# Installation Microgrid Planner (MGP) unter Windows (11)

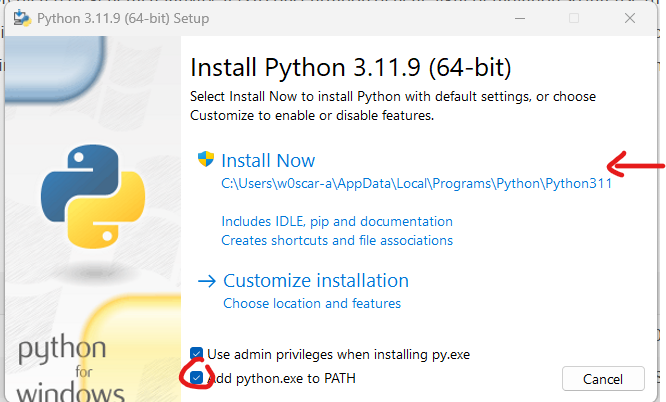
1. Schauen, ob Python installiert ist und Version ermitteln

Mit rechter Maustaste auf Windows-Symbol  in der Taskleiste und Terminal (Adminstrator) starten. Dort folgendes eingeben und mir Return bestätigen: python --version

Falls Python mit Version 3.11 installiert ist, kann der folgende Schritt entfallen. Wegen der Kompatibilität vielleicht aber doch lieber die aktuelle Python-Version deinstallieren (über Systemsteuerung wie üblich: rechte Maustaste auf Windows-Symbol, ganz oben: Installierte Apps)) und Python 3.11 installieren (Schritt 2). Mit neueren Versionen gab es Probleme.

1. Python Version 3.11 herunterladen und installieren

<https://www.python.org/ftp/python/3.11.9/python-3.11.9-amd64.exe>



Unbedingt „add python to PATH“ anklicken.

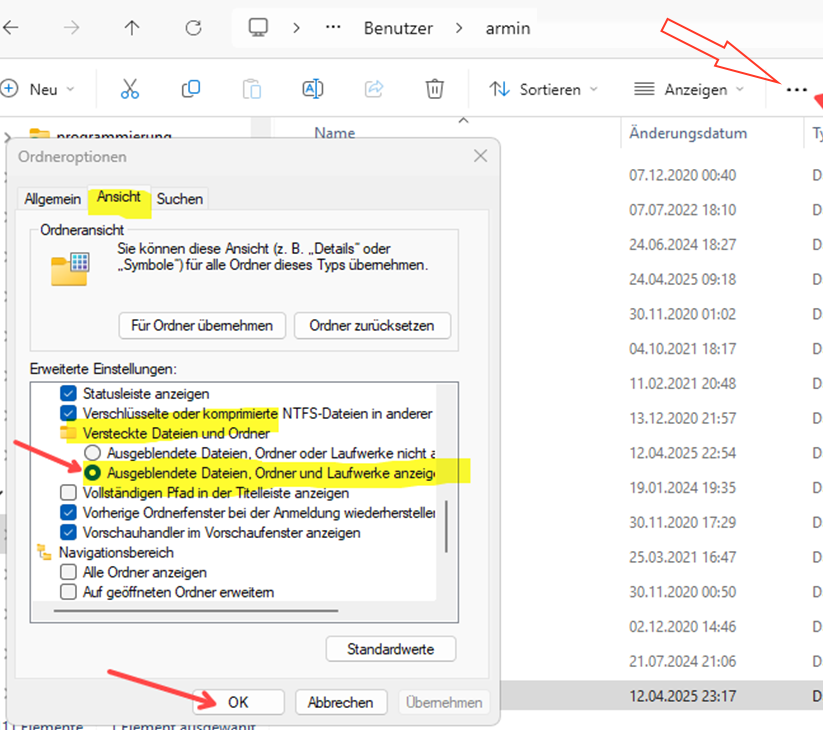
1. MySQL-Datenbank installieren

Herunterladen unter: <https://dev.mysql.com/downloads/installer/>

Aktuelle Version: [mysql-installer-community-8.0.41.0.msi](https://dev.mysql.com/downloads/file/?id=536788)

Nächste Seite: “No thanks, just start my download.”

