



Search Medium



Write



2



♦ Member-only story

# The Dawn of AI-Powered Software Development with GPT-engineer

And explore how GPT-Engineer is changing the future of software development.

Josep Ferrer · [Follow](#)

Published in Geek Culture · 7 min read · Jun 18



451



3



...

# GPT-engineer



## The Dawn of Software Development

Self-made image. GPT-engineer and the dawn of software development.

In the realm of software development, the winds of change are blowing.

The driving force behind this transformation is none other than GPT-Engineer, an AI-powered tool that is redefining the landscape of code generation and customization.

This revolutionary tool, with its ability to generate entire codebases based on your specifications, is poised to become an indispensable ally in the developer's toolkit.

It has already been viral in GitHub with more than 11k starts and will most likely be a buzzword in the coming weeks!

*Intrigued about it?*

Let's explore this exciting new frontier together and get ahead! 

## The Rise of GPT-Engineer

GPT-Engineer is a cutting-edge AI-powered development tool designed to revolutionize code generation and customization.

*I know most of you might be wondering... why?*

It simplifies the process of creating software. All you need to do is provide a prompt, and GPT-engineer will ask for clarification as needed before generating a whole codebase customized to your chosen coding style and functionality.

*Sounds crazy right?*

This is a game-changer in software development, as it allows:

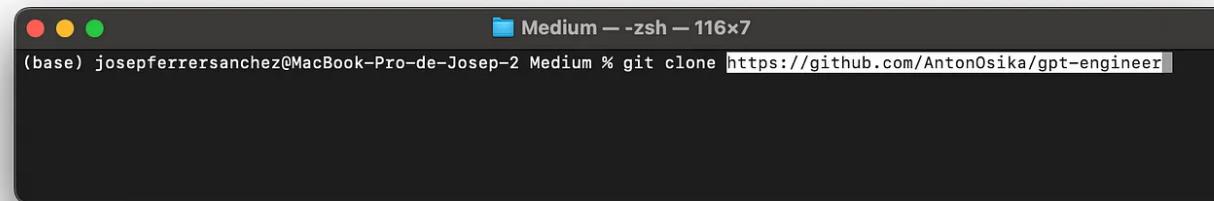
- Adaptable and dynamic development.
- Iterating over several outputs to obtain the best coding app – in just minutes!
- Personalize your AI-agent to understand your coding preferences.

*And the best part of it all?*

Setting up the environment for using GPT-Engineer is straightforward!

You just need to:

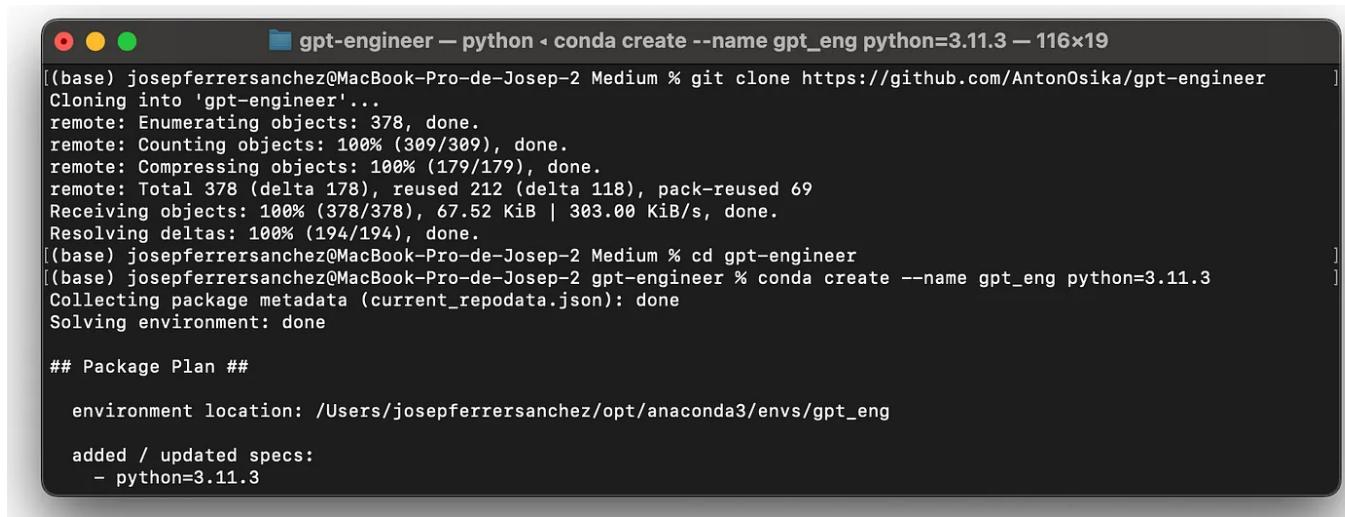
1. Clone the GPT-Engineer repository contained in the following [GitHub](#).



```
Medium — zsh — 116x7
(base) josepferrersanchez@MacBook-Pro-de-Josep-2 Medium % git clone https://github.com/antonosika/gpt-engineer
```

Self-made screenshot. Executing the git clone command on my terminal.

