# Emoveo Version 2.8.4

## PPMetrics – Installation and Administration

Document Version: 2.4
Document Release Date: February 2026

# Table of Contents

# 1. Introduction

This document provides information about installation and administration of the Emoveo application. It was written for:

- Emoveo administrators
- Emoveo technical support personnel

# 2. Solution Architecture

The high-level architecture diagram below depicts the main components of the Emoveo solution and their communication with other components.

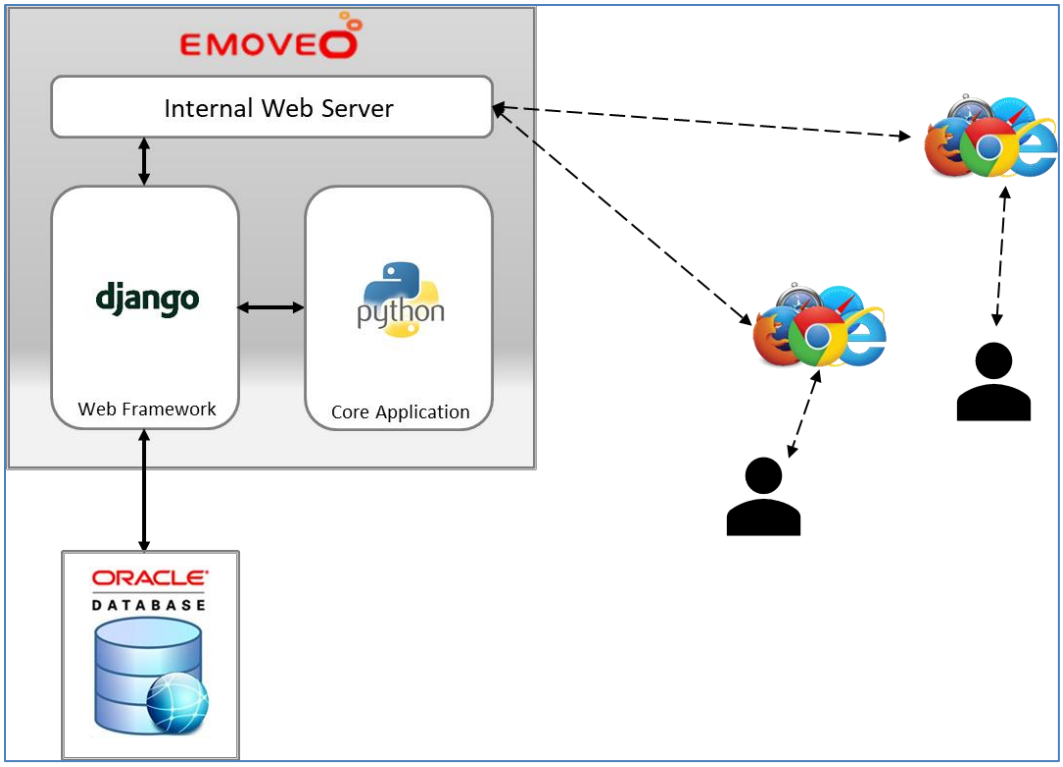


*Fig. 1 – Emoveo solution architecture*

The application tier of the solution consists of the following components:

- The core application is written in Python and relies upon the Django Web Framework. Django and all other required components are already included and configured by the installer.
- The application sets up and uses its own Python virtual environment, so it is effectively isolated from any other Python site packages already in the system.
- By default, an internal Python web server is used to deploy the application. For high performance scenarios, it is recommended to use an external web server. Any web server supported by Django 2.2 may be used, such as nginx, gunicorn or Apache.

The database tier of the solution is comprised of:

- A standard Oracle database 11G, 12C, or higher that is used to store the Emoveo data and configurations.

➤ **The database tier may be a separate server or can be on the same server as the application tier.**

# 3. Pre-Installation Tasks

This section outlines all the pre-requisite execution steps and information that should be gathered prior to beginning the installation and configuration of Emoveo.

- **It is recommended to install Emoveo into a brand-new operating system user account. It is not recommended to install Emoveo as root, however sudo permissions will be required if Python and the pre-requisite modules are not already installed.**

| Task | Details |
|---|---|
| **Verify system compatibility** | Please refer to the section “System Requirements and Compatibility Matrix” below for details |
| **Download and install Python** | Download the latest current stable version of Python. At the time of this document’s release date, we recommend that you use Python 3.9.x. *For illustration purposes, this document may contain references to Python version 3.7 in sample commands and script output.*<br><br>To verify your Python version, run:<br><br>`python3 -V`<br><br>If Python is not installed, see the following examples of how to install and compile Python as well as pre-requisite modules by downloading them from the web. If Internet access is limited, download the following files separately and copy to the server first. If wget is installed on the server you can download them as shown:<br><br>`wget https://www.python.org/ftp/python/3.7.2/Python-3.7.2.tgz`<br><br>You will need to have a development toolchain installed to compile Python, including optional SSL support. To ensure this support is available, install the SSL libraries for your system before installing Python. For example, in Red Hat Enterprise Linux (RHEL) you would use:<br><br>`sudo yum install openssl-devel`<br><br>Then, you can install the pre-requisites by running these commands on the directory where the files are copied (sudo permissions will be required for the last step only). Make sure you have installed zlib-devel installed on your system, or run the following (for RHEL). Please note the root or sudo su privileges to root need to be available to install “Development tools”<br><br>`yum groupinstall "Development tools"`<br><br>`yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel` |

