MIFOS

# Installment enviroment :

1. Mifos server and web interface : <https://sungateway.xyz>

2. Android Apps : <https://drive.google.com/open?id=1exM4pnGmwv8yzu6Bk0efvEROEGDSTM8Q>

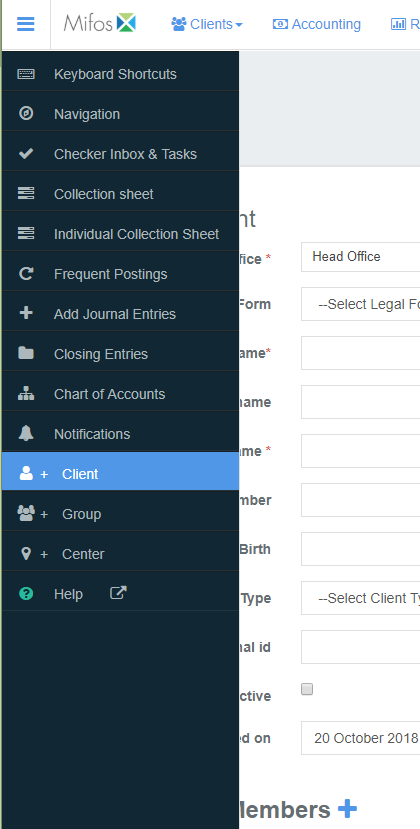
# Usage :

Login : Login account :

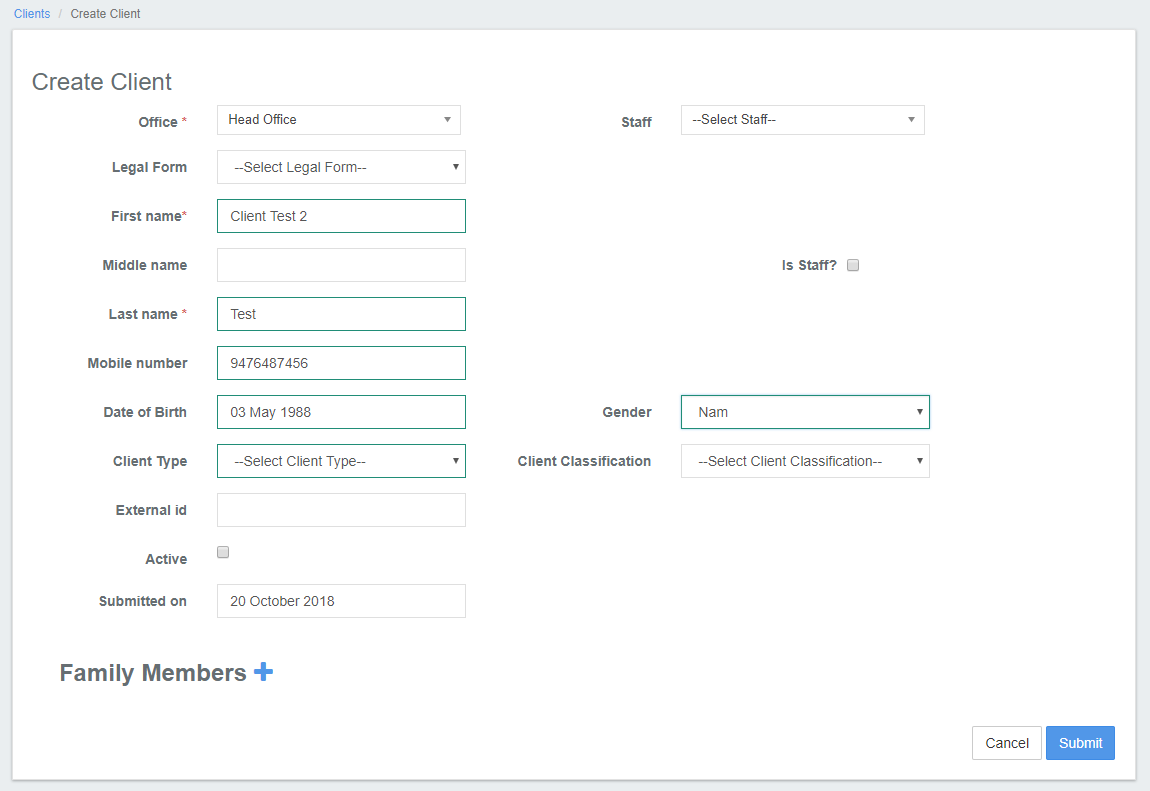
* Login : mifos
* Password : 123456789

Create a new client (on website) as with android apps:

