

模擬環境 for Windows

by Wade @ 20220612

準備工作

1. 環境：

- a. 安裝 git for windows, 安裝時除了勾選『On the Desktop』外, 一律以預設值安裝(一直按Next), 安裝後桌面圖示如右。
- b. 啟動 git for windows 後如次頁, 稱之為 git shell 或 bash
- c. git clone <https://github.com/wade-fs/webui-dist>
如果是前端開發者, 應該是透過 <https://github.com/wade-fs/webui>



2. 以後稱執行 q8server 的目錄為 root/ 或是 Q8Path

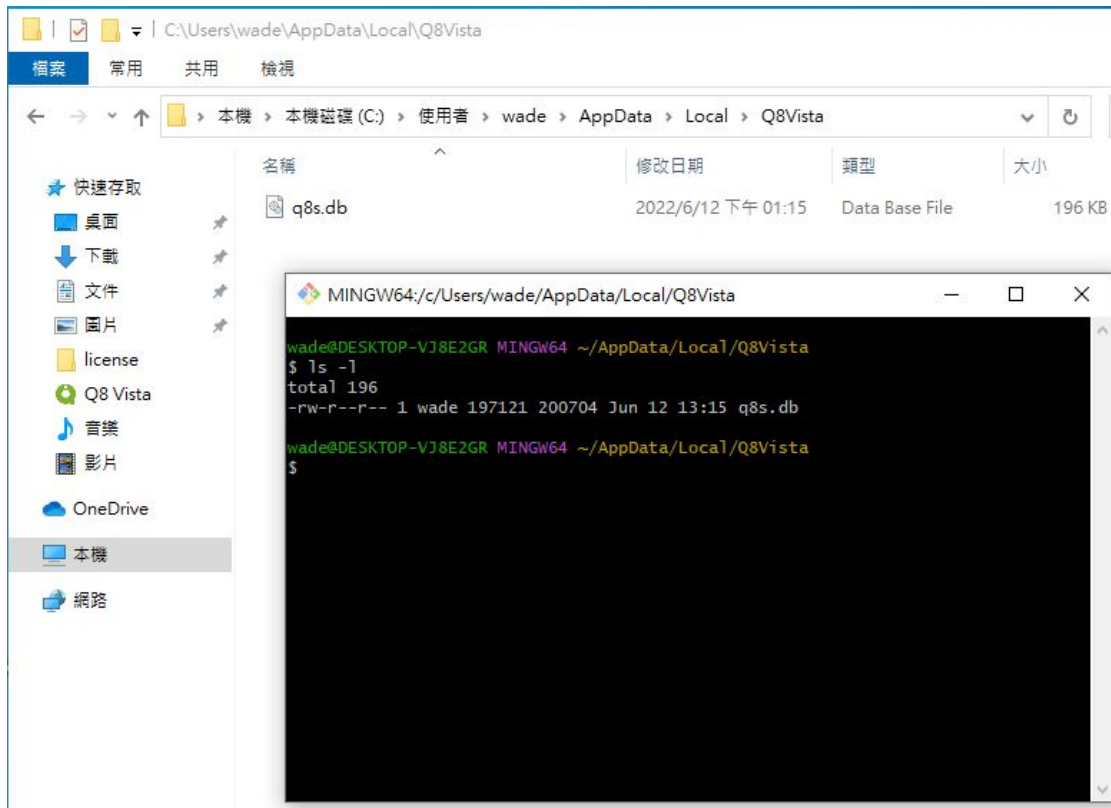
- a. 使用 webui-dist 的話, 此目錄即為 root/
- b. 使用 webui 的話, webui 為原始碼, 編譯後在 q8server/ 才是 root/