2. Move to the gpt-engineer folder, create a new Conda environment and activate it.



```
gpt-engineer -- python -m conda create --name gpt_eng python=3.11.3 -n 116x19
(base) josepferrersanchez@MacBook-Pro-de-Josep-2 Medium % git clone https://github.com/AntonOsika/gpt-engineer
Cloning into 'gpt-engineer'...
remote: Enumerating objects: 378, done.
remote: Counting objects: 100% (309/309), done.
remote: Compressing objects: 100% (179/179), done.
remote: Total 378 (delta 178), reused 212 (delta 118), pack-reused 69
Receiving objects: 100% (378/378), 67.52 KiB | 303.00 KiB/s, done.
Resolving deltas: 100% (194/194), done.
(base) josepferrersanchez@MacBook-Pro-de-Josep-2 Medium % cd gpt-engineer
(base) josepferrersanchez@MacBook-Pro-de-Josep-2 gpt-engineer % conda create --name gpt_eng python=3.11.3
Collecting package metadata (current_repodata.json): done
Solving environment: done

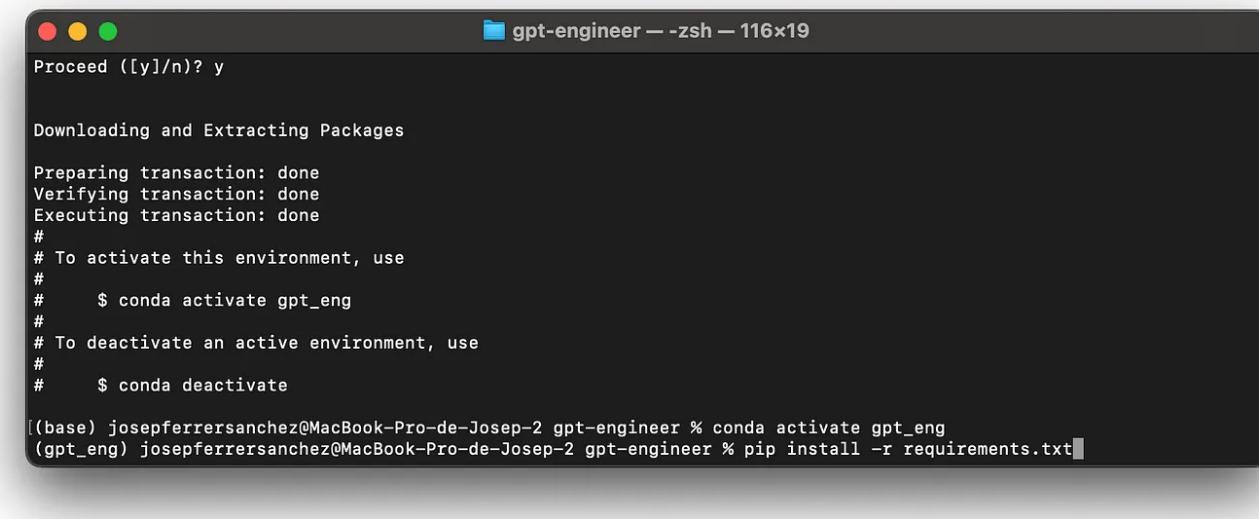
## Package Plan ##

environment location: /Users/josepferrersanchez/opt/anaconda3/envs/gpt_eng

added / updated specs:
- python=3.11.3
```

Self-made screenshot. Creating a new Conda environment.

