

MagicMirror²

The open source modular smart mirror platform.

MagicMirror

Vorwort

Diese Anleitung beschreibt die Installation eines MagicMirror auf einem Raspberry Pi 3 B. Du findest hier Informationen zur Systemvorbereitung, Installation und Konfiguration des MagicMirror, Einrichtung des Prozessmanagers PM2, Integration verschiedener Module und Einrichtung der iCloud-Synchronisation für Kalenderinformationen.

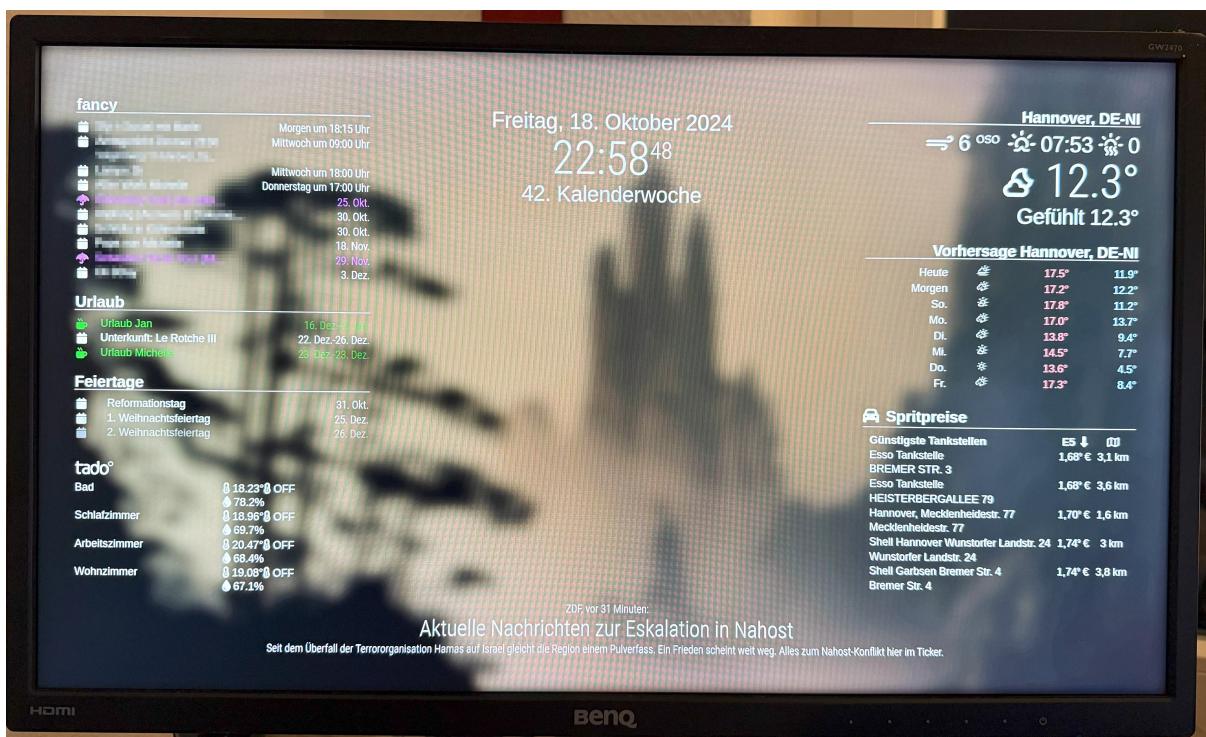
Die Anleitung ist in mehrere Abschnitte unterteilt:

- Systemvorbereitung und Software-Installation
- MagicMirror-Installation und -Konfiguration
- PM2-Einrichtung
- Modulintegration
- iCloud-Synchronisation
- Energieeinstellungen
- Troubleshooting

Die PDF-Version eignet sich nicht gut für Code-Bausteine. Deshalb habe ich einen GitHub-Link bereitgestellt, wo die einzelnen Schritte deutlich übersichtlicher und leichter nachzuvollziehen sind. Die HTML-Version ist ebenfalls gut lesbar.

GitHub:

<https://github.com/janklmm/UZ4iE1MagYUt>



Vorbereitung

System aktualisieren

```
sudo apt update
sudo apt upgrade
```

Deaktivieren des Power Management (WLAN)

Wenn der Pi über WLAN betrieben wird, schaltet sich das WLAN nach einer gewissen Zeit automatisch aus. Dies geschieht, wenn keine Daten empfangen oder gesendet werden. Der Pi geht dann in einen Ruhemodus, wodurch das WLAN abgeschaltet wird.

