

Getting Started with APEX

A friendly guide for people who have never used GitHub before

This guide explains what open source is, how GitHub works, what "cloning" means, and how to install **APEX — the AI-Powered MT5 EA Optimizer** on your own computer. It assumes zero programming experience. If a word looks technical, there's a glossary at the end.

Part 1 — What is open source?

Imagine a great chef writes down a recipe and posts it on the wall outside their restaurant. They say: *"Anyone can read this. Anyone can cook it at home. Anyone can change it. Anyone can even sell their version of the dish — as long as they say I came up with the original."*

That's open source software. The "recipe" is the code. The "wall" is a website called **GitHub**. The "anyone" is — well — anyone in the world with an internet connection.

A few things this means in practice:

- **It's free.** You don't pay to download or use it.
- **You can read every line of how it works.** No hidden behaviour, no spyware.
- **You can change it.** If you want it to do something differently, you can edit the recipe.
- **You can share your version.** If you make it better, you can post your improved version too.

Open source isn't charity. The chef benefits because their recipe spreads, people learn their style, they get a reputation, and they can still sell their *premium* dishes (or run a restaurant). For software, the same logic — the original creator builds trust by giving the basic version away, and may sell a hosted version, premium features, or support on top.

APEX is open source under what's called the MIT License, which is the most permissive version of "the recipe on the wall" — you can do basically anything with it.

Part 2 — What is GitHub?

GitHub is the website where the recipes live. Think of it as a giant shared filing cabinet for code. Every project has its own drawer (called a **repository**, or **repo** for short).

Three things you need to know GitHub does:

1. **Stores the code.** Every file, every change, going all the way back to the first day of the project.
2. **Tracks history.** Anyone can see who changed what, when, and why. This is called the project's "commit history" — like a logbook.
3. **Lets people copy the project.** With one command, you can copy the entire project — including all of its history — to your own computer. That copy is yours. You can use it, change it, and run it locally without affecting the original.

The act of making that personal copy is called **cloning**. We'll do this in Part 4.

Part 3 — What is a repository (repo)?

A "repo" is just a folder, but a very organised folder. APEX's repo lives at:

https://github.com/tonnylegacy/MT5_Optimizer

If you visit that link in any web browser, you'll see:

- A **README.md** at the bottom — like the cover sheet of a book. It explains what the project is.
- A list of folders and files (the actual code).
- A green **< > Code** button — that's the button you click to copy the project.
- Tabs at the top: **Code**, **Issues** (where users report bugs), **Pull requests** (where people suggest changes), **Actions**, etc. You can ignore most of these as a new user.

You don't need a GitHub account to download APEX. You only need an account if you want to *contribute back* to the project (suggest changes, report bugs).

Part 4 — What does "cloning" mean?

Cloning means: *"Make a complete personal copy of this entire project on my computer, including its full history."*

Once cloned, the copy on your computer is yours. You can:

- Run the software.
- Change the code if you want.
- Pull new updates from the original whenever you like (with a single command).

It's like photocopying a book — once you have the photocopy, the bookstore can't take it back, and you don't need an internet connection to read it.

There are two ways to clone APEX. The easy way (no Git knowledge needed) and the proper way (so you can get future updates).

Part 5 — What you need to install first

Before you clone APEX, you need three things on your computer. You only do this **once**, ever.

5.1 — Git (the cloning tool)

Git is the program that does the actual copying. To install it:

- **Windows:** Download from <https://git-scm.com/download/win>. Run the installer, click "Next" on every screen, accept the defaults. It's safe.
- **Mac:** Open Terminal, type `git --version` and press Enter. If it says "command not found", a popup will offer to install it. Click yes.
- **Linux:** Run `sudo apt install git` (Ubuntu/Debian) or `sudo dnf install git` (Fedora) in a terminal.

To check it worked: open a terminal (Windows: search "Command Prompt"; Mac/Linux: open Terminal) and type:

```
git --version
```

You should see something like `git version 2.45.0`. Done.

5.2 — Python 3.11 or newer

APEX is written in Python. You need Python installed to run it.

- **Windows:** Download from <https://www.python.org/downloads/>. **Important:** on the first screen of the installer, tick the checkbox "Add Python to PATH". This is the easiest mistake to forget.
- **Mac:** Type `python3 --version` in Terminal. If it's older than 3.11, install via <https://www.python.org/downloads/macos/> or with Homebrew (`brew install python@3.11`).
- **Linux:** Most Linux distributions already have Python 3.