3. Install the prerequisites contained within the requirements.txt



```
Proceed ([y]/n)? y

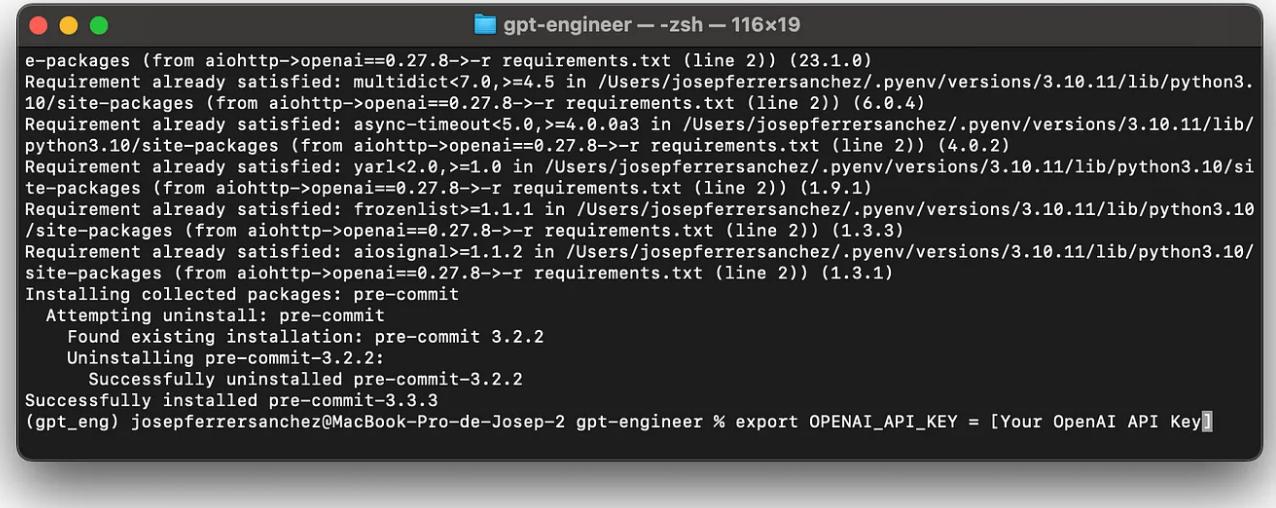
Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate gpt_eng
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) josepferrersanchez@MacBook-Pro-de-Josep-2 gpt-engineer % conda activate gpt_eng
(gpt_eng) josepferrersanchez@MacBook-Pro-de-Josep-2 gpt-engineer % pip install -r requirements.txt
```

Self-made screenshot. Installing the requirement.txt document.

4. Set up your OpenAI API Key. No idea how to get it? No worries, I'll explain in a sec 😊



```
e-packages (from aiohttp->openai==0.27.8->-r requirements.txt (line 2)) (23.1.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /Users/josepferrersanchez/.pyenv/versions/3.10.11/lib/python3.10/site-packages (from aiohttp->openai==0.27.8->-r requirements.txt (line 2)) (6.0.4)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in /Users/josepferrersanchez/.pyenv/versions/3.10.11/lib/python3.10/site-packages (from aiohttp->openai==0.27.8->-r requirements.txt (line 2)) (4.0.2)
Requirement already satisfied: yarl<2.0,>=1.0 in /Users/josepferrersanchez/.pyenv/versions/3.10.11/lib/python3.10/site-packages (from aiohttp->openai==0.27.8->-r requirements.txt (line 2)) (1.9.1)
Requirement already satisfied: frozenlist>=1.1.1 in /Users/josepferrersanchez/.pyenv/versions/3.10.11/lib/python3.10/site-packages (from aiohttp->openai==0.27.8->-r requirements.txt (line 2)) (1.3.3)
Requirement already satisfied: aiosignal>=1.1.2 in /Users/josepferrersanchez/.pyenv/versions/3.10.11/lib/python3.10/site-packages (from aiohttp->openai==0.27.8->-r requirements.txt (line 2)) (1.3.1)
Installing collected packages: pre-commit
  Attempting uninstall: pre-commit
    Found existing installation: pre-commit 3.2.2
    Uninstalling pre-commit-3.2.2:
      Successfully uninstalled pre-commit-3.2.2
Successfully installed pre-commit-3.3.3
(gpt_eng) josepferrersanchez@MacBook-Pro-de-Josep-2 gpt-engineer % export OPENAI_API_KEY = [Your OpenAI API Key]
```

Self-made screenshot. Defining the OpenAI API key.

You can directly copy all the commands in the following GitHub gist!

```
1 # Clone the GPT-Engineer repository
2 git clone <repository_url>
3
4 # Navigate to the cloned repository
5 cd gpt-engineer
6
7 # Create a new Conda environment
8 conda create --name gpt-eng python=3.11.3
9
10 # Activate the Conda Environment
11 conda activate gpt-eng
12
13 # Install the prerequisites
14 pip install -r requirements.txt
15
16 # Set up your OpenAI API Key
17 export OPENAI_API_KEY=[your_api_key]
```

GPT-engineer.py hosted with ❤️ by GitHub

[view raw](#)

Once you've successfully configured the environment, you can start using GPT-Engineer.

So now we are all set-up... so let's start with the real fun! 💥

## The Power of GPT-Engineer

GPT-Engineer's ability to generate entire codebases based on prompts is a game-changer.

For example, if you want to create a snake game, you can provide a detailed description of the game's rules, features, and requirements in a single — and super long — prompt.

GPT-Engineer will then process this prompt and generate the code for the game.

*Sounds easy... right?*

So let's see it in action!

