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

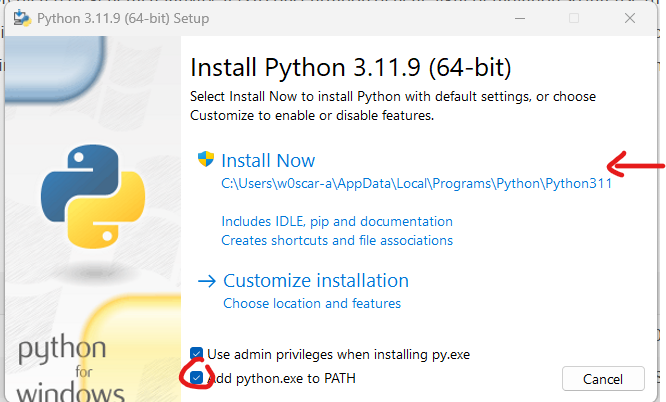
1. Check whether Python is installed and determine the version

Right-click on the Windows icon in the taskbar and start the terminal (administrator). Enter the following there and confirm with Return: python --version

If Python version 3.11 is installed, the following step can be omitted. For compatibility reasons, however, it may be better to uninstall the current Python version (via Control Panel as usual: right mouse button on Windows icon, at the top: Installed apps)) and install Python 3.11 (step 2). There were problems with newer versions

1. Download and install Python version 3.11

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



Be sure to click on "add python to PATH".

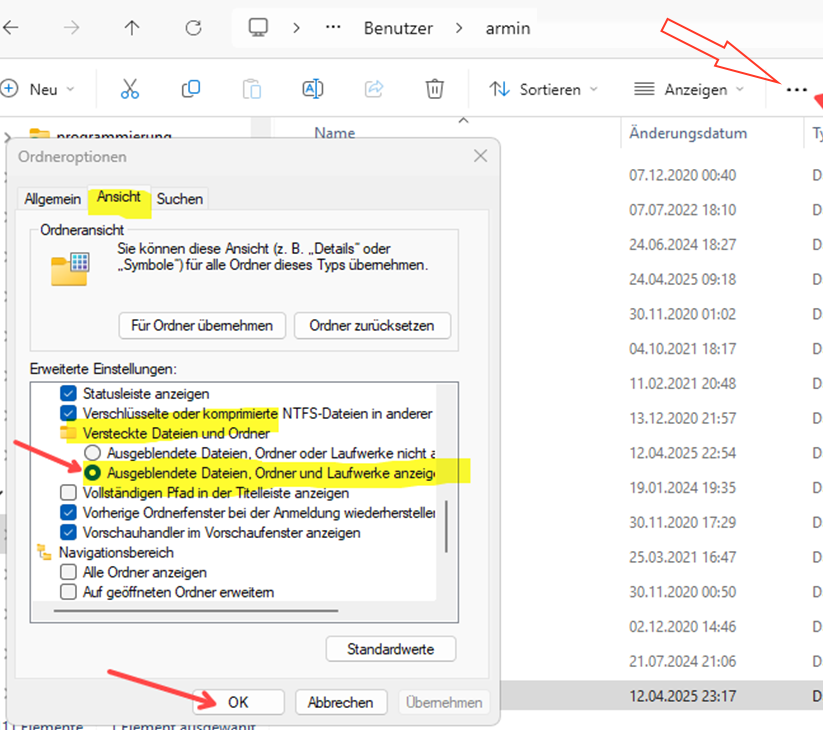
1. Install MySQL database

Download at: <https://dev.mysql.com/downloads/installer/>

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

Next page: "No thanks, just start my download."

Run as administrator and install;

* Start the installation program and set the installation options as follows. Select "Server only" as the installation type. Then press "Next". Then "Execute".
* Once the installation is complete, you must configure the server. Use the standard options and click on "Next".
* Leave "Type and Networking" unchanged, "Next"
* "Authentication Method": "Use Strong Password Encryption for Authentication", as preset, "Next".
* "Accounts and Roles": Enter the desired root password twice and write it down. It will be used later in the database-\*.env files. Then press "Next".
* "Windows Service", "Server File Permissions": Keep the default options and press "Next".
* "Apply Configuration": Finish with "Execute". Then Finish.
* The server should now be running. If you need to restart the server, go to "Services" (left mouse button on Windows icon and then select Services) and search for the entry "MySQL80" and click "Start" or "Restart".
* For troubleshooting, please read the [MySQL documentation for installation via the Windows installation program](https://dev.mysql.com/doc/refman/8.3/en/windows-installation.html).
* It is also best to download and install the MySQL Workbench.  
  <https://dev.mysql.com/downloads/workbench/>
* Unrecoverable problems with the database:
  + Uninstall MySQL Server 8.0 and MySQL Installer, if necessary also the Workbench
  + Open Explorer, change to the user directory
  + In Explorer, click on the 3 dots above the file list, View and then display the hidden files and folders
  + After confirmation, additional folders appear. Follow the path "AppData\Roaming" here. If available, delete the MySQL folder there. Enter all requested deletion confirmations.
  + Now reinstall the database server as described at the beginning of step 3.
  + The database should be refilled by starting the backend according to section 8.

