

# Ntfy-desktop

## Build-Anleitung mit Erläuterungen

---

### Vorbemerkung:

Bei vielen Windows-Installationen sind zwei Besonderheiten für verschiedene Build-Probleme verantwortlich:

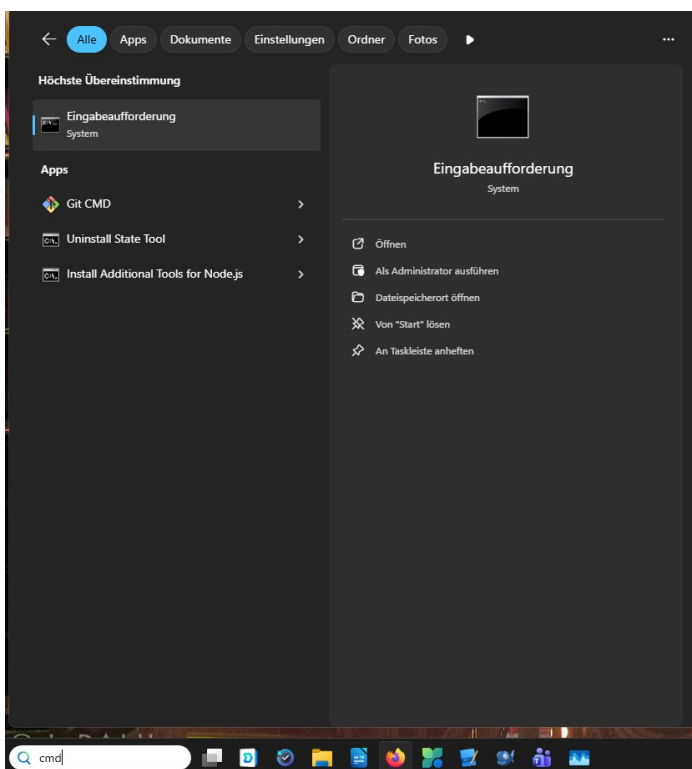
1. **NODE\_ENV=production ist bei dir dauerhaft gesetzt.** npm interpretiert das als Signal "installiere nur Produktions-Abhängigkeiten" und überspringt automatisch alle devDependencies – dort steht aber electron selbst.
  2. **PowerShell 7 als interaktive Shell.** Wenn build.bat (ein cmd-Batch-Skript) aus einer PowerShell-7-Sitzung heraus gestartet wird, erbt der intern aufgerufene powershell-Unterprozess (Windows PowerShell 5.1) eine PSModulePath-Variablen, die für ihn nicht passt – dadurch scheitert Compress-Archive.
- 

### Vorbereitung:

Siehe: <https://github.com/Aetherinox/ntfy-desktop#windows>

Falls Node.js benötigt wird, dann hier (<https://nodejs.org/en/download>) den Installer herunterladen, ausführen, gut.