3. 執行前, 請將 DocPath/q8s.db 進行備份, DocPath 見次頁

Windows 目錄常常是被隱藏, 參考[網頁上的說明](#)以便顯示隱藏目錄/檔案

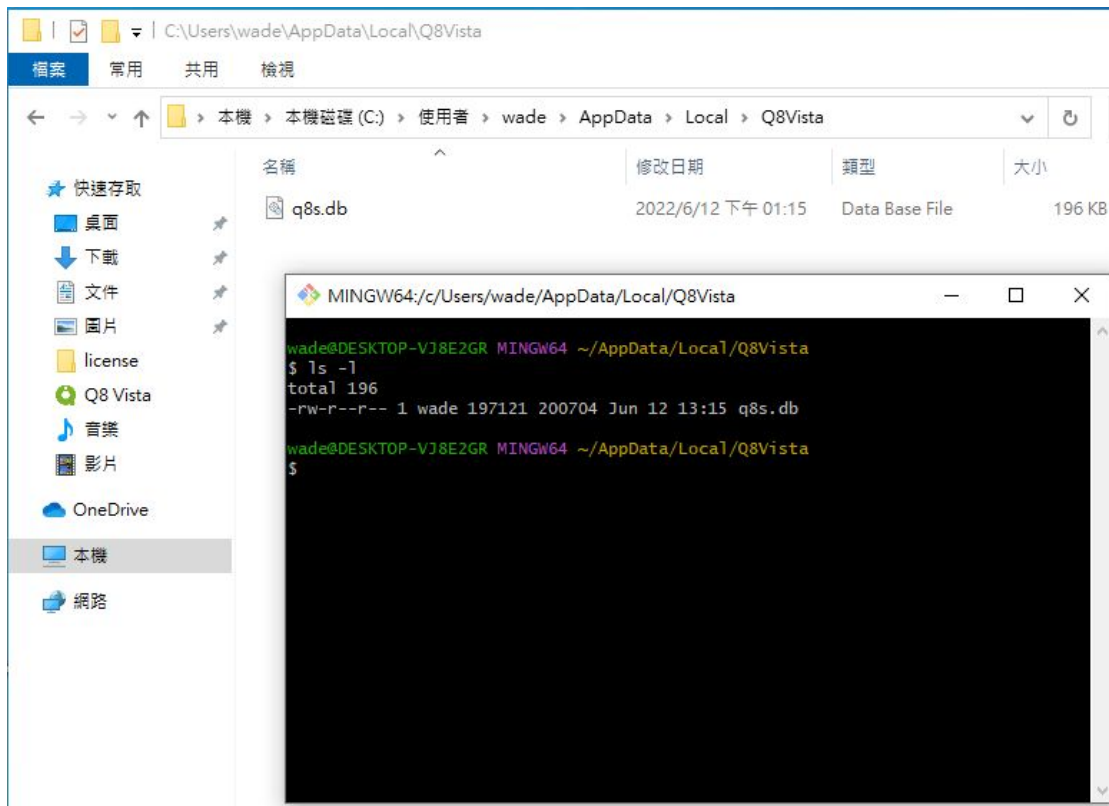
git for windows

1. git for windows 執行環境是透過 Mingw64
2. 它類似 Linux 環境, 執行 bash, 因此也可以稱之為 git shell 或是 bash
3. 資料庫存放目錄稱為「DocPath」, 在
~/AppData/Local/Q8Vista



更換資料庫 - q8s.db

1. 首先要知道，資料庫存在 DocPath/q8s.db
2. 確定停止 q8server, 再對它進行處理
3. root/q8s.db 只是作為參考，實際執行時以 DocPath/q8s.db 為準
4. 透過 <http://localhost:8088> 來確認 q8server 是否有被執行

