|  |
| --- |
| MOTECH Suite – Steps for MOTECH Platform Development setup |

# Overview

Developers who want to make development on top of MOTECH platform, first need to set up the whole development environment for MOTECH. Process of whole set up is a bit lengthy and complex and moreover it runs only on Ubuntu (LTS 64 bit platform); hence this document is written to provide a simple and easy to follow steps for doing whole setup including adding up new modules over the core platform.

Detailed reference for MOTECH setup is available on its portal itself -

<http://docs.motechproject.org/en/latest/development/dev_setup/dev_install.html> .

More references that are used are mentioned at the end of the document.

# Platform & Software Prerequisite

To get up and running with MOTECH, dependent software and systems need to be installed and then the MOTECH packages over that.

MOTECH makes use of a number of third-party open source systems, each of which needs to be installed for MOTECH to function correctly.

Following are software requirements to install and build MOTECH in Linux (Ubuntu) platform

1. OS required – **Ubuntu Desktop  14.04 LTS 64bit**
2. Dependent software
   1. Java 8
   2. Git
   3. CURL
   4. Node js
   5. NPM
   6. Bower
   7. Gulp
3. Dependent Systems/Platform
   1. [Tomcat7](http://tomcat.apache.org/download-70.cgi)
   2. [ActiveMQ](http://activemq.apache.org/activemq-551-release.html)
   3. [MySQL](http://dev.mysql.com/downloads/mysql/5.5.html)
   4. Quartz

MOTECH uses **Tomcat** as application container, **ActiveMQ** as message queue, **Quartz** for timers and **MySQL** for database.

# Setting Up a Development Environment

**Installing On Ubuntu Platform**

1. **Install Ubuntu Desktop 14.04 LTS 64bit (or any later version).**

Instruction for installation can referred in below link:

[**http://www.ubuntu.com/download/desktop/install-ubuntu-desktop**](http://www.ubuntu.com/download/desktop/install-ubuntu-desktop)

**Note:** 64-bit is required for MOTECH’s installation

1. **Install Maven, Git, Curl, ActiveMQ, NodeJS, NPM, Bower, Gulp and MySQL**

***(All commands to be typed on terminal)***

* 1. On a fresh Ubuntu installation, run the following first

***sudo apt-get update***

* 1. In terminal, type

***sudo apt-get install curl git maven activemq npm***

***curl -sL https://deb.nodesource.com/setup\_4.x | sudo -E bash -***

***sudo apt-get install -y nodejs***

***sudo npm install -g bower gulp***

1. **Configure ActiveMQ**

Type the following in terminal and run

***sudo ln -s /etc/activemq/instances-available/main /etc/activemq/instances-enabled/main***

For ActiveMQ scheduled delivery to work, set the attribute **schedulerSupport=“true”** in activemq.xml

* 1. Find the file activemq.xml. This file should be located at (active-mq-folder)/conf/activemq.xml.
  2. Go to that directory in terminal
  3. Open the file using following command

***nano activemq.xml***

* 1. Find the broker element
  2. Add (if not there) the attribute **schedulerSupport**
  3. Set its value to **“true”**
  4. Save the changes (Ctrl+X) and quit

1. **Install JDK 8**
   1. Go to [The Java JDK Download Page](http://www.oracle.com/technetwork/java/javase/downloads)
   2. Accept License Agreement and download the jdk-8u73-linux-x64.tar.gz (or latest stable version)
   3. Extract the file into your home directory
   4. Set the proper Java environment and change maven options:
      1. Start a new terminal session
      2. Edit .profile file (below steps a to g)
         1. Open the .profile file using

***nano ~/.profile***

* + - 1. Append the following at the end of the file:

export PATH="$HOME/jdk1.8.0\_73/bin:$PATH"

export JAVA\_HOME=$HOME/jdk1.8.0\_73

export MAVEN\_OPTS="-Xmx512m -XX:MaxPermSize=128m"

export CATALINA\_OPTS="-Xms1024m -Xmx2048m -XX:MaxPermSize=1024m"

* + - 1. Save the changes (Ctrl+X) and quit
      2. Confirm the settings are right
    1. In a new terminal, type

java -version **&&** env | grep "\(MAVEN\_OPTS\|CATALINA\_OPTS\)"

Version will show as:

*java version "1.8.0\_73"*

*Java(TM) SE Runtime Environment (build 1.8.0\_73-b02)*

*Java HotSpot(TM) 64-Bit Server VM (build 25.73-b02, mixed mode)*

*MAVEN\_OPTS=-Xmx512m -XX:MaxPermSize=128m*

*CATALINA\_OPTS=-Xms1024m -Xmx2048m -XX:MaxPermSize=1024m*

1. **Install Tomcat 7**
   1. Go to [Tomcat’s download page](http://tomcat.apache.org/download-70.cgi)
   2. Under 7.0.52 (or the latest stable version)- Binary Distributions-Core, click on tar.gz
   3. Once downloaded, expand the file to the home directory
   4. Edit the *tomcat.users.xml* file to add an admin user.
      1. In the terminal type to open the file

***nano ~/apache-tomcat-7.0.52/conf/tomcat-users.xml***

* + 1. Insert a line similar to the following before the closing </tomcat-users> tag:

*<user username="\*<usrnm>\*" password="\*<pwd>\*" roles="manager-gui"/>*

* + 1. Save the changes (Ctrl+X) then quit
    2. Edit the web.xml of the manager application(located under \webapps\manager\WEB-INF\web.xml):

nano ~/apache-tomcat-7.0.68/webapps/manager/WEB-INF/web.xml

* + 1. Edit the lines in multipart-config defining the max upload value. Change it from 50MB to a bit more, 70MB should suffice:

<!-- Before changes -->

<multipart-config>

<!-- 50MB max -->

<max-file-size>52428800</max-file-size>

<max-request-size>52428800</max-request-size>

<file-size-threshold>0</file-size-threshold>

</multipart-config>

<!-- After changes -->

<multipart-config>

<!-- 70MB max -->

<max-file-size>71680000</max-file-size>

<max-request-size>71680000</max-request-size>

<file-size-threshold>0</file-size-threshold>

</multipart-config>

* + 1. Save the changes by hitting Ctrl+X then quit
  1. Now edit *~/.bashrc* to setup tomcat’s environment variable
     1. In the terminal type to open the file

***nano ~/.bashrc***

* + 1. Append the following line:

*export CATALINA\_HOME=$HOME/apache-tomcat-7.0.52*

* + 1. Save the changes (Ctrl+X) then quit
  1. Start a new terminal session or type

***source ~/.bashrc***

1. **Setup MySQL**
   1. In MOTECH source root directory, type in the terminal:

***mysql -u root –p***

* 1. Then type:

***sql> create database motechquartz;***

***sql> create database motechschema;***

***sql> create database motechdata;***

***sql> exit;***

***sql> create user 'quartz'@'localhost' identified by 'quartz2123';***

***sql> grant all privileges on motechquartz.\* to 'quartz'@'localhost';***

***sql> exit;***

* 1. Then type:

***mysql -u root -p MOTECHquartz < <MOTECH-dir>/modules/scheduler/sql/create\_db\_schema\_quartz\_v2.1.sq***

1. **Start Tomcat**
   1. In terminal, type:

***~/apache-tomcat-7.0.52/bin/catalina.sh jpda start***

* 1. Logs in terminal will be seen as :

*Using CATALINA\_BASE: /home/\*<user>\*/apache-tomcat-7.0.52*

*Using CATALINA\_HOME: /home/\*<user>\*/apache-tomcat-7.0.52*

*Using CATALINA\_TMPDIR: /home/\*<user>\*/apache-tomcat-7.0.52/temp*

*Using JRE\_HOME: /home/\*<user>\*/jdk1.7.0\_51*

*Using CLASSPATH: /home/\*<user>\*/apache-tomcat-7.0.52/bin/bootstrap.jar:/home/\*<user>\*/...*

* 1. Confirm tomcat was started by going to [**http://localhost:8080**](http://localhost:8080/) in a browser.

# Building and Installing MOTECH

Following steps needs to be followed to setup and build MOTECH Backend.

