DATRO Wiki

Latest Release | Version 0.0.0

Author(s): DATRO Consortium

Jul 26, 2021

CONTENTS

CHAPTER

ONE

HOME > HOME > HOME

Welcome to the DATRO Wiki! Our mission here at DATRO is to free, democratize & decentralize our ability to communicate digitally e.g. access to the internet. Right now we pay, so it's an affordable privilege, our communications mostly goes through a corporate/privately owned internet service provider, who dictate the service and terms to the public, which has usually been dictated to them by a central Government and/or bank - And in a lot of cases, to the users complete detriment and dissatisfaction. [Learn More](https://datro.world)

To join this revolution, begin with [HotspotBnB](http://hbnb.datro.world/): Our free & universal software upgrade for your wireless access point. (Also works on most other Linux Debian supported devices, such as the Raspberry Pi's (all models)). This software upgrade will allow you to install and host applications on your wireless access point. The HotspotBnB App Store features a range of cutting edge web app and a stylish and modern dashboard to access and manage them. With a few apps installed your wireless access point will transform into a smart home hub, featuring gaming, iptv & media center entertainment, energy & ipcctv monitoring and more. HotspotBnB also features an opt-in cryptocurrency mining feature, used for the Wave application.

Wave is a Decentralized Application (DApp) on the HotspotBnB App Store. It enables you to forward the cryptocurrency being mined on your home network, to your Internet Service Provider (ISP). This DApp is subsequently able to reduce your monthly internet bill. In some cases by as much as 100% e.g. free internet access. It all depends on your monthly usage of your newly upgrade wireless access point.

Then we have a few other solution which help secure, decentralize and democratize our software solutions. Here's a summary of these additional solutions DATRO is developing:

- To-Go USB - A disk image for a USB Dongle. It creates a pre-configured work environment, giving anyone anywhere the tools and development environment they need to collaborate on this project. No more need to change settings or install work software onto your laptop, just copy our persistent live disk image to a usb dongle and reboot your machine and it's just like you've visited our offices and booted up one of our workstations.
- 'Scottish Bay' Decentralised Autonomous Society (DAS) - Revenue is mined from our network and distributed to ISP's and developers fully autonomously using cryptocurrency smart contracts and the blockchain. These types of autonomous organisations are commonly known as DAO's (Decentralised Autonomous Organisations), but our stakeholders exceed 5,000 and subsequently constitute this DAO being a society and not just an organisation, hence the abbreviation DAS. The term Scottish Bay comes from the Caribbean coastline from where this technology is being developed and trialed.
- Proxy Cache - A disk image for use on offline networks. Our Proxy Cache simulates internet access, tricking the autonomous self-building HotspotBnB NetInstaller, to think there's internet when there isn't, in order for it to upgrade a wireless access point, install a mesh network application and join a nearby mesh network, and/or download Wave, to get free internet access from the second the new user joins the network.
- Monorepo - We're proud to announce that everything listed above and more, including compiled technical documents and our websites, training videos etc, are all stored in a single public directory (repository), which uses industry standard methodologies to manage changes and release (CHANGELOG, Semantic version etc)
-

CHAPTER

TWO

HBNB > APPS > PIHOLE

Manual Install

```
curl -sSL https://install.pi-hole.net | bash
```

During the setup it asks for lightpad to be installed. Say no. We will use apache as our webserver.

http://HOSTNAMEorIP/admin to access the PiHole dashboard

Go from there.