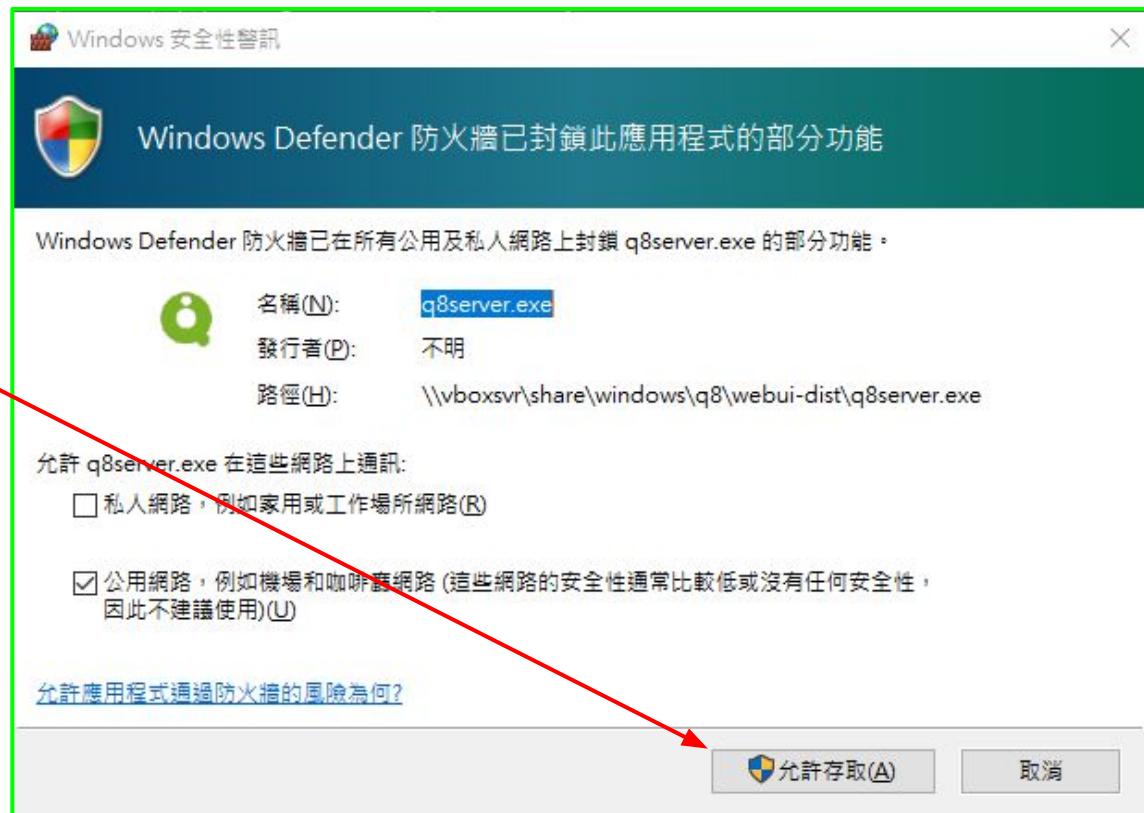


第一次執行

1. 為了方便統一說明，本文件一律以 git shell 當操作環境，而不是 windows 的 cmd 的命令視窗
2. 記得前面說的 root/，此即工作目錄，也就是要先 cd root/
 - a. 採用 webui-dist 的話: cd webui-dist
 - b. 採用 webui 的話: cd webui/q8server
 - c. 以後不會再囉嗦，只會說 `cd root/` 或是 `在 root/ 下`
3. 請執行 `./q8server.exe`，如果有跳出防火牆畫面如次頁，請按『允許存取』關於防火牆，請自行上網查詢怎麼管理，必要時可以關閉防火牆
4. 執行 `./q8server.exe` 會跑出很多訊息，此即命令視窗的特色，容易除錯，畫面參考『命令視窗執行畫面』
5. 如果使用 windows cmd 的命令視窗，畫面很類似，但是只要滑鼠有點到命令視窗，則會停止 q8server，請小心使用。
6. 請用瀏覽器瀏覽 `http://localhost:8088`，如『WebUI 畫面』

執行 q8server 的防火牆畫面

1. 防火牆是因為 q8server 需要 8088 port 及 8085 port 的網頁伺服器
2. 請點擊「允許存取」
3. 請自行研究防火牆管理，這邊純粹簡化說明至此。



命令視窗執行畫面

```
wade@DESKTOP-VJ8E2GR MINGW64 /z/windows/Q8/webui-dist
$ ./q8server.exe
E: 10:30:30 NO RESTOREDB Z:\windows\Q8\webui-dist\RESTOREDB
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.
```

```
[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env:  export GIN_MODE=release
- using code: gin.SetMode(gin.ReleaseMode)
```