1. Get the latest code of **MOTECH core platform** via Git cloning from MOTECH Repositories
   1. In the terminal, go to directory where cloning MOTECH code need to be done.
   2. Then type and run the following

***Motech Platform:***

***git clone*** [***https://github.com/motech/motech***](https://github.com/motech/motech)

***MOTECH Modules***

***git clone https://github.com/motech/modules***

* 1. This will clone the latest MOTECH code into the local folder

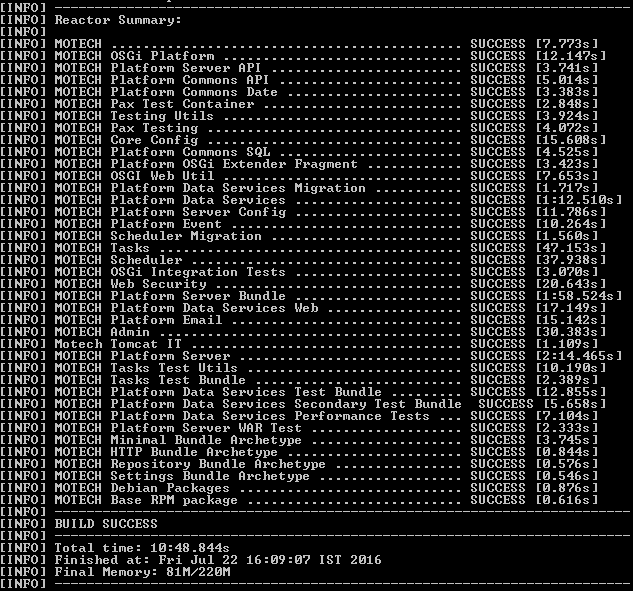
1. **Build MOTECH**
   1. Go to the directory where MOTECH code is cloned. Say the path is /usr/MOTECH, then type in terminal:

***cd /usr/MOTECH***

* 1. Then type

***mvn install***

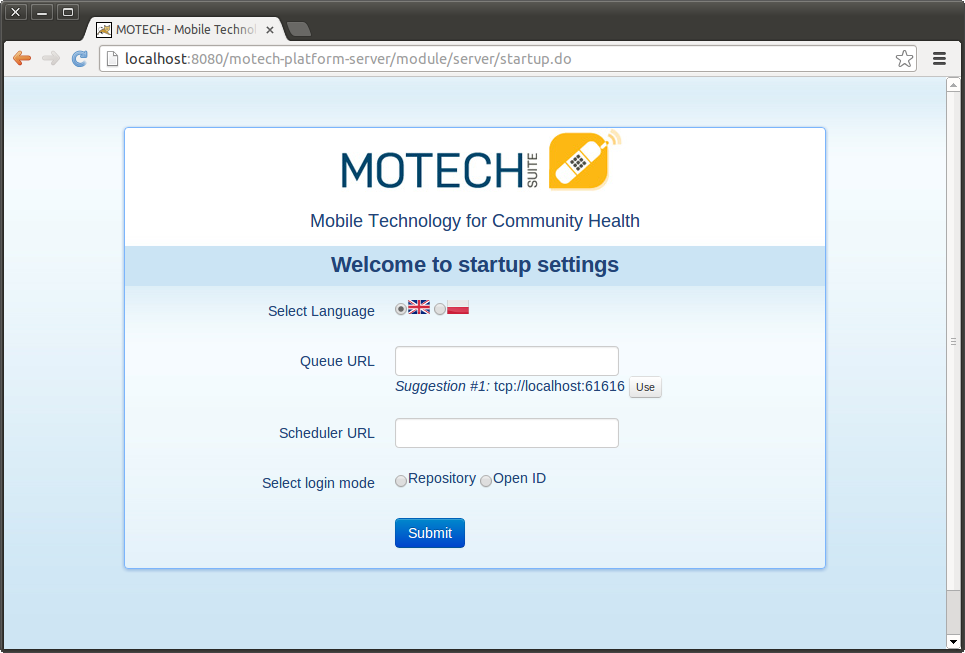
* 1. Logs in terminal will be seen as



* 1. Install MOTECH
     1. In a browser, go to [http://localhost:8080](http://localhost:8080/)
     2. Click on Manager App
     3. Type the user/password you used in tomcat-users.xml
     4. In the Tomcat Web Application Manager, scroll down to the Deploy section and the WAR file to deploy subsection, click on Browse and select or navigate to

~/MOTECH/platform/server/target/MOTECH-platform-server.war

* + 1. Click on Deploy. Depending on your machine it could take a while for MOTECH-platform-server to deploy
    2. In the Tomcat Web Application Manager page, click on /MOTECH-platform-server, you get the MOTECH initial user screen as below :

[](http://docs.motechproject.org/en/latest/_images/motech-initial-user.png)

**Note:** While setting up MOTECH, one may get a java.lang.OutOfMemoryError exception. To resolve this one may try the following:

1. Try to adjust JAVA\_OPTS environment variable to set higher maximum permanent generation size:  
   *export JAVA\_OPTS="-XX:MaxPermSize=512m"*
2. Increase –Xmx in MAVEN\_OPTS. So something like -Xmx1024m might work.