Überprüfen des Power Management

```
iwconfig
```

Output

```
jan@mirrorpi:~ $ iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

wlan0   IEEE 802.11 ESSID:"FRITZ!Box 6660 Cable NA"
        Mode:Managed Frequency:2.467 GHz Access Point: 09
        Bit Rate=72.2 Mb/s Tx-Power=31 dBm
        Retry short limit:7 RTS thr:off Fragment thr:off
        Power Management:on
        Link Quality=64/70 Signal level=-46 dBm
        Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
        Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

Deaktivieren des Power Management

```
sudo nano /etc/rc.local
```

```
# der Code wird unter # By default this script does nothing.
# /sbin/iwconfig wlan0 power off
```

Beispiel rc.local

```

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the exit
# bits.
#
# By default this script does nothing.
/sbin/iwconfig wlan0 power off
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

exit 0

```

Ist der Code eingefügt, das Dokument speichern und schließen anschließend

```
sudo reboot
```

Überprüfen nach Änderung

```

jan@mirrorpi:~ $ iwconfig
lo      no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:"FRITZ!Box 6660 Cable NA"
          Mode:Managed  Frequency:2.467 GHz  Access Point: 09
          Bit Rate=72.2 Mb/s  Tx-Power=31 dBm
          Retry short limit:7  RTS thr:off  Fragment thr:off
          Power Management:off

```

```
Link Quality=64/70  Signal level=-46 dBm
Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid f
Tx excessive retries:0  Invalid misc:0    Missed beacon:0
```

Sollte der Raspberry Pi nach einer gewissen Zeit dennoch das WLAN-Signal kappen, findet man ganz unten im Abschnitt "Troubleshooting" einen entsprechenden Lösungsansatz.

Deaktivieren IPv6

```
sudo nano /etc/sysctl.conf
```

```
# Ganz nach unten in das Dokument und diese zwei Zeilen einfügen
#Deaktivieren IPv6
net.ipv6.conf.all.disable_ipv6 = 1
```

```
sysctl -p
```

```
sudo nano /etc/sysctl.d/90-disable-ipv6.conf
```

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

```
sudo reboot
```

NPM

```
sudo apt install npm
```

NodeJS

Vorbereitung für nodejs

 Die "normale" Installation von nodejs installiert v19.x, MagicMirror benötigt 20.x - 22.x
Deswegen nehme ich 20.x

```
sudo apt install -y ca-certificates curl gnupg
```

```
curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo..
```

```
NODE_MAJOR=20
```

```
echo "deb [signed-by=/usr/share/keyrings/nodesource.gpg] http
```

```
sudo apt update
```

Installation nodejs

```
sudo apt install nodejs
```

MagicMirror

Installation MagicMirror

```
git clone https://github.com/MagicMirrorOrg/MagicMirror
```

```
cd MagicMirror && npm install
```

Konfigurieren der Config.js Datei

```
# Erstellen der config.js Datei
```

```
cp ~/MagicMirror/config/config.js.sample ~/MagicMirror/config
```

```
# Bearbeiten der config.js Datei  
nano ~/MagicMirror/config/config.js
```

Bearbeitung der Config.js



Will man mit einem Webbrower auf den Mirror zugreifen, muss "0.0.0.0" als HostAdresse eingegeben werden. Greif die Whitelist Regelung

Beispiel Config.js IP Einstellung

```
address: "0.0.0.0",      // Address to listen on, can be:  
                        // - "localhost", "127.0.0.1", "::1"  
                        // - another specific IPv4/6 to liste  
                        // - "0.0.0.0", ":" to listen on any  
                        // Default, when address config is le  
port: 8080,  
basePath: "/",           // The URL path where MagicMirror2 is  
                        // you must set the sub path here. ba  
ipWhitelist: ["127.0.0.1",  
                "::ffff:127.0.0.1",  
                "::1",  
                "::ffff:192.168.178.64",  
                "::ffff:192.168.178.49",  
                "192.168.178.1/24",  
                "192.168.178.49"],
```

PM2 (Prozessmanager)

Installation PM2

```
sudo npm install -g pm2
```

```
pm2 startup
```



Es gibt eine Befehlsausgabe, diesen ausgegebenen Befehl kopieren, einfügen und mit Enter drücken

