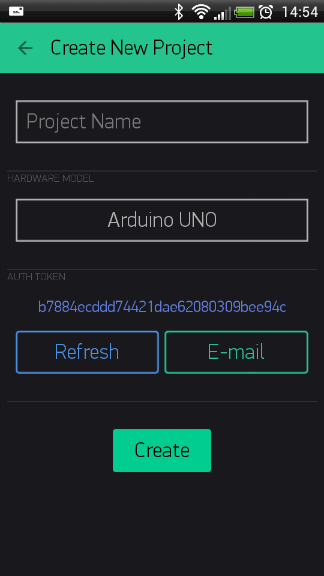
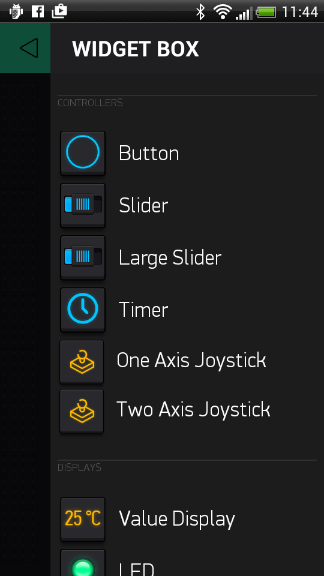
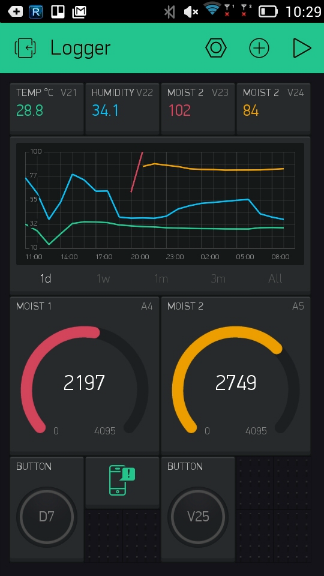
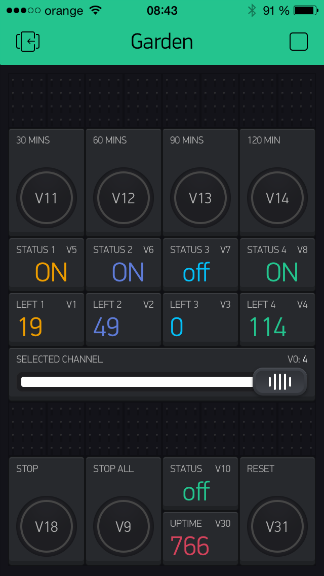
**GITHUB link :** [GitHub - blynkkk/blynk-server: Blynk is an Internet of Things Platform aimed to simplify building mobile and web applications for the Internet of Things. Easily connect 400+ hardware models like Arduino, ESP8266, ESP32, Raspberry Pi and similar MCUs and drag-n-drop IOT mobile apps for iOS and Android in 5 minutes](https://github.com/blynkkk/blynk-server#blynk-server)

[GitHub - blynkkk/blynk-server: Blynk is an Internet of Things Platform aimed to simplify building mobile and web applications for the Internet of Things. Easily connect 400+ hardware models like Arduino, ESP8266, ESP32, Raspberry Pi and similar MCUs and drag-n-drop IOT mobile apps for iOS and Android in 5 minutes](https://github.com/blynkkk/blynk-server)

**What is Blynk?**

Blynk is a platform with iOS and Android apps to control Arduino, ESP8266, Raspberry Pi and the likes over the Internet.  
You can easily build graphic interfaces for all your projects by simply dragging and dropping widgets. If you need more information, please follow these links:

* [Blynk site](https://www.blynk.io/)
* [Blynk docs](http://docs.blynk.cc/)
* [Blynk community](https://community.blynk.cc/)
* [Blynk Examples generator](https://examples.blynk.cc/)
* [Facebook](http://www.fb.com/blynkapp)
* [Twitter](http://twitter.com/blynk_app)
* [App Store](https://itunes.apple.com/us/app/blynk-control-arduino-raspberry/id808760481?ls=1&mt=8)
* [Google Play](https://play.google.com/store/apps/details?id=cc.blynk)
* [Blynk library](https://github.com/blynkkk/blynk-library)
* [Kickstarter](https://www.kickstarter.com/projects/167134865/blynk-build-an-app-for-your-arduino-project-in-5-m/description)