| | |
|---|---|
| | `gdbm-devel db4-devel libpcap-devel xz-devel libffi-devel`<br><br>Now you can continue with python installation (root privileges not required except for the last step):<br><br>`tar xvfz Python-3.7.2.tgz`<br><br>`cd Python-3.7.2`<br><br>`./configure --prefix=/usr/local LDFLAGS="-Wl,-rpath /usr/local/lib"  --enable-shared`<br><br>`make`<br><br>`sudo make altinstall`<br><br>The above process will avoid overwriting the existing Python installation. |
| **Install Cryptography package in Python** | Assuming there is internet access, we can install Cryptography using the pip manager:<br><br>`sudo pip3 install cryptography`<br><br>If this does not work, please upgrade your pip first, as that is the single most common cause of installation problems:<br><br>`sudo pip3 install --upgrade pip`<br><br>To verify installation of the Cryptography package, run:<br><br>`python3 -m pip show cryptography` |
| **Download and install Oracle Client** | The Oracle Client 11.2 or higher is required.<br><br>The following instructions are for Red Hat Enterprise Linux. Before installing the RPMs, check the official Oracle documentation to make sure you have all the pre-requisites installed. The same link contains information about installation in other operating systems.<br><br>https://docs.oracle.com/database/121/LADBI/pre_install.htm#LADBI7534<br><br>If additional libraries such as libaio are required by the client in your system, you can install them like this:<br><br>`sudo yum install libaio`<br><br>In the Emoveo distribution the RPMs needed for Oracle 12.1 are provided in the install/pre/ directory, if required, you can download and copy to the server the **basic**, **sqlplus** and **devel** RPMs from this page (according with the desired version of Oracle): |

| | |
|---|---|
| | https://oracle.github.io/odpi/doc/installation.html#oracle-instant-client-rpm Install the RPMs using:<br><br>`sudo rpm -ivh install/pre/oracle-instantclient*-basic-*.x86_64.rpm`<br><br>`sudo rpm -ivh install/pre/oracle-instantclient*-devel-*.x86_64.rpm`<br><br>`sudo rpm -ivh install/pre/oracle-instantclient*-sqlplus-*.x86_64.rpm`<br><br>Append the following entry into the .bash_profile file for the install user<br><br>`export ORACLE_HOME=/usr/lib/oracle/12.1/client64`<br>`export PATH=$ORACLE_HOME/bin:$PATH`<br>`export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib:$LD_LIBRARY_PATH`<br><br>➤ **The Oracle client library "libclntsh*.so" should be in the ORACLE_HOME variable.** |
| **Available Oracle database for Emoveo** | Please refer to the section "System Requirements and Compatibility Matrix" below for details |

*Table 1 – Pre-installation tasks*

# 4. Installation

In order to install Emoveo, a systems administrator with access to the application server and a DBA with access to the database will need to execute the steps outlined in the sections below.

## *4.1. Creating Oracle Schema*

Please have the DBA create a new Oracle schema for Emoveo with these permissions:

- Login (Create Session + Write on Tablespace);
- Create Tables, DB Links, Sequences and Triggers.

The following script is an example, you may replace "*emoveo*" with any other valid schema name:

```
DROP USER emoveo CASCADE;
CREATE USER emoveo IDENTIFIED BY emoveo;

GRANT CREATE SESSION TO emoveo;
GRANT UNLIMITED TABLESPACE TO emoveo;
GRANT CREATE TABLE TO emoveo;
GRANT CREATE DATABASE LINK TO emoveo;
GRANT CREATE SEQUENCE TO emoveo;
GRANT CREATE TRIGGER TO emoveo;
```

Please have the DBA record the following information for use during the application install.

- Obtain Oracle Service Name or SID
- Oracle Hostname and Port
- Oracle Schema Username and Password

## *4.2. Installing Emoveo*

1. Copy and extract the Emoveo distribution files to the directory where you want to install the application. In this example, we will assume a directory named ‘emoveo.’

   ```
   tar xvfz emoveo-*.tar.gz
   cd emoveo
   ```

2. Execute the configure.py script

   ```
   python3 configure.py
   ```

3. The `configure.py` script will provide a menu where you can enter settings that will then generate configuration files for Emoveo, these include the `local_settings.py` and the `database.conf` files under the folder /Emoveo. Be sure to update both the database connections and application settings. You should have available your schema information from the prior ‘Creating Oracle Schema’ section. Here is a sample of the `configure.py` script:

   ```
   Choose a set of options to configure:
   ```

```
1. Database connection.
2. Application settings.
0. Exit configuration tool.
>> 1
Provide information about your Oracle database.
DB Hostname: orcl.c74zpz4vi62p.us-east-2.rds.amazonaws.com
DB Port (default 1521):
DB Service Name (leave blank to use SID):
DB SID: orcl
DB User: emoveoinst
DB Password: emoveoinst
Database configuration saved to: /opt/emoveo/emoveo/database.conf

Choose a set of options to configure:
1. Database connection.
2. Application settings.
0. Exit configuration tool.
>> 2
Warning: These settings are already configured.
To prevent data loss, backup /opt/emoveo/emoveo/local_settings.py
before proceeding.
You may also modify that file directly.
Please answer the following application questions:
Base URL (include port, default localhost:8080): ec2-35-205-157-
57.compute-1.amazonaws.com:8080
SMTP host for notifications:
SMTP port:
SMTP email user (example: emoveo@example.com):
SMTP password for user:
SMTP default "from" email (default is same as user):
Choose an encryption service for SMTP:
1. SSL.
2. TLS.
0. None (default).
>>
Using unencrypted email.
Application configuration saved to:
/opt/emoveo/emoveo/local_settings.py

Choose a set of options to configure:
1. Database connection.
2. Application settings.
0. Exit configuration tool.
>> 0
Terminating configuration script.
```

4. After the required configuration files have been generated, run the primary installation script.

   ```
   python3 install.py
   ```

   This will perform all the remaining steps to complete the installation:

   - Creating the virtual environment (only if it doesn't exist, to overwrite it use --clean).

- Installing all the packages (wheel files) required by Emoveo (if one of the included wheels is not compatible with your architecture or OS, you can use the --allow-index option to download other versions from the internet).
- Validating the folder structure (for ex. log and media directories).
- Running database migrations (this creates/updates all the database schema objects)
- Installing seed data (Out of the Box Object Types, Custom Field Groups, Workflows and Environments).
- Collecting and compress static assets (speeds up web page access)

If any exceptions occur during the install, you can try utilizing PyPI (Python Package Index) to install any missing dependencies from the standard internet repository. Utilize the following command with the –clean parameter to overwrite any virtual environment that might have been created:

```
python3 install.py --pypi --clean
```

