# MagicMirror – Backup auf NAS

## Ziel

Der MagicMirror-Ordner wird auf ein NAS gesichert.

Es gibt zwei Varianten:

- **Manuelles Backup:** Start bei Bedarf per Befehl
- **Automatisches Backup:** jeden Sonntag um 01:00 Uhr

Eigenschaften:

- Backup als komprimiertes `tar.gz`
- `node_modules` werden nicht gesichert
- SHA-256-Prüfsumme zur Integritätskontrolle
- NAS muss tatsächlich erreichbar und gemountet sein
- unvollständige Backups bleiben als `_INCOMPLETE` erhalten
- automatische Backups: maximal **4 Stück**
- Erfolg/Fehler wird über **ioBroker → Telegram** gemeldet

## Verwendete Pfade

| **Funktion** | **Pfad / Ort** |
|---|---|
| Manuelles Backup-Script | `/home/pi/bin/backup_magicmirror.sh` |
| Wöchentliches Backup-Script | `/home/pi/bin/backup_magicmirror_weekly.sh` |
| ioBroker-Zugangsdaten | `/home/pi/.config/magicmirror-backup/iobroker.conf` |
| Backup-Logs | `/home/pi/magicmirror-backup-logs/` |
| systemd-Service | `/etc/systemd/system/magicmirror-backup-weekly.service` |
| systemd-Timer | `/etc/systemd/system/magicmirror-backup-weekly.timer` |
| NAS-Zugangsdaten | `/etc/samba/credentials-nasbackup` |
| NAS-Mount-Konfiguration | `/etc/fstab` |
| NAS-Mountpunkt | `/mnt/nasbackup` |
| Manuelle Backups | `/mnt/nasbackup/Manuelle_Backups/` |
| Automatische Backups | `/mnt/nasbackup/Automatische_Backups/` |
| ioBroker Telegram-Script | `ioBroker → Skripte → JavaScript` |
| ioBroker Backup-Datenpunkte | `0_userdata.0.MagicMirror.Backup.*` |

# 1. NAS auf dem Raspberry Pi einbinden

## Pakete installieren

```
sudo apt update
```

```
sudo apt install cifs-utils
```

## Gruppe anlegen

```
sudo groupadd nasusers
sudo usermod -aG nasusers <PI_USER>
```

Danach neu anmelden oder neu starten.

UID und GID ermitteln:

```
id -u <PI_USER>
getent group nasusers
```

## Mountpunkt anlegen

```
sudo mkdir -p /mnt/nasbackup
```

## NAS-Zugangsdaten speichern

```
sudo mkdir -p /etc/samba
sudo nano /etc/samba/credentials-nasbackup
```

Inhalt:

```
username=<NAS_USER>
password=<NAS_PASSWORT>
```

Datei schützen:

```
sudo chown root:root /etc/samba/credentials-nasbackup
sudo chmod 600 /etc/samba/credentials-nasbackup
```

## /etc/fstab

Vorher sichern:

```
sudo cp /etc/fstab /etc/fstab.backup
sudo nano /etc/fstab
```

Eintrag:

```
//<NAS_HOST>/<NAS_SHARE> /mnt/nasbackup cifs
```

```
credentials=/etc/samba/credentials-
nasbackup,vers=3.0,iocharset=utf8,uid=<UID>,gid=<GID>,file_mode=0660,dir_mod
e=0770,_netdev,nofail,x-systemd.automount,x-systemd.mount-timeout=10s 0 0
```

systemd neu laden:

```
sudo systemctl daemon-reload
sudo systemctl start mnt-nasbackup.automount
```

Status:

```
systemctl status mnt-nasbackup.automount --no-pager
```

Erwartet:

```
Active: active (waiting)
```

## NAS testen

Mount auslösen:

```
ls -la /mnt/nasbackup
```

Prüfen:

```
findmnt /mnt/nasbackup
df -h /mnt/nasbackup
```

Schreibtest:

```
touch /mnt/nasbackup/test.txt
rm /mnt/nasbackup/test.txt
```

# 2. Manuelles MagicMirror-Backup

Das manuelle Backup:

- wird per Befehl gestartet
- läuft danach im Hintergrund
- erzeugt ein geprüftes TAR.GZ-Archiv
- erzeugt zusätzlich eine SHA-256-Prüfsumme
- sichert keine `node_modules`
- bricht ab, wenn das NAS nicht korrekt gemountet ist
- meldet Erfolg oder Fehler über ioBroker / Telegram

Die ioBroker-Zugangsdaten werden aus

```
~/.config/magicmirror-backup/iobroker.conf
```

gelesen.