[](https://github.com/blynkkk/blynk-server/blob/master/docs/overview/dash_settings.png) [](https://github.com/blynkkk/blynk-server/blob/master/docs/overview/widgets_box.png) [](https://github.com/blynkkk/blynk-server/blob/master/docs/overview/dash.png) [](https://github.com/blynkkk/blynk-server/blob/master/docs/overview/dash2.png)

**Content**

* [Download](https://github.com/blynkkk/blynk-server#blynk-server)
* [Requirements](https://github.com/blynkkk/blynk-server#requirements)
* [Quick Local Server setup](https://github.com/blynkkk/blynk-server#quick-local-server-setup)
* [Enabling mail on Local server](https://github.com/blynkkk/blynk-server#enabling-mail-on-local-server)
* [Quick local server setup on Raspberry PI](https://github.com/blynkkk/blynk-server#quick-local-server-setup-on-raspberry-pi)
* [Docker container setup](https://github.com/blynkkk/blynk-server#docker-container-setup)
* [Enabling server auto restart on unix-like systems](https://github.com/blynkkk/blynk-server#enabling-server-auto-restart-on-unix-like-systems)
* [Enabling server auto restart on Windows](https://github.com/blynkkk/blynk-server#enabling-server-auto-restart-on-windows)
* [Update instruction for unix-like systems](https://github.com/blynkkk/blynk-server#update-instruction-for-unix-like-systems)
* [Update instruction for Windows](https://github.com/blynkkk/blynk-server#update-instruction-for-windows)
* [App and sketch changes for Local Server](https://github.com/blynkkk/blynk-server#app-and-sketch-changes)
* [Advanced local server setup](https://github.com/blynkkk/blynk-server#advanced-local-server-setup)
* [Administration UI](https://github.com/blynkkk/blynk-server#administration-ui)
* [HTTP/S RESTful API](https://github.com/blynkkk/blynk-server#https-restful)
* [Enabling sms on local server](https://github.com/blynkkk/blynk-server#enabling-sms-on-local-server)
* [Enabling raw data storage](https://github.com/blynkkk/blynk-server#enabling-raw-data-storage)
* [Automatic Let's Encrypt Certificates](https://github.com/blynkkk/blynk-server#automatic-lets-encrypt-certificates-generation)
* [Manual Let's Encrypt SSL/TLS Certificates](https://github.com/blynkkk/blynk-server#manual-lets-encrypt-ssltls-certificates)
* [Generate own SSL certificates](https://github.com/blynkkk/blynk-server#generate-own-ssl-certificates)
* [Install java for Ubuntu](https://github.com/blynkkk/blynk-server#install-java-for-ubuntu)
* [How Blynk Works?](https://github.com/blynkkk/blynk-server#how-blynk-works)
* [Blynk Protocol](https://github.com/blynkkk/blynk-server#blynk-protocol)

**GETTING STARTED**

**Blynk server**

Blynk Server is an Open-Source [Netty](https://github.com/netty/netty) based Java server, responsible for forwarding messages between Blynk mobile application and various microcontroller boards and SBCs (i.e. Arduino, Raspberry Pi. etc).

**Download latest server build**[**here**](https://github.com/blynkkk/blynk-server/releases)**.**

**Requirements**

* Java 8/11 required (OpenJDK, Oracle)
* Any OS that can run java
* At least 30 MB of RAM (could be less with tuning)
* Open ports 9443 (for app and hardware with ssl), 8080 (for hardware without ssl)

[Ubuntu java installation instruction](https://github.com/blynkkk/blynk-server#install-java-for-ubuntu).

For Windows download Java [here](https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html) and install.

**Quick local server setup**

* Make sure you are using Java 11
* java -version
* Output: java version "11"
* Run the server on default 'hardware port 8080' and default 'application port 9443' (SSL port)
* java -jar server-0.41.15.jar -dataFolder /path

That's it!

**NOTE: /path should be real existing path to folder where you want to store all your data.**

* As an output you should see something like that:
* Blynk Server successfully started.
* All server output is stored in current folder in 'logs/blynk.log' file.