Check it worked:

```
python --version
```

Should say `Python 3.11.x` or higher.

5.3 — MetaTrader 5 (only if you want to run real backtests)

If you just want to see the AI loop work without actually backtesting trading strategies, you can skip this — APEX has a **demo mode** that fakes the backtest results.

If you want real backtests:

- Download MT5 from your broker, or from <https://www.metatrader5.com/en/download>.
- Install it normally.
- Note the path to `terminal64.exe` — usually `C:\Program Files\MetaTrader 5\terminal64.exe`. You'll need this in Part 7.

Part 6 — Cloning APEX (step by step)

Option A — The graphical way (no terminal needed)

If terminals scare you, this is fine.

1. Visit https://github.com/tonnylegacy/MT5_Optimizer in your browser.
2. Click the green < > **Code** button (top right of the file list).
3. Click **Download ZIP** at the bottom of the dropdown.
4. Save the ZIP file somewhere you'll remember — like your Desktop or Documents folder.
5. Unzip it. You'll get a folder called `MT5_Optimizer-main`.

That's it. You have APEX. The downside: when the project updates next week, you have to re-download the ZIP. The Git method (Option B) updates with one command.

Option B — The Git way (recommended, gives you updates)

1. Open a terminal:

- **Windows:** Press the Windows key, type "Command Prompt", press Enter.
- **Mac/Linux:** Open the **Terminal** app.

2. Navigate to where you want APEX to live. For example, your Desktop:

```
cd Desktop
```

(If you want it somewhere else, replace `Desktop` with that folder name. Use `cd ..` to go up one level.)

3. Run the clone command. **Copy this exactly:**

```
git clone https://github.com/tonnylegacy/MT5_Optimizer.git
```

4. Press Enter. You'll see a few lines of progress. After 10–30 seconds it'll finish.

5. You now have a folder called `MT5_Optimizer` containing the entire project. Move into it:

```
cd MT5_Optimizer
```

You're cloned. From now on, when the project updates, you can get the latest version by going into the folder and running:

```
git pull
```

That's it. No re-downloading ZIPs.

Part 7 — Setting up APEX after cloning

You have the code. Now you need to install Python's helper packages and configure a few things. About 5 minutes.

7.1 — Install the Python packages

In the same terminal, while inside the `MT5_Optimizer` folder, run:

```
pip install -r requirements.txt
```

This downloads all the libraries APEX needs (Flask, Pandas, Anthropic SDK, etc.). It takes a couple of minutes the first time and prints a lot of text — that's normal.

If you get an error about `pip` not being found, try `pip3 install -r requirements.txt`. If that still fails, your Python install is missing — go back to Part 5.2.

7.2 — Create your config file

APEX comes with a template. Copy it to a real config:

- **Windows:** `copy config.example.yaml config.yaml`

- **Mac/Linux:** `cp config.example.yaml config.yaml`

Open `config.yaml` in any text editor (Notepad is fine on Windows; TextEdit on Mac).

7.3 — Add your Anthropic API key (optional, but powerful)

The AI features need an Anthropic API key. **APEX works without one** — you just don't get the live Claude reasoning. To get a key:

1. Visit <https://console.anthropic.com/>.
2. Sign up (free).
3. New accounts get free credits (about \$5, enough for ~50 optimization runs).
4. Click **API Keys** in the sidebar, then **Create Key**. Copy the key it gives you (starts with `sk-ant-...`).

In `config.yaml`, find this line:

```
anthropic_api_key: ${ANTHROPIC_API_KEY}
```

Replace `${ANTHROPIC_API_KEY}` with your real key in quotes:

```
anthropic_api_key: "sk-ant-api03-your-real-key-here"
```

Save the file.

(If you'd rather not paste keys into files, skip this step and instead set an environment variable. APEX will pick it up automatically. See the README for details.)

7.4 — Tell APEX where MetaTrader 5 lives (only if using real backtests)

Still in `config.yaml`, find the `mt5:` section and update the paths:

```
mt5:
  terminal_exe: C:/Program Files/MetaTrader 5/terminal64.exe
  appdata_path: C:/Users/<YOU>/AppData/Roaming/MetaQuotes/Terminal/<HASH>
  mql5_files_path: C:/Users/<YOU>/AppData/Roaming/MetaQuotes/Tester/<HASH>
```

Replace `<YOU>` with your Windows username and `<HASH>` with the long random folder name MT5 created when you installed it. To find it:

- Open MT5
- Go to **File** → **Open Data Folder** in the MT5 menu
- The folder that opens — copy its full path. That's your `appdata_path`.