Here is sample output of the `install.py` script:

```
[ec2-user@emoveo]$ python3 install.py
Found virtual environment: /opt/emoveo/env
Checking for valid pip installation...
Requirement already up-to-date: pip in ./env/lib/python3.7/site-
packages
Installing wheels using requirements.txt...
Looking in links: /opt/emoveo/install/wheels
Requirement already satisfied: pip>=18.0.0 in ./env/lib/python3.7/site-
packages (from -r requirements.txt (line 1)) (19.0.2)
Requirement already satisfied: setuptools>=20.3.1 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 2))
(40.6.2)
Requirement already satisfied: Django==1.11.20 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 3))
(1.11.20)
Requirement already satisfied: django-debug-toolbar>=1.4 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 4))
(1.11)
Requirement already satisfied: django-extensions>=1.6.1 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 5))
(2.1.5)
Requirement already satisfied: django-tables2>=1.1.2 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 6))
(2.0.4)
Requirement already satisfied: py>=1.4.31 in ./env/lib/python3.7/site-
packages (from -r requirements.txt (line 7)) (1.7.0)
Requirement already satisfied: pyasn1>=0.1.9 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 8))
(0.4.5)
Requirement already satisfied: pytest>=2.9.0 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 9))
(4.2.1)
Requirement already satisfied: Django-Select2>=5.8.4 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 10))
(6.3.1)
```

```
Requirement already satisfied: wheel>=0.29.0 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 11))
(0.33.0)
Requirement already satisfied: django-tabbed-admin>=1.0.4 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 12))
(1.0.4)
Requirement already satisfied: django-jquery>=3.1.0 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 13))
(3.1.0)
Requirement already satisfied: rsa>=3.3 in ./env/lib/python3.7/site-
packages (from -r requirements.txt (line 14)) (4.0)
Requirement already satisfied: six>=1.10.0 in ./env/lib/python3.7/site-
packages (from -r requirements.txt (line 15)) (1.12.0)
Requirement already satisfied: sqlparse>=0.1.19 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 16))
(0.2.4)
Requirement already satisfied: pycryptodome>=3.7.3 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 17))
(3.7.3)
Requirement already satisfied: ecdsa>=0.13 in ./env/lib/python3.7/site-
packages (from -r requirements.txt (line 18)) (0.13)
Requirement already satisfied: paramiko>=1.16.0 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 19))
(2.4.2)
Requirement already satisfied: pytest-django>=2.9.1 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 20))
(3.4.7)
Requirement already satisfied: whitenoise>=4.1.2 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 21))
(4.1.2)
Requirement already satisfied: django-sslserver>=0.19 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 22))
(0.20)
Requirement already satisfied: djangorestframework>=3.6.2 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 23))
(3.9.1)
Requirement already satisfied: django-compressor>=2.1.1 in
./env/lib/python3.7/site-packages (from -r requirements.txt (line 24))
(2.2)
Requirement already satisfied: pytz in ./env/lib/python3.7/site-
packages (from Django==1.11.20->-r requirements.txt (line 3)) (2018.9)
Requirement already satisfied: atomicwrites>=1.0 in
./env/lib/python3.7/site-packages (from pytest>=2.9.0->-r
requirements.txt (line 9)) (1.3.0)
Requirement already satisfied: more-itertools>=4.0.0; python_version >
"2.7" in ./env/lib/python3.7/site-packages (from pytest>=2.9.0->-r
requirements.txt (line 9)) (6.0.0)
Requirement already satisfied: attrs>=17.4.0 in
./env/lib/python3.7/site-packages (from pytest>=2.9.0->-r
requirements.txt (line 9)) (18.2.0)
Requirement already satisfied: pluggy>=0.7 in ./env/lib/python3.7/site-
packages (from pytest>=2.9.0->-r requirements.txt (line 9)) (0.8.1)
Requirement already satisfied: django-appconf>=0.6.0 in
./env/lib/python3.7/site-packages (from Django-Select2>=5.8.4->-r
requirements.txt (line 10)) (1.0.2)
```

```
Requirement already satisfied: pynacl>=1.0.1 in
./env/lib/python3.7/site-packages (from paramiko>=1.16.0->-r
requirements.txt (line 19)) (1.3.0)
Requirement already satisfied: cryptography>=1.5 in
./env/lib/python3.7/site-packages (from paramiko>=1.16.0->-r
requirements.txt (line 19)) (2.5)
Requirement already satisfied: bcrypt>=3.1.3 in
./env/lib/python3.7/site-packages (from paramiko>=1.16.0->-r
requirements.txt (line 19)) (3.1.6)
Requirement already satisfied: rcssmin==1.0.6 in
./env/lib/python3.7/site-packages (from django-compressor>=2.1.1->-r
requirements.txt (line 24)) (1.0.6)
Requirement already satisfied: rjsmin==1.0.12 in
./env/lib/python3.7/site-packages (from django-compressor>=2.1.1->-r
requirements.txt (line 24)) (1.0.12)
Requirement already satisfied: cffi>=1.4.1 in ./env/lib/python3.7/site-
packages (from pynacl>=1.0.1->paramiko>=1.16.0->-r requirements.txt
(line 19)) (1.12.0)
Requirement already satisfied: asn1crypto>=0.21.0 in
./env/lib/python3.7/site-packages (from cryptography>=1.5-
>paramiko>=1.16.0->-r requirements.txt (line 19)) (0.24.0)
Requirement already satisfied: pycparser in ./env/lib/python3.7/site-
packages (from cffi>=1.4.1->pynacl>=1.0.1->paramiko>=1.16.0->-r
requirements.txt (line 19)) (2.19)
Looking in links: /opt/emoveo/install/wheels
Requirement already satisfied: cx_Oracle==6.0.2 in
./env/lib/python3.7/site-packages (from -r install/oracle_driver.txt
(line 1)) (6.0.2)
Wheels installed. Environment setup complete.
Validating all required directories exist...
Directories validated. Running manage.py tasks...
Operations to perform:
  Apply all migrations: admin, auth, config, contenttypes, deploy,
sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying config.0001_initial... OK
  Applying deploy.0001_initial... OK
  Applying sessions.0001_initial... OK
Seeding  SpecialCommand
Loading file /opt/emoveo/install/seed_data/SpecialCommand.zip
- SpecialCommand 1 valid
- SpecialCommand 2 valid
- SpecialCommand 3 valid
```

```
- SpecialCommand 4 valid
- SpecialCommand 5 valid
- SpecialCommand 6 valid
- SpecialCommand 7 valid
- SpecialCommand 8 valid
Import complete.
Seeding  FieldGroup
Loading file /opt/emoveo/install/seed_data/FieldGroup.zip
- FieldGroup 1 valid
- FieldGroup 2 valid
Import complete.
Seeding  ObjectType
Loading file /opt/emoveo/install/seed_data/ObjectType.zip
- ObjectType 67 valid
- ObjectType 1 valid
- ObjectType 106 valid
- ObjectType 104 valid
Import complete.
Seeding  Environment
Loading file /opt/emoveo/install/seed_data/Environment.zip
- Environment 62 valid
- Environment 81 valid
- Environment 61 valid
Import complete.
Seeding  Workflow
Loading file /opt/emoveo/install/seed_data/Workflow.zip
- Workflow 2 valid
- Workflow 21 valid
Import complete.
Seeding  ReportType
Loading file /opt/emoveo/install/seed_data/ReportType.zip
- ReportType 1 valid
Import complete.
Missing seed data imported.
Validating CustomFieldGroup
..
Found 0 errors.
Validating ObjectType
.................
Found 0 errors.
Validating WorkflowStepCommand

Found 0 errors.
Validating WorkflowStep
........
Found 0 errors.
Validating Workflow
..
Found 0 errors.
Total errors found: 0
Found another file with the destination path
'django_select2/django_select2.js'. It will be ignored since only the
first encountered file is collected. If this is not what you want, make
sure every static file has a unique path.
```

```
0 static files copied to '/opt/emoveo/static', 234 unmodified.
Install complete.
```

# 5. Starting and Stopping the Emoveo Application

Once the server components have been successfully installed, the Emoveo application may be started and stopped as required.