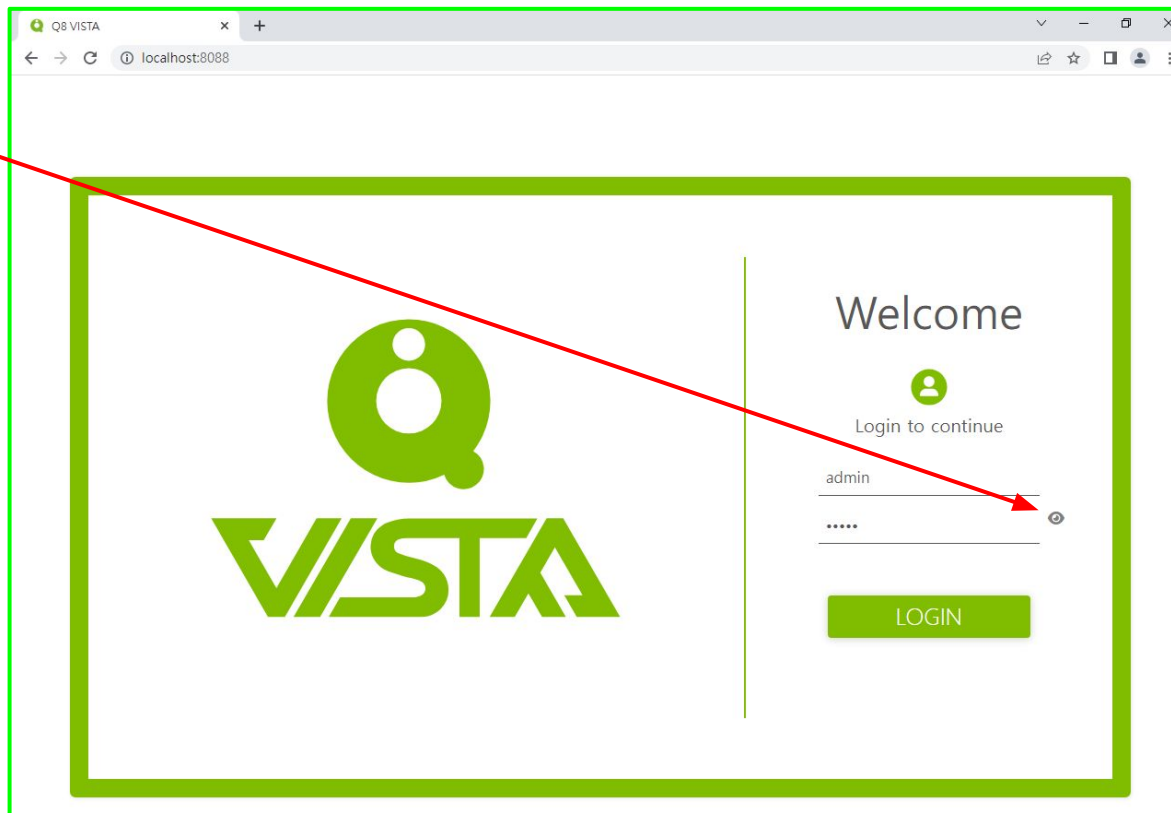
```
[GIN-debug] GET    /api/log/server/:start/:end/:level/:class --> q8server/api/log.ginGetServerLogs (5 handlers)
[GIN-debug] GET    /api/log/terminal/:start/:end/:level/:class/:tid --> q8server/api/log.ginGetTerminalLogs (5 handlers)
[GIN-debug] GET    /api/log/server/download --> q8server/api/log.ginGetServerDownloadLogs (5 handlers)
[GIN-debug] GET    /api/log/terminal/download/:tid --> q8server/api/log.ginGetTerminalDownloadLogs (5 handlers)
[GIN-debug] DELETE /api/log/terminal/:tid --> q8server/api/log.ginDelLogTerm (5 handlers)
[GIN-debug] DELETE /api/log/server --> q8server/api/log.ginDelLogSrv (5 handlers)
```

```
[GIN-debug] POST   /api/other/aduser/tlist --> q8server/api/aduser.ginPostAduserList (5 handlers)
[GIN-debug] POST   /api/other/aduser/verify --> q8server/api/aduser.ginPostAduserVerify (5 handlers)
[GIN-debug] GET    /api-q8c/dl/:target/:fn --> github.com/gin-contrib/timeout.New.func1 (5 handlers)
[GIN-debug] POST   /api-q8c/ul/:target --> q8server/api-q8c/fileop.ginPostUl (5 handlers)
[GIN-debug] GET    /api-q8c/term/:mac/:man/:model --> q8server/api-q8c/term.ginGetTerm (5 handlers)
[GIN-debug] POST   /api-q8c/term/:mac/:man/:model --> q8server/api-q8c/term.ginPostTerm (5 handlers)
[GIN-debug] POST   /api-q8c/create/:mac/:man/:model --> q8server/api-q8c/term.ginPostCreate (5 handlers)
[GIN-debug] POST   /api-q8c/replace/:mac --> q8server/api-q8c/term.ginPostReplace (5 handlers)
[GIN-debug] POST   /api-q8c/status/:mac --> q8server/api-q8c/status.ginPostStatus (5 handlers)
I: 10:30:30 q8vista version 2dc9b13 run at db: C:\Users\wade\AppData\Local\Q8Vista, Work: Z:\windows\Q8\webui-dist
[GIN-debug] I: 10:30:30 Checking Z:\windows\Q8\webui-dist/www/index.html
[GIN-debug] GET    / --> main.run_api.func1 (5 handlers)
[GIN-debug] GET    /assets/*filepath --> github.com/gin-gonic/gin.(*RouterGroup).createStaticHandler.func1 (5 handlers)
[GIN-debug] HEAD   /assets/*filepath --> github.com/gin-gonic/gin.(*RouterGroup).createStaticHandler.func1 (5 handlers)
[GIN-debug] GET    /fp/*filepath --> github.com/gin-gonic/gin.(*RouterGroup).createStaticHandler.func1 (5 handlers)
[GIN-debug] HEAD   /fp/*filepath --> github.com/gin-gonic/gin.(*RouterGroup).createStaticHandler.func1 (5 handlers)
I: 10:30:30 serve @ http://localhost:8088
[GIN-debug] Listening and serving HTTP on :8088
```