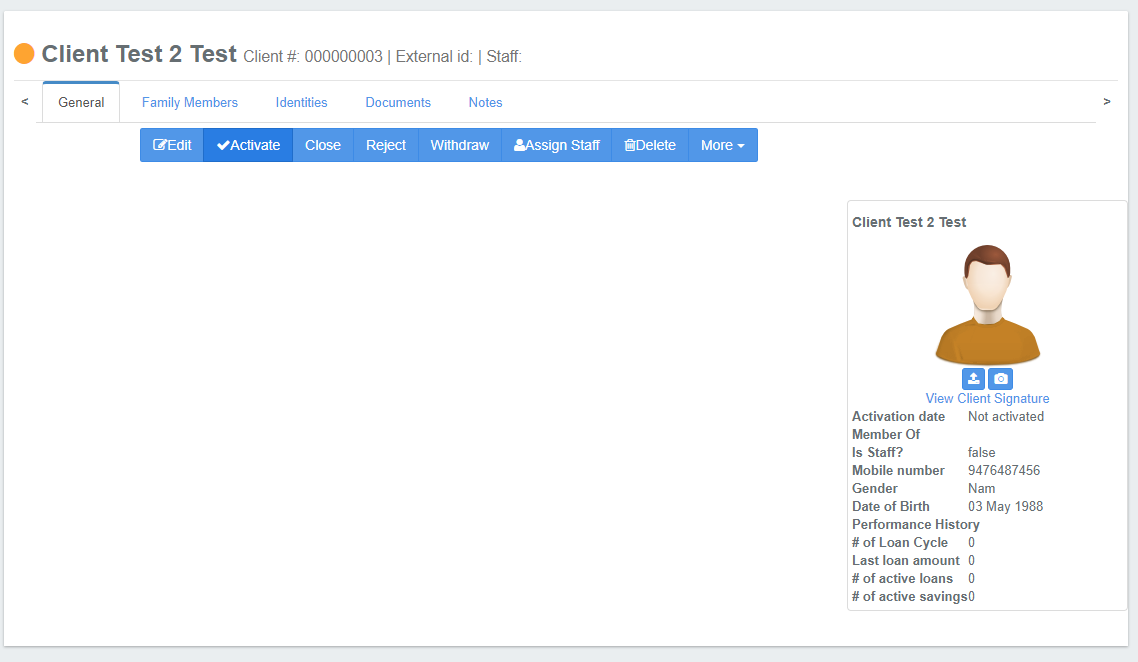
1. Create a new client menu :



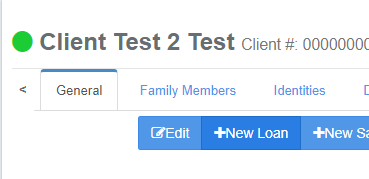
2. Fill in the information and submit :



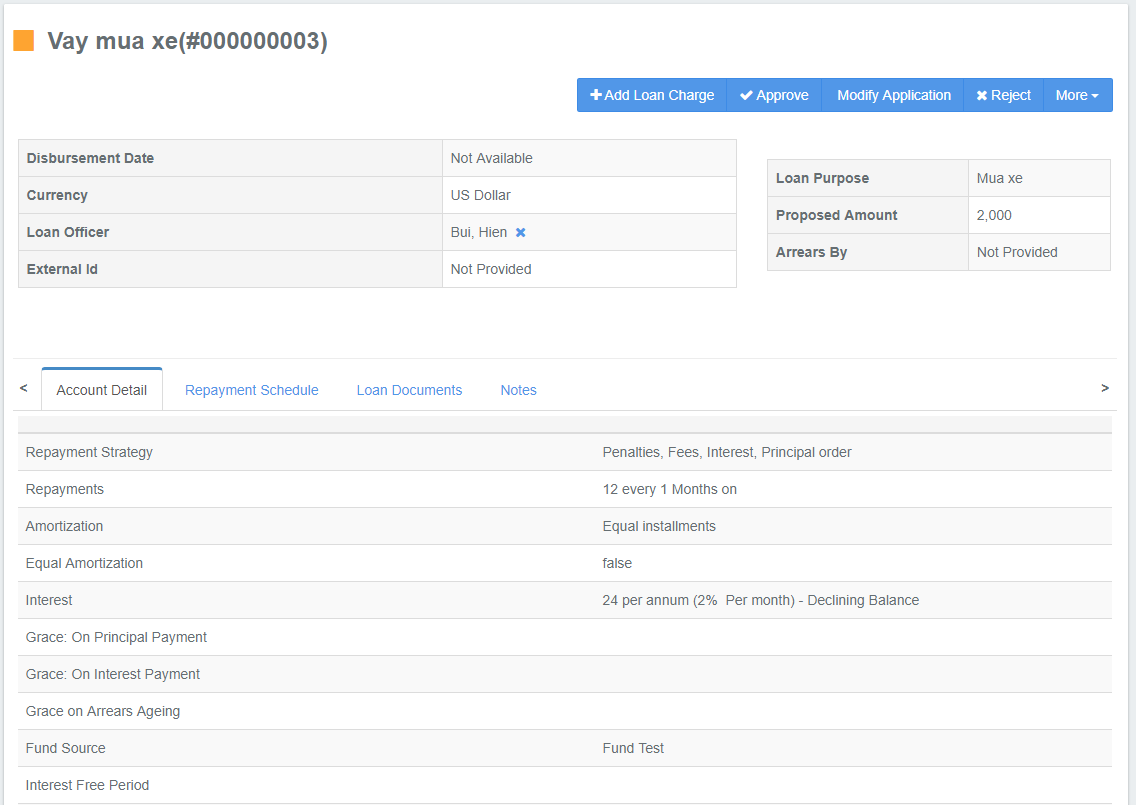
3. After create a client, choose « Activate » to activate this client.



