SOP: MedalHub Installation

# Purpose and scope

In this document, you will learn how to install and configure medalHub on a Raspberry PI.

LocalData is the relay between the tablets, serves as storage, and syncs the data with the National server (mainData). The purpose of LocalData is 3-fold:

1. Relay between the tablets, thus allowing to start a consultation on a tablet and finish it on another one (Client server)
2. Local database for the clinical data
3. Hub for the data synchronisation with the National server (mainData)

# Glossary/Definitions

|  |  |
| --- | --- |
| **SOP** | Standard Operating Procedure: Simple instructions to complete necessary tasks / functions for the study. |
| **DYT** | Dynamic Tanzania |
| **RasPI** | Raspberry PI 4 (Latest Version) |
|  |  |

# Responsibilities and Procedures

**All of the following procedures must be executed by the IT Coordinator**

1. **Set up router and WIFI and configure username and password** (SOP: DYR\_IT\_07\_Health\_Facility\_IT\_Physical\_deployment.docx)
2. **Material Needed**

* Raspberry PI 4 (For convenience and ease, it will be shortened to RasPi) [www.amazon.com/dp/B07TC2BK1X](http://www.amazon.com/dp/B07TC2BK1X/)
* A MicroSD card   
  [ww.amazon.com/dp/B06XWMQ81P](https://www.amazon.com/dp/B06XWMQ81P)
* An SD card reader for the PC  
  [www.amazon.com/dp/B07G5JV2B5](http://www.amazon.com/dp/B07G5JV2B5/)
* An HDMI to micro-HDMI cable (NOT mini)  
  [www.amazon.com/dp/B00Z07JYLE](http://www.amazon.com/dp/B00Z07JYLE/)
* A PowerBank (5v & 3A outputs)  
  [www.amazon.com/dp/B01JIWQPMW](http://www.amazon.com/dp/B01JIWQPMW/)
* A Power Adapter (USB + Power)  
  [www.amazon.com/dp/B07K1K1F65](http://www.amazon.com/dp/B07K1K1F65)
* An internet connection (either cable or Wi-Fi)
* A PC connected to internet, with admin rights
* A screen with HDMI connection
* A USB keyboard

## SD Card setup

* Download “BalenaEtcher” from: <https://www.balena.io/etcher/>
* Download the HypriotOS image: <https://blog.hypriot.com/downloads/>
* Plug in your SD card reader with the SD inside
* SD card already formatted (accept default format option for SD card if not formatted, use quickflash)
* Start BalenaEtcher and select the SD Card and the .img (or .img.xz) for HypriotOS
* Click “Flash” and wait

## Setup

* Put the SD card back in the RasPI
* Connect the RasPI to the following:
  + Connect to HDMI (0)

A picture containing text, electronics

Description automatically generated

* + Ethernet (if present)
  + A Keyboard
  + A Screen (Micro-HDMI)
  + The PowerBank (Plugged in the power Brick)
* Wait until you see “login” ls a blinking underscore (after about 4 minutes) that means you are ready to be used, If no login then repeat steps above

5. Wi-Fi Setup  
Enter the credentials (press enter between each and be careful: the keyboard is UK):

* Username: pirate
* Password: hypriot

The next step is to create the wpa\_supplicant.conf file that contains the SSIDs and passwords (do not forget the –a option for the second execution of the tee command).

wpa\_passphrase health\_facility\_dynamic SbqCdzBoW8amSx | sudo tee /etc/wpa\_supplicant.conf

If in offline facility also configure (see Dynamic\_Study SOP (Wireless connectivity))

wpa\_passphrase mobile\_dynamic yicoxZ8afL2MX6 | sudo tee –a /etc/wpa\_supplicant.conf

After that, we have to give a name to each of the network configurations. In order to do that, open /etc/wpa\_supplicant.conf (sudo nano /etc/ wpa\_supplicant.conf) **and add the two id\_str lines (do not change the rest of the content):**

network={

ssid=”health\_facility\_dynamic”