Als Administrator ausführen und installieren;

* Starten Sie das Installationsprogramm und setzen Sie die Installationsoptionen wie folgt. Wählen Sie als Installationstyp „Server only“. Dann „Next“ drücken. Danach „Execute“.
* Sobald die Installation abgeschlossen ist, müssen Sie den Server konfigurieren. Verwenden Sie die Standardoptionen und klicken Sie auf “Next”.
* “Type and Networking” unverändert lassen, “Next”
* “Authentication Method”: „Use Strong Password Encryption for Authentication“, wie voreingestellt, “Next”.
* „Accounts and Roles“: Gewünschtes Root-Passwort doppelt eingeben und aufschreiben. Es wird später in den database-\*.env-Dateien verwendet. Dann „Next“ drücken.
* „Windows Service“, „Server File Permissions“: Standardoptionen beibehalten und „Next“ drücken.
* “Apply Configuration”: Abschließen mit “Execute”. Dann Finish.
* Der Server sollte jetzt laufen. Wenn Sie den Server erneut starten müssen, gehen Sie zu „Dienste“ (linke Maustaste auf Windows-Symbol und dann Dienste auswählen) und dort den Eintrag „MySQL80“ suchen und „starten“ oder „neu starten“ klicken.
* Zur Fehlerbehebung lesen Sie bitte die [MySQL-Dokumentation für die Installation über das Windows-Installationsprogramm](https://dev.mysql.com/doc/refman/8.3/en/windows-installation.html).
* Am besten außerdem die MySQL Workbench herunterladen und installieren.  
  <https://dev.mysql.com/downloads/workbench/>
* Nicht behebbare Probleme mit der Datenbank:
  + MySQL Server 8.0 und MySQL Installer deinstallieren, ggf. besten auch die Workbench
  + Explorer öffnen, in das Userverzeichnis wechseln
  + Im Explorer über der Dateiliste auf die 3 Punkte klicken, Ansicht und dann die versteckten Dateien und Ordner anzeigen lassen
  + Nach Bestätigung erscheinen zusätzliche Ordner. Hier dem Pfad „AppData\Roaming“ folgen. Falls vorhanden, dort den Ordner MySQL löschen. Dabei alle erfragten Löschbestätigungen geben.
  + Nun den Datenbank-Server, wie zu Beginn von Schritt 3 beschrieben, neu installieren.
  + Durch Starten des Backend laut Abschnitt 8 sollte die Datenbank neu gefüllt werden.

1. Microgrid-Planner installieren

* Projektseite: <https://github.com/reichd/MicrogridPlanner>   
  Direktlink zum Quellcode: <https://github.com/reichd/MicrogridPlanner/archive/refs/heads/main.zip>
* ZIP-Archiv am Installationsort entpacken
* Es muss folgende Verzeichnisstruktur entstanden sein:

Ein Bild, das Text, Screenshot, Software, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

* Alle Dateien mit Endung „.template“ kopieren und so umbenennen, dass „.template“ fehlt. So wird z.B. aus „config.ini.template“ eine Datei „config.ini“. Nachfrage wegen geänderter Endung mit „ja“ bestätigen. *Alternatives Vorgehen:* Datei mit einem Texteditor wie [Notepad++](https://notepad-plus-plus.org/downloads/)  öffnen und unter dem neuen Namen ohne „.template“ abspeichern.
* Bei Standardeinstellung in Windows werden die neuen Endungen nicht angezeigt, wenn das System sie kennt (z.B. „.ini“).

1. Microgrid-Planner mit SQL-Datenbank verbinden

* Secret Key erzeugen: Dazu das Terminal / Eingabeaufforderung öffnen und zum Ort wechseln, an dem sich das Hauptverzeichnis von MGP befindet. Dann dort „python generate\_secret\_key.py“ aufrufen.