### Schritt 1: Eingabeaufforderung (cmd.exe) öffnen – nicht PowerShell



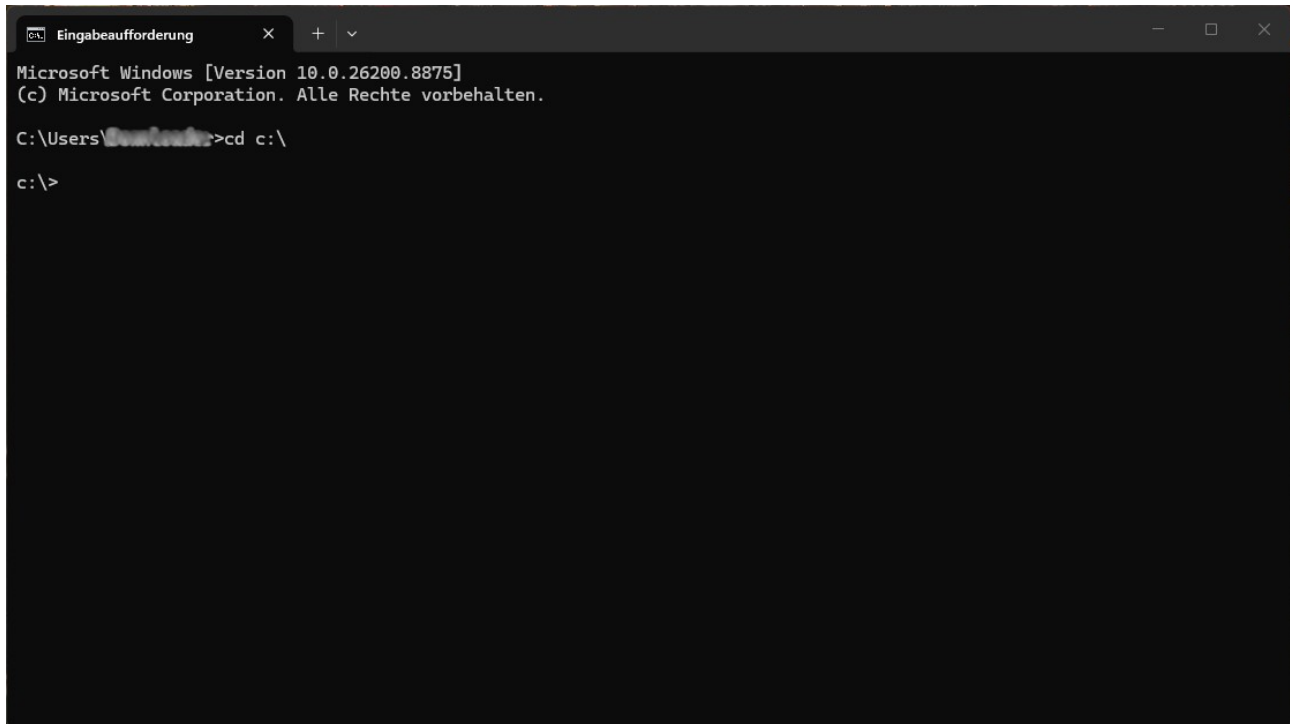
Warum? Weil cmd.exe keine PS7-spezifische PSModulePath an die späteren powershell-Unterprozess-Aufrufe vererbt.

Task-Leiste (Suche) → „nach „cmd“ suchen: → „Eingabeaufforderung“ (bzw. cmd) öffnen.

## Schritt 2: In das Projektverzeichnis wechseln

In das Grundverzeichnis der Festplatte wechseln:

```
cd C:\
```



```
mkdir ntfy-desktop
cd ntfy-desktop
```

Warum? Alle folgenden Befehle (npm, build.bat) müssen relativ zum Projekt-Root ausgeführt werden, da sie mit relativen Pfaden arbeiten.

## Schritt 3: Root-Abhängigkeiten installieren

```
npm install
npm install -g electron
```

Warum? Das ist der offizielle erste Schritt laut Projekt-README. Er installiert die (wenigen) Abhängigkeiten der Wurzel-package.json sowie global das electron-CLI-Tool.

## Schritt 4: Abhängigkeiten der eigentlichen App installieren – mit --include=dev

```
cd src
npm install --include=dev
cd ..
```

Warum genau --include=dev: Die eigentliche Electron-App (inkl. der electron-Abhängigkeit selbst, Version ^37.0.0) liegt in src/package.json, und electron steht dort unter devDependencies. Wegen NODE\_ENV=production würde ein normales npm install

genau diese Abhängigkeiten überspringen — das war der Grund für den Fehler `Cannot find module 'electron/package.json'`. `--include=dev` erzwingt die Installation trotzdem, ohne die Umgebungsvariable dauerhaft zu ändern.

### Kontrolle (optional):

```
dir src\node_modules\electron
```

Wenn dort eine `package.json` liegt, war der Schritt erfolgreich.

## Schritt 5: Build-Skript ausführen

```
build.bat
```

### Warum das funktioniert:

- Das Skript wechselt intern nach `src` und ruft `npx @electron/packager` auf — das findet jetzt die installierte `electron`-Abhängigkeit und paketierte die Anwendung für `Windows/Linux/macOS`.
- Anschließend ruft es `powershell Compress-Archive` auf, um die paketierte Ordner zu `.zip/.tar.gz` zu packen. Da wir aus `cmd.exe` heraus gestartet haben, bekommt dieser Unterprozess eine saubere `PSModulePath` und lädt das Archiv-Modul korrekt.

