# Documentation for ThingsBoard

## Introuction

This documentation describes how to set up a ThingsBoard server instance on an Ubuntu server (whether it's a VPS or a baremetal server). Later, we also go into the details of how to integrate existing IoT devices into ThingsBoard using the MQTT Gateway API.

0. Pre-requisites
    - Set up Ubuntu Environment

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer
```

    - Set up PostgreSQL

```
sudo apt-get update
sudo apt-get install postgresql postgresql-contrib
sudo service postgresql enable
sudo service postgresql start
```

    - Add ThingsBoard DB to PostgreSQL

    Run this command in Terminal: `psql -U postgres -d postgres -h 127.0.0.1 -W`

    And then,

```
CREATE DATABASE thingsboard;
\q
```

    - Also, change the username and password of the default Postgres user

1. Download ThingsBoard
    - Run this command in Terminal :

```
wget
https://github.com/thingsboard/thingsboard/releases/download/v2.0.3/thingsboard-
2.0.3.deb
```

2. Install ThingsBoard

```
sudo dpkg -i thingsboard*.deb
```

3. Configure database in Thingsboard

- First, enter in terminal:

```
sudo gedit /etc/thingsboard/conf/thingsboard.yml
```

- Find the block of code which says `HSQLDB DAO Configuration`

  It will look like this.

```
# HSQLDB DAO Configuration
#spring:
#  data:
#    jpa:
#      repositories:
#        enabled: "true"
#  jpa:
#    hibernate:
#      ddl-auto: "validate"
#    database-platform: "org.hibernate.dialect.HSQLDialect"
#  datasource:
#    driverClassName: "${SPRING_DRIVER_CLASS_NAME:org.hsqldb.jdbc.JDBCDriver}"
#    url:
"${SPRING_DATASOURCE_URL:jdbc:hsqldb:file:${SQL_DATA_FOLDER:/tmp}/thingsboardDb;sql
.enforce_size=false}"
#    username: "${SPRING_DATASOURCE_USERNAME:sa}"
#    password: "${SPRING_DATASOURCE_PASSWORD:}"
```

- Make sure that it is commented by placing a # character before each line

  This ensures that the inbuilt HSQL instance is disabled.

- Now we need to enable the PostgreSQL instance. For this, find `PostgreSQL DAO Configuration block`

  Uncomment all the lines so that it looks similar to this

```
# PostgreSQL DAO Configuration
spring:
  data:
    jpa:
      repositories:
        enabled: "true"
  jpa:
    hibernate:
      ddl-auto: "validate"
    database-platform: "org.hibernate.dialect.PostgreSQLDialect"
  datasource:
    driverClassName: "${SPRING_DRIVER_CLASS_NAME:org.postgresql.Driver}"
    url: "${SPRING_DATASOURCE_URL:jdbc:postgresql://localhost:5432/thingsboard}"
    username: "${SPRING_DATASOURCE_USERNAME:postgres}"
    password: "${SPRING_DATASOURCE_PASSWORD:postgres}"
```

### NOTE Remember to change the password of the `postgres` user to the one that you changed to in Step 0

4. Finally, we are ready to Install thingsboard:

```
# --loadDemo option will load demo data: users, devices, assets, rules, widgets.
sudo /usr/share/thingsboard/bin/install/install.sh --loadDemo
```

**Disable the loadDemo option if you do not want the demo data!**

5. It is ready! Start the ThingsBoard service!

```
sudo service thingsboard start
```

**If you want ThingsBoard to start automatically upon a reboot, you must enable it:**

```
sudo service thingsboard enable
```

Congratulations! You have successfully set up the ThingsBoard server. You can now see the web ui on (within 5 minutes)

`http://localhost:8080`

Now, we are ready to set up the GPSAdapter to act as a bridge between the GPS Data that comes into the system through TCP and ThingsBoard which needs MQTT protocol.

For this, use the code at gitlab.com/reisub0/gpsAdapter

Since this is written in GoLang, we need to set up a few steps beforehand.

1. First, install Go

```
snap install --classic go
```

2. Set GOPATH environment variable: Add this to .bash_profile

```
export PATH="$PATH:/root/.go"
```

3. Reboot the system for the path changes to take effect.
4. Now, install the Dep GoLang package to help with the dependencies

```
curl https://raw.githubusercontent.com/golang/dep/master/install.sh | sh
```

5. We are ready to download the gpsAdapter package.

   ```
   go get -u gitlab.com/reisub0/gpsAdapter
   ```

6. Now to ensure all the dependencies are correct

   ```
   cd $GOPATH
   cd src/gitlab.com/reisub0/gpsAdapter
   dep ensure
   ```

7. Now we can install gpsAdapter by running (in the same directory)

   ```
   go install
   ```

8. Start the GPSAdapter at any time by running `gpsAdapter`

   By default, it accepts connections on port 8000 and connects to the MQTT Broker on the same localhost. All this can be configured by changing the values of constants in the code of main.go in the package gpsAdapter.