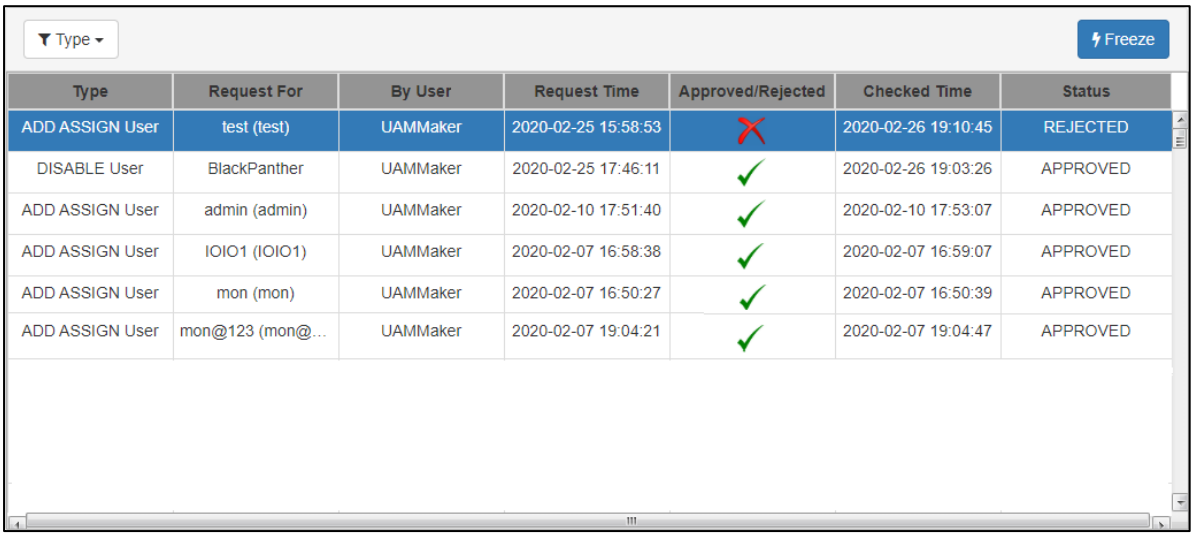


Figure: Columns of Pending Request Tab

1. **Type:** It shows the type of the request asked for the approval.
2. **Requested For:** It contains the name of the user for whom the request is made.
3. **By User:** It shows the name of the user who raised the request for approval.
4. **Request Time:** It contains the date and time when The UAMMaker has requested.
5. **Approve:** Click  icon to approve the request for that user.
6. **Reject:** Click  icon to reject the request for that user.

For the rejected requests, you can check the comments which are provided by the approver.

Click  icon, and the following modal with the comments is opened.

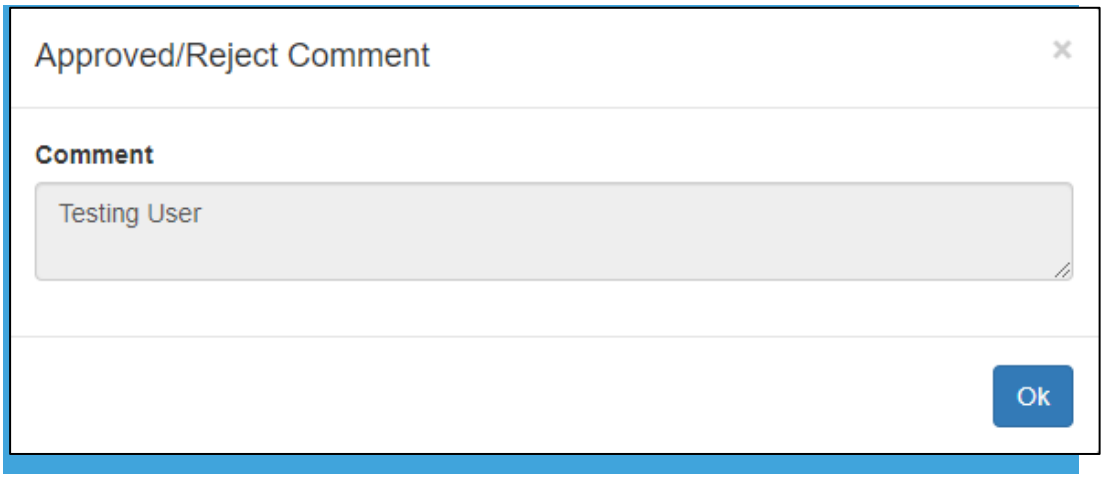


Figure: Rejected Comments

7. **Checked Time:** It contains the time at which the approver saw the request.
8. **Status:** It shows the status of the request, whether the request is approved or rejected.

11.3 UAMChecker Logout from Management Framework Architecture

Click the user account menu located on the top right corner.

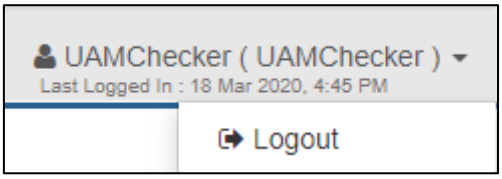


Figure: Logout from The UAMChecker Console

Click "Logout" to logout from The UAMChecker Console.

