GPT Engineer

gpt-engineer 3225 members

Stars 43k

Follow @antonosika

Specify what you want it to build, the AI asks for clarification, and then builds it.

GPT Engineer is made to be easy to adapt, extend, and make your agent learn how you want your code to look. It generates an entire codebase based on a prompt.

Demo

Project philosophy

- Simple to get value
- Flexible and easy to add new own "AI steps". See `steps.py`.
- Incrementally build towards a user experience of:
    1. high level prompting
    2. giving feedback to the AI that it will remember over time
- Fast handovers back and forth between AI and human
- Simplicity, all computation is "resumable" and persisted to the filesystem

Usage

Choose either **stable** or **development**.

For **stable** release:

- `pip install gpt-engineer`

For **development**:

- `git clone https://github.com/AntonOsika/gpt-engineer.git`
- `cd gpt-engineer`
- `pip install -e .`
    - (or: `make install && source venv/bin/activate` for a venv)

Setup

With an OpenAI API key (preferably with GPT-4 access) run:

- `export OPENAI_API_KEY=[your api key]`

Alternative for Windows

- `set OPENAI_API_KEY=[your api key]` on cmd
- `$env:OPENAI_API_KEY="[your api key]"` on powershell

Run:

- Create an empty folder. If inside the repo, you can run:
  - `cp -r projects/example/ projects/my-new-project`
- Fill in the `prompt` file in your new folder
- `gpt-engineer projects/my-new-project`
  - (Note, `gpt-engineer --help` lets you see all available options. For example `--steps use_feedback` lets you improve/fix code in a project)

By running gpt-engineer you agree to our terms.

Results

- Check the generated files in `projects/my-new-project/workspace`

Getting Started with GitHub Codespaces

To get started, create a codespace for this repository by clicking this 👇

Open in GitHub Codespaces

Features

You can specify the "identity" of the AI agent by editing the files in the `preprompts` folder.

Editing the `preprompts`, and evolving how you write the project prompt, is currently how you make the agent remember things between projects.

Each step in `steps.py` will have its communication history with GPT4 stored in the logs folder, and can be rerun with `scripts/rerun_edited_message_logs.py`.

Contributing

The gpt-engineer community is building the **open platform for devs to tinker with and build their personal code-generation toolbox**.

If you are interested in contributing to this, we would be interested in having you!

You can check for good first issues here.
Contributing document here.

We are currently looking for more maintainers and community organisers. Email anton.osika@gmail.com if you are interested in an official role.

If you want to see our broader ambitions, check out the roadmap, and join
discord
to get input on how you can contribute to it.

Example

https://github.com/AntonOsika/gpt-engineer/assets/4467025/6e362e45-4a94-4b0d-973d-393a31d92d9b