# Building and Adding Module into MOTECH Core Platform

Following steps needs to be followed to build and add a module into MOTECH Backend.

1. Get the latest code of **MOTECH Modules** via Git cloning from MOTECH Repositories
   1. In the terminal, go to directory where you want to clone MOTECH modules code
   2. Then type and run the following

***git clone https://github.com/MOTECH/modules***

* 1. This will clone the latest MOTECH-Module code into your local folder

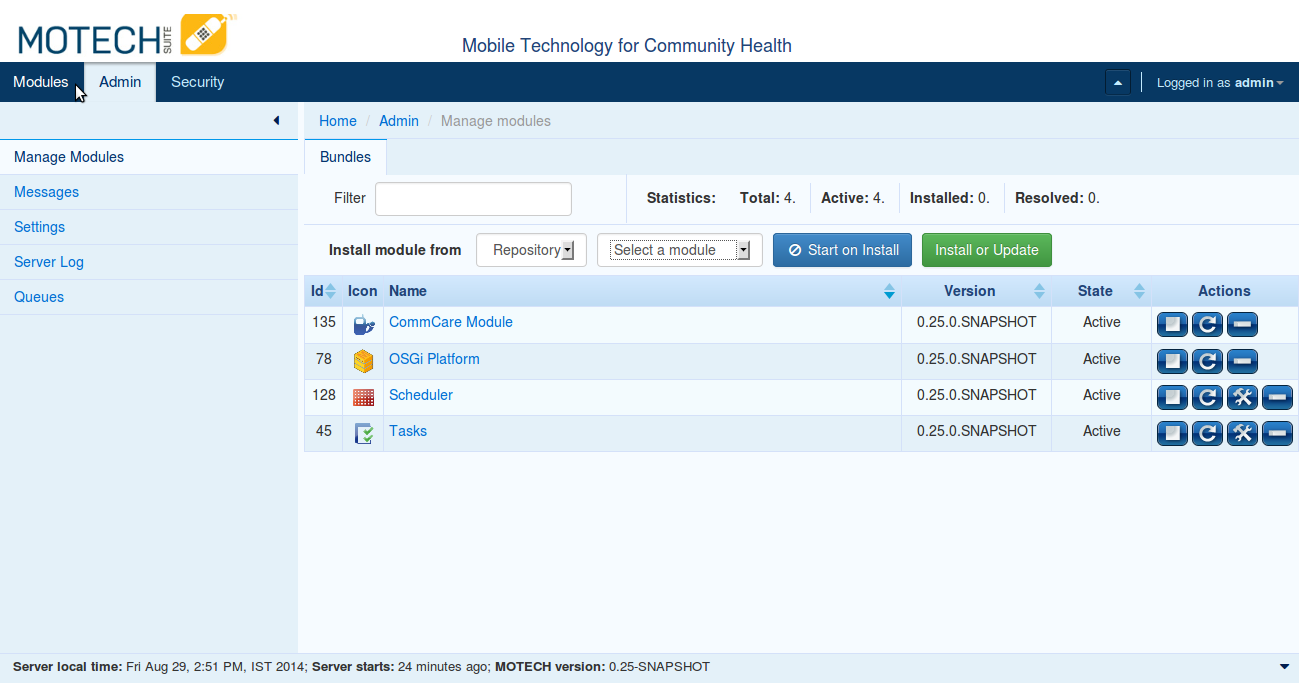
1. Go to the folder of the particular module you want to add.