CHAPTER

THREE

HBNB > APPS > RETROPIE > BGM

Background audio solution is BGM (google it)

CHAPTER

FOUR

HBNB > APPS > RETROPIE > ES_INPUT.CFG

``` <inputConfig type="keyboard" deviceName="Keyboard" deviceGUID="-1" deviceNbAxes="0" deviceNbHats="0" deviceNbButtons="120">

<input name="a" type="key" id="115" value="1" code="168" /> <input name="b" type="key" id="97" value="1" code="168" /> <input name="down" type="key" id="1073741905" value="1" code="168" /> <input name="hotkey" type="key" id="32" value="1" code="168" /> <input name="left" type="key" id="1073741904" value="1" code="168" /> <input name="pagedown" type="key" id="1073741902" value="1" code="168" /> <input name="pageup" type="key" id="1073741899" value="1" code="168" /> <input name="right" type="key" id="1073741903" value="1" code="168" /> <input name="select" type="key" id="32" value="1" code="168" /> <input name="start" type="key" id="13" value="1" code="168" /> <input name="up" type="key" id="1073741906" value="1" code="168" />

```

``` <inputConfig type="joystick" deviceName="Virtual gamepad" deviceGUID="030000000300000003000000020000000" deviceNbAxes="2" deviceNbHats="0" deviceNbButtons="8">

<input name="a" type="button" id="0" value="1" code="304" /> <input name="b" type="button" id="1" value="1" code="305" /> <input name="down" type="axis" id="1" value="1" code="1" /> <input name="hotkey" type="button" id="6" value="1" code="314" /> <input name="left" type="axis" id="0" value="-1" code="0" /> <input name="pagedown" type="button" id="5" value="1" code="311" /> <input name="pageup" type="button" id="4" value="1" code="310" /> <input name="right" type="axis" id="0" value="1" code="0" /> <input name="select" type="button" id="6" value="1" code="314" /> <input name="start" type="button" id="7" value="1" code="315" /> <input name="up" type="axis" id="1" value="-1" code="1" /> <input name="x" type="button" id="2" value="1" code="307" /> <input name="y" type="button" id="3" value="1" code="308" />

```

CHAPTER

FIVE

HBNB > APPS > RETROPIE-MANAGER

RetroPie-Manager

Errors & Solution

The Makefile can bring up a myriad of issues, so go through it manually to fulfill it's requirements and deal with the issues 1 by 1 as they arrive. Here are some examples below:

Error: bad magic number in 'application': b'x03xf3rn' **Solution:** Remove *–no-site-packages* from manage.py

Error: bad magic number in 'project': b'x03xf3rn' **Solution:** *find . -name *.pyc -delete* **Source:** https://github.com/Miserlou/Zappa/issues/854

Error: TypeError: expected str, bytes or os.PathLike object, not NoneType **Solution:** apt-get -y update && apt-get install -y build-essential g++ gcc make git zip unzip libopenblas-dev cmake python3-dev python3-pip **Source:** https://github.com/IBM/mimkl/issues/6

CHAPTER

SIX

HBNB > APPS > VIRTUAL-GAMEPAD

Install Node.js

` curl -sL https://deb.nodesource.com/setup_9.x | sudo bash - sudo apt install -y build-essential python-dev nodejs npm npm install -g npm ` (Mines *node -v* = v14.13.0, *nodejs -v* = v10.21.0, *npm -v* = 6.14.8)

You may also need development tools to build native addons: ` sudo apt-get install gcc g++ make curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add - echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list sudo apt-get update && sudo apt-get install yarn `

Then run:

` sudo npm cache clean -f sudo npm install -g n sudo n 9 sudo npm install -g npm ` (Mines now *node -v* = v9.11.2 - nodejs & npm are the same)

Install Virtual Gamepad (Must Be Run As Root!)