**Enabling mail on Local server**

To enable mail notifications on Local server you need to provide your own mail credentials. Create file mail.properties within same folder where server.jar is. Mail properties:

mail.smtp.auth=true

mail.smtp.starttls.enable=true

mail.smtp.host=smtp.gmail.com

mail.smtp.port=587

mail.smtp.username=YOUR\_EMAIL\_HERE

mail.smtp.password=YOUR\_EMAIL\_PASS\_HERE

Find example [here](https://github.com/blynkkk/blynk-server/blob/master/server/notifications/email/src/main/resources/mail.properties).

WARNING : only gmail accounts are allowed.

NOTE : you'll need to setup Gmail to allow less secured applications. Go [here](https://www.google.com/settings/security/lesssecureapps) and then click "Allow less secure apps".

**Quick local server setup on Raspberry PI**

* Login to Raspberry Pi via ssh;
* Install java 8:
* sudo apt install openjdk-8-jdk openjdk-8-jre
* Make sure you are using Java 8
* java -version
* Output: java version "1.8"
* Download Blynk server jar file (or manually copy it to Raspberry Pi via ssh and scp command):
* wget "https://github.com/blynkkk/blynk-server/releases/download/v0.41.13/server-0.41.13-java8.jar"
* Run the server on default 'hardware port 8080' and default 'application port 9443' (SSL port)
* java -jar server-0.41.13-java8.jar -dataFolder /home/pi/Blynk

That's it!

* As output you will see something like that:
* Blynk Server successfully started.
* All server output is stored in current folder in 'logs/blynk.log' file.

**Docker container setup**

**Quick Launch**

* Install [Docker](https://docs.docker.com/install/)
* Run Docker container
* docker run -p 8080:8080 -p 9443:9443 mpherg/blynk-server

**Quick Launch on Raspberry Pi**

* Install [Docker](https://docs.docker.com/engine/install/debian/)
* Run Docker container
* docker run -p 8080:8080 -p 9443:9443 linuxkonsult/rasbian-blynk

**Full customisation**

* Check [README](https://github.com/blynkkk/blynk-server/blob/master/server/Docker) in docker folder

**Enabling server auto restart on unix-like systems**

* To enable server auto restart find /etc/rc.local file and add:
* java -jar /home/pi/server-0.41.13-java8.jar -dataFolder /home/pi/Blynk &
* Or if the approach above doesn't work, execute
* crontab -e

add the following line

@reboot java -jar /home/pi/server-0.41.13-java8.jar -dataFolder /home/pi/Blynk &

save and exit.

**Enabling server auto restart on Windows**

* Create bat file:
* start-blynk.bat
* Put in it one line:
* java -jar server-0.41.13.jar -dataFolder /home/pi/Blynk
* Put bat file to windows startup folder

You can also use [this](https://github.com/blynkkk/blynk-server/tree/master/scripts/win) script to run server.

**Update instruction for unix-like systems**

**IMPORTANT** Server should be always updated before you update Blynk App. To update your server to a newer version you would need to kill old process and start a new one.

* Find process id of Blynk server
* ps -aux | grep java
* You should see something like that
* username 10539 1.0 12.1 3325808 428948 pts/76 Sl Jan22 9:11 java -jar server-0.41.13.jar
* Kill the old process
* kill 10539

10539 - blynk server process id from command output above.

* Start new server [as usual](https://github.com/blynkkk/blynk-server#quick-local-server-setup)

After this steps you can update Blynk app. Server version downgrade is not supported.

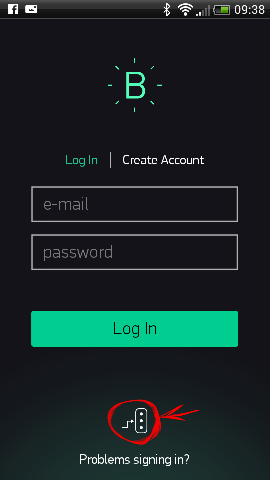
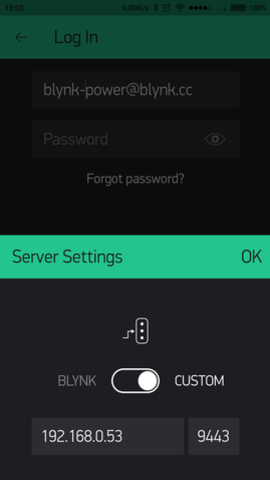
**WARNING!** Please **do not** revert your server to lower versions. You may loose all of your data.