這邊的命令視窗是在 git
shell 環境下。
以後不會再特別提醒。

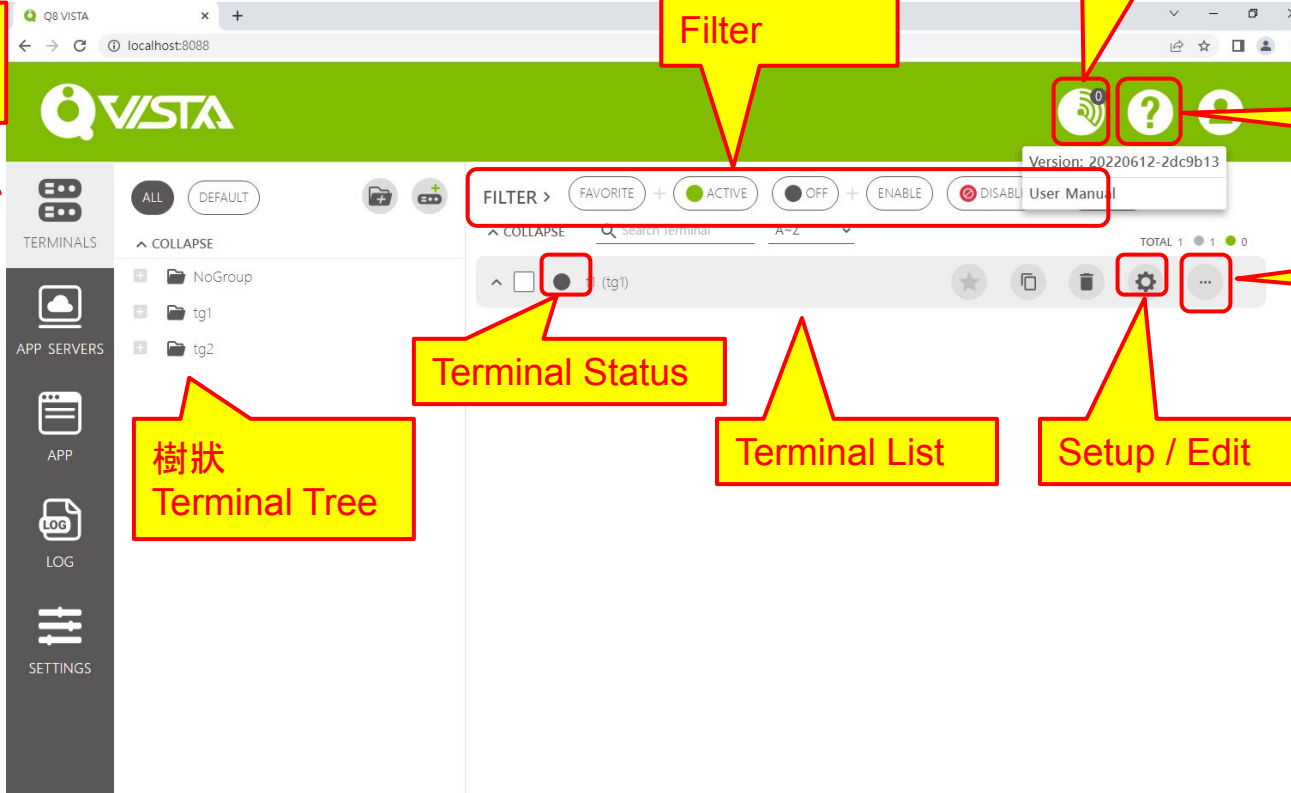
WebUI 畫面

1. admin 的密碼可以在登入後更改, 預設為 admin
2. 直接按 LOGIN 即可看到次頁的畫面



Q8 Vista 控制台 - Q8 WebUI

主功能頁
首頁



雷達

Filter

可以看版本

Terminal Status

Terminal List

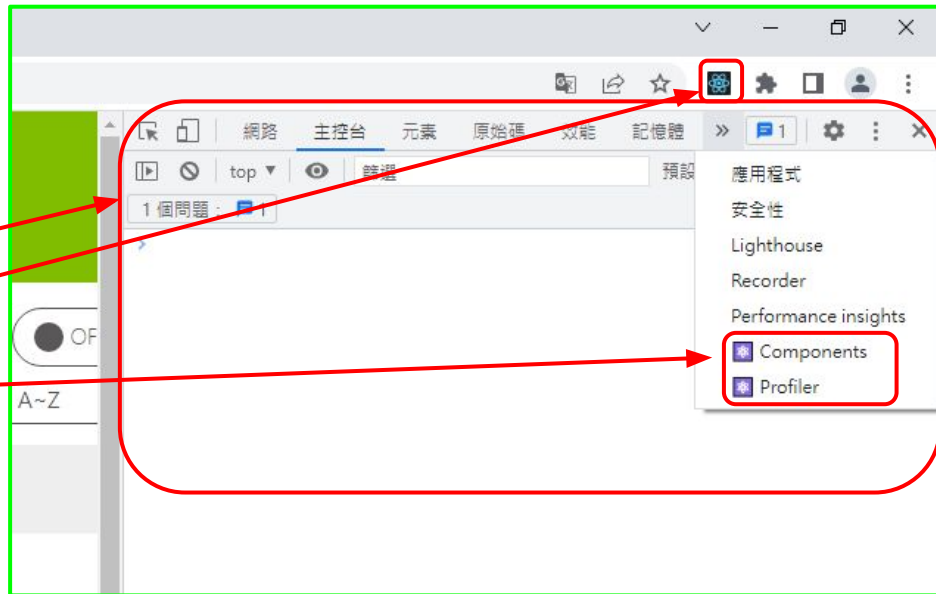
Setup / Edit

More / Op

樹狀
Terminal Tree

除錯環境 - chrome

1. 只針對前端開發者，為了方便除錯，請安裝 [React Developer Tools](#)
2. 除錯時，請按 F12，稱之為 debug console, 或 javascript console
3. 安裝完 RDT 之後，會有圖示
4. 搭配 F12 參考畫面如右
5. 關於 debug console 或 RDT 請自行上網研究



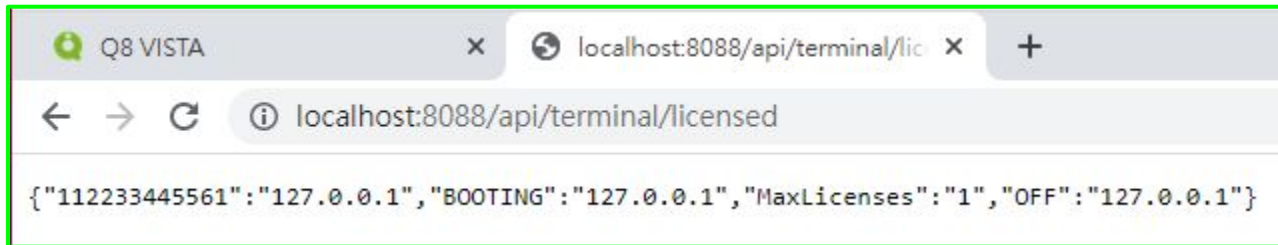
terminal > Setup

The screenshot displays the 't1-OFF' terminal setup interface. A red box highlights the top navigation bar, which includes the title 't1-OFF' and a series of menu items: ABOUT, CONFIGURATION, MODULE, SHADOW, SCHEDULE, and LOG. A yellow callout box labeled 'More / Op' points to a three-dot menu icon on the right side of this bar. Another yellow callout box labeled '功能列' (Function List) points to the main content area of the interface. The main content area is divided into several sections: 'TERMINAL INFO' (Terminal ID: 6, Firmware Version, Firmware Package: modeldefault, Hardware Model: Arista BoxPC-138-G01, Hardware Boot ROM Version, Up Time), 'CPU INFO' (Vendor, Name, Clock Speed, Cache Size, MIPS, Load Percentage), 'IP MENU SETTINGS' (IP Address, Network Router, Subnet Mask), 'PCI INFO' (Video PCI ID, Network PCI ID), and 'MEMORY INFO' (Total Memory, Free Memory).