1. Install Microgrid-Planner

* Project page: [https:](https://github.com/reichd/MicrogridPlanner)  
  Direct link to the source code: <https://github.com/reichd/MicrogridPlanner/archive/refs/heads/main.zip>
* Unpack the ZIP archive at the installation location
* The following directory structure must have been created:

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

KI-generierte Inhalte können fehlerhaft sein.

* Copy all files with the extension ".template" and rename them so that ".template" is missing. For example, "config.ini.template" becomes a "config.ini" file. Confirm request for changed ending with "yes". *Alternative procedure:* Open the file with a text editor such as [Notepad++](https://notepad-plus-plus.org/downloads/) and save it under the new name without ".template".
* The default setting in Windows does not display the new extensions if the system recognizes them (e.g. ".ini").

1. Connect Microgrid-Planner with SQL database

* Generate secret key: To do this, open the terminal / command prompt and change to the location where the main directory of MGP is located. Then call "python generate\_secret\_key.py" from there.

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

KI-generierte Inhalte können fehlerhaft sein.

* Enter the secret key and the frontend URL in the "config.ini" in the main directory of MGP and save ([Notepad++](https://notepad-plus-plus.org/downloads/) is particularly suitable for this):

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

KI-generierte Inhalte können fehlerhaft sein.

* The file "database-authentication.env" is generated from the template and changed as follows: In the red field, enter the root password assigned in step 3 during the MySQL installation. The address with which the application is later called up in the browser (default value) is defined in the yellow highlighted area. Leave everything as it is for the user and password. The port must remain at the value "3306".

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

KI-generierte Inhalte können fehlerhaft sein.

1. Install MGP frontend

* Again, create the config.ini file as a renamed copy of the temporary version and edit it with a text editor. Only set the host to 127.0.0.1 for the time being.

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

KI-generierte Inhalte können fehlerhaft sein.

* In the directory "...\MicrogridPlanner-main\frontend\webapp\static\js" change the file "paths.js.template" to "paths.js".
* Switch to the frontend path in the Windows terminal/command prompt (see above):   
  "cd ...\MicrogridPlanner-main\frontend"
* Now first update PIP there: enter "python.exe -m pip install --upgrade pip" and confirm with RETURN.
* Now enter "pip3 install ." (the final point is important) and confirm with RETURN. A lengthy installation process will then run, which will hopefully be (essentially) error-free. If the software is to be further developed, pip3 install . -e" must be entered.

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

* Now start the web server. To do this, enter "python webapp/app.py" and confirm. It should now look like this:

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

* You can now call up 127.0.0.1:8000 in the browser:

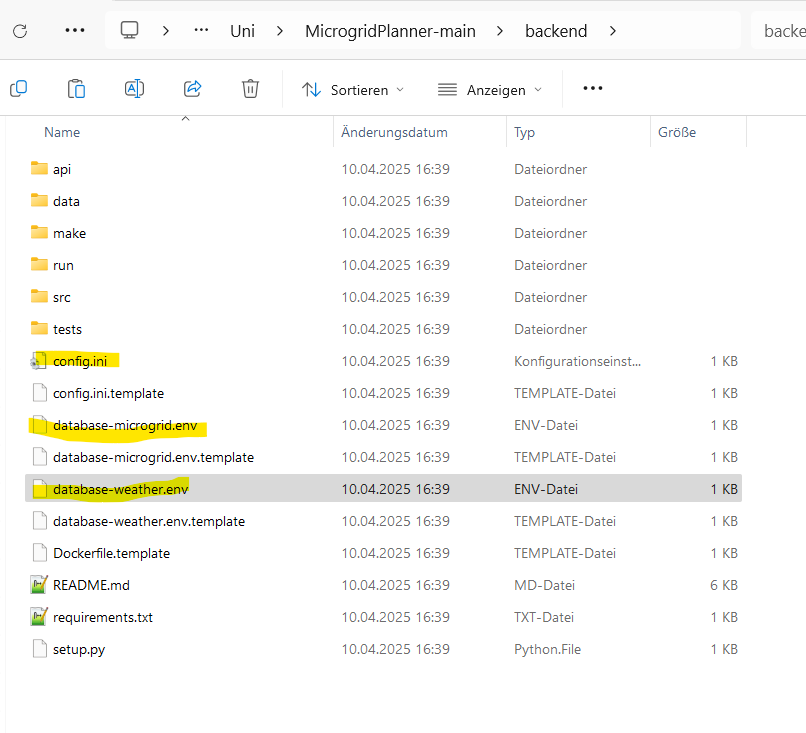
Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