## Schritt 6: Ergebnis prüfen

```
dir dist
```

Erwartetes Ergebnis in `dist\`:

- `ntfy-desktop-windows-x64.zip`, `-ia32.zip`, `-arm64.zip`
- `ntfy-desktop-linux-x64.zip`, `-arm64.zip`, `-armv7l.zip`
- `ntfy-desktop-mac-x64.tar.gz`, `-arm64.tar.gz`

---

## Welche Dateien brauchst du für den Betrieb?

Für den **Betrieb** (nicht für weitere Builds) brauchst du nur **einen** der paketierte Ordner bzw. das daraus entpackte ZIP — je nachdem, welche Plattform/Architektur du tatsächlich nutzt. Für `Windows x64` ist das:

`dist\ntfy-desktop-windows-x64.zip`

bzw. nach dem Entpacken (oder direkt, falls du das Zwischenverzeichnis behalten hast):

`build\ntfy-desktop-win32-x64\`

**Wichtig:** Das ist ein **kompletter Electron-App-Ordner** — alle darin enthaltenen Dateien gehören zusammen und werden zur Laufzeit gebraucht (u. a. `ntfy-desktop.exe`, diverse `.dll`-Dateien wie

ffmpeg.dll/vulkan-1.dll, resources.pak, resources\, locales\, icudtl.dat, snapshot\_blob.bin, v8\_context\_snapshot.bin). Das ist die Electron/Chromium-Laufzeitumgebung, nicht "Bonus-Material" – **einzelne Dateien daraus zu löschen kann die App unbrauchbar machen**. Diesen ganzen Ordner kannst du z. B. nach C:\Program Files\ntfy-desktop\ verschieben oder als Ganzes an einen Ort deiner Wahl kopieren; er ist eigenständig lauffähig, ohne Node.js, npm oder das Ursprungs-Repo.

Einzige Ausnahme, die man theoretisch entfernen kann, ohne die App zu zerstören:

- LICENSES.chromium.html – reine Lizenztext-Datei, wird zur Laufzeit nicht benötigt (aus rechtlichen Gründen aber sinnvoll, sie zu behalten, wenn du die App weitergibst).
- Sprachdateien in locales\, die du sicher nie brauchst (z. B. alle außer de.pak und en-US.pak) – spart etwas Platz, ist aber nicht nötig und birgt das Risiko, versehentlich eine benötigte Datei zu löschen.