For example adding the CommCare module into the MOTECH Core Platform, code is cloned the MOTECH-modules code into, */usr/modules/.* So for navigating to the CommCare module folder, type in terminal:

***cd /usr/modules/commcare***

1. Then type :

***mvn clean install***

1. This will build the modules and build success message will show on terminal.
2. Undeploy the MOTECH-platform-server project from Tomcat Web Application Manager.
3. Deploy the war file again following the steps in Step no. 3 in previous section.
4. The module will be then added into the MOTECH core. New module can be seen in the module list in the MOTECH portal under the tab “*Admin >> Manage Modules*” as in the below screenshot

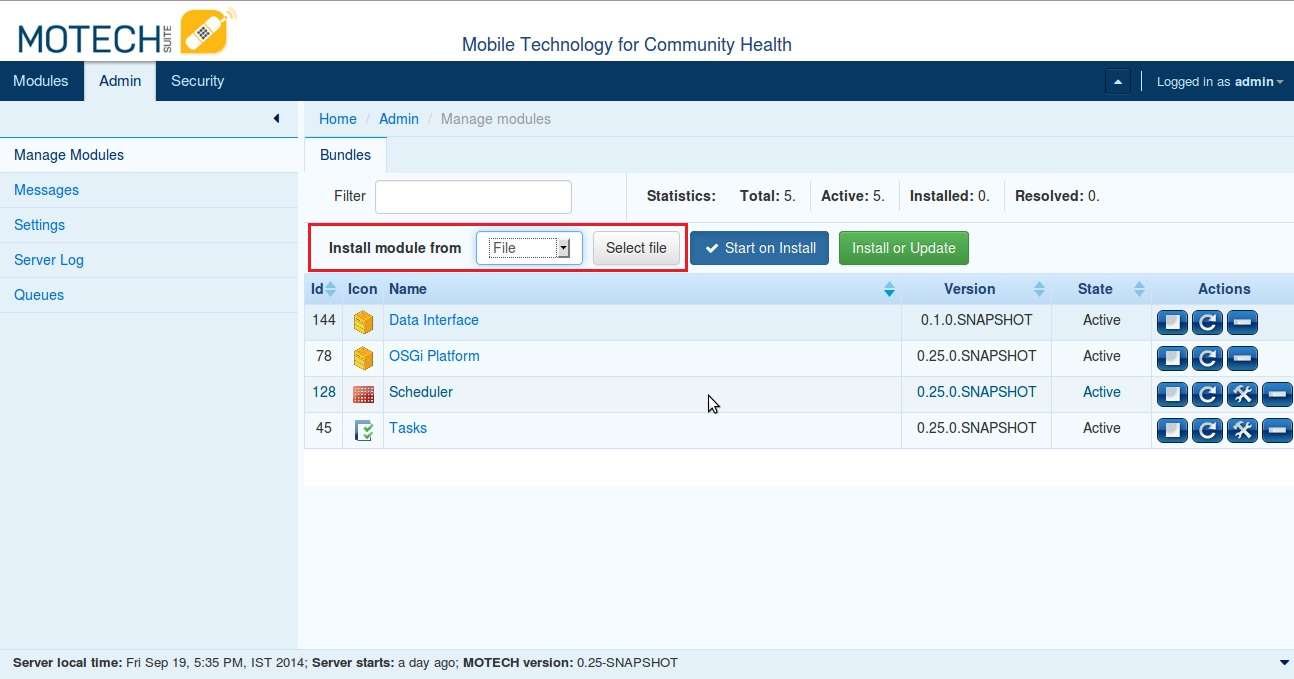
# Creating and Adding a New Custom Module

Custom module can be created to specify custom functionality and integrate with MOTECH Core Platform. To create a new module, below links can be referred:

* <http://docs.motechproject.org/en/latest/demos/hello_world.html>
* <http://docs.motechproject.org/en/latest/development/archetype.html>

**Following are the steps to integrate this new module with MOTECH Core:**

1. Go to the directory where new module is created.
2. In the terminal, type: *mvn clean install*
3. This will create a jar file in the target folder of the module.
4. Then login to the MOTECH Portal and go to “*Admin >> Manage Modules*”.
5. In the field “*Install Module From*”, select from the dropdown “*File*” (highlighted in the screenshot below)
6. Click on “*Select file*” button. Browse and select the jar file created
7. Click on the “*Install or Update*” button.
8. The module will be then added into the MOTECH core and visible in the module list.



# References

* [http://docs.MOTECHproject.org/en/latest/development/dev\_setup/index.html](http://docs.motechproject.org/en/latest/development/dev_setup/index.html)
* http://docs.motechproject.org/en/latest/get\_started/installing.html
* <http://docs.motechproject.org/en/latest/demos/hello_world.html>
* <http://docs.motechproject.org/en/latest/development/archetype.html>

**~~ end of doc ~~**