## How to run GPT-engineer

We can easily run GPT-engineer following these steps:

### #1. How to get the OpenAI API key

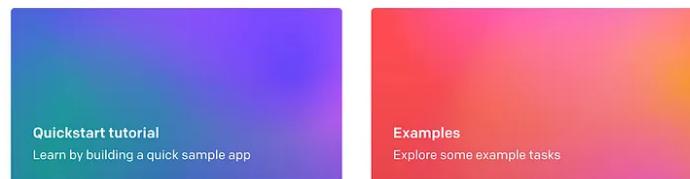
In case you do not have an OpenAI account yet, go to its [website](#) and create one. Once your account is created, go to your [account overview page](#).

You should see something like follows:

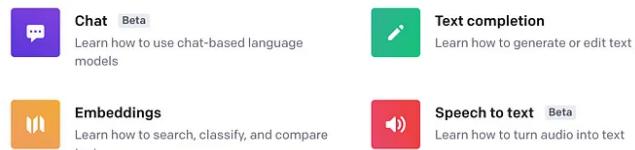


## Welcome to OpenAI

### Start with the basics



### Build an application



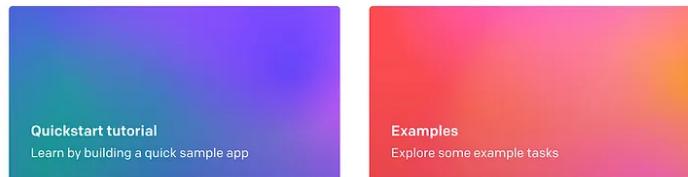
OpenAI API main view.

Now click on your *personal account profile*.

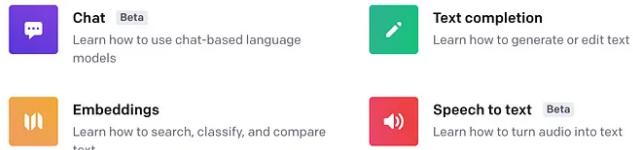


## Welcome to OpenAI

### Start with the basics

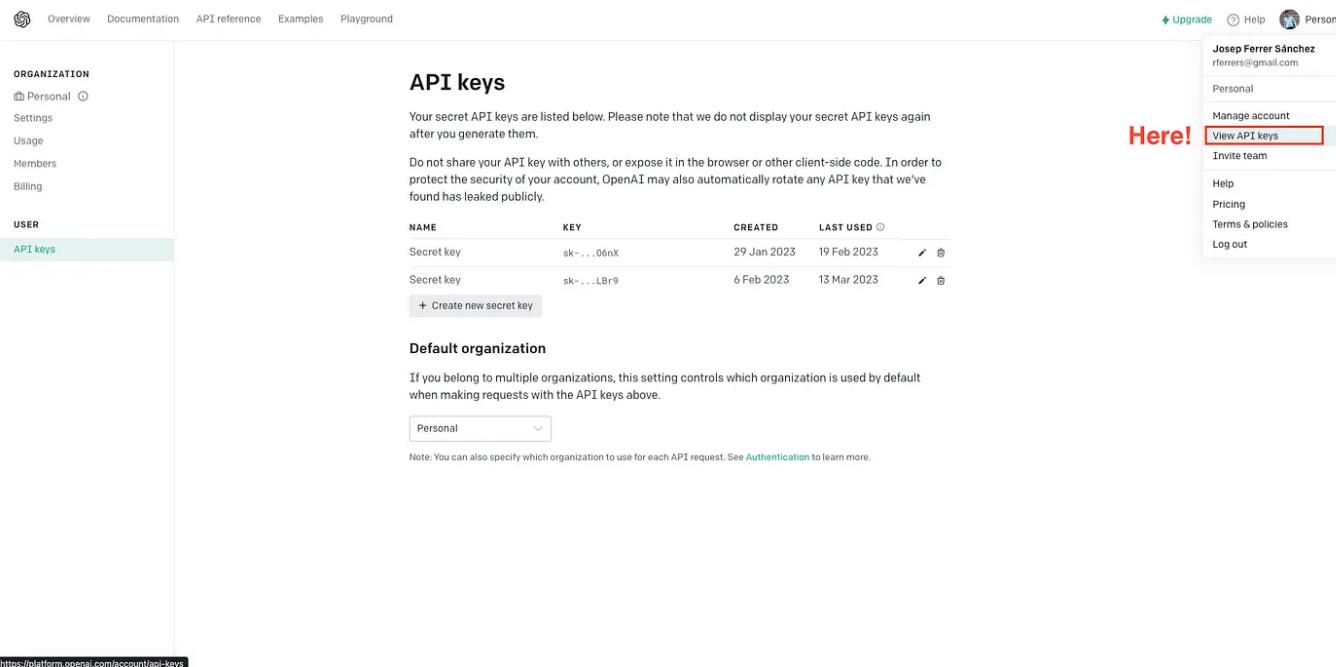


### Build an application



OpenAI API main view.