Starting:

On the Emoveo application server, run the following command from the Emoveo directory

```
python3 start.py
```

The output should resemble the following:

```
[ec2-user@emoveo]$ python3 start.py
[ec2-user@emoveo]$ Performing system checks...

System check identified no issues (0 silenced).
January 14, 2022 - 23:23:13
Django version 2.2.20, using settings 'emoveo.settings'
Starting development server at http://emoveo.com:8080/
```

Stopping:

On the Emoveo application server, run the following command from the Emoveo directory

```
python3 stop.py
```

Status:

To determine the status of the Emoveo application, go to the /emoveo directory and cat the pid.txt file. This file will show the latest Emoveo pid running. To check the process on the server run "`ps -a`" to match it up against the running processes. To forcibly kill the process run "`kill -9 [pid]`".

# 6. Making Changes to Emoveo Configuration

## *6.1. Application Configuration*

To change the Emoveo configuration (for example, if database connectivity parameters need to be changed):

1. Stop the Emoveo server.
2. Run the python `configure.py` script and provide responses to the configuration items.
3. Start the server.

## *6.2. Local Configuration Settings*

There are specific settings that can be made to an individual instance install. The parameters to modify to adjust these settings are in [HOME]/emoveo/local_settings.py.

Listed below are some configurations parameters that can be changed per install:

| Name | Description | Values |
|---|---|---|
| ADD_LINES_TO_NEW_GROUP | Allows the application to choose the default value of a new package line. | **"True"** will default a new line into a new group. (DEFAULT)<br><br>**"False"** will add a new line to the latest created group. |
| CUSTOM_GROUP_PERMISSIONS | Enables the "Permissions" section of the security group screen and allows for the setting of application access by security group | **"True"** will allow for the setting of permission by security group (DEFAULT)<br><br>**"False"** will turn off the permissions access settings per security group and limit the difference in users to standard and admin. |
| SESSION_COOKIE_AGE | Sets the timeout of an inactive session | This parameter is set in seconds and the DEFAULT setting is: **120*60** |
| ENV_CHECK_DB_TIMEOUT | Limits the time an environment check will run before timing out and failing | This parameter is set in seconds and the DEFAULT setting is: **10** |
| NUMBER_OF_ DISALLOWED_ PREVIOUS_PASSWORDS | When the user changes their password, Emoveo checks the last specified number of passwords used, and ensures that there is no repetition | This parameter is set in whole numbers and the DEFAULT setting is: **1000** |

| | | |
|---|---|---|
| PASSWORD_EXPIRATION | Determines whether non-admin passwords should have an expiration date | This parameter accepts **True** and **False** values and the DEFAULT setting is **False** |
| PASSWORD_EXPIRATION_DAYS | If non-admin passwords are set to have an expiration date, this parameter defines a limit for their duration. When the limit is exceeded, the user is prompted to change their password upon login | This parameter is set in whole numbers representing days and the DEFAULT setting is: **-1** |
| BB_MAX_FAILURES | Sets a limit on the number of failed login attempts. When the limit is exceeded, Emoveo locks the account | This parameter is set in whole numbers and the DEFAULT setting is: **1000** |
| BB_BLOCK_INTERVAL | Defines the duration of time in minutes a user account should remain locked upon a string of unsuccessful login attempts | This parameter is set in whole numbers representing minutes and the DEFAULT setting is: **1** |

*Table 2 – Available parameters for configuration*

## *6.3. Additional User Password Enforcement Settings*

In addition to the password parameters defined in *local_settings.py*, optional Django-provided validators can be added at the end of the file. The default validators are listed in the table below, followed by an example of how to enable and configure them in *local_settings.py*.

| Name | Description |
|---|---|
| UserAttributeSimilarityValidator | Check the similarity between the password and a set of attributes of the user |
| MinimumLengthValidator | Check whether the password meets a minimum length. This validator is configured with a custom option: it now requires the minimum length to be nine characters, instead of the default eight |
| CommonPasswordValidator | Check whether the password occurs in a list of common passwords. By default, it compares to an included list of 20,000 common passwords. Additional passwords that should be included in the check for common passwords |
| NumericPasswordValidator | Check whether the password is not entirely numeric |