## Script

Das Script liegt unter:

```
~/bin/backup_magicmirror.sh
```

Bearbeiten:

```
nano ~/bin/backup_magicmirror.sh
```

Das Script besitzt intern zwei wichtige Betriebsarten:

| Modus | Verwendung |
|---|---|
| `–worker` | erstellt ausschließlich das Backup; wird auch vom automatischen Weekly-Backup verwendet |
| `–manual-worker` | führt den Backup-Worker aus und meldet anschließend Erfolg oder Fehler an ioBroker |

Dadurch erzeugt das automatische Backup keine doppelte Telegram-Nachricht.

## Backup starten

```
~/bin/backup_magicmirror.sh
```

Das Backup läuft anschließend im Hintergrund.

## Status / Log ansehen

```
tail -f "$(ls -t ~/magicmirror-backup-logs/*.log | head -1)"
```

Mit `Ctrl+C` wird nur die Log-Anzeige beendet.

Das Backup selbst läuft weiter.

## Erfolgreiches Backup

Ein erfolgreiches Backup enthält:

```
MagicMirror.tar.gz
```

```
MagicMirror.tar.gz.sha256
```

Das Log enthält:

```
BACKUP ERFOLGREICH
```

Zusätzlich wird über ioBroker eine Telegram-Nachricht gesendet.

## Speicherort

```
/mnt/nasbackup/Manuelle_Backups/
```

Beispiel:

```
MagicMirror_2026-10-08_21-30-00/
├── MagicMirror.tar.gz
└── MagicMirror.tar.gz.sha256
```

## Backup prüfen

```
cd /mnt/nasbackup/Manuelle_Backups/MagicMirror_YYYY-MM-DD_HH-MM-SS

sha256sum -c MagicMirror.tar.gz.sha256
```

Erwartet:

```
MagicMirror.tar.gz: OK
```

# 3. ioBroker für Backup-Meldungen

## REST-API

In ioBroker den Adapter **REST API** installieren.

Empfohlene Einstellungen:

- Authentifizierung: **Ein**
- Keine GUI: **Ein**
- Commands erlauben: **Aus**
- Admin-Commands erlauben: **Aus**
- API nur im internen Netzwerk erreichbar machen
- kein Port-Forwarding ins Internet

# Benutzer und Gruppe

ioBroker-Benutzer anlegen:

```
magicmirror
```

Gruppe anlegen:

```
magicmirror_backup
```

Benutzer der Gruppe hinzufügen.

Benötigte Gruppenrechte:

- Objekte: Auflisten + Lesen
- Zustände: Auflisten + Lesen + Schreiben
- HTTP: Ein
- Execute/SendTo: Aus

# Datenpunkte

Folgende Datenpunkte werden benötigt:

```
0_userdata.0.MagicMirror.Backup.Status
0_userdata.0.MagicMirror.Backup.Message
0_userdata.0.MagicMirror.Backup.LastRun
0_userdata.0.MagicMirror.Backup.BackupPath
```

Wichtig:

Der Benutzer `magicmirror` sollte Besitzer dieser vier Datenpunkte sein.

Beispiel auf dem ioBroker-Host:

```
iobroker object chown magicmirror magicmirror_backup
0_userdata.0.MagicMirror.Backup.Status
iobroker object chown magicmirror magicmirror_backup
0_userdata.0.MagicMirror.Backup.Message
iobroker object chown magicmirror magicmirror_backup
0_userdata.0.MagicMirror.Backup.LastRun
iobroker object chown magicmirror magicmirror_backup
0_userdata.0.MagicMirror.Backup.BackupPath
```

# Telegram-JavaScript

Im ioBroker-JavaScript-Adapter:

```
const BASE = '0_userdata.0.MagicMirror.Backup';

createState(`${BASE}.Status`, 'idle', {
    type: 'string',
    role: 'text',
    read: true,
    write: true
});

createState(`${BASE}.Message`, '', {
    type: 'string',
    role: 'text',
    read: true,
    write: true
});

createState(`${BASE}.LastRun`, '', {
    type: 'string',
    role: 'text',
    read: true,
    write: true
});

createState(`${BASE}.BackupPath`, '', {
    type: 'string',
    role: 'text',
    read: true,
    write: true
});


on({
    id: `${BASE}.Status`,
    change: 'ne'
}, obj => {

    const status = String(obj.state.val);

    if (status !== 'success' && status !== 'error') return;

    const message = String(getState(`${BASE}.Message`).val || '');
    const lastRun = String(getState(`${BASE}.LastRun`).val || '');
    const backupPath = String(getState(`${BASE}.BackupPath`).val || '');

    let text;

    if (status === 'success') {
        text = `□ MagicMirror Backup erfolgreich\n\n${message}`;
    } else {
        text = `□ MagicMirror Backup FEHLGESCHLAGEN\n\n${message}`;
    }
```

```
    text += `\n\nZeit: ${lastRun}`;

    if (backupPath) {
        text += `\nBackup: ${backupPath}`;
    }

    sendTo('telegram.0', text);
});
```