* **Troubleshooting:** Reinstallation of the front end (if something does not work or updates)
  + If necessary, pack new file versions into the corresponding subdirectory.
  + Delete the following subdirectory (bdist.win-amd64 may be different):   
    "..\frontend\build\bdist.win-amd64\wheel\microgrid\_frontend-0.0.0.dist-info"
  + Restart the computer to clear file blockages.
  + Switch to "...\frontend" with terminal window and call "pip3 install." again. Call up to regenerate the frontend software ("build").

1. Install MGP backend

* As before, copy the template files and rename them without the ".template" extension:

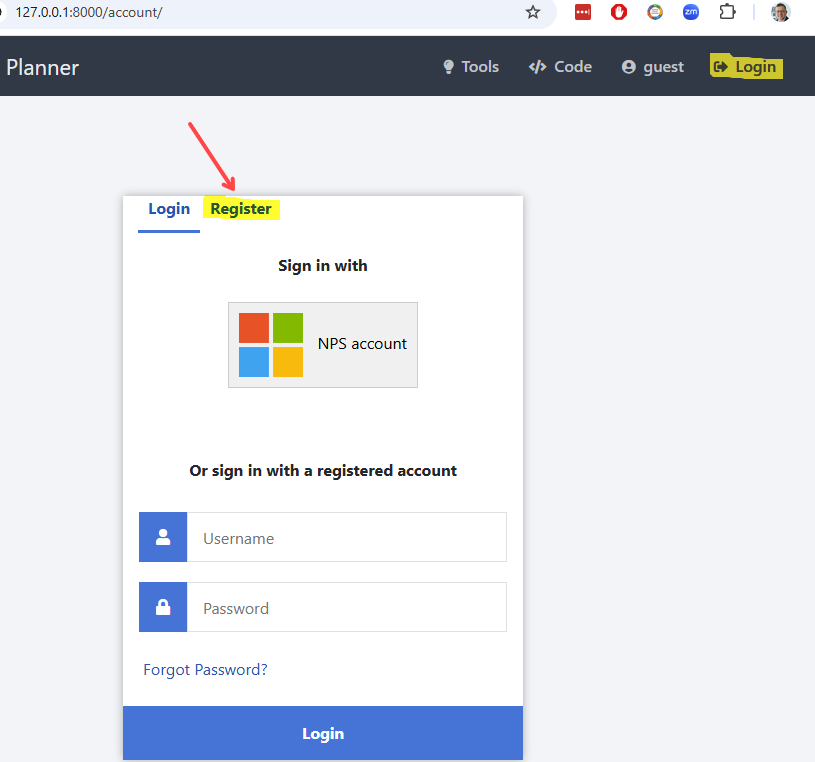


* Change the following in the "config.ini":
  + DEFAULT section: Directory in which scripts should run on the local computer (can be left blank for now)
  + SECURITY: Set and enter ADMIN\_PASSWORD for login to the application as administrator (overwrites the password in "data/mysql/data.sql")
  + API: Set "HOST=127.0.0.1" here.
  + QUOTA: User restrictions (not required, so leave unchanged)
  + SSH: Path to the ssh key file for authentication on the SLURM server (not required)
  + SLURM: Configuration for Slurm jobs (not required, so leave unchanged)
* Changes in "database-microgrid.env":
  + MYSQL\_ROOT\_PASSWORD: Enter the root password for MySQL defined in step 3.
  + MYSQL\_USER, MYSQL\_PASSWORD and MYSQL\_DATABASE remain.
  + MYSQL\_PORT: must remain at "3306"
  + MYSQL\_HOST: Enter 127.0.0.1
* Changes in "database-weather.env":
  + MYSQL\_ROOT\_PASSWORD: Enter the root password for MySQL defined in step 3.
  + MYSQL\_USER, MYSQL\_PASSWORD and MYSQL\_DATABASE remain.
  + MYSQL\_PORT: must remain at "3306"
  + MYSQL\_HOST: Enter 127.0.0.1
* Convert the template files in the folder "..\backup\data\csv\weather" to "locations.csv" and "locations.xlsx". These files provide the location and weather data. They can be filled manually or with the API of a weather service or as described in step 8 (4th step).