And go to the *View API Keys*.



The screenshot shows the 'API keys' section of the OpenAI API website. On the left, there's a sidebar with 'ORGANIZATION' and 'USER' sections. Under 'USER', 'API keys' is selected. The main area is titled 'API keys' and contains a table of two secret keys:

NAME	KEY	CREATED	LAST USED
Secret key	sk-...06nX	29 Jan 2023	19 Feb 2023
Secret key	sk-...LBr9	6 Feb 2023	13 Mar 2023

A red box highlights the '+ Create new secret key' button at the bottom of the table.

**Default organization**

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

A dropdown menu is set to 'Personal'.

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

<https://platform.openai.com/account/api-keys>

OpenAI API website. API keys view.

Click into the *Create new secret key*

**ORGANIZATION**

- [Personal](#)
- [Settings](#)
- [Usage](#)
- [Members](#)
- [Billing](#)

**USER**

- [API keys](#)

## API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED		
Secret key		29 Jan 2023	19 Feb 2023		
Secret key		6 Feb 2023	13 Mar 2023		
AutoGPT		25 Apr 2023	25 Apr 2023		

[+ Create new secret key](#)

**Default organization**

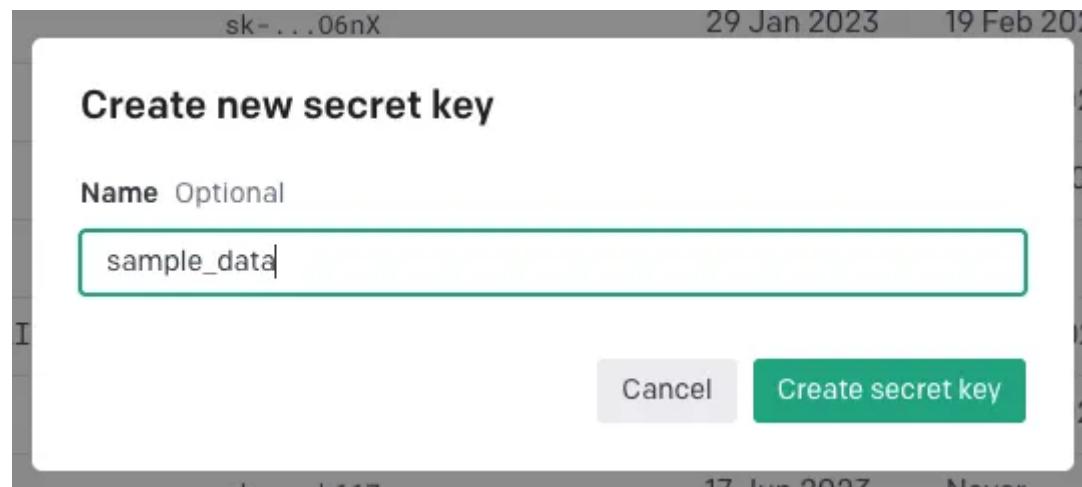
If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