12. Frequently Asked Questions

Following are some of the Frequently Asked Questions about Management Framework Architecture:

1. **Is it possible to configure multiple versions of the Application Server? How will it work with one version of the Management Framework Architecture?**

   There is a possibility that any organization has multiple Application Servers across its multiple branches. In such cases, a question is raised that whether all these different versions of Application Servers can work with a single Management Framework Architecture. So, the answer is "No". As of now, it is not possible that a single Management Framework Architecture can manage multiple Application Servers with different versions. Management Framework Architecture is compatible with Ameyo Server 4.7 and later versions.

2. **How will the activities like updates, upgrades or other activities be performed on the Management Framework Architecture?**

   Any activity such as updating Management Framework Architecture, upgrading its services, or any other management related activities can be performed on it with usual steps. These activities have a similar process as that of the Application Server. For any such activity that requires stopping the Management Framework Architecture application, you have to stop the Management Server application, and then you can perform above such activities. For the activities which do not require stopping the Management Framework Architecture, then you can perform those activities without stopping Management Framework Architecture.

3. **What if the Management Framework Architecture stopped working?**

   If, Management Framework Architecture stopped working or someone stopped it for any activity, then some services of the Application Server will also stop working. Know more...

4. **What are some of the basic commands to manage the Management Server application?**

Following are some of the basic commands to manage the Management Framework Architecture services:

- Command to install the Management Server

  ```
  rpm -ivh <management_server>.rpm
  ```

- Command to update the Management Server

  ```
  rpm -Uvh <management_server>.rpm
  ```

- Command to stop the Management Server

  ```
  ameyoctl service ameyomanagementserver stop
  ```

- Command to start the Management Server

  ```
  ameyoctl service ameyomanagementserver start
  ```

- Command to check the status of the Management Server

  ```
  ameyoctl service ameyomanagementserver status
  ```

- Location of log files of the Management Server start script

  ```
  /dacx/var/ameyo/dacxdata/log/scripts/AmeyoManagementServerStart.log
  ```

- Location of log files for other Management Server logs

  ```
  /dacx/var/ameyo/dacxdata/ameyo.management.server.product/logs/<Dated_Folder>/ameyo_server.log
  ```

5. **Is the configuration of the Management Framework Architecture One-time setup?**

   Yes, once the Management Framework Architecture is enabled on Application Servers, then the user has only to monitor the Management Server.

6. **Can we deploy a Management Framework Architecture with already running Application Server setups?**

   Yes, it is possible to configure the Management Framework Architecture with already running Application Servers. As of now, the configurational steps are manual.

7. **How and what procedure has to be followed to create the Application Servers, if Management Server is integrated with already running Ameyo?**

There is a possibility that Management Framework Architecture is integrating with already running Application Server. In such case, the user can create Application Server from Management Framework Architecture Interface, like a normal process. Know more...

8. **When deploying Management Framework Architecture on already running Application Server, then what happen with the already created Tenants?**

   If Management Framework Architecture is integrated with already running Application server, then the present Application Servers must have the created tenants previously. For those tenants, the user has to sync those tenants manually to Management Framework Architecture. Know more...

9. **Is the Management Framework Architecture being compatible with the VAPT setup?**

   Yes, the Management Framework Architecture from 4.7 GA version onwards can be configured for Application Servers integrated with VAPT setups.

10. **What is the difference between Management Server(MS) and Management Framework Architecture (MFA)?**

    MFA is the architecture of the Management Server. Management Server is a service that performs centrally user creation, user authentication, tenant creation, and synchronizing of users, call servers, and tenants from the Management Framework Architecture to the Application Server and Application Servers to a Management Server. Know more...

11. **What is the difference between CRM and WC_crm?**

    CRM is a third-party service that is used to store the information about Customers, and through which Ameyo integrates its services and provides the feature of calling. For example: Fresh Desk Mint, Microsoft Dynamics, and so on.

    Whereas, WC_crm is a database through which the customer sends its data to Ameyo with Push and Pull techniques.

12. **Is it possible to disable the Management Framework Architecture and work only with Application servers?**

    Yes, it is possible to disable the Management Framework Architecture and to work with the Application Server, though it is not recommended. As, disabling Management

Framework Architecture results in disabling the services provided by it, and hence, the possibility for issues can be raised. Therefore, it is not recommended to disable the Management Framework Architecture.

13. How to disable the Management Framework Architecture and allow Application Servers to work separately?

Perform the following steps to disable Management Framework Architecture:

- Execute the following command to login to the database:

  ```
  psql -U postgres <Application_Server_Database_Name>
  ```

- Run the following queries to delete the commands that provide privileges to the Application Server to authenticate from the Management Server.

  ```
  Delete from system_configuration_parameter where value = 'auth.type.ms';
  ```

- Run the following query to delete system mode entry from the Application Server.

  ```
  DELETE from system_configuration_parameter where value ='ms';
  ```

- Run the following query to delete "Roo"" and "PowerUser" entry from Application Server.

  ```
  Delete from system_configuration_parameter where name = 'override.authentication.scheme';
  ```

- Now, run the following command to exit from the database.

  ```
  \q
  ```