**Update instruction for Windows**

* Open Task Manager;
* Find Java process;
* Stop process;
* Start new server [as usual](https://github.com/blynkkk/blynk-server#quick-local-server-setup)

**App and sketch changes**

* Specify custom server path in your application

[](https://github.com/blynkkk/blynk-server/blob/master/docs/login.png) [](https://github.com/blynkkk/blynk-server/blob/master/docs/custom.png)

* Change your ethernet sketch from
* Blynk.begin(auth);

to

Blynk.begin(auth, "your\_host", 8080);

or to

Blynk.begin(auth, IPAddress(xxx,xxx,xxx,xxx), 8080);

* Change your WIFI sketch from
* Blynk.begin(auth, SSID, pass));

to

Blynk.begin(auth, SSID, pass, "your\_host", 8080);

or to

Blynk.begin(auth, SSID, pass, IPAddress(XXX,XXX,XXX,XXX), 8080);

* Change your rasp PI javascript from
* var blynk = new Blynk.Blynk(AUTH, options = {connector : new Blynk.TcpClient()});

to

var blynk = new Blynk.Blynk(AUTH, options= {addr:"xxx.xxx.xxx.xxx", port:8080});

* or in case of USB when running blynk-ser.sh provide '-s' option with address of your local server
* ./blynk-ser.sh -s you\_host\_or\_IP

**IMPORTANT** Blynk is being constantly developed. Mobile apps and server are updated often. To avoid problems during updates either turn off auto-update for Blynk app, or update both local server and blynk app at same time to avoid possible migration issues.

**IMPORTANT** Blynk local server is different from Blynk Cloud server. They are not related at all. You have to create new account when using Blynk local server.

**Advanced local server setup**

For more flexibility you can extend server with more options by creating server.properties file in same folder as server.jar. Example could be found [here](https://github.com/blynkkk/blynk-server/blob/master/server/core/src/main/resources/server.properties). You could also specify any path to server.properties file via command line argument -serverConfig. You can do the same with mail.properties via -mailConfig and sms.properties via -smsConfig.

For example:

java -jar server-0.41.13-java8.jar -dataFolder /home/pi/Blynk -serverConfig /home/pi/someFolder/server.properties

Available server options:

* Blynk app, https, web sockets, admin port
* https.port=9443
* Http, hardware and web sockets port
* http.port=8080
* For simplicity Blynk already provides server jar with built in SSL certificates, so you have working server out of the box via SSL/TLS sockets. But as certificate and it's private key are in public this is totally not secure. So in order to fix that you need to provide your own certificates. And change below properties with path to your cert. and private key and it's password. See how to generate self-signed certificates [here](https://github.com/blynkkk/blynk-server#generate-ssl-certificates)
* #points to cert and key that placed in same folder as running jar.
* server.ssl.cert=./server\_embedded.crt
* server.ssl.key=./server\_embedded.pem
* server.ssl.key.pass=pupkin123

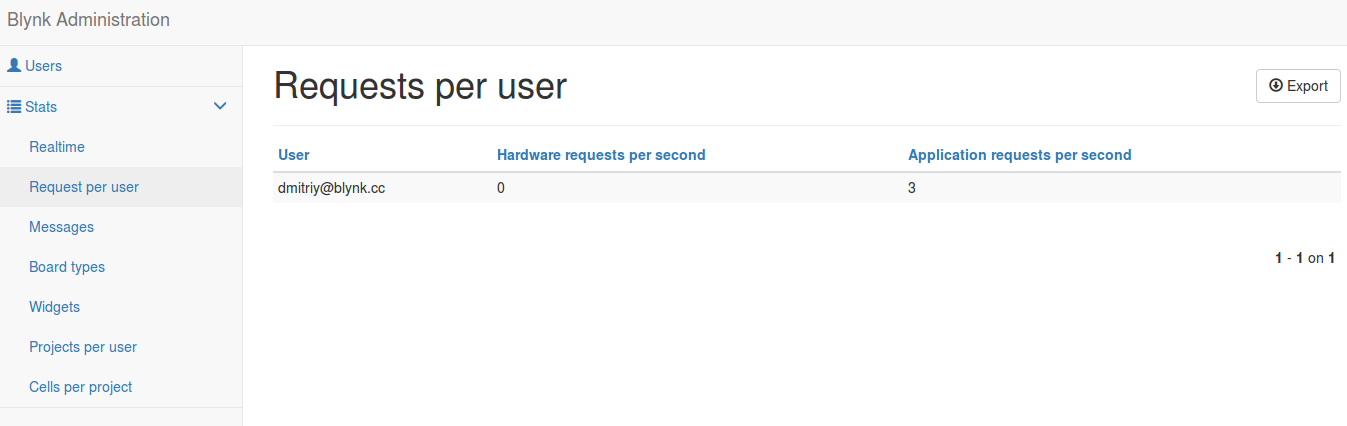
**Note**: if you use Let's Encrypt certificates you'll have to add #define BLYNK\_SSL\_USE\_LETSENCRYPT before #include <BlynkSimpleEsp8266\_SSL.h> in the Arduino Sketch for your hardware.