```
sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 start mm.sh
```

```
cd ~
```

Erstellen der Autostart Datei

```
sudo nano mm.sh
```

```
# Festlegen wie MM starten soll, auf welchem Display
cd ./MagicMirror/
DISPLAY=:0 npm start
```

```
sudo chmod +x mm.sh
```

```
pm2 start mm.sh
```

```
pm2 save
```

Module

Diese Module sind die, die ich aktuell auf meinem MagicMirror verwende. Sie können frei angepasst werden, basierend auf den Dokumentationen in den jeweiligen GitHub-Repositories.

▼ Übersicht über ThirdParty-Module

<https://kristjanesperanto.github.io/MagicMirror-3rd-Party-Modules/>
<https://github.com/MagicMirrorOrg/MagicMirror/wiki/3rd-party-modules>

▼ MMM-Admin-Interface

<https://github.com/ItayXD/MMM-Admin-Interface>

```
cd ~/MagicMirror/modules/
```

```
git clone https://github.com/ItayXD/MMM-Admin-Interface.git
```

```
cd MMM-Admin-Interface && npm install --only=production
```

```
// in config.js
{
  "module": "MMM-Admin-Interface"
},
```

```
<HOST>:8080/MMM-Admin-Interface/
```

▼ MMM-Fuel

<https://github.com/fewieden/MMM-Fuel>

```
cd ~/MagicMirror/modules/
```

```
git clone https://github.com/fewieden/MMM-Fuel.git
```

```
cd MMM-Fuel && npm install --only=production
```

```
// in config.js
{
  "module": "MMM-Fuel",
  "position": "top_right",
```

```
        "config": {
            "lat": 52.4091985,
            "api_key": "APIKEY",
            "lng": 9.6577755,
            "radius": 5,
            "max": 5,
            "sortBy": "E5",
            "shortenText": 20,
            "rotate": false,
            "showOpenOnly": true,
            "colored": true,
            "types": [
                "e5"
            ]
        },
    },
```

▼ MMM-Tado

<https://github.com/WouterEekhout/MMM-Tado>

```
cd ~/MagicMirror/modules/
```

```
git clone https://github.com/WouterEekhout/MMM-Tado
```

```
cd MMM-Tado && npm install --only=production
```

```
// in config.js
{
    "module": "MMM-Tado",
    "header": "",
    "position": "top_left",
    "config": {
        "username": "EMAIL",
        "password": "PASSWORD",
        "updateInterval": 300000
    }
}
```

```
        },  
    },
```

▼ MMM-Wallpaper

<https://github.com/kolbyjack/MMM-Wallpaper>

```
cd ~/MagicMirror/modules/
```

```
git clone https://github.com/kolbyjack/MMM-Wallpaper.git
```

```
cd MMM-Wallpaper && npm install --only=production
```

```
// in config.js  
    {  
        "module": "MMM-Wallpaper",  
        "header": "",  
        "position": "fullscreen_below",  
        "config": {  
            "source": "chromecast",  
            "slideInterval": 6900000,  
            "maximumEntries": 30  
        }  
    },
```

iCloud Sync

Vorbereitung

```
sudo apt install libxml2 libxslt1.1 zlib1g python3 #sollte sc
```

```
sudo apt install vdirsyncer
```

Erstellen Ordner und Dateien

```
mkdir /home/$USER/MagicMirror/modules/calendars
```

```
mkdir ~/.vdirsyncer  
touch ~/.vdirsyncer/config  
nano ~/.vdirsyncer/config
```

Config für iCloud Sync



Um sich bei Apple anzumelden, benötigt man ein App-spezifisches Passwort. Dieses wird dann für die Konfiguration verwendet..
<https://account.apple.com/>



Der zu nutzende Kalender muss öffentlich sein. Öffne dazu auf dem iPhone die Kalender-App, wähle den gewünschten Kalender aus (tippe auf das "i"), scrolle nach unten und aktiviere "Öffentlicher Kalender".

Trotzdem hat niemand von außen Zugriff auf den Kalender, solange der generierte Link nicht weitergegeben wird.

```
# vdirsyncer configuration for MagicMirror.  
#  
# Move it to ~/.vdirsyncer/config or ~/.config/vdirsyncer/con-  
# Run `vdirsyncer --help` for CLI usage.  
#  
# Optional parameters are commented out.  
# This file doesn't document all available parameters, see  
# http://vdirsyncer.pimutils.org/ for the rest of them.  
  