## Was du nach einem erfolgreichen Build löschen/behalten kannst

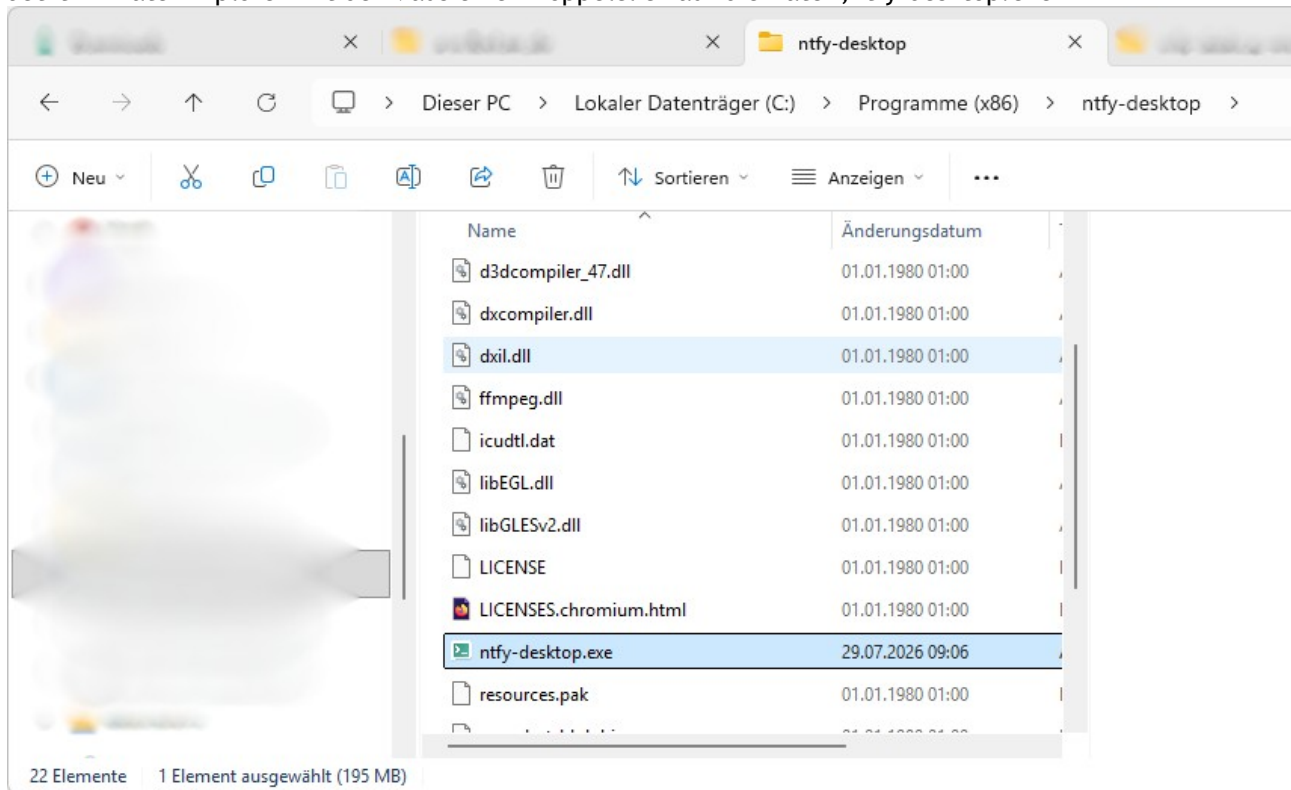
| Verzeichnis/Datei                                                                                                              | Löschen möglich?                                           | Begründung                                                                                                                                                                                                                                                                                           |
|--------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| dist\ntfy-desktop-windows-x64.zip (bzw. die für dich relevante Plattform)                                                      | <b>Nein, behalten</b>                                      | Das ist dein fertiges Endprodukt zum Betrieb/Weitergeben.                                                                                                                                                                                                                                            |
| build\ntfy-desktop-win32-x64\                                                                                                  | Nein, solange du es zum Betrieb nutzt                      | Enthält dieselben Dateien wie im ZIP, nur schon entpackt – praktisch, wenn du nicht extra entpacken willst.                                                                                                                                                                                          |
| build\ – Ordner anderer Plattformen/Architekturen (z. B. ntfy-desktop-win32-ia32, ntfy-desktop-linux-*, ntfy-desktop-darwin-*) | <b>Ja</b> , falls du sie nicht brauchst                    | Du hast bereits das gepackte ZIP/TAR.GZ in dist\ – die entpackten Zwischenordner sind danach redundant.                                                                                                                                                                                              |
| dist\ – ZIPs/TARs für Plattformen, die du nicht nutzt                                                                          | <b>Ja</b>                                                  | Reine Bequemlichkeit für andere Systeme; wenn du z. B. nur Windows x64 brauchst, sind Linux/macOS-Pakete überflüssig.                                                                                                                                                                                |
| node_modules\ (Root und src\)                                                                                                  | <b>Ja, für den Betrieb</b> – nicht nötig für erneute Runde | Nur zum <b>Bauen</b> der App nötig (Build-Werkzeuge, Electron-Binaries für den Packager). Die fertige .exe enthält bereits alles, was sie zur Laufzeit braucht. Lösche sie nur, wenn du sicher bist, nicht erneut bauen zu wollen – sonst müsstest du npm install wieder komplett neu laufen lassen. |
| Der gesamte geklonte Quellcode (C:\ntfy-desktop inkl. .git, src\, README.md, Tests etc.)                                       | <b>Ja, für den Betrieb</b>                                 | Sobald du den paketierten Ordner/ZIP an anderer Stelle gesichert hast, brauchst du den Quellcode nur noch, wenn du später eine neue Version bauen willst.                                                                                                                                            |

**Kurz gesagt:** Für den reinen Betrieb brauchst du ausschließlich den kompletten Inhalt des Ordners ,c:\ntfy-desktop\build\ntfy-desktop-win32-x64\‘ (zum Beispiel oder das entsprechende ZIP aus dem Ordner ,c:\ntfy-desktop\dist\‘). Diesen Ordner kopierst du an einem beliebigen Ort. – Alles andere (Quellcode, node\_modules, andere Plattform-Builds) dient nur dem Bauprozess und kann danach gelöscht werden.