4. Choose « New Loan » to create new loan



5. Follow instructions to create a new loan for this client.



6. Pick « Approve » to confirm this loan .

Setup Fineract on Ubuntu 14.04  
$ ssh root@128.199.134.53  
$ sudo apt-get update  
### https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-14-04  
# sudo adduser khanh                        # As `root` create a new user, enter information when asked  
# gpasswd -a khanh sudo                     # As `root` run this command to add user to group sudo  
# su - khanh                                # As `root` switch to user `khanh`  
$ mkdir .ssh                                # As `khanh`, at /home/khanh create a new folder .ssh  
$ chmod 700 .ssh                            # Restrict permission to folder .ssh  
$ nano .ssh/authorized\_keys                 # Open file `authorized\_keys` for adding new SSH public key, paste key there and Ctrl-X, Y, Enter  
$ chmod 600 .ssh/authorized\_keys            # Restrict the permissions of the authorized\_keys file  
$ exit                                      # Exit `khanh` and return `root` account  
# service ssh restart                       # Restart SSH service  
# exit                                      # Exit `root` session  
### From now working with normal user to do all setup for the server, using `sudo` command when needed  
$ ssh khanh@128.199.134.53                  # Login as normal user  
### Installing the Oracle JDK 8  
$ sudo add-apt-repository ppa:webupd8team/java  
$ sudo apt-get update  
$ sudo apt-get install oracle-java8-installer       # Start installation  
$ sudo update-alternatives --config java            # Check installation & config  
$ sudo nano /etc/environment                        # Open the file and add this to a new line: JAVA\_HOME="/usr/lib/jvm/java-8-oracle"  
$                                                   # Ctrl-X, Y, Enter to close and save the file  
$ source /etc/environment                           # Reload the setting file  
$ echo $JAVA\_HOME                                   # Verify the setting  
$ java -version                                     # Check if Java has been setup successful  
### Installing MySQL 5.6 # https://askubuntu.com/questions/1029634/install-mysql-5-6-on-ubuntu-18-04-lts  
$ cd /tmp  
$ curl -OL https://dev.mysql.com/get/Downloads/MySQL-5.6/mysql-server\_5.6.41-1ubuntu14.04\_amd64.deb-bundle.tar  
$ tar -xf mysql-server\_5.6.41-1ubuntu14.04\_amd64.deb-bundle.tar  
$ sudo dpkg -i mysql-common\_5.6.41-1ubuntu14.04\_amd64.deb  
$ sudo apt-get install libaio1  
$ sudo dpkg -i mysql-community-client\_5.6.41-1ubuntu14.04\_amd64.deb  
$ sudo dpkg -i mysql-client\_5.6.41-1ubuntu14.04\_amd64.deb  
$ sudo dpkg -i mysql-community-server\_5.6.41-1ubuntu14.04\_amd64.deb  
$ sudo dpkg -i mysql-server\_5.6.41-1ubuntu14.04\_amd64.deb  
$ sudo apt-get install -f  
$ mysql --version  
### https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-14-04  
\*\*\* The mysql\_install\_db command is deprecated as of MySQL 5.7.6. If you're using version 5.7.6 or later, you should use mysqld --initialize instead.  
$ sudo mysql\_install\_db  
$ service mysql status                      # check if the service is running  
$ sudo service mysql start                  # If MySQL isn't running, start it  
$ mysqladmin -p -u root version             # do some more check  
### Change `root` password to `mysql`  
$ sudo mysql -u root -p                     # enter the current password  
mysql> update mysql.user SET password=PASSWORD('mysql') WHERE user ='root';  
mysql> flush privileges;  
mysql> exit;  
$ sudo mysql -u root -p                     # enter password: mysql  
mysql> CREATE DATABASE `mifosplatform-tenants`;  
mysql> CREATE DATABASE `mifostenant-default`;  
mysql> show databases;  
mysql> exit  
### Install Tomcat 7  
### https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-7-on-ubuntu-14-04-via-apt-get  
$ cd /tmp  
$ sudo wget http://mirror.downloadvn.com/apache/tomcat/tomcat-7/v7.0.91/bin/apache-tomcat-7.0.91.tar.gz  