* User profiles folder. Folder in which all users profiles will be stored. By default System.getProperty("java.io.tmpdir")/blynk used. Will be created if not exists
* data.folder=/tmp/blynk
* Folder for all application logs. Will be created if it doesn't exist. "." is dir from which you are running script.
* logs.folder=./logs
* Log debug level. Possible values: trace|debug|info|error. Defines how precise logging will be. From left to right -> maximum logging to minimum
* log.level=trace
* Maximum allowed number of user dashboards.
* user.dashboard.max.limit=100
* 100 Req/sec rate limit per user. You also may want to extend this limit on [hardware side](https://github.com/blynkkk/blynk-library/blob/f4e132652906d63d683abeed89f5d6ebe369e37a/Blynk/BlynkConfig.h#L42).
* user.message.quota.limit=100
* this setting defines how often you can send mail/tweet/push or any other notification. Specified in seconds
* notifications.frequency.user.quota.limit=60
* Maximum allowed user profile size. In Kb's.
* user.profile.max.size=128
* Number of strings to store in terminal widget (terminal history data)
* terminal.strings.pool.size=25
* Maximum allowed number of notification queue. Queue responsible for processing email, pushes, twits sending. Because of performance issue - those queue is processed in separate thread, this is required due to blocking nature of all above operations. Usually limit shouldn't be reached
* notifications.queue.limit=5000
* Number of threads for performing blocking operations - push, twits, emails, db queries. Recommended to hold this value low unless you have to perform a lot of blocking operations.
* blocking.processor.thread.pool.limit=6
* Period for flushing all user DB to disk. In millis
* profile.save.worker.period=60000
* Specifies maximum period of time when hardware socket could be idle. After which socket will be closed due to non activity. In seconds. Leave it empty for infinity timeout
* hard.socket.idle.timeout=15
* Mostly required for local servers setup in case user want to log raw data in CSV format. See [raw data] (#raw-data-storage) section for more info.
* enable.raw.data.store=true
* Url for opening admin page. Must start from "/". For "/admin" url path will look like that "<https://127.0.0.1:9443/admin>".
* admin.rootPath=/admin
* Comma separated list of administrator IPs. Allow access to admin UI only for those IPs. You may set it for 0.0.0.0/0 to allow access for all. You may use CIDR notation. For instance, 192.168.0.53/24.
* allowed.administrator.ips=0.0.0.0/0
* Default admin name and password. Will be created on initial server start
* admin.email=admin@blynk.cc
* admin.pass=admin
* Host for reset password redirect and certificate generation. By default current server IP is taken from "eth" network interface. Could be replaced with more friendly hostname. It is recommended to override this property with your server IP to avoid possible problems of host resolving.
* server.host=blynk-cloud.com
* Email used for certificate registration, could be omitted in case you already specified it in mail.properties.
* contact.email=pupkin@gmail.com

**Administration UI**

Blynk server provides administration panel where you can monitor your server. It is accessible at this URL:

https://your\_ip:9443/admin

[](https://github.com/blynkkk/blynk-server/blob/master/docs/admin_panel.png)

**WARNING** Please change default admin password and name right after login to admin page. **THIS IS SECURITY MEASURE**.

**WARNING** Default allowed.administrator.ips setting allows access for everyone. In other words, administration page available from any other computer. Please restrict access to it via property allowed.administrator.ips.

**Turn off chrome https warning on localhost**

* Paste in chrome
* chrome://flags/#allow-insecure-localhost
* You should see highlighted text saying: "Allow invalid certificates for resources loaded from localhost". Click enable.