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

* Den Secret-Key und die Frontend-URL in der “config.ini” im Hauptverzeichnis von MGP eintragen und abspeichern (dazu ist [Notepad++](https://notepad-plus-plus.org/downloads/) besonders gut geeignet):

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

KI-generierte Inhalte können fehlerhaft sein.

* Die Datei „database-authentication.env“ wird aus dem Template erzeugt und wie folgt geändert: An der roten Stelle geben Sie das in Schritt 3 bei der MySQL-Installation vergebene Root-Passwort an. An der gelb unterlegten Stelle wird die Adresse festgelegt, mit der die Anwendung später im Browser aufgerufen wird (Standard-Wert). Bei User und Passwort lassen sie alles stehen. Der Port muss beim Wert „3306“ bleiben.

Ein Bild, das Text, Screenshot, Software, Webseite enthält.

KI-generierte Inhalte können fehlerhaft sein.

1. Frontend von MGP installieren

* Auch hier die Datei config.ini als umbenannte Kopie der temporären Version erzeugen und mit Texteditor bearbeiten. Dabei erst einmal nur den Host auf 127.0.0.1 setzen.

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

KI-generierte Inhalte können fehlerhaft sein.

* Im Verzeichnis „...\MicrogridPlanner-main\frontend\webapp\static\js“ die Datei „paths.js.template“ zu „paths.js“ machen.
* Im Terminal/Eingabeaufforderung von Windows (s. oben) zum Frontend-Pfad wechseln:   
  „cd …\MicrogridPlanner-main\frontend“
* Nun dort erst einmal PIP aktualisieren: „python.exe -m pip install --upgrade pip“ eingeben und mit RETURN bestätigen.
* Nun „pip3 install .“ (der finale Punkt ist wichtig) eingeben und mit RETURN bestätigen. Dann läuft ein längerer Installationsprozess durch, der hoffentlich (im Wesentlichen) fehlerfrei ist. Wenn die Software weiterentwickelt werden soll, muss pip3 install . -e“ eingegeben werden.

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

* Nun den Webserver starten. Dazu „python webapp/app.py“ eingeben und bestätigen. Das sollte nun so aussehen:

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

* Man kann jetzt schon mal 127.0.0.1:8000 im Browser aufrufen:

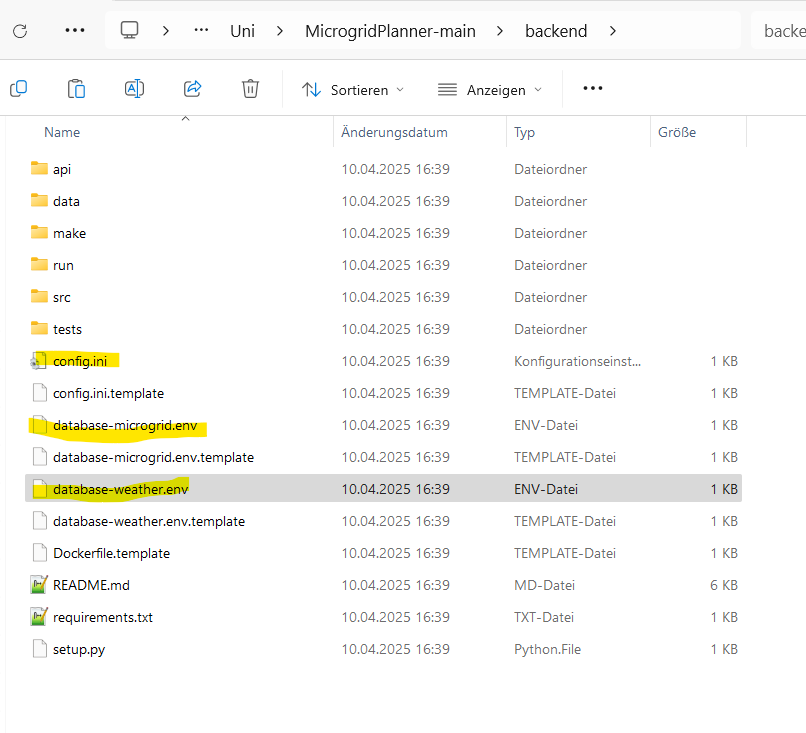
Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