$ sudo tar xzvf apache-tomcat-\*  
$ sudo mv apache-tomcat-7.0.91 /opt/tomcat  
$ cd /opt/tomcat  
$ sudo groupadd tomcat  
$ sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat  
$ sudo chgrp -R tomcat /opt/tomcat  
$ sudo chmod -R g+r conf  
$ sudo chmod g+x conf  
$ sudo chown -R tomcat webapps/ work/ temp/ logs/  
$ sudo update-java-alternatives -l              # /usr/lib/jvm/java-8-oracle  
$ sudo nano /etc/systemd/system/tomcat.service  # content below  
--BEGIN OF FILE--  
[Unit]  
Description=Apache Tomcat Web Application Container  
After=network.target  
[Service]  
Type=forking  
Environment=JAVA\_HOME=/usr/lib/jvm/java-8-oracle  
Environment=CATALINA\_PID=/opt/tomcat/temp/tomcat.pid  
Environment=CATALINA\_HOME=/opt/tomcat  
Environment=CATALINA\_BASE=/opt/tomcat  
Environment='CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'  
Environment='JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'  
ExecStart=/opt/tomcat/bin/startup.sh  
ExecStop=/opt/tomcat/bin/shutdown.sh  
User=tomcat  
Group=tomcat  
UMask=0007  
RestartSec=10  
Restart=always  
[Install]  
WantedBy=multi-user.target  
--END OF FILE--  
$ cd /opt/tomcat/lib  
$ sudo wget http://central.maven.org/maven2/org/drizzle/jdbc/drizzle-jdbc/1.3/drizzle-jdbc-1.3.jar  
$ cd /tmp  
$ sudo wget https://dev.mysql.com/get/archives/mysql-connector-java-5.1/mysql-connector-java-5.1.36.tar.gz  
$ tar xzf mysql-connector-java-5.1.36.tar.gz  
$ cd mysql-connector-java-5.1.36  
$ cp mysql-connector-java-5.1.36-bin.jar /opt/tomcat/lib  
### To enable tomcat SSL (https), generate and store a key: (set password: xyz123)  
$ cd /opt/tomcat  
$ sudo keytool -genkey -keyalg RSA -alias tomcat -keystore tomcat.keystore  
### Updating server.xml configuration file  
$ cd /opt/tomcat/conf  
$ sudo nano server.xml  
--BEGIN OF FILE--  
<?xml version='1.0' encoding='utf-8'?>  
<Server port="8005" shutdown="SHUTDOWN">  
    <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />  
    <Listener className="org.apache.catalina.core.JasperListener" />  
    <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />  
    <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />  
    <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />  
    <GlobalNamingResources>  
        <Resource name="UserDatabase" auth="Container"  
            type="org.apache.catalina.UserDatabase"  
            description="User database that can be updated and saved"  
            factory="org.apache.catalina.users.MemoryUserDatabaseFactory"  
            pathname="conf/tomcat-users.xml" />  
        <Resource type="javax.sql.DataSource"  
            name="jdbc/mifosplatform-tenants"  
            factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"  
            driverClassName="org.drizzle.jdbc.DrizzleDriver"  
            url="jdbc:mysql:thin://localhost:3306/mifosplatform-tenants"  
            username="root"  
            password="mysql"  
            initialSize="3"  
            maxActive="10"  
            maxIdle="6"  
            minIdle="3"  
            validationQuery="SELECT 1"  
            testOnBorrow="true"  
            testOnReturn="true"  
            testWhileIdle="true"  
            timeBetweenEvictionRunsMillis="30000"  
            minEvictableIdleTimeMillis="60000"  
            logAbandoned="true"  
            suspectTimeout="60" />  
    </GlobalNamingResources>  
    <Service name="Catalina">   
        <Connector protocol="org.apache.coyote.http11.Http11Protocol"  
            port="8443" maxThreads="200" scheme="https"  
            secure="true" SSLEnabled="true"  
            keystoreFile="/opt/tomcat/tomcat.keystore"  
            keystorePass="xyz123"  
            clientAuth="false" sslProtocol="TLS"  
            URIEncoding="UTF-8"  
            compression="force"  
            compressableMimeType="text/html,text/xml,text/plain,text/javascript,text/css"/>  
        <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />  