**HTTP/S RESTful**

Blynk HTTP/S RESTful API allows to easily read and write values to/from Pins in Blynk apps and Hardware. Http API description could be found [here](http://docs.blynkapi.apiary.io/).

**Enabling sms on local server**

To enable SMS notifications on Local Server you need to provide credentials for SMS gateway (currently Blynk server supports only 1 provider - [Nexmo](https://www.nexmo.com/). You need to create file sms.properties within same folder where server.jar is.

nexmo.api.key=

nexmo.api.secret=

And fill in the above properties with the credentials you'll get from Nexmo. (Account -> Settings -> API settings). You can also send SMS over email if your cell provider supports that. See [discussion](http://community.blynk.cc/t/sms-notification-for-important-alert/2542) for more details.

**Enabling raw data storage**

By default raw data storage is disabled (as it consumes disk space a lot). When you enable it, every Blynk.virtualWrite command will be saved to DB. You will need to install PostgreSQL Database (**minimum required version is 9.5**) to enable this functionality:

**1. Enabling raw data on server**

Enable raw data in server.properties :

enable.db=true

enable.raw.db.data.store=true

**2. Install PostgreSQL. Option A**

sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb\_release -cs`-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'

wget -q https://www.postgresql.org/media/keys/ACCC4CF8.asc -O - | sudo apt-key add -

sudo apt-get update

sudo apt-get install postgresql postgresql-contrib

**2. Install PostgreSQL. Option B**

sudo apt-get update

apt-get --no-install-recommends install postgresql-9.6 postgresql-contrib-9.6

**3. Download Blynk DB script**

wget https://raw.githubusercontent.com/blynkkk/blynk-server/master/server/core/src/main/resources/create\_schema.sql

wget https://raw.githubusercontent.com/blynkkk/blynk-server/master/server/core/src/main/resources/reporting\_schema.sql

**4. Move create\_schema.sql and reporting\_schema.sql to temp folder (to avoid permission problems)**

mv create\_schema.sql /tmp

mv reporting\_schema.sql /tmp

Result:

/tmp/create\_schema.sql

/tmp/reporting\_schema.sql

Copy it to clipboard from your console.

**5. Connect to PostgreSQL**

sudo su - postgres

psql

**6. Create Blynk DB and Reporting DB, test user and tables**

\i /tmp/create\_schema.sql

\i /tmp/reporting\_schema.sql

/tmp/create\_schema.sql - is path from step 4.

You should see next output:

postgres=# \i /tmp/create\_schema.sql

CREATE DATABASE

You are now connected to database "blynk" as user "postgres".

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE TABLE

CREATE ROLE

GRANT

GRANT

**Quit**

\q

Now start your server and you should see next text in postgres.log file :

2017-03-02 16:17:18.367 - DB url : jdbc:postgresql://localhost:5432/blynk?tcpKeepAlive=true&socketTimeout=150

2017-03-02 16:17:18.367 - DB user : test

2017-03-02 16:17:18.367 - Connecting to DB...

2017-03-02 16:17:18.455 - Connected to database successfully.

WARNING: Raw data may consume your disk space very quickly!

**CSV data format**

Data format is:

value,timestamp,deviceId

For example:

10,1438022081332,0

Where 10 - value of pin. 1438022081332 - the difference, measured in milliseconds, between the current time and midnight, January 1, 1970 UTC. To display the date/time in excel you may use formula:

=((COLUMN/(60\*60\*24)/1000+25569))

0 - device id

**Automatic Let's Encrypt certificates generation**

Latest Blynk server has super cool feature - automatic Let's Encrypt certificates generation. However, it has few requirements:

* Add server.host property in server.properties file. For example :
* server.host=myhost.com

IP is not supported, this is the limitation of Let's Encrypt. Also have in mind that myhost.com should be resolved by public DNS severs.

* Add contact.email property in server.properties. For example :
* contact.email=test@gmail.com
* You need to start server on port 80 (requires root or admin rights) or make [port forwarding](https://github.com/blynkkk/blynk-server#port-forwarding-for-https-api) to default Blynk HTTP port - 8080.

That's it! Run server as regular and certificates will be generated automatically.