* **Troubleshooting:** Neuinstallation des Frontends (wenn etwas nicht klappt oder Updates)
  + Evtl. neue Dateiversionen in das entsprechende Unterverzeichnis packen.
  + Das folgende Unterverzeichnis löschen (bdist.win-amd64 kann anders lauten):   
    „..\frontend\build\bdist.win-amd64\wheel\microgrid\_frontend-0.0.0.dist-info“
  + Rechner neu starten, um Dateiblockaden zu lösen.
  + Mit Terminalfenster zu „...\frontend“ wechseln und erneut „pip3 install.“ Aufrufen, um die Frontend-Software neu zu generieren („build“).

1. Backend von MGP installieren

* Wie zuvor die template-Dateien kopieren und ohne Endung „.template“ umbenennen:

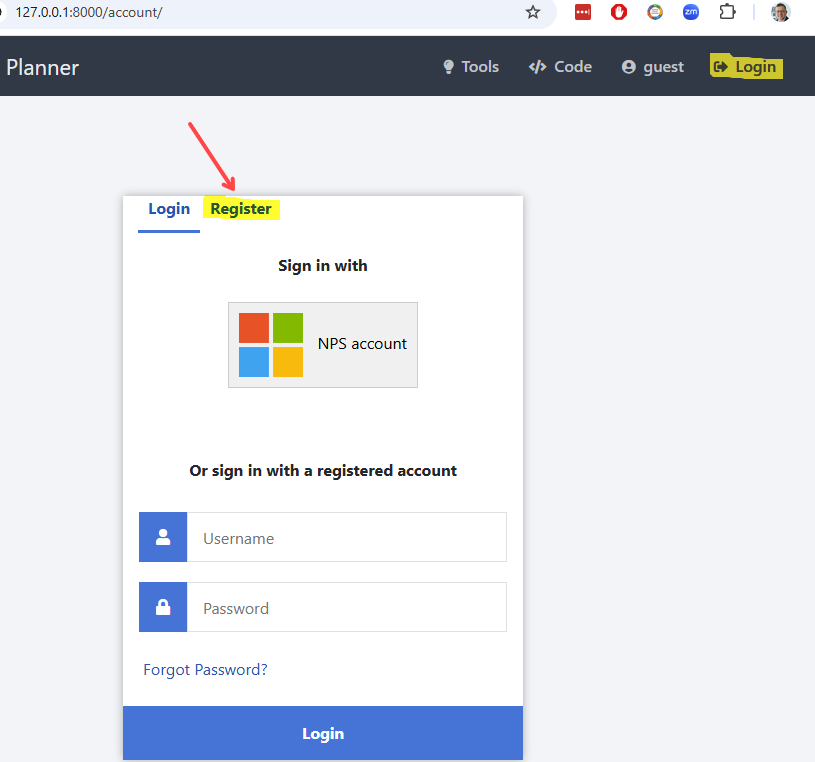


* In der „config.ini“ folgendes ändern:
  + Abschnitt DEFAULT: Verzeichnis, in dem auf dem lokalen Computer Scripts laufen sollen (kann man erst einmal freilassen)
  + SECURITY: ADMIN\_PASSWORD für den Login in die Anwendung als Administrator festlegen und eintragen (überschreibt das Passwort in “data/mysql/data.sql”)
  + API: Hier “HOST=127.0.0.1” setzen.
  + QUOTA: Benutzerbeschränkungen (nicht benötigt, also unverändert lassen)
  + SSH: Pfad zum ssh key file für Authentifizierung am SLURM server (braucht man nicht)
  + SLURM: Konfiguration für Slurm jobs (nicht benötigt, also unverändert lassen)
* Änderungen in „database-microgrid.env“:
  + MYSQL\_ROOT\_PASSWORD: Das in Schritt 3 festgelegte Root-Passwort für MySQL eintragen.
  + MYSQL\_USER, MYSQL\_PASSWORD and MYSQL\_DATABASE bleiben.
  + MYSQL\_PORT: muss bei “3306” bleiben
  + MYSQL\_HOST: 127.0.0.1 eintragen
* Änderungen in “database-weather.env“:
  + MYSQL\_ROOT\_PASSWORD: Das in Schritt 3 festgelegte Root-Passwort für MySQL eintragen.
  + MYSQL\_USER, MYSQL\_PASSWORD and MYSQL\_DATABASE bleiben.
  + MYSQL\_PORT: muss bei “3306” bleiben
  + MYSQL\_HOST: 127.0.0.1 eintragen
* Im Ordner “..\backup\data\csv\weather die Template-Dateien in „locations.csv“ und „locations.xlsx“ umwandeln. Diese Dateien liefern die Standort- und Wetterdaten. Sie können manuell oder mit der API eines Wetterdienstes oder wie in Schritt 8 (4. Anstrich) beschrieben gefüllt werden.

1. Backend lokal ausführen

