

Folder Structure & Experiment Manager

Eine Electron-App zur Verwaltung von Projekt-Vorlagen mit Ordnerstrukturen und Metadaten für Experimente.

Features

Ordner-Vorlagen

- Erstelle wiederverwendbare Ordnerstrukturen
- Unterstützung für verschachtelte Ordner (2 Leerzeichen = 1 Ebene tiefer)
- Automatische Datei- und Ordnererstellung
- Projekt-basierte Organisation

Experiment-Vorlagen

- Ordnerstruktur + JSON-Metadaten
- Grafischer Metadaten-Editor
- Verschiedene Feldtypen: Text, Zahl, Datum, Dropdown, Checkbox, Textbereich
- JSON-Import/Export für Templates

Installation & Start

```
bash
```

```
# Dependencies installieren
```

```
npm install
```

```
# App im Entwicklungsmodus starten
```

```
npm run dev
```

```
# Produktionsversion starten
```

```
npm start
```

```
# App für verschiedene Plattformen bauen
```


```
npm run build-win    # Windows
```

```
npm run build-mac    # macOS
```

```
npm run build-linux  # Linux
```

Verwendung

Ordner-Vorlage erstellen

1. **Template-Typ wählen:** Klicke auf " Ordner" in der Seitenleiste

2. **Neue Vorlage:** Klicke auf "Neue Vorlage"



3. **Details eingeben:**

- Name der Vorlage
- Beschreibung (optional)
- Ordnerstruktur

Ordnerstruktur-Syntax:

```
src/                # Ordner (mit / am Ende)
  components/       # Unterordner (2 Leerzeichen eingerückt)
    Button/
    Modal/
  assets/
    images/
    css/
      main.css      # Datei (ohne / am Ende)
      responsive.css
  utils/
docs/
  README.md         # Datei im docs Ordner
package.json        # Datei im Root
```

Experiment-Vorlage erstellen


1. **Template-Typ wählen:** Klicke auf "  Experimente"
2. **Neue Vorlage:** Klicke auf "Neue Vorlage"
3. **Typ auswählen:** "  Experiment (Ordner + Metadaten)"
4. **Struktur-Tab:** Ordnerstruktur definieren
5. **Metadaten-Tab:**
 - Felder hinzufügen mit "+ Feld hinzufügen"
 - Feldtypen auswählen
 - Optionen für Dropdowns definieren
 - JSON-Vorschau wird automatisch aktualisiert

Verfügbare Metadaten-Feldtypen:

- **Text:** Einfacher Text-Input
- **Zahl:** Numerischer Input (mit Min/Max Werten)
- **Datum:** Datums-Picker
- **Textbereich:** Mehrzeiliger Text

- **Dropdown:** Auswahlliste (Optionen komma-getrennt eingeben)
- **Checkbox:** Ja/Nein Auswahl

Projekt erstellen

1. **Vorlage auswählen:** Klicke auf eine Vorlage in der Seitenleiste
2. **Basis-Verzeichnis:** Wähle den Ordner wo das Projekt erstellt werden soll
3. **Projekt-Name:** Gib einen Namen für dein Projekt ein
4. **Projekt erstellen:** Klicke auf "  Projekt erstellen"

Das System erstellt automatisch:

- Hauptprojekt-Ordner mit dem angegebenen Namen
- Alle Unterordner und Dateien aus der Vorlage
- `metadata.json` Datei (bei Experiment-Vorlagen)



Technische Details

Ordnerstruktur-Format

- **2 Leerzeichen** = eine Ebene tiefer
- **Ordner:** Namen mit `/` am Ende
- **Dateien:** Namen ohne `/` am Ende
- **Verschachtelung:** Beliebig tief möglich

Beispiel:

```
projekt/  
  src/  
    components/  
      Button/  
        Button.js  
        Button.css  
      Modal/  
        index.js  
  docs/  
    README.md  
  package.json
```

Metadaten-System

Experiment-Vorlagen erstellen eine `metadata.json` Datei mit folgender Struktur:

json

```
{
  "experiment_name": {
    "label": "Experiment Name",
    "type": "text",
    "value": ""
  },
  "researcher": {
    "label": "Forscher",
    "type": "text",
    "value": ""
  },
  "start_date": {
    "label": "Start Datum",
    "type": "date",
    "value": "2025-05-27"
  }
}
```

Fehlerbehebung

Ordner werden nicht korrekt erstellt

- Prüfe die Indentation (genau 2 Leerzeichen pro Ebene)
- Stelle sicher, dass Ordernamen mit `/` enden
- Überprüfe die Berechtigung im Zielverzeichnis

JSON-Import funktioniert nicht

- Datei muss gültiges JSON-Format haben
- Stelle sicher, dass die Feldstruktur korrekt ist
- Prüfe die Konsole auf Fehlermeldungen (F12)

App startet nicht

- Überprüfe ob Node.js installiert ist
- Führe `npm install` erneut aus
- Prüfe auf Fehlermeldungen in der Konsole

Projektstruktur

```
/
├── main.js           # Electron Main Process
├── preload.js        # Secure IPC Communication
├── index.html         # UI & Frontend Logic
├── package.json       # Dependencies & Scripts
└── assets/           # Icons & Resources
    ├── icon.png
    ├── icon.ico
    └── icon.icns
```

Updates & Erweiterungen

Die App speichert Vorlagen lokal und kann einfach erweitert werden:

- Neue Metadaten-Feldtypen hinzufügen
- Template-Sharing implementieren
- Cloud-Synchronisation einbauen
- Erweiterte Ordnerstruktur-Validierung

Support

Bei Problemen oder Fragen:

1. Prüfe diese Anleitung
2. Schaue in die Browser-Konsole (F12) für Fehlermeldungen
3. Erstelle ein Issue im Projekt-Repository