1. Execute backend locally

The following commands must be started in the terminal/command prompt in the directory "...\MicrogridPlanner-main\backend":

* "pip3 install ." (or "pip3 install -e ." for developers)
* "python -m pytest tests/" to check whether the application is running.  
  The database contents are probably still missing. Only then execute the fourth coat of paint and repeat the test.
* "python api/app.py" to start the web server  
  The database is initially filled at the first start.
* **User administration:** In principle, you can access the database with existing users, such as "user1" with the password "usera". However, it is recommended to create at least one admin user. This is done as follows:
  + Ein Bild, das Text, Software, Computersymbol, Webseite enthält.

    KI-generierte Inhalte können fehlerhaft sein.Open MySQL-Workbench, click on "Schemas" in the list at the bottom left. Then click on "authentication" and then on "Tables" with the black arrow to the left. Move the mouse over "settings" and click on the table icon on the right. Now search for the entry "twofactor\_protection" and set the value to "false": right-click on Start of false, then "Open Value in Editor", "Text tab", replace "true" with "false" and click "Apply" three times at the bottom right.
  + Now you can register a new user under LOGIN while the application is running (see section 10). As soon as this user appears in the database (if necessary, press the refresh icon with the two blue arrows), their "role" can be changed to Admin
* If you want to use the default location data (recommended at the beginning, but only after the database has been set up):
  + In "..\backend\make" copy the file "make-data.yaml" from the template and rename it.
  + In the "make-data.yaml" file, change the "data\_dev" entry to True.
  + Open the Windows command prompt/terminal and switch to the "...\backend" folder. There call "python make/make\_data.py -c make/make\_data.yaml":
  + Now "python -m pytest tests/" in the backend directory should work.

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

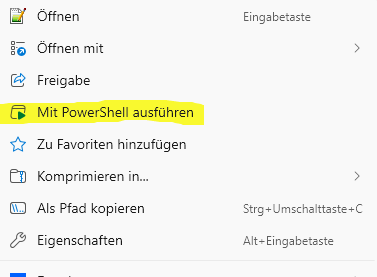
1. Location database and scripts

* Locations can be downloaded (see instructions in "..\backend\data\csv\weather\ README.md").
  + Get an API key here via your university e-mail address. This will be sent by e-mail.  
    https://developer.nrel.gov/signup/
  + Create the "nsrdb\_api.ini" file from the template in the "..\backend\src\data\csv" directory. Enter the API key and the e-mail address specified there.
  + In "locations.csv" in the "\backend\data\csv\weather" folder, enter a line for each location for which data is required. Latitude and longitude can be obtained by right-clicking on a location in Google Maps.
  + Open question: Where do you get the next measuring point from NREL? Do you need it?
  + Now navigate to the "..\backend" directory in the terminal and   
    start "python src\data\csv\nsrdb\_download\_data.py". Leave the terminal open and run the script until the cursor jumps to a new terminal line. The script triggers the process at NREL, pauses if necessary because you have a quota at NREL and waits until the downloads are offered.
* Scripts in the "..\backend\run" directory can be executed, e.g.   
  python run/compute.py -m "simulate" -c run/simulate/compute.yml  
  Instructions for executing analysis methods can be found in the file ...\backend\run\README.md".

1. Operation of backend and frontend

* Open two separate prompts/terminals and leave these windows open:
  + Frontend: Start "python webapp/app.py" in the "..\frontend" folder in a terminal.
  + Backend: Start "python api/app.py" in the "...\backend" folder in the other terminal.
* If the database server is not running or not running properly (error when trying to link to the database), it should be restarted: Go to "Services" (left mouse button on Windows icon and then select or search for services) and there search for the entry "MySQL80" and click "start" or "restart".
* Now you can open the application in the browser with the URL "127.0.0.1:8000" and the tools should also work.
* If you press CTRL-C in one or both terminals or close the windows, the corresponding services are closed.
* To set up users and other location data, please refer to the corresponding readme files.

**Automation through Power Shell script**

* Create a text file with the extension ".ps1" in a suitable location (e.g. main directory of the Microgridplanner or user directory), e.g. "mgp.ps1".
* Create the following 5 lines in it:  
  $AL1 = "-d :\ ...\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
* Save file (without additional extension ".txt")
* Start: right-click on the "ps1 file and select "Run with Powershell" (see right).
* The script performs all the tasks listed at the beginning of the chapter.
* **ATTENTION:** The paths must be adapted to your own directory structure.   
  Example: MGP is located 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 "