



# Handbuch: Wie nutze ich die Master-GUI für ein neues Projekt?

Herzlich willkommen zu deiner neuen Master-GUI! Dieses Handbuch erklärt dir detailliert und verständlich, wie du die GUI für dein eigenes Projekt einsetzen kannst.

---



## 1. Voraussetzungen

Du hast bereits die fertige Master-GUI erhalten und möchtest sie nun individuell an dein neues Projekt anpassen und benutzen.

---



## 2. Projektstruktur vorbereiten

Erstelle einen neuen Projektordner. Die empfohlene Struktur sieht so aus:

```
Neues_Projekt/  
├─ main.py  
├─ config.json  
├─ scripts/  
├─ assets/  
│   ├── icons/  
│   │   └─ placeholder.png  
│   └─ styles/  
│       └─ default.qss  
├─ core/  
├─ ui/  
├─ util/  
├─ logs/  
├─ gui_tools/ (optional)  
└─ tests/ (optional)
```



## 3. Dateien in den Projektordner kopieren

Kopiere folgende Dateien aus der bestehenden Master-GUI in deinen neuen Ordner:

- main.py
- config.json
- Ordner core/
- Ordner ui/
- Ordner util/
- Ordner assets/

- requirements.txt
- Ordner gui\_tools/ (optional)
- Ordner tests/ (optional)

Erstelle zusätzlich:

- Einen leeren Ordner scripts/
- Einen leeren Ordner logs/



## 4. Startkonfiguration einrichten (config.json)

Verwende eine leere Standardkonfiguration, um neu zu beginnen:

```
{
  "buttons": [],
  "theme": {
    "stylesheet": "assets/styles/default.qss",
    "background": ""
  },
  "window_title": "Mein neues Projekt"
}
```



## 5. GUI starten

- Öffne eine Eingabeaufforderung im Projektordner und führe:

```
python main.py
```

- Alternativ erstelle eine ausführbare Datei (EXE):

```
pyinstaller main.py --noconsole --onefile --icon=assets/icons/placeholder.ico
```

Die fertige EXE findest du dann in dist/master\_gui.exe .



## 6. Buttons verwalten

**Neuen Button anlegen:**

- Starte die GUI.
- Gehe im Menü auf "**Button-Manager**".
- Klicke "**Neu**".
- Gib einen **Namen/ID** ein.
- Wähle einen **Aktionstyp**:

- **SCRIPT** (führt ein Python-Skript aus)
- **LINK** (öffnet eine URL)
- **FILE** (öffnet eine Datei)
- **FOLDER** (öffnet ein Verzeichnis)
- **MENU** (erstellt Unterbuttons)
- Wähle **Payload** (z. B. `scripts/beispiel.py`).
- Wähle ein **Icon**.
- Klicke **"Speichern"**.

### Buttons positionieren:

- Nutze die Option **"Verschieben"** im Button-Manager.
- Ziehe die Buttons einfach auf die gewünschte Position.



## 7. Skripte hinzufügen

- Speichere deine Python-Skripte im Ordner `scripts/`.
- Beispiel für ein einfaches Python-Skript (`scripts/beispiel.py`):

```
for i in range(101):
    print(f"[{i}%] Verarbeitung läuft...")
```

- Verknüpfe das Skript anschließend über den Button-Manager.



## 8. Hintergrundbild und Theme anpassen

- Gehe im Menü auf **"Einstellungen"**.
- Hier kannst du den Fenstertitel, das Hintergrundbild und das Stylesheet (`.qss`) ändern.



## 9. Task-Dashboard nutzen

- Öffne das Task-Dashboard über das Menü.
- Hier siehst du alle laufenden und abgeschlossenen Jobs inklusive Fortschritt und Log-Ausgabe.



## 10. Best Practices

- Lege Skripte stets im Ordner `scripts/` ab.
- Nutze relative Pfade (`scripts/mein_script.py`).
- Icons gehören in `assets/icons/`.
- Speichere regelmäßig deine Konfiguration.
- Erstelle Unterbuttons mit Aktionstyp `MENU`.
- Führe stets einfache Tests der Buttons und Skripte durch.

## 11. Backup und Release

Sichere dein Projekt regelmäßig:

```
git commit -am "Release: v1.0.0 - Erste stabile Version"  
git tag v1.0.0
```

---

## Zusammenfassung

Du hast nun alles, was du brauchst, um deine Master-GUI erfolgreich für neue Projekte einzusetzen. Folge den Schritten, um schnell und effizient zu starten. Bei Fragen kannst du jederzeit Unterstützung anfordern.

Viel Erfolg mit deiner neuen Master-GUI!