➤ **Add the following to the end of your local_settings.py file and adjust as needed**

```
# Custom password validators
from django.core.exceptions import ValidationError
class CustomValidator:
```

```
    DEFAULT_MSG = ''

    def __init__(self, message=None, special_chars=['!','?','$','.','%'], dictionary=[]):
        self.message = message if message else self.DEFAULT_MSG
        self.special_chars = special_chars
        self.dictionary = dictionary

    def get_help_text(self):
        return self.message

    def repeat_values(self, password):
        for i in range(len(password) - 1):
            if password[i] == password[i+1]:
                return True

    def validate(self, password, user=None):
        if not any(chr.isdigit() for chr in password):
            raise ValidationError("Your password must have at least one number", code='missing_numeric')
        if not any(chr.isupper() for chr in password):
            raise ValidationError("Your password must have at least one upper case letter",
code='missing_upper_case')
        if not any(chr.islower() for chr in password):
            raise ValidationError("Your password must have at least one lower case letter",
code='missing_lower_case')
        if not any(chr in self.special_chars for chr in password):
            raise ValidationError(f"Your password must have at least one special character {',
'.join(self.special_chars)}", code='missing_special')
        if self.repeat_values(password):
            raise ValidationError(f"Your password must not have repeating consecutive characters",
code='repeating_characters')
        if password in self.dictionary:
            raise ValidationError(f"Your password must not be in the dictionary of common passwords. Please
contact you sysadmin for further information or choose a more complex password.", code='dictionary')

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
        'OPTIONS': {'user_attributes': ['username', 'first_name', 'last_name', 'email']}
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
        'OPTIONS': {'min_length': 12}
    },
    {'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator'},
    {'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator'},
    {'NAME': 'config.models.users.PreviousPasswordValidator'},
    # Change special_chars and dictionary in order to customize valid special characters and disallowed
passwords
    {'NAME': 'emoveo.local_settings.CustomValidator', 'OPTIONS': {
        'special_chars': ['!','?','$','.','%','#','@','&','*','^','{','}','{','}','|'],
        'dictionary': ["password", "Password123", "Password123!"]}}
]
```

- *Additional validators for configuration can be found here:* (https://docs.djangoproject.com/en/2.2/topics/auth/passwords/#enabling-password-validation)

# 7. Upgrading Emoveo

Every major Emoveo upgrade is accompanied by a technical manual produced by PPMetrics. Please follow the instructions in the PPMetrics release manual when you are ready to perform the upgrade.

# 8. Uninstalling Emoveo

To uninstall the Emoveo application, perform the following tasks:

1. Stop the Emoveo server.
2. Delete the Emoveo directory on the application server(s).
3. Drop the Emoveo database schema.

# 9. Ongoing Maintenance

Emoveo requires very little ongoing maintenance. Nevertheless, please keep in mind the following:

1. The Emoveo process must be up and running for end users to be able to use it. Since various IT events such as database/OS upgrades may involve a need to bring down the Emoveo process, you may need to manually restart it at times.
2. The Emoveo log files may grow large over time, and it may be valuable to periodically delete their contents. All log files are stored in the “log” directory within the Emoveo installation directory.
   2.1. Under typical usage, this task isn’t be needed more frequently than every 2-3 months.
   2.2. Deleting application logs has no effect on workflow step logs and reports, which are stored in the “media” directory. Deleting the contents of the media directory will make it impossible for Emoveo to show logs of previous package executions and reports.

- **User transactions most relevant for auditing are stored in the “media” directory so exercise caution if any logs are to be deleted from the directory.**

# 10. Defining Users

- **By default, one user account is created after installation. The 'admin' user is an administrative account that effectively provides superuser access and the default password for this account is 'admin'.**

## *10.1. Creating Users*

When admins need to create users to participate and perform different deployment activities in Emoveo, they need to execute the following steps:

1. Log on to Emoveo as an admin user.
2. From the menu bar, select Configure > Users
3. Click the + ADD USER button in the lower right.
4. Enter the information in the following sections for the user:

| Prompt | Description |
|---|---|
| Username: | The username that the new user will use to log onto Emoveo. Required field |
| New Password:<br>Confirm Password: | The password of the new user. Required field |
| First Name: | The first name of the new user |
| Last Name: | The last name of the new user |
| Email address: | The email address of the new user |
| Active | Whether to enable this user |
| Security Groups: | To link the user to the security groups that provide functional roles and permissions, click individual or multiple security groups from the Available Security Groups table and then click the right arrow to add them to the Chosen Security Groups table |