If you're not running MT5, leave these as they are — demo mode bypasses them.

Part 8 — Running APEX

You're configured. Time to launch.

Option A — Demo mode (no MT5 needed, recommended for first run)

In the terminal, while inside `MT5_Optimizer`:

```
python -m demo.run_demo
```

You'll see a banner, then "Launching APEX server at <http://localhost:5000>". A browser tab will pop open showing the dashboard.

This mode generates fake (but realistic) backtest results so you can see the entire flow — exploration, AI iteration, validation, verdict — without needing MT5 installed. Perfect for learning the tool.

Option B — Real mode (with MT5)

```
python app.py
```

Open <http://localhost:5000> in your browser. Click **New Run** in the sidebar, register your EA on the Setup page, set thresholds, and click **Start Optimization**.

To stop the server later, go back to the terminal and press **Ctrl + C**.

Part 9 — When things go wrong (common problems)

"git: command not found"

Git isn't installed (Part 5.1) or your terminal doesn't know about it. On Windows, restart Command Prompt after installing Git.

"python: command not found"

Same problem with Python. On Windows, you probably forgot to tick "Add Python to PATH" during install. Reinstall Python and tick that box.

"ModuleNotFoundError: No module named 'flask'"

You skipped `pip install -r requirements.txt`. Run it.

Browser shows "This site can't be reached"

The server didn't start. Look at the terminal — there'll be an error message. Most common: port 5000 is already used by another program. Either close that program, or change the port in `app.py` (find the `port=5000` line at the bottom).

"AI insights are off — no Anthropic API key set"

You skipped Part 7.3. APEX still works for backtesting, but the AI features are silent. Add a key in **Settings** (gear icon in the sidebar) any time you want.

Nothing happens when I click Start

Check the bottom-left of the dashboard for the **MT5 Connection** indicator. If it says Disconnected, your MT5 paths in `config.yaml` are wrong. Double-check Part 7.4.

Part 10 — Updating APEX

When the project gets updates (bug fixes, new features), you can pull them:

If you used **Option A (ZIP)**: download the ZIP again, replace your old folder.

If you used **Option B (git clone)**:

```
cd MT5_Optimizer
git pull
```

That's it. Your `config.yaml` is preserved (it's git-ignored, meaning the project doesn't overwrite it).

Part 11 — Glossary

Word	Meaning
Open source	Software where the code is published publicly under a license that lets anyone use, modify, and share it.
License	The rules that say what you can and can't do with the code. APEX uses the MIT License , which is very permissive — you can do almost anything.
GitHub	The website where most open-source projects live. Owned by Microsoft.
Repository (repo)	A single project's folder on GitHub. Includes all the code, all the history, the README, etc.
Cloning	Making a complete personal copy of a repo on your computer, including its history.
Pull (git pull)	Updating your cloned copy with the latest changes from GitHub.
Commit	A single saved snapshot of changes. Like a checkpoint in a video game.
Branch	A parallel line of development. You can mostly ignore this as a beginner.
Pull request (PR)	A formal proposal to add your changes to someone else's project.
README	The introduction file at the top of every repo. The first thing visitors read.
Issue	A bug report or feature request, filed on the project's GitHub page.
Terminal / Command Prompt	The text-based way to run commands on your computer. Where you type <code>git clone ...</code> , <code>cd ...</code> , <code>python ...</code> , etc.
Python	The programming language APEX is written in.
pip	Python's package installer — what you use to download Python libraries.
Flask	A small web server library — APEX uses it to power the dashboard.
MIT License	The most permissive open-source license. Anyone can use the code for anything, including selling it, as long as they keep the copyright notice.

Part 12 — What to do next

You have APEX cloned, installed, and running. From here:

1. **Play with demo mode** for 10 minutes. Click around. Hit "Best Net Profit" to see the AI's evolution path. Try the Compare feature on /reports. Get a feel for what the dashboard does.
2. **Read the main README.md** in the repo for the architecture overview and the live event reference.
3. **Set up MT5 + a real EA** when you're ready for actual backtests.
4. **Star the repo on GitHub** if you like it — that's how open-source projects get noticed.
5. **Open an issue** on GitHub if you find a bug or have a question. Other users will see your question and the answer, which helps everyone.

Welcome to open source. The whole world's code is now yours to read.

Last updated 2026-04-25 · APEX version: see git log · Questions? Open an issue at <https://github.com/tonnylegacy/MT5 Optimizer/issues>