[general]  
# A folder where vdirsyncer can store some metadata about each  
status_path = "~/.vdirsyncer/status/"
```

```

# CALDAV Sync
[pair iCloud_to_MagicMirror]
a = "Mirror"
b = "iCloud"
collections = ["HERE-GOES-THE-UUID-OF-THE-CALENDAR-YOU-WANT-TO-SYNC"]

# Calendars also have a color property
metadata = ["displayname", "color"]

[storage Mirror]
# We need a single .ics file for use with the mirror (Attention: This will sync both ways)
type = "singlefile"
# We'll put the calendar file to a readable location for the mirror
path = "/home/$USER/MagicMirror/modules/calendars/%s.ics"

[storage iCloud]
type = "caldav"
url = "https://caldav.icloud.com/"
# Authentication credentials
username = "YOUR-ICLOUD-EMAIL-ADDRESS"
password = "HERE-GOES-YOUR-APP-SPECIFIC-ICLOUD-PASSWORD"
# We only want to sync in the direction TO the mirror, so we set read_only = true
read_only = true
# We only want to sync events
item_types = ["VEVENT"]
# We need to keep the number of events low, so we'll just sync 100
# Adjust this to your needs
start_date = "datetime.now() - timedelta(days=1)"
end_date = "datetime.now() + timedelta(days=365)"

```

Erstellen automatisches Synchronisieren

```
curl https://raw.githubusercontent.com/pimutils/vdirsyncer/master/vdirsyncer.py
```

```
curl https://raw.githubusercontent.com/pimutils/vdirsyncer/master/vdirsyncer.py
```

```
# Falls 15 Minuten zu lang oder zu wenig sind kann es hier angepasst werden
sudo nano /etc/systemd/user/vdirsyncer.timer
# bei Start 2m, im Betrieb 5
```

```
#Aktivieren des Timers
systemctl --user enable vdirsyncer.timer
```

Prüfen ob der Synchronisation funktioniert

```
vdirsyncer discover
```

Wenn erfolgreich werden die Kalender angezeigt

```
Discovering collections for pair iCloud_to_MagicMirror
Mirror:
iCloud:
- "25CB285C-E163-4EB7C00" ("Arbeit")
- "9221FEE8-E8B4-4D09919EC" ("Urlaub")
- "953A5477-E405-4EACC95" ("Alles andere")
warning: No collection "HERE-GOES-THE-UUID-OF-THE-CALENDAR-YOU-WANT-TO-SYNC"
Should vdirsyncer attempt to create it? [y/N]:  

# mit N abbrechen, da es ein Verbindungstest ist und wir gleich rausfinden wollten
```

Einfügen der Kalender in ~/.vdirsyncer/config



collections = ["HERE-GOES-THE-UUID-OF-THE-CALENDAR-YOU-WANT-TO-SYNC"]

Beispiel der collections



```
collections = ["3A56A-1C35-4183-9137-D6138", "392AD-278C-4EAB-AF16-94459"]
```

Erstellen der .ics Datei

```
vdirsyncer discover

# mit y bestätigen
# je mehr Kalender in "Collections" eingefügt wurden, desto höher
# muss man mit y bestätigen
```

```
# die erste Synchronisation der/des Kalender/s
vdirsyncer sync
```

```
# speicherort der erstellen Kalender
/home/$USER/MagicMirror/modules/calendars/*
```

MagicMirror/Config.js

<https://docs.magicmirror.builders/modules/calendar.html>

```
{
    "module": "calendar",
    "header": "fancy",
    "position": "top_left",
    "config": {
        "calendars": [
            {
                "symbol": "calendar",
                "url": "<HOST>:8080/modules/calendars"
            }
        ],
        "colored": true,
        // Events mit bestimmten Keywords, werden speziell hervorgehoben
        "customEvents": [
            ...
        ]
    }
}
```

```

        {
            "keyword": "Jan allein",
            "symbol": "umbrella",
            "color": "#e066ff"
        },
        {
            "keyword": "Michelle weg",
            "symbol": "umbrella",
            "color": "#e066ff"
        }
    ],
    "fade": false,
    "showLocation": true,
    "fetchInterval": 300000,
    "maximumEntries": 10,
    "dateFormat": "Do MMM"
}
},
{
    "module": "calendar",
    "header": "Urlaub",
    "position": "top_left",
    "config": {
        "calendars": [
            {
                "symbol": "calendar",
                "url": "<HOST>:8080/modules/calendars"
            }
        ],
        "colored": true,
        "customEvents": [
            {
                "keyword": "Urlaub Jan",
                "symbol": "mug-hot",
                "color": "#228b22"
            },
            {
                "keyword": "Urlaub Michelle",

```