        <Engine name="Catalina" defaultHost="localhost">  
            <Realm className="org.apache.catalina.realm.LockOutRealm">  
            <Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase"/></Realm>  
            <Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">  
                <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"  
                    prefix="localhost\_access\_log." suffix=".txt"  
                    pattern="%h %l %u %t &quot;%r&quot; %s %b" />  
            </Host>  
        </Engine>  
    </Service>  
</Server>  
--END OF FILE--  
$ cd /etc/init.d/  
$ sudo nano tomcat  
--BEGIN OF FILE--  
# Tomcat auto-start  
#  
# description: Auto-starts tomcat  
# processname: tomcat  
# pidfile: /var/run/tomcat.pid  
case $1 in  
start)  
sh /opt/tomcat/bin/startup.sh  
;;  
stop)  
sh /opt/tomcat/bin/shutdown.sh  
;;  
restart)  
sh /opt/tomcat/bin/shutdown.sh  
sh /opt/tomcat/bin/startup.sh  
;;  
esac  
exit 0  
--END OF FILE--  
$ sudo chmod 755 /etc/init.d/tomcat  
$ sudo ln -s /etc/init.d/tomcat /etc/rc1.d/K99tomcat  
$ sudo ln -s /etc/init.d/tomcat /etc/rc2.d/S99tomcat  
$ sudo nano /opt/tomcat/conf/tomcat-users.xml  
<user username="admin" password="aionsigma123" roles="manager-gui,admin-gui"/>  
### Install Git  
$ sudo apt-get install git  
$ mkdir ~/aionsigma  
$ git clone https://github.com/thkhanh/fineract  
$ git clone https://github.com/thkhanh/community-app  
### Install ActiveMQ: https://cwiki.apache.org/confluence/display/FINERACT/Setting+up+ActiveMQ+for+Fineract+1.0  
### https://activemq.apache.org/activemq-5151-release.html  
$ wget https://archive.apache.org/dist/activemq/5.15.1/apache-activemq-5.15.1-bin.tar.gz  
$ tar xzf apache-activemq-5.15.1-bin.tar.gz  
$ sudo mv apache-activemq-5.15.1 /opt/activemq  
$ cd /opt/activemq/bin  
$ ./activemq start  
$ ./activemq status  
$ ./activemq stop  
### Build and start services  
$ cd ~/aionsigma/fineract  
$ ./gradlew build  
$ ./gradlew migrateTenantListDB -PdbName=mifosplatform-tenants  
$ ./gradlew migrateTenantDB -PdbName=mifostenant-default  
$ ./gradlew clean integrationTest  
$ sudo cp build/libs/fineract-provider.war /opt/tomcat/webapps  
$ sudo /etc/init.d/tomcat start  
$ sudo /etc/init.d/tomcat stop  
\* https://localhost:8443/fineract-provider/  
### Install and build the Community app  
$ cd ~/aionsigma/community-app  
$ sudo apt-get install software-properties-common  
### https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-an-ubuntu-14-04-server  
### How To Install Using NVM - https://github.com/creationix/nvm  
$ cd ~/aionsigma/community-app/  
$ sudo apt-get update  
$ sudo apt-get install build-essential libssl-dev  
$ curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.11/install.sh | bash  
$ nvm --version                             # to verify the installation, if not found try to exit and reconnect  
$ nvm install 8.12.0                        # install node version 8.12.0 & npm  
$ node -v                                   # check node version  
$ npm -v                                    # check npm version  
$ npm install -g bower                      # install bower  
$ npm install -g grunt-cli                  # install grunt-cli  
$ sudo apt-get update  
$ sudo apt-get install ruby-full            # install ruby-full  
$ sudo apt-get install ruby-bundler         # If this not working, try the following commands to install  
$ sudo apt-get update  
$ gem --version  
$ sudo gem install bundler   
$ bower install  
$ npm install  
$ bundler install  
$ grunt validate                            # validate the source code  
$ grunt serve                               # Run for local testing  
$ grunt prod                                # or build for deployment  
### Deploy the community app to webapps  
$ sudo /etc/init.d/tomcat stop              # stop tomcat service  
$ sudo cp -a dist/. /opt/tomcat/webapps/    # move the community-app folder to /tomcat/webapps  
$ sudo /etc/init.d/tomcat start             # start tomcat service