TERMINAL INFO	CPU INFO	IP MENU SETTINGS
TERMINAL ID 6	VENDOR	IP ADDRESS
FIRMWARE VERSION	NAME	NETWORK ROUTER
FIRMWARE PACKAGE modeldefault	CLOCK SPEED	SUBNET MASK
HARDWARE MODEL Arista BoxPC-138-G01	CACHE SIZE	
HARDWARE BOOT ROM VERSION	MIPS	
UP TIME	LOAD PERCENTAGE	

MEMORY INFO	PCI INFO
TOTAL MEMORY	VIDEO PCI ID
FREE MEMORY	NETWORK PCI ID

檢查 License



1. 在操作時, 請先確認目前 License 是多少, 預設是 1
2. 建議改為 10
 - a. 主畫面 > SETTINGS > UPLOAD LICENSE > INSTALL
 - b. 挑選 root/license/ 的檔
 - c. CONFIRM
3. 參考次頁

上傳 License

The screenshot displays the Q8 VISTA web interface with the following components and annotations:

- 1) 選 SETTINGS**: Points to the SETTINGS icon in the bottom-left sidebar.
- 2) 選 INSTALL**: Points to the green 'INSTALL' button with an upward arrow in the 'LICENSE' section.
- 3) 選 license 路徑**: Points to the file path 'webui-dist > license' in the Windows File Explorer window.
- 4) 選 license 檔案**: Points to the 'Q8License-10.lic' file selected in the file list of the Windows File Explorer window.
- 5) 上傳檔案後**: Points to the 'Q8License-10.lic' file icon in the 'LICENSE' section of the web interface.