# 4. Zugangsdaten für ioBroker auf dem Pi

```
mkdir -p ~/.config/magicmirror-backup
nano ~/.config/magicmirror-backup/iobroker.conf
```

Inhalt:

```
IOBROKER_URL='http://<IOBROKER_HOST>:<REST_PORT>'
IOBROKER_USER='magicmirror'
IOBROKER_PASSWORD='<PASSWORT>'
```

Datei schützen:

```
chmod 600 ~/.config/magicmirror-backup/iobroker.conf
```

# 5. Automatisches wöchentliches Backup

## Script

```
nano ~/bin/backup_magicmirror_weekly.sh
```

Inhalt:

```
#!/usr/bin/env bash

BACKUP_SCRIPT="$HOME/bin/backup_magicmirror.sh"

NAS="/mnt/nasbackup"
AUTO_BASE="$NAS/Automatische_Backups"

LOGDIR="$HOME/magicmirror-backup-logs"

CONFIG="$HOME/.config/magicmirror-backup/iobroker.conf"
```

```
KEEP_BACKUPS=4

STATE_BASE="0_userdata.0.MagicMirror.Backup"


if [[ ! -r "$CONFIG" ]]; then
    echo "FEHLER: $CONFIG fehlt."
    exit 1
fi

source "$CONFIG"


set_iobroker_state() {

    local ID="$1"
    local VALUE="$2"

    curl \
        --fail \
        --silent \
        --show-error \
        --max-time 10 \
        --user "${IOBROKER_USER}:${IOBROKER_PASSWORD}" \
        --get \
        --data-urlencode "value=${VALUE}" \
        "${IOBROKER_URL}/v1/state/${ID}" \
        >/dev/null
}


notify_iobroker() {

    local STATUS="$1"
    local MESSAGE="$2"
    local BACKUP_PATH="${3:-}"

    set_iobroker_state "${STATE_BASE}.Message" "$MESSAGE" || return 1
    set_iobroker_state "${STATE_BASE}.BackupPath" "$BACKUP_PATH" || return 1
    set_iobroker_state "${STATE_BASE}.LastRun" "$(date -Iseconds)" || return
1

    # Status zuletzt setzen: dieser löst Telegram aus.
    set_iobroker_state "${STATE_BASE}.Status" "$STATUS"
}


mkdir -p "$LOGDIR"

STAMP="$(date '+%Y-%m-%d_%H-%M-%S')"
```

```
DEST="$AUTO_BASE/MagicMirror_Auto_$STAMP"

LOG="$LOGDIR/MagicMirror_Auto_$STAMP.log"


set_iobroker_state "${STATE_BASE}.Status" "running" || true


"$BACKUP_SCRIPT" --worker "$DEST" >"$LOG" 2>&1

RESULT=$?


if [[ $RESULT -eq 0 ]]; then

    SIZE="$(du -h "$DEST/MagicMirror.tar.gz" | awk '{print $1}')"

    mapfile -t BACKUPS < <(
        find "$AUTO_BASE" \
            -mindepth 1 \
            -maxdepth 1 \
            -type d \
            -name 'MagicMirror_Auto_*' \
            ! -name '*_INCOMPLETE' \
            -printf '%f\n' \
            | sort -r
    )

    if (( ${#BACKUPS[@]} > KEEP_BACKUPS )); then

        for ((i=KEEP_BACKUPS; i<${#BACKUPS[@]}; i++)); do
            rm -rf -- "$AUTO_BASE/${BACKUPS[$i]}"
        done

    fi


    MESSAGE="Automatisches MagicMirror-Backup erfolgreich. Größe: ${SIZE}.
Es werden ${KEEP_BACKUPS} automatische Backups aufbewahrt."

    notify_iobroker "success" "$MESSAGE" "$DEST" || true

    exit 0
fi


MESSAGE="Automatisches MagicMirror-Backup fehlgeschlagen. Exit-Code:
${RESULT}. Log: ${LOG}"

notify_iobroker "error" "$MESSAGE" "${DEST}_INCOMPLETE" || true
```

```
exit "$RESULT"
```