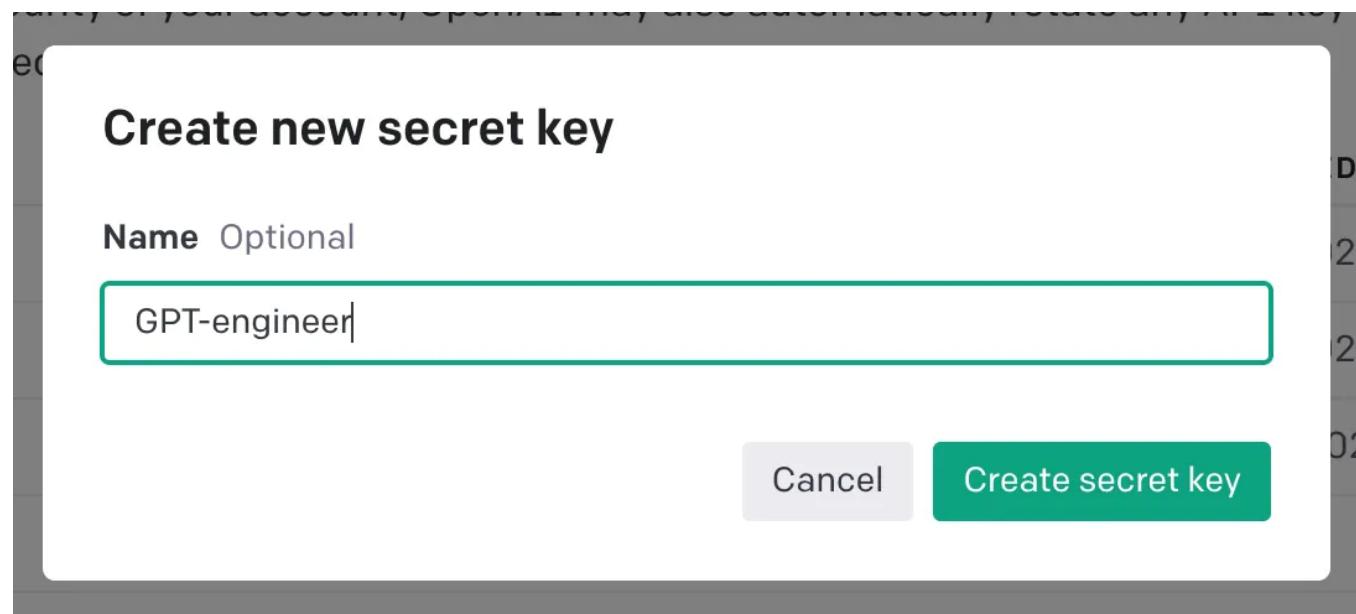
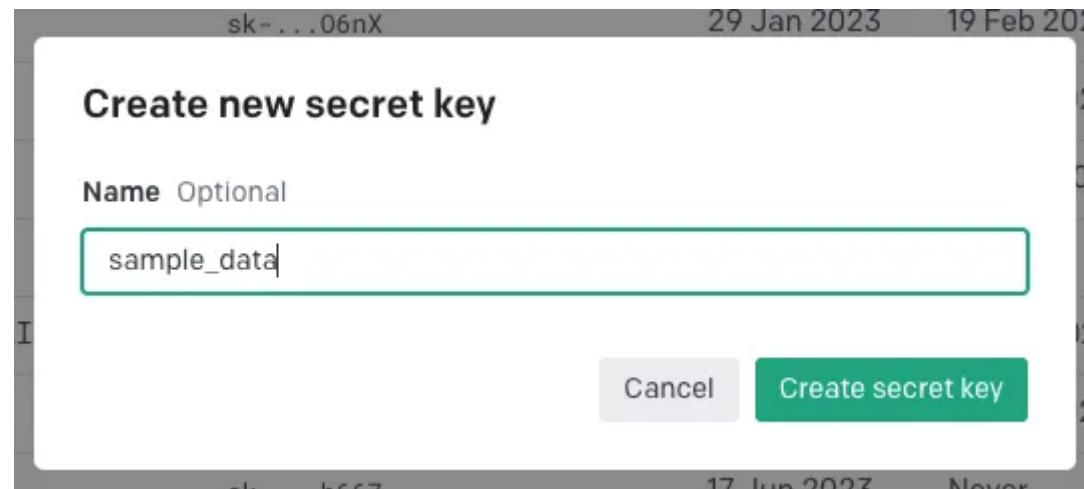
Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

OpenAI API website. API keys view.

And name it.





OpenAI API website. Creating a new API key.

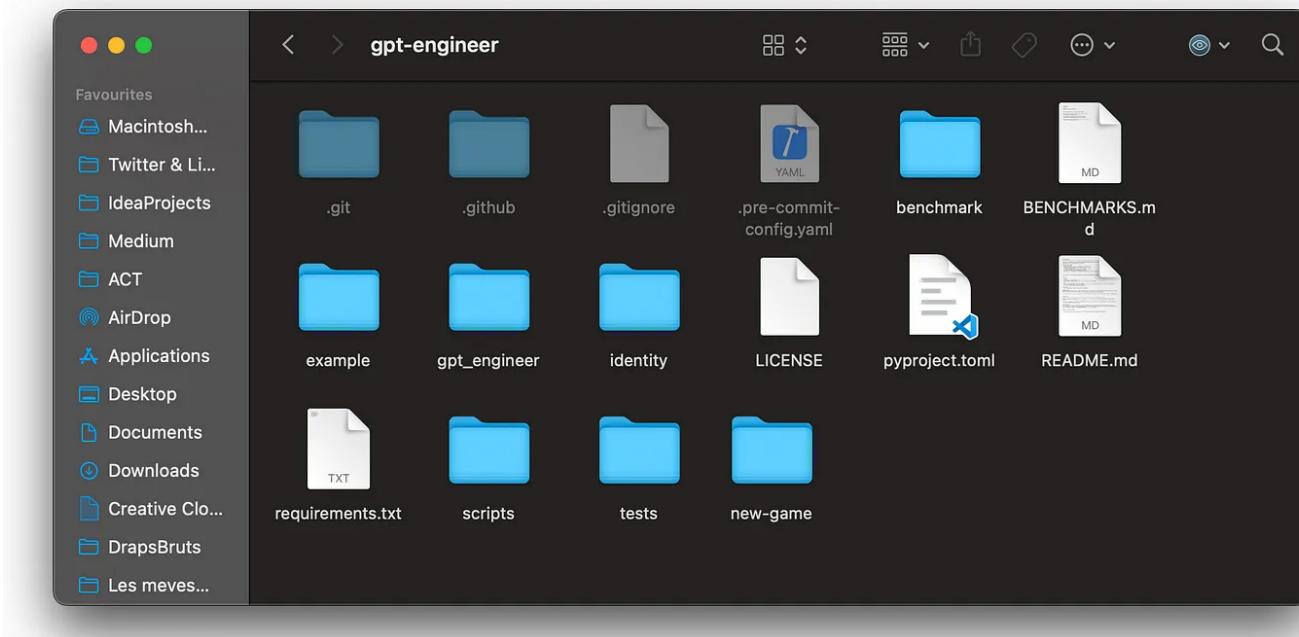
⚠ Note: Keep a copy of this key because you can't retrieve it from the web interface and we are needing it later on!