Die folgenden Befehle müssen im Terminal/Eingabeaufforderung im Verzeichnis „…\MicrogridPlanner-main\backend“ gestartet werden:

* “pip3 install .” (oder „pip3 install -e .“ für Entwickler)
* “python -m pytest tests/”, um zu schauen, ob die Anwendung läuft.  
  Wahrscheinlich fehlen noch die Datenbankinhalte. Dann erst den vierten Anstrich ausführen und Test wiederholen.
* “python api/app.py” um den Webserver zu starten  
  Beim erstmaligen Start wird die Datenbank initial befüllt.
* **Userverwaltung:** Grundsätzlich kann man auf die Datenbank mit bestehenden Users zugreifen, etwa „user1“ mit Passwort „usera“. Es ist aber zu empfehlen sich zumindest noch einen Admin-User zu erstellen. Dies geht wie folgt:
  + MySQL-Workbench öffnen, in der linken Liste unten auf „Schemas“ klicken. Dann „authentication“ und darunter „Tables“ mit dem schwarzen Pfeil links daneben aufblättern. Mit der Maus über „settings“ fahren und rechts das Tabellen-Icon anklicken. Nun den Eintrag „twofactor\_protection“ suchen und den Wert auf „false“ setzen: rechte Maustaste auf Start von false, dann „Open Value in Editor“, „Reiter Text“, „true“ durch „false“ ersetzen und dreimal rechts unten auf „Apply“ drücken.

Ein Bild, das Text, Software, Computersymbol, Webseite enthält.

KI-generierte Inhalte können fehlerhaft sein.

* + Nun kann man bei laufender Anwendung (s. Abschnitt 10) unter LOGIN einen neuen Benutzer registrieren. Sobald dieser in der Datenbank erscheint (ggf. das Aktualisierungssymbol mit den zwei blauen Pfeilen drücken), kann seine „role“ auf Admin geändert werden.
* Wenn man die vorgegebenen Standort-Daten nutzen will (am Anfang zu empfehlen, allerdings erst nachdem die Datenbank einmal grundlegend angelegt wurde):
  + In „..\backend\make“ die Datei „make-data.yaml“ aus dem Template kopieren und umbenennen.
  + In der Datei „make-data.yaml“ den Eintrag „data\_dev“ auf True ändern.
  + Die Eingabeaufforderung/Terminal von Windows öffnen und in den Ordner „…\backend“ wechseln. Dort “python make/make\_data.py -c make/make\_data.yaml” aufrufen:
  + Jetzt sollte “python -m pytest tests/” im backend-Verzeichnis funktionieren.