```

        "symbol": "mug-hot",
        "color": "#228b22"
    },
],
"fade": false,
"showEnd": true,
"showEndsOnlyWithDuration": false,
"timeFormat": "absolute",
"maximumEntries": 8,
"dateEndFormat": "MMM Do",
"getRelative": 0,
"urgency": 0
}
},

```

Besonderheiten

Ich habe insgesamt drei separate Kalender eingerichtet. Neben meinem geteilten Hauptkalender gibt es einen zweiten Urlaubskalender, in dem meine Freundin und ich unsere Urlaube eintragen.

In der Konfiguration gibt es eine Funktion (showEnd), die das Ende eines Ereignisses – in diesem Fall des Urlaubs – anzeigen soll. Leider funktioniert diese nicht wie erwartet. Meine Anfrage im Forum brachte keine zufriedenstellende Lösung.

Die Anzeige in MM sieht wie folgt aus:

Jan Urlaub - 16. Dez



Fehleranalyse:

Das Problem tritt bei ganztägigen Ereignissen auf, die sich über mehrere Tage oder Wochen erstrecken. In der Übersicht wird dann nur der Starttermin angezeigt.

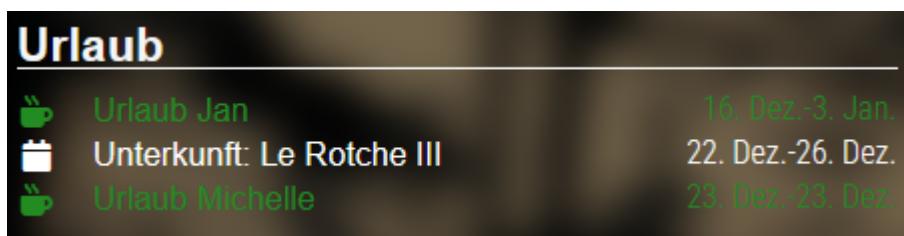
Lösungsvorschlag:

Statt ganztägiger Ereignisse präzise Zeitangaben verwenden.

Beispiel: 16.12.2024 00:05 - 03.01.2025 23:55

Nach dieser Änderung sieht die Anzeige wie folgt aus:

Jan Urlaub - 16. Dez - 03. Jan



Diese Methode zeigt in der Übersicht genau das an, was ich haben möchte.

Energieeinstellungen

Um den Stromverbrauch des Raspberry Pi zu reduzieren, nutze ich die Automatisierung von Cron.d für nächtliche Abschaltungen.

Das System soll sich zu einer festgelegten Uhrzeit selbst herunterfahren. Zuvor wird der MagicMirror ordnungsgemäß beendet, um mögliche Schäden zu vermeiden.

Crontab System

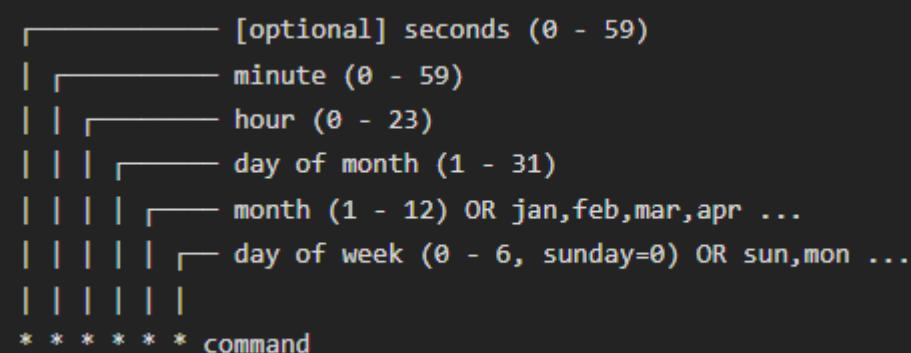
```
sudo crontab -e
```

Bei der ersten Verwendung wird nach einem bevorzugten Texteditor für die Bearbeitung des Crontabs gefragt. Option 1 (Nano) ist ein benutzerfreundlicher und einfach zu bedienender Texteditor. Nach der Auswahl öffnet sich die Crontab-Datei zur Bearbeitung.

Um den Raspberry Pi zu einer festgelegten Zeit herunterzufahren, ist folgende Zeile in die Cron-Konfiguration einzufügen:

```
# System wird jeden Tag runtergefahren  
56 21 * * * sudo systemctl poweroff  
  
# System wird Montag bis Donnerstag runtergefahren  
56 21 * * 1-4 sudo systemctl poweroff  
# System wird Freitag bis Sonntag runtergefahren  
31 22 * * fri,sat,sun sudo systemctl poweroff
```

Das System verarbeitet die Zeitangaben von rechts nach links. Demnach steht **55** für die Minute und **21** für die Stunde.



Ein Online-Tool zur Umrechnung und Anzeige von Cron-Ausdrücken ist hier verfügbar: <https://it-tools.tech/crontab-generator>

Das Tool ist auch als Offline-Variante zum Selbsthosten verfügbar – ich kann es nur empfehlen.

<https://github.com/CorentinTh/it-tools>

Crontab MagicMirror

PM2 bietet zwar eine eigene Cron-Funktionalität, diese beschränkt sich jedoch auf das Neustarten von Prozessen. Für unseren Anwendungsfall ist diese Funktion nicht erforderlich.

Aus diesem Grund wird dafür ebenfalls ein Cronjob erstellt, allerdings mit einer Besonderheit: Es ist wichtig, diesen Job ohne sudo zu erstellen. "Normale" Cron-Jobs werden mit sudo erstellt, da sie direkt aufs System zugreifen sollen.

Beispiel

Wenn man den Befehl

```
pm2 restart mm
```

eingibt, geschieht genau das, was man erwartet: MagicMirror startet neu. Gibt man den Befehl jedoch mit sudo ein, erscheint folgende Meldung:

```
[PM2][ERROR] Process or Namespace mm not found
```

Da die MagicMirror-Instanz ohne sudo erstellt wurde, kann sie auch nicht als Superuser geöffnet werden.

Erstellen Crontab PM2

```
nano mmstop.sh
```

```
pm2 restart mm
```

```
sudo chmod +x mmstop.sh
```

```
crontab -e
```

```
# System neustarten jeden tag  
55 21 * * * sh /home/jan/mmstop.sh
```

```
# System stop jeden Tag von Montag - Donnerstag  
55 21 * * 1-4 sh /home/jan/mmstop.sh  
# System stop jeden Tag Freitag - Sonntag  
30 22 * * fri,sat,sun sh /home/jan/mmstop.sh
```

Dies stoppt MM. Um MagicMirror neu zu starten, ersetze "stop" durch "restart" im Befehl. Es können auch mehrere Befehle verwendet werden. Wichtig ist, dass jeder Befehl in einer eigenen Zeile steht.

Um die Logdateien von PM2 täglich zu bereinigen, kann ein zusätzlicher Befehl in die `mm.sh`-Datei eingefügt werden.

```
pm2 flush mm
```

Dieser Befehl muss an erster Stelle stehen. Dadurch werden bei jedem Neustart des MagicMirrors die Logdateien bereinigt.

Alternativ kann auch ein Shell-Skript geschrieben und dieses in Crontab eingefügt werden. Siehe `mmstop.sh`

Troubleshooting

Um nicht ständig Logdateien über das Terminal auslesen zu müssen, bietet WinSCP eine praktische Alternative: <https://winscp.net/eng/docs/lang:de>

PM2

Alle wichtigen Befehle

```
pm2 ls          # Zeigt alle verfügbaren Instanzen  
pm2 restart id # Lädt die Instanz neu  
pm2 delete id  # Löscht eine Instanz  
pm2 info id    # Info über Instanz
```

Auslesen der PM2 Logs, um eventuelle Fehler in der MM-Konfiguration zu finden oder zu überprüfen, welche Plugins nicht korrekt funktionieren

```
cat /home/$USER/.pm2/logs/mm-error.log # Auslesen des Error-Log  
cat /home/$USER/.pm2/logs/mm-out.log   # Auslesen des Event-Log
```

vdirsyncer

Überprüfen, ob vdirsyncer ordnungsgemäß funktioniert, Anfragen an iCloud sendet und gegebenenfalls die .ics-Datei(en) aktualisiert

```
journalctl --user -u vdirsyncer -r
```

WORK IN PROGRESS — Neustart mit Cron.d — WORK IN PROGRESS

<https://www.dzombak.com/blog/2023/12/Maintaining-a-solid-WiFi-connection-on-Raspberry-Pi.html>

<https://github.com/cdzombak/dotfiles/blob/master/linux/pi/wifi-check.sh>

```
nano /usr/local/bin/wifi-check.sh
```