## #2. Make a New Empty Folder

Make a new folder at the location you want. This can be done manually or via the command line. To create a new folder named “new-game” in the current directory, for example, type:

```
mkdir new-game
```

You can always do it manually as well!



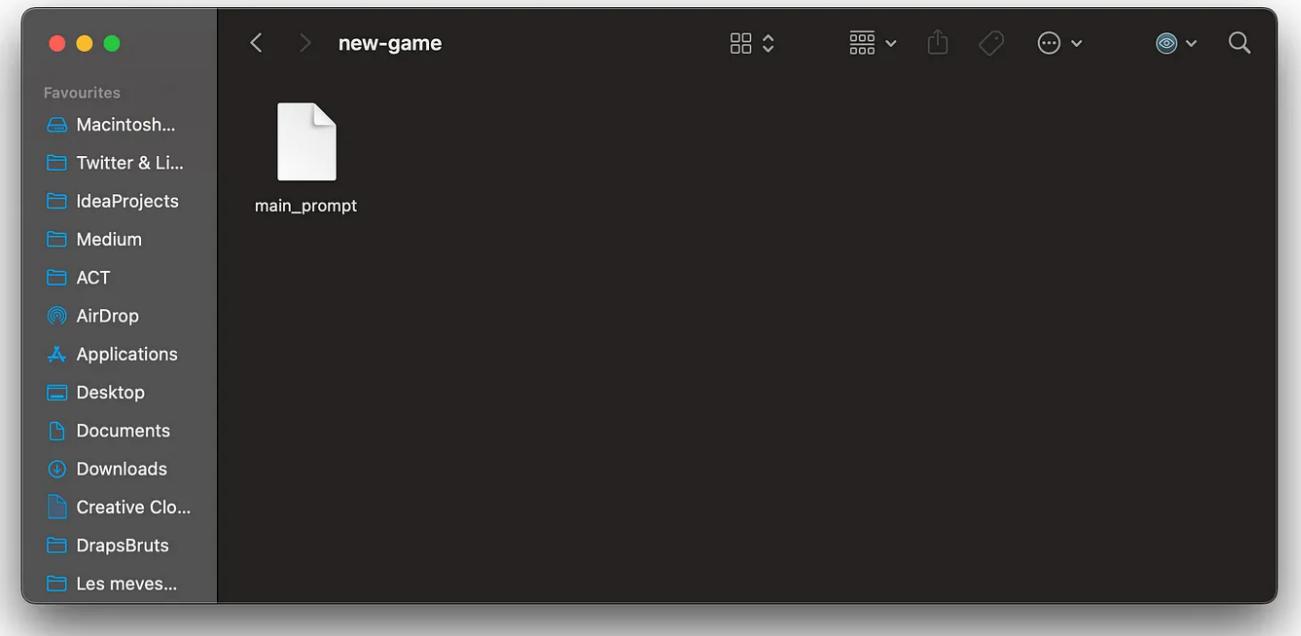
Self-made screenshot. GPT-engineer folder.

### #3. Copy Example Files

To begin with an example project structure, copy the contents of the `example` folder into your newly created folder. To copy the files and folders, use the following command.

```
cp -r example/* my-new-project/
```

Or you can do it manually as well! :)



Self-made screenshot. Creating the `main_prompt` file within our new `new-game` project folder.