#psk=”SbqCdzBoW8amSx”

psk=…

id\_str=”hf”

}

network={

ssid=”mobile\_dynamic”

#psk=”yicoxZ8afL2MX6”

psk=…

id\_str=”mobile”

}

Save file using 1. Control X 2. Confirm Y 3. Press Enter

After that, we have to link the wpa\_supplicant.conf configuration to the wlan0 interface. Open /etc/network/interfaces (sudo nano /etc/network/interfaces) and edit the file to make it exactly as below:

#source /etc/network/interfaces.d/\*

allow-hotplug wlan0

iface wlan0 inet manual

wpa-roam /etc/wpa\_supplicant.conf

iface hf inet dhcp

iface mobile inet dhcp

Save file using 1. Control X 2. Confirm Y 3. Press Enter

After that, we can activate the wlan0 interface:

sudo ifup wlan0

The device should now be connected to the internet. Run:

sudo ping google.com

in order to verify this. (Control C to exit ping command).

Now that the connection is established, we have to take note of some of the connection details in order to establish a connection with a static IP. Run:

ifconfig

and take note of the following values (under **wlan0** interface)

* inet (the assigned IP)
* netmask
* broadcast address

These values will be useful later. Run:

cat /etc/resolv.conf

and take note of the nameserver address (the first address that appears).

Run:

sudo nano /etc/network/interfaces

Make the necessary changes (under hf interface):

#source /etc/network/interfaces.d/\*

allow-hotplug wlan0

iface wlan0 inet manual

wpa-roam /etc/wpa\_supplicant.conf

iface hf inet static

address 178.17.54.136

netmask 255.255.255.0

network 178.17.54.0

broadcast 178.17.54.255

gateway 178.17.54.1

dns-nameservers 178.17.54.1

iface mobile inet dhcp

Replace the relevant values with the ones from the previous step. The network value is the bitwise AND of address and netmask. The gateway value is usually the network value with a 1 at the end. The dns-nameservers address is the address that we saw in /etc/resolv.conf.

Make sure to save the changes and reboot

sudo shutdown –r now

The connection should now be established (again).

## Application setup

* 1. Clone the repository and checkout the *master* branch

git clone <https://bitbucket.org/wavemind_swiss/liwi-local-data.git>

cd liwi-local-data

git checkout master

* 1. Create the environment file and set the health facility ID.

cp .env.example .env

nano .env

* 1. Install the project nginx, php and other requirements of the project

sudo apt update

sudo apt upgrade

sudo apt install nginx php composer php-zip php-xml php-fpm php-pgsql

* 1. Start the docker container, install dependencies and prepare the database

docker-compose up –d

composer install

php artisan key:generate

php artisan migrate

* 1. Open the nginx configuration file (sudo nano /etc/nginx/nginx.conf) and insert the following server block (right after the # Virtual Host Configs)

The two #include statements after the server block should be commented out

#include /etc/nginx/conf.d/\*.conf;

#include /etc/nginx/sites-enabled/\*;

server {

listen 80;

listen 443 ssl;

ssl\_certificate /home/pirate/liwi-local-data/docker-compose/nginx/srv.crt;

ssl\_certificate\_key /home/pirate/liwi-local-data/docker-compose/nginx/srv.key;

#server\_name example.com;

root /home/pirate/liwi-local-data/public;

add\_header X-Frame-Options "SAMEORIGIN";

add\_header X-Content-Type-Options "nosniff";

index index.php;

charset utf-8;

location / {

try\_files $uri $uri/ /index.php?$query\_string;

}

location = /favicon.ico { access\_log off; log\_not\_found off; }

location = /robots.txt { access\_log off; log\_not\_found off; }

error\_page 404 /index.php;

client\_max\_body\_size 0;

location ~ \.php$ {

fastcgi\_pass unix:/var/run/php/php7.3-fpm.sock;

fastcgi\_param SCRIPT\_FILENAME $realpath\_root$fastcgi\_script\_name;

include fastcgi\_params;