*Table 3 – User parameters*

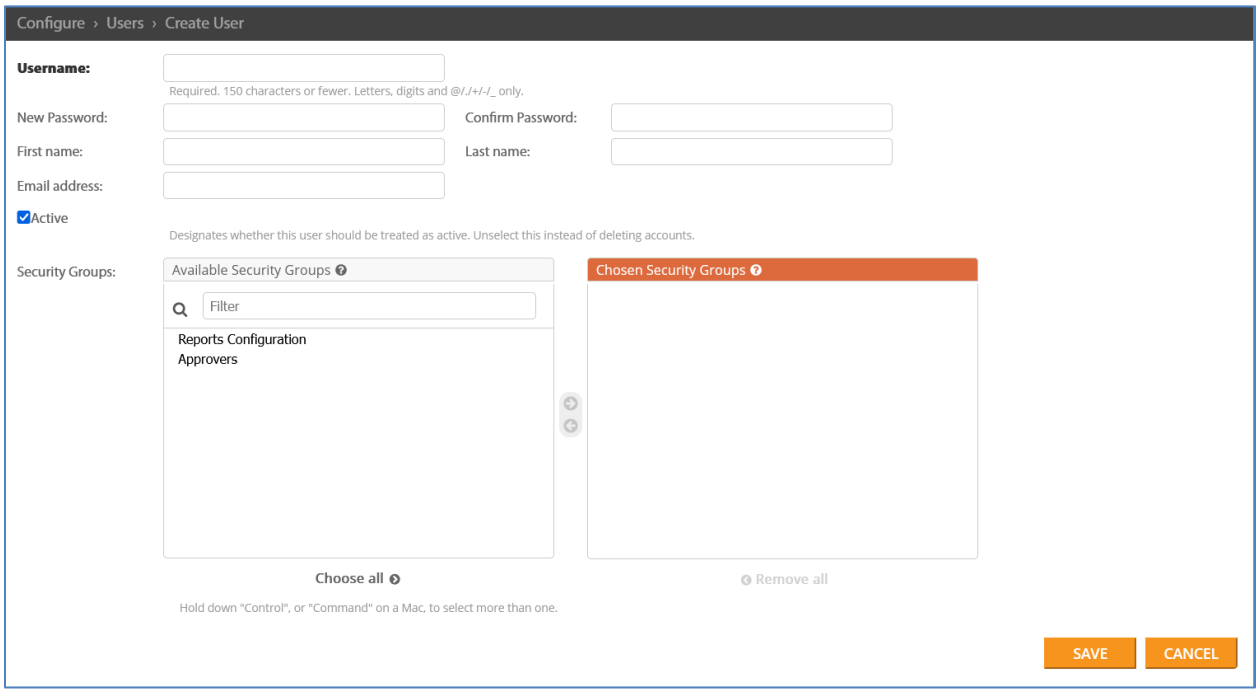


*Fig 2 – Create User screen*

5. Click the Save button when completed.

## 10.2. Copying Users

To create a user by copying an existing user:

1. Log on to Emoveo as an admin user.
2. From the menu bar, select Configure > Users
3. Click the checkbox next to the username that you'd like to copy.
4. From the Action: dropdown, choose Copy selected Users and then hit GO.

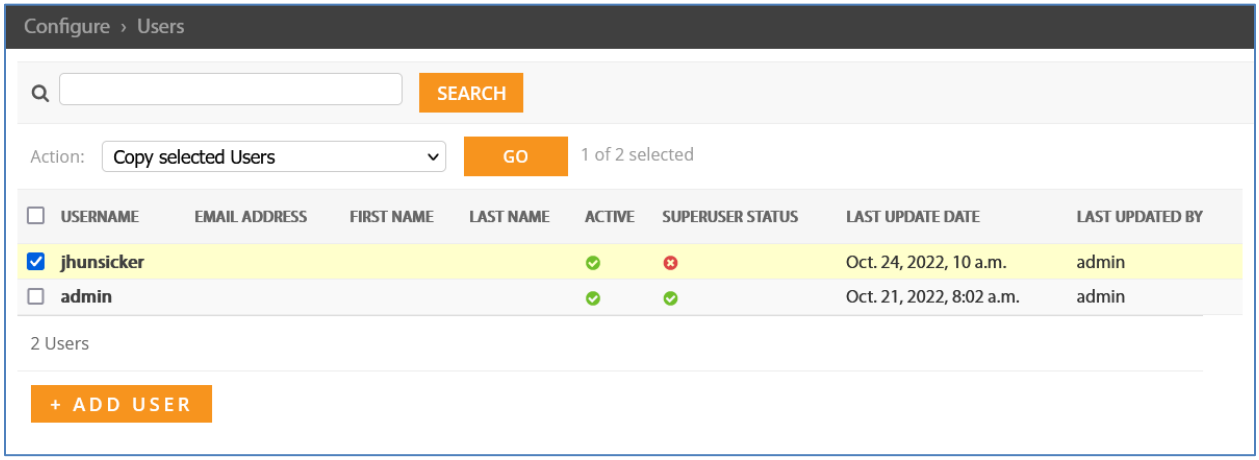


*Fig 3 – Configure Users screen and copy action*

5. A new user will be created with the same username you selected with a "_1" appended.
6. Click the new username to modify the user's profile.

## 10.3. Deleting/Disabling Users

If a user was only created for testing purposes and has not been associated with migration data, then Emoveo will permit you to delete it:

1. Log on to Emoveo as an admin user.
2. From the menu bar, select Configure > Users
3. Click the Username that you want to delete.
4. From the Action: dropdown, choose Delete selected Users and then hit GO.