The Windows File Explorer window shows the following table of files:

名稱	修改日期	類型	大小
Q8License-0.lic	2022/6/12 下午 12:54	License	
Q8License-1.lic	2022/6/12 下午 12:54	License	
Q8License-2.lic	2022/6/12 下午 12:54	License	
Q8License-3.lic	2022/6/12 下午 12:54	License	
Q8License-4.lic	2022/6/12 下午 12:54	License	
Q8License-5.lic	2022/6/12 下午 12:54	License	
Q8License-6.lic	2022/6/12 下午 12:54	License	
Q8License-7.lic	2022/6/12 下午 12:54	License	
Q8License-8.lic	2022/6/12 下午 12:54	License	
Q8License-9.lic	2022/6/12 下午 12:54	License	
Q8License-10.lic	2022/6/12 下午 12:54	License	

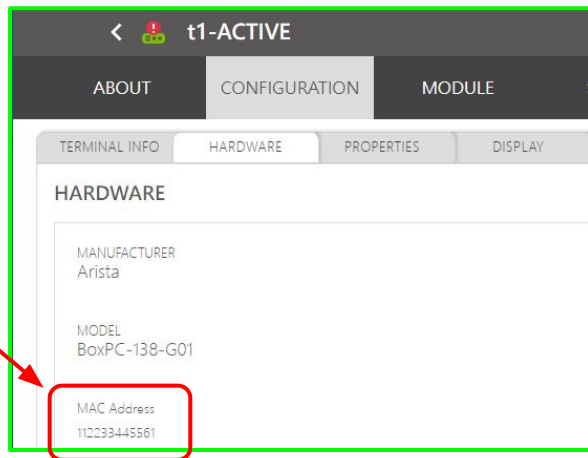
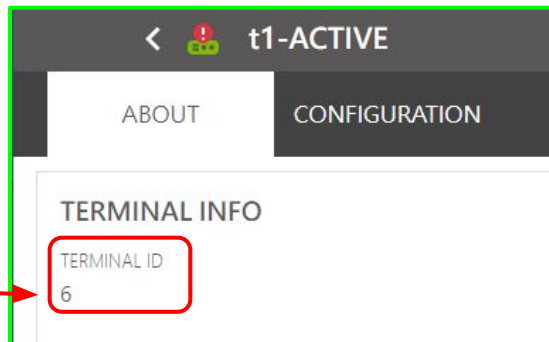
模擬硬體動作

1. 有幾個核心需要知道

- terminal.sh 本檔案是核心工具, 例如次頁畫面的
./terminal.sh -s -M 112233445561
- 查看 terminal 的資訊, 請按 Setting
 - tid
 - MAC