}

location ~ /\.(?!well-known).\* {

deny all;

}

}

* 1. We can verify that the nginx configuration is valid using the following command:

sudo nginx -t

* 1. After that, we have to change the permissions of the project folder so that each component of the application (the web server and the database) can access the required files.

sudo chown -R www-data:www-data /home/pirate/liwi-local-data

sudo chown -R root:root /home/pirate/liwi-local-data/postgres-data

sudo usermod -a -G www-data pirate

* 1. Restart the docker container

docker-compose down

docker-compose up -d

* 1. Restart nginx

sudo systemctl restart nginx

* 1. Go to http://<server-ip>/api/medical\_cases and make sure the server is running (it should return an empty JSON array [])

***Server is now up and running. The instructions below are for maintenance setup:***

**Update script**

We are now going to create a script to update the application from the git repository

touch ~/update.sh

chmod +x ~/update.sh

nano ~/update.sh

The content of the script should look like this:

#!/bin/sh

cd /home/pirate/liwi-local-data

docker-compose down

git pull

docker-compose up -d

composer install

php artisan key:generate

php artisan migrate

**Crontab setup**

We need to configure cron jobs so that Laravel can run synchronization jobs and that the update script gets executed everyday. Open the crontab with sudo crontab –e and add the following jobs:

\* \* \* \* \* php /home/pirate/liwi-local-data/artisan schedule:run

30 16 \* \* \* /home/pirate/update.sh >> /home/pirate/update.log 2>&1

The exact time of the update can be adjusted based on the health facility requirements (in this example it is set to update at 16:30 everyday).

## Sync configuration

Open the configuration file for synchronization (nano config/synchronization.php) and choose the appropriate configuration options.

* offline\_mode and daily\_sync\_time
  + Choose false for offline\_mode if a reliable connection is expected to be available throughout the day. In that case, the hub makes one attempt of pushing the cases to main data every day at a fixed time. The exact time at which the synchronization happens can be configured with daily\_sync\_time.
  + Choose true if a reliable connection is only available from time to time, e.g. during a few minutes each day. In that case, the hub will attempt to push the cases as soon as a connection can be established with the main data server.
* sync\_url and sync\_domain
  + These values are used to reach the main data server.
* offline\_outofsync\_thr and online\_outofsync\_thr
  + This value is the number of hours after which the cases are considered out-of-sync (starting from creation time). Synchronization will be triggered as soon as any closed case is out-of-sync and connection is available. The value for online mode should to be much higher than the one for offline mode because a reliable connection is expected to be available in online mode. For this reason, cases in online mode only become out-of-sync when the daily scheduled synchronization cannot take place for some reason.

## Synchronization using mobile hotspot

In the case where no connection is available at the health facility, it is possible to perform a synchronization with the help of a mobile phone with a mobile internet connection that can act as a hotspot. The procedure is the following:

* The hotspot should be configured with the ssid and the password specified in wpa\_supplicant.conf. The raspberry will automatically connect to a network with this configuration.
* As soon as the connection is established, the hub will push the cases to the main server.
* In order to make sure that the transaction completed successfully, logs are available under liwi\_local\_data/storage/logs/laravel.log

# Administrative information

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Signature** | **Date** |
| **Author** | Surbeck Léon |  | 08.12.2020 |
| **Revised by** | Maillard Benoît |  | 10.02.2021 |
| **Approved by** |  |  |  |
|  |  |  |  |

**Document change control**:

| **Version** | **Changes** | **Name** | **Date** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Distribution of SOP:**

| **Name** | **Role** | **Health facility / Institute** | **Date** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*Note: When printing and distributing the SOP, the administrative information page does not need to be included.