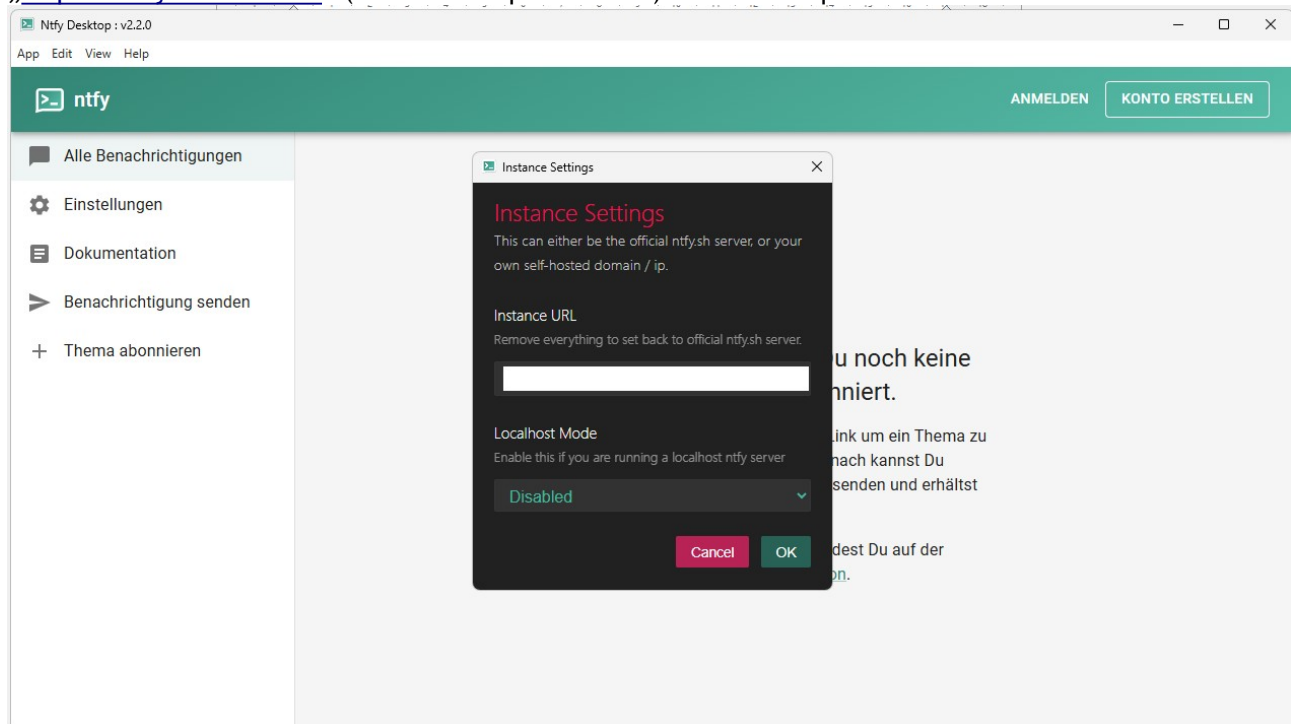
Ich beispielsweise habe im Verzeichnis ,C:\Programme (x86)\‘ einen Ordner ,ntfy-desktop\‘ erstellt, da hinein den Inhalt von ,c:\ntfy-desktop\build\ntfy-desktop-win32-x64\‘ kopiert und anschließend das gesamte Verzeichnis ,c:\ntfy-desktop\‘ gelöscht.

## Programm starten

Jetzt im Datei-Explorer mit der Maus einen Doppelclick auf die Datei ,ntfy-desktop.exe‘



Server einstellen: App → Settings → Instance. Hier unter ,Instance URL‘ die Serveradresse eingeben „<https://ntfy.domain.com>“ (nur eine Beispiel-Domain) und mit OK quittieren.



Als nächstes „anmelden“ und die Benutzerdaten eingeben, einloggen, fertig. Jetzt das Programm mit „x“ schließen und es verschwindet im System-Tray („^“) unten rechts auf Eurem Bildschirm und ist von dort aus jederzeit aufrufbar.

---

## Programm als „autostart“ festlegen

Jetzt macht Ihr mit der Maus einen „Rechtsklick“ auf Eure ‚ntfy-desktop.exe‘, dann ‚weitere Optionen anzeigen‘ → ‚Senden an‘ → ‚Desktop (Verknüpfung erstellen)‘. Geht im Datei-Explorer in das Verzeichnis ‚Desktop‘, schneidet die Datei ‚ntfy-desktop.exe - Verknüpfung‘ aus (rechte Maustaste) und geht in das Verzeichnis ‚C:\ProgramData\Microsoft\Windows\Startmenü\Programme\Autostart‘. Dort fügt Ihr die Verknüpfungsdatei wieder ein. Das müsst Ihr im Zweifelsfall einmal bestätigen, weil Ihr nicht als Administrator angemeldet seid. Dann sei dem eben so.

Beim nächsten Systemstart seid ihr automatisch eingeloggt und das Programm ist betriebsbereit. Eventuell müsst Ihr es nur wieder „klein“ klicken.

**Alles unklar? Gut!**