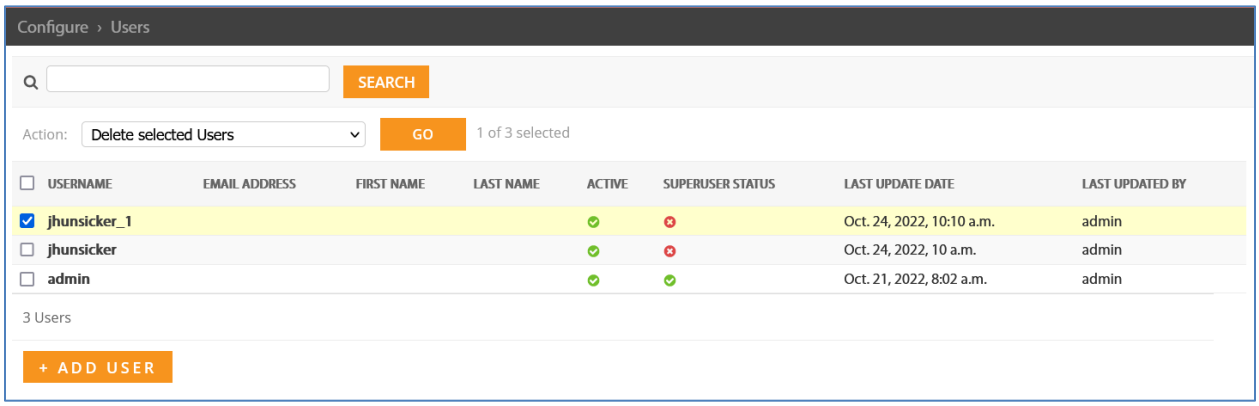


*Fig 4 – Configure Users screen and delete action*

If you attempt to delete a user that has executed any transactions in the system prior, Emoveo will not permit you to delete it. This is done to preserve governance and avoid auditing issues. In this case, you may disable the user:

1. Log on to Emoveo as an admin user.
2. From the menu bar, select Configure > Users
3. Click the Username that you want to disable.
4. From the user profile, uncheck the Active checkbox.
5. Click the Save button when completed.

# 11. Creating Security Groups

To control access to specific sections of the Emoveo user interface and its functionality, you create security groups, optionally configure their permissions, and then specify their members. To create a security group:

1. Log on to Emoveo as an admin user.
2. From the menu bar, select Configure > Security Groups
3. Click the + ADD SECURITY GROUP button in the lower right.
4. Enter the information in the following sections for the Security Group:

| Prompt | Description |
|---|---|
| Name: | The name of the new security group. Required field |
| Description: | Optional description of the new security group |
| Permissions: | To add permissions to the new security group for members to inherit, click individual or multiple permissions from the Available permissions table and then click the right arrow to add them to the Chosen permissions table |

*Table 4 – Security Group parameters*

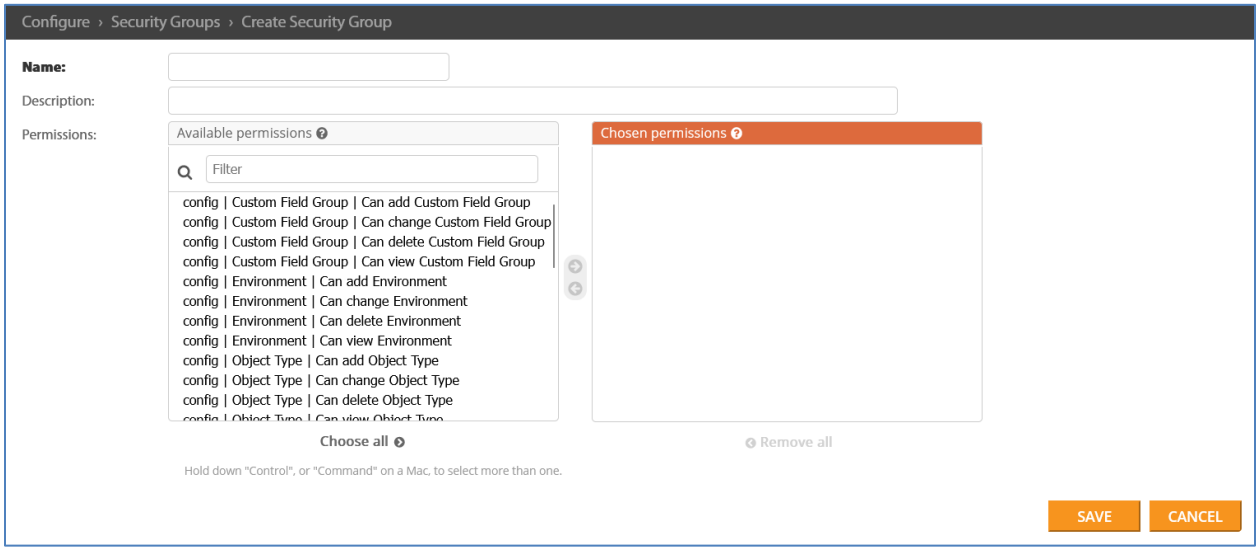


*Fig 5 – Create Security Groups screen*

5. Click the Save button when completed.

# 12. System Requirements and Compatibility Matrix

Software listed below has been tested by PPMetrics to ensure compatibility with current versions of the Emoveo.

| Environment | Component Type | Details | Notes |
|---|---|---|---|
| Application server | Operating System | • Red Hat Linux 6.x<br>• Red Hat Linux 7<br>• Oracle Enterprise Linux 6<br>• Oracle Solaris 11 | Red Hat Enterprise Linux recommended |
| Application server | Available RAM | 8GB | |
| Application server | Available Disk Space | 10GB | |
| Application server | Python Codebase | 3.9 | Higher versions are suitable |
| Database server | Available Disk Space | 10GB | |
| Database | Oracle | • 12C<br>• 19C | Higher versions are suitable |

*Table 5 – Emoveo system requirements and compatibility*