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

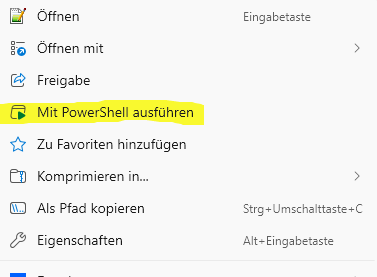
1. Standortdatenbank und Skripte

* Standorte kann man downloaden (s. Anleitung in “..\backend\data\csv\weather\ README.md”).
  + Hier einen API Key besorgen via Uni-Email-Adresse. Dieser kommt per E-Mail.  
    https://developer.nrel.gov/signup/
  + Im Verzeichnis „..\backend\src\data\csv“ die Datei „nsrdb\_api.ini“ aus dem Template erzeugen. Dort den API-Key und die dazu angegebene E-Mail-Adresse angeben.
  + In „locations.csv“ im Ordner „\backend\data\csv\weather” eine Zeile für jeden Standort eintragen, für den Daten benötigt werden. Breitengrad (latitude) und Längengrad longitude) erhält man, wenn man in Google Maps auf einen Ort rechts klickt.
  + Offene Frage: Woher bekommt man den nächsten Messpunkt von NREL? Braucht man den?
  + Nun im Terminal das Verzeichnis “..\backend“ ansteuern und   
    „python src\data\csv\nsrdb\_download\_data.py“ starten. Solange das Terminal offen und das Skript laufen lassen, bis der Cursor auf eine neue Terminalzeile springt. Das Skript stößt den Prozess bei NREL an, macht ggf. Pausen, weil man eine Quota bei NREL hat und wartet bis die Downloads angeboten werden.
* Skripte im Verzeichnis „..\backend\run“ sind ausführbar, z.B.   
  python run/compute.py -m "simulate" -c run/simulate/compute.yml  
  In der Datei …\backend\run\README.md” finden sich Anleitungen für das Ausführen von Analysemethoden.

1. Betrieb von Backend und Frontend

* Zwei getrennte Eingabeaufforderungen/Terminals öffnen und diese Fenster offen lassen:
  + Frontend: In einem Terminal „python webapp/app.py“ im Ordner „..\frontend“ starten.
  + Backend: Im anderen Terminal „python api/app.py“ im Ordner „...\backend“ starten.
* Wenn der Datenbankserver nicht oder nicht vernünftig läuft (Fehler beim Versuch sich mit der DB zu verlinken), müsste dieser neu gestartet werden: Gehen Sie zu „Dienste“ (linke Maustaste auf Windows-Symbol und dann Dienste auswählen oder suchen) und dort den Eintrag „MySQL80“ suchen und „starten“ oder „neu starten“ klicken.
* Nun kann man im Browser die Applikation mit der URL „127.0.0.1:8000“ öffnen und es sollten auch die Tools funktionieren.
* Wenn man in einem oder beiden Terminals STRG-C drückt oder die Fenster schließt, werden die entsprechenden Services geschlossen.
* Für die Einrichtung von Benutzern und weiteren Standortdaten bitte in die entsprechenden readme-files schauen.

**Automatisierung durch Power-Shell-Skript**

* An geeigneter Stelle (z.B. Hauptverzeichnis des Microgridplanner oder User-Verzeichnis) Text-Datei mit Endung „.ps1“ anlegen, z.B. „mgp.ps1“
* Darin folgende 5 Zeilen erzeugen:  
  $AL1 = "-d c:\ …\frontend python webapp/app.py"  
  $AL2 = "-d c:\ …\backend python api/app.py"  
  Start wt.exe -ArgumentList $AL1  
  Start wt.exe -ArgumentList $AL2  
  Start chrome.exe 127.0.0.1:8000
* Datei abspeichern (ohne zusätzliche Endung „.txt“)
* Starten: mit rechter Maustaste auf die „ps1-Datei zeigen und „Mit Powershell ausführen“ wählen“ (s. rechts).
* Das Skript übernimmt alle Aufgaben, die am Anfang des Kapitels aufgeführt sind.
* **ACHTUNG:** Die Pfade müssen auf die eigene Verzeichnisstruktur angepasst werden.   
  Beispiel: MGP liegt in D:\user\heiner\microgridplanner  
  $AL1 = "-d d:\user\heiner\microgridplanner\frontend python webapp/app.py"  
  $AL2 = "-d d:\user\heiner\microgridplanner\ backend python api/app.py "