` sudo -i cd / git clone https://github.com/miroof/node-virtual-gamepads cd node-virtual-gamepads npm install `

Test it out ` sudo node main.js `

Make the gamepad load at startup

` sudo npm install pm2 -g sudo pm2 start main.js # full path e.g. /home/pi/node-virtual-gamepad/main.js etc sudo pm2 startup sudo pm2 save `

EmulationStation Controller Config (in */opt/retropie/configs/all/retroarch-joypads/Virtualgamepad.cfg*)

` input_device = "Virtual gamepad" input_driver = "udev" input_r_btn = "5" input_save_state_btn = "5" input_start_btn = "7" input_exit_emulator_btn = "7" input_l_btn = "4" input_load_state_btn = "4" input_up_axis = "-1" input_a_btn = "0" input_b_btn = "1" input_reset_btn = "1" input_down_axis = "+1" input_right_axis = "+0" input_state_slot_increase_axis = "+0" input_x_btn = "2" input_menu_toggle_btn = "2" input_select_btn = "6" input_enable_hotkey_btn = "6" input_y_btn = "3" input_left_axis = "-0" input_state_slot_decrease_axis = "-0" `

Prevent “Welcome - No Gamepad Detected” on boot-up

You just need to configure a keyboard as a gamepad once. Then it seems to stop asking you on boot-up.

Troubleshooting

npm audit fix

Running this command can actually cause the gamepad not to run. So don’t run it when it suggests, not unless *sudo node main.js* fails.

other errors

Most other errors are solved by simply removing the *node_modules* directory and performing *sudo npm install* again.

CHAPTER

SEVEN

HBNB > CONFIGS > CONFIG.TXT

Below is where you'll find the default config.txt - Although looking at this, this isn't it - so don't depend on it.

```
disable_splash=1 display_default_lcd=0 disable_overscan=1 hdmi_force_hotplug=1 hdmi_ignore_edid=0xa5000080 hdmi_drive=2 config_hdmi_boost=4 dtparam=i2c_arm=on dtparam=audio=on gpu_mem_256=128 gpu_mem_512=256 gpu_mem_1024=384 gpu_mem_2048=512 gpu_mem_4096=512

[pi3] program_usb_boot_mode=1 enable_uart=1

[pi4] dtoverlay=vc4-fkms-v3d max_framebuffers=2

[all] dtoverlay=vc4-fkms-v3d
```

CHAPTER

EIGHT

HBNB > CONFIGS > FSTAB

If the MicroSD Card corrupts, use fsck/ dosfsck.

1. Place it into a Linux machine
2. Get device location with this command *lsblk* e.g. # /dev/sdb

3. Check for errors with this command *sudo dosfsck -w -r -l -a -v -t /dev/sdb1* 3. This command also works *sudo fsck -y /dev/sdb1*

CHAPTER

NINE

HBNB > CONFIGS > SYSTEM-ANALYZE

The command *systemd-analyze blame* lists bootup items and time. This is helpful for troubleshooting boot-up speed issues.

CHAPTER

TEN

HBNB > CONFIGS > SYSTEM.CONF

The system.conf file is kept in this directory */etc/systemd/*

You can append *arm_64bit=1* to invoke the 64 bit kernel

CHAPTER

ELEVEN

HBNB > CONFIGS > WPA_SUPPLICANT.CONF

1. Create *wpa_supplicant.conf* in the */boot/* directory using this command:
   sudo nano wpa_supplicant.conf
2. Past this text below into the file and fill out the Country Code, SSID & PSK:

```
country=US ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev update_config=1
network={  ssid="NETWORK-NAME" psk="NETWORK-PASSWORD"
}
```

3. Then *CTL-X* then *Y* then *ENTER* to save the file.
4. *umount ./* to unmount

CHAPTER

TWELVE

HBNB > MANUAL-BUILD-PROCESS

Install Dependencies

sudo apt-get install git

Install GUI

Get the GUI and place it onto your webserver e.g. /var/www/html/

cd /var/www/html

sudo svn co –depth infinity https://github.com/unclehowell/datro/branches/gh-pages/static/gui/ (the online interactive demo (https://datro.xyz/static/gui) source files, are the same source files as the actual GUI)

Get Started

1. visit *http://hostname/*
2. For installation instructions for the application, search for the corresponding documentation using the search tool.

CHAPTER

THIRTEEN

DOCUMENT AUTHOR(S):

13.1 DATRO Consortium