Ausführbar machen:

```
chmod +x ~/bin/backup_magicmirror_weekly.sh
```

Manueller Test:

```
~/bin/backup_magicmirror_weekly.sh
```

Bei Erfolg:

- Backup liegt unter `Automatische_Backups`
- Telegram-Meldung wird gesendet
- Exit-Code ist `0`

# 6. systemd-Timer

## Service

```
sudo nano /etc/systemd/system/magicmirror-backup-weekly.service
```

```
[Unit]
Description=Wöchentliches MagicMirror Backup auf NAS
Wants=network-online.target
After=network-online.target

[Service]
Type=oneshot
User=<PI_USER>
Group=<PI_USER>
Environment=HOME=/home/<PI_USER>
WorkingDirectory=/home/<PI_USER>

ExecStart=/home/<PI_USER>/bin/backup_magicmirror_weekly.sh

TimeoutStartSec=infinity
```

## Timer

```
sudo nano /etc/systemd/system/magicmirror-backup-weekly.timer
```

```
[Unit]
Description=MagicMirror Backup jeden Sonntag um 01:00 Uhr
```

```
[Timer]
OnCalendar=Sun *-*-* 01:00:00
Persistent=true
AccuracySec=1min

[Install]
WantedBy=timers.target
```

## Aktivieren

```
sudo systemctl daemon-reload
sudo systemctl enable --now magicmirror-backup-weekly.timer
```

Kontrolle:

```
systemctl list-timers --all | grep magicmirror
```

Timer-Status:

```
systemctl status magicmirror-backup-weekly.timer --no-pager
```

Erwartet:

```
Active: active (waiting)
```

## Service manuell testen

```
sudo systemctl start magicmirror-backup-weekly.service
```

Danach:

```
systemctl status magicmirror-backup-weekly.service --no-pager
```

# 7. Backups prüfen

## SHA-256

```
cd /mnt/nasbackup/Manuelle_Backups/MagicMirror_YYYY-MM-DD_HH-MM-SS
sha256sum -c MagicMirror.tar.gz.sha256
```

Erwartet:

```
MagicMirror.tar.gz: OK
```

### Archivinhalt ansehen

```
tar -tzf MagicMirror.tar.gz | less
```

# 8. Wiederherstellung

## Auf gleichem System

Archiv prüfen:

```
sha256sum -c MagicMirror.tar.gz.sha256
```

Dann:

```
tar -xzf MagicMirror.tar.gz -C ~
```

## Auf einem neuen Raspberry Pi

Bei einem neuen Pi sollte MagicMirror und Node.js zunächst frisch installiert werden.

Aus dem Backup werden anschließend insbesondere übernommen:

- `config/config.js`
- `css/custom.css`
- Drittmodule und deren Konfiguration
- eigene Bilder, Skripte und Dateien

`node_modules` werden bewusst nicht wiederhergestellt.

Die Abhängigkeiten werden auf dem neuen System neu installiert.

# 9. Alltag

## Manuelles Backup starten

```
~/bin/backup_magicmirror.sh
```

Bei Erfolg oder Fehler wird automatisch eine Telegram-Nachricht über ioBroker gesendet.

## Aktuelles Backup-Log

```
tail -f "$(ls -t ~/magicmirror-backup-logs/*.log | head -1)"
```

## Automatisches Backup

Das automatische Backup läuft:

```
jeden Sonntag um 01:00 Uhr
```

Es werden maximal:

```
4 automatische Backups
```

aufbewahrt.

Manuelle Backups werden dadurch nicht gelöscht.

## Timer prüfen

```
systemctl list-timers --all | grep magicmirror
```

## NAS prüfen

```
findmnt /mnt/nasbackup
df -h /mnt/nasbackup
```

## Backup prüfen

Im jeweiligen Backup-Verzeichnis:

```
sha256sum -c MagicMirror.tar.gz.sha256
```

Erwartet:

```
MagicMirror.tar.gz: OK
```

From:
https://wiki.mahlen.biz/ - **Smart-Home Wiki**

Permanent link:
**https://wiki.mahlen.biz/doku.php?id=magicmirror:magicmirror_tipps:backup**

Last update: **08.10.2026**