**Manual Let's Encrypt SSL/TLS Certificates**

* First install [certbot](https://github.com/certbot/certbot) on your server (machine where you going to run Blynk Server)
* wget https://dl.eff.org/certbot-auto
* chmod a+x certbot-auto
* Generate and verify certificates (your server should be connected to internet and have open 80/443 ports)
* ./certbot-auto certonly --agree-tos --email YOUR\_EMAIL --standalone -d YOUR\_HOST

For example

./certbot-auto certonly --agree-tos --email pupkin@blynk.cc --standalone -d blynk.cc

* Then add to your server.properties file (in folder with server.jar)
* server.ssl.cert=/etc/letsencrypt/live/YOUR\_HOST/fullchain.pem
* server.ssl.key=/etc/letsencrypt/live/YOUR\_HOST/privkey.pem
* server.ssl.key.pass=

**Generate own SSL certificates**

* Generate self-signed certificate and key
* openssl req -x509 -nodes -days 1825 -newkey rsa:2048 -keyout server.key -out server.crt
* Convert server.key to PKCS#8 private key file in PEM format
* openssl pkcs8 -topk8 -v1 PBE-SHA1-2DES -in server.key -out server.enc.key

If you connect hardware with [USB script](https://github.com/blynkkk/blynk-library/tree/master/scripts) you have to provide an option '-s' pointing to "common name" (hostname) you did specified during certificate generation.

As an output you'll retrieve server.crt and server.pem files that you need to provide for server.ssl properties.

**Install java for Ubuntu**

sudo add-apt-repository ppa:openjdk-r/ppa \

&& sudo apt-get update -q \

&& sudo apt install -y openjdk-11-jdk

or if above doesn't work:

sudo apt-add-repository ppa:webupd8team/java

sudo apt-get update

sudo apt-get install oracle-java8-installer

**Port forwarding for HTTP/S API**

sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080

sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 9443

**Enabling QR generation on server**

sudo apt-get install libxrender1

**Behind wifi router**

If you want to run Blynk server behind WiFi-router and want it to be accessible from the Internet, you have to add port-forwarding rule on your router. This is required in order to forward all of the requests that come to the router within the local network to Blynk server.

**How to build**

Blynk has a bunch of integration tests that require DB, so you have to skip tests during build.

mvn clean install -Dmaven.test.skip=true

**How Blynk Works?**

When hardware connects to Blynk cloud it opens either keep-alive ssl/tls connection on port 443 (9443 for local servers) or keep-alive plain tcp/ip connection on port 8080. Blynk app opens mutual ssl/tls connection to Blynk Cloud on port 443 (9443 for local servers). Blynk Cloud is responsible for forwarding messages between hardware and app. In both (app and hardware) connections Blynk uses own binary protocol described below.

**Blynk protocol**

**Hardware side protocol**

Blynk transfers binary messages between the server and the hardware with the following structure:

| **Command** | **Message Id** | **Length/Status** | **Body** |
| --- | --- | --- | --- |
| 1 byte | 2 bytes | 2 bytes | Variable |

Command and Status definitions: [BlynkProtocolDefs.h](https://github.com/blynkkk/blynk-library/blob/7e942d661bc54ded310bf5d00edee737d0ca44d7/src/Blynk/BlynkProtocolDefs.h)

**Mobile app side protocol**

Blynk transfers binary messages between the server and mobile app with the following structure:

| **Command** | **Message Id** | **Length/Status** | **Body** |
| --- | --- | --- | --- |
| 1 byte | 2 bytes | 4 bytes | Variable |

**Websockets web side protocol**

Blynk transfers binary messages between the server and websockets (for web) with the following structure:

| **Websocket header** | **Command** | **Message Id** | **Body** |
| --- | --- | --- | --- |
|  | 1 byte | 2 bytes | Variable |

When command code == 0, than message structure is next:

| **Websocket header** | **Command** | **Message Id** | **Response code** |
| --- | --- | --- | --- |
|  | 1 byte | 2 bytes | 4 bytes |

[Possible response codes](https://github.com/blynkkk/blynk-server/blob/master/server/core/src/main/java/cc/blynk/server/core/protocol/enums/Response.java#L12). [Possible command codes](https://github.com/blynkkk/blynk-server/blob/master/server/core/src/main/java/cc/blynk/server/core/protocol/enums/Command.java#L12)

Message Id and Length are [big endian](http://en.wikipedia.org/wiki/Endianness#Big-endian). Body has a command-specific format.