2. terminal status 有

- OFF
- ACTIVE
- BOOTING
- PENDING
- CREATING
- ERROR
- NeedToRestart



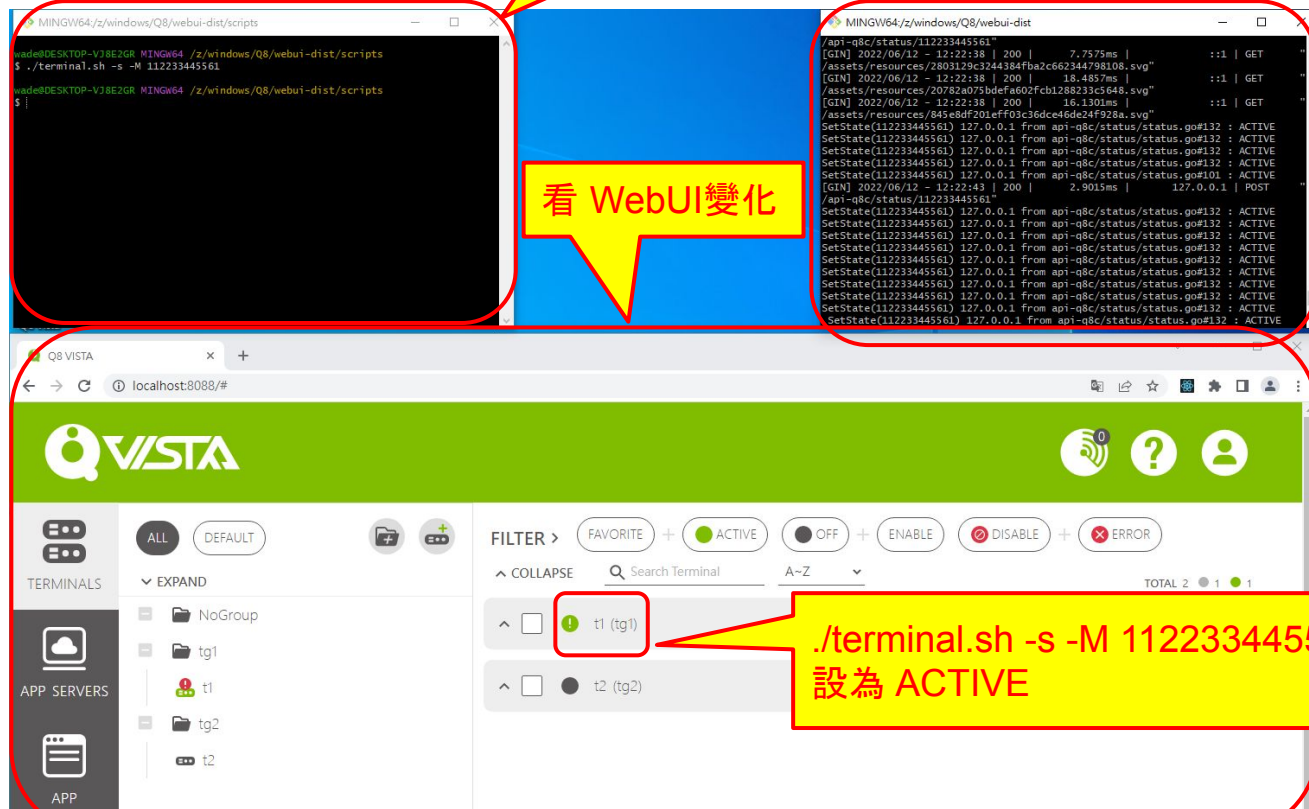
模擬硬體畫面

執行模擬命令

看 WebUI 變化

執行 q8server

`./terminal.sh -s -M 112233445561`
設為 ACTIVE



terminal.sh 語法摘要

1. 主要有幾項功能

- a. -g 取得 terminal 設定, 後述
- b. -s 設定 terminal 狀態
- c. -c Create terminal, 雷達相關
- d. -r Replace terminal

2. 參數 -l 指向 q8server 機器的 IP, 因為是同一台機器, 所以都不必給, 預設值 localhost

3. 狀態只能設

OFF/BOOTING/ACTIVE

```
wade@DESKTOP-VJ8E2GR MINGW64 /z/windows/Q8/webui-dist/scripts
$ ./terminal.sh
Usage:
./terminal.sh -g -I IP -M MAC # 取得 terminal config
case 1) 已定義, 取回目前定義值, UI 該 term 會變橘
case 2) 未定義且未出現在雷達: 傳回 replace list
case 3) 未定義, 但是出現在雷達, 傳回[]
./terminal.sh -s -I IP -M MAC [-S STATUS] # 設定 terminal status
status(預設 ACTIVE), terminal 有可能因為 timeout 變 OFF
status == OFF, 會讓實際是 active 的 terminal 暫時變 OFF
./terminal.sh -c -I IP -C COUNT
Create Terminal with COUNT of Random mac, 讓該 mac 出現在 UI 的雷達
./terminal.sh -c -I IP -M MAC
Create Terminal with MAC
./terminal.sh -r -I IP -M MAC -T TID # replace terminal TID with MAC
```


terminal.sh 語法摘要

1. 主要有幾項功能

- a. -g 取得 terminal 設定, 後述
- b. -s 設定 terminal 狀態
- c. -c Create terminal, 雷達相關
- d. -r Replace terminal

2. 參數 -l 指向 q8server 機器的 IP, 因為是同一台機器, 所以都不必給, 預設值 localhost

3. 狀態只能設 OFF/BOOTING/ACTIVE

1. `./terminal.sh -c -M 112233445563`

在雷達會出現 11 22 33 44 55 63

參考 **terminal.sh -c** 頁說明

2. `./terminal.sh -s -M 112233445561 -S OFF`

將會 t1 狀態變為 OFF, 如圖右

3. `./terminal.sh -r -M 112233445563`



將 tid 為 3 的 terminal 設定 MAC

換 q8s.db-t3-nomac, 參考「更換資料庫」

雷達如果有 112233445563 會消失

因為它已被建立, 畫面見 **terminal.sh -r** 頁

terminal.sh -c

1) 執行模擬命令

2) q8server 會API 資訊

3) 雷達會出現該 Mac

The image shows a terminal window and a web interface. The terminal window displays the command `./terminal.sh -c -M 112233445563` being executed. The web interface shows the Q8 VISTA dashboard with a list of terminals. A red box highlights the terminal entry for `t1` with IP `127.0.0.1` and status `ACTIVE`. A red box also highlights the terminal entry for `t2` with IP `127.0.0.1` and status `PENDING`. The terminal entry for `t1` is highlighted with a red box, indicating it is the active terminal.

Q8 VISTA

localhost:8088/api/terminal/lic

localhost:8088/#

Q8 VISTA

ALL DEFAULT

TERMINALS

APP SERVERS

FILTER > FAVORITE + ACTIVE OFF + ENABLE + DISABLE + ERROR

COLLAPSE Search Terminal A~Z

TOTAL 2 0 1

t1

t2

**務必更換資料庫
或確認可被更換的
terminal ID**

[illegible]

5) terminal 清單 tid=3 的狀態變成 BOOTING

4) 雷達該Mac消失