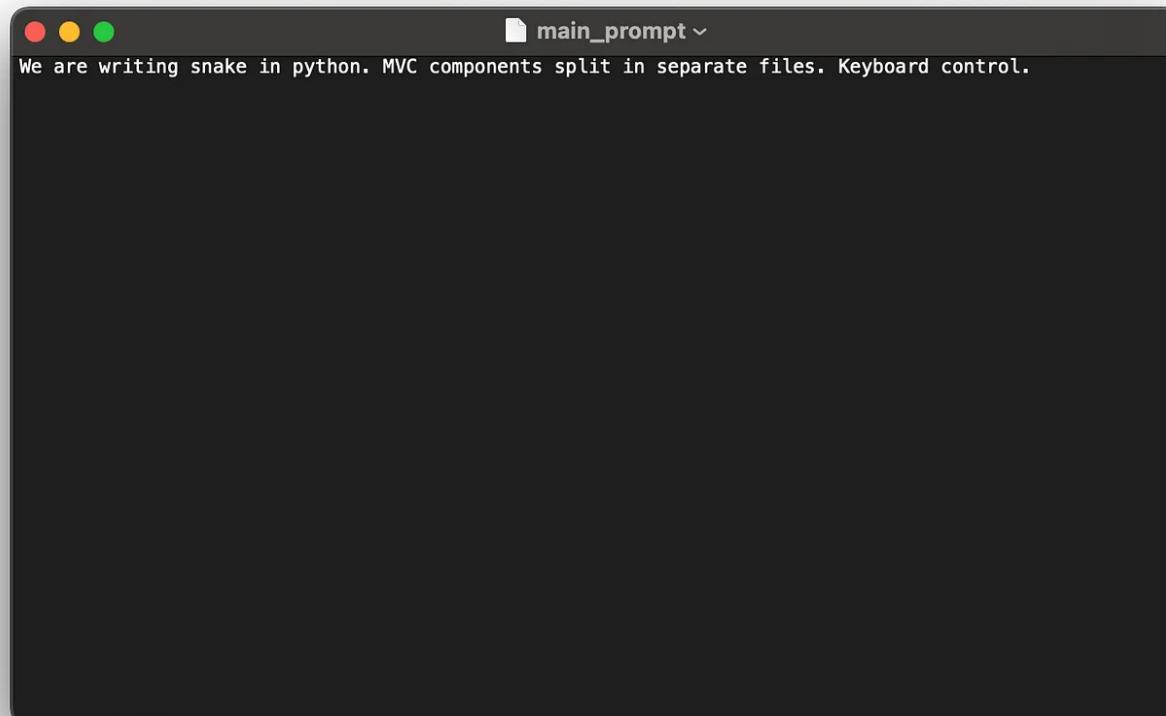
### #3. Fill in the Main Prompt

Open a text editor and navigate to the `main_prompt` file in your `new-game` folder.

Replace the present content with the code-generating prompt of your choice. Make certain that your prompt properly specifies the necessary functionality or code structure.

*No ideas...?*

Let's try the easiest one... following the example of the snake game.



Self-made screenshot. The sample prompt that comes in the repository.

We need to spice it up a bit right?

*No idea what to write or how to define the snake game?*

That's alright! We can easily ask to ChatGPT to generate a prompt for the Snake game 😊

The screenshot shows a dark-themed ChatGPT interface. At the top, it says "Model: GPT-4". The user asks, "Please, explain the rules of the snake game for a developer." The AI responds, "Sure, I'll explain the rules of the classic snake game and also provide some insights about its game mechanics from a developer's perspective:". Below this, under the heading "Game Rules:", the AI lists six points: 1. Playfield: The game is typically set in a confined 2D grid. The grid size can be defined based on the game difficulty and device display. 2. The Snake: At the beginning of the game, the snake is usually a small unit (or a couple of units) that moves in a particular direction. The snake's position is typically represented as a list or array of coordinates on the grid. 3. The Movement: The snake moves continuously, and the player controls the direction of the snake's head - up, down, left, or right. The snake can't move backwards, meaning it can't flip 180 degrees. 4. The Food: Food items appear randomly within the game grid at one spot at a time. The position of food must be validated to ensure it's not appearing inside the snake's body. 5. Eating the Food: If the snake's head touches a food item, the snake grows longer by one unit and the player's score increases. The new food item is then randomly placed on the game grid. 6. The End of the Game: The game ends if the snake touches the grid boundaries or if it touches its own body. Under the heading "From a Developer's Perspective:", the AI lists one point: 1. Game Loop: You'll need a game loop that continuously updates the game state (snake's position, food's position, game over condition, etc.) and renders it on the screen.

Self-made screenshot. ChatGPT main view.

We copy and paste it into the main-prompt. And that's it! :)

## #4. Run the GPT Engineer Script

Navigate to the GPT-engineer root directory, which is the folder containing `main.py`. To run GPT Engineer and produce code depending on your main prompt, enter the following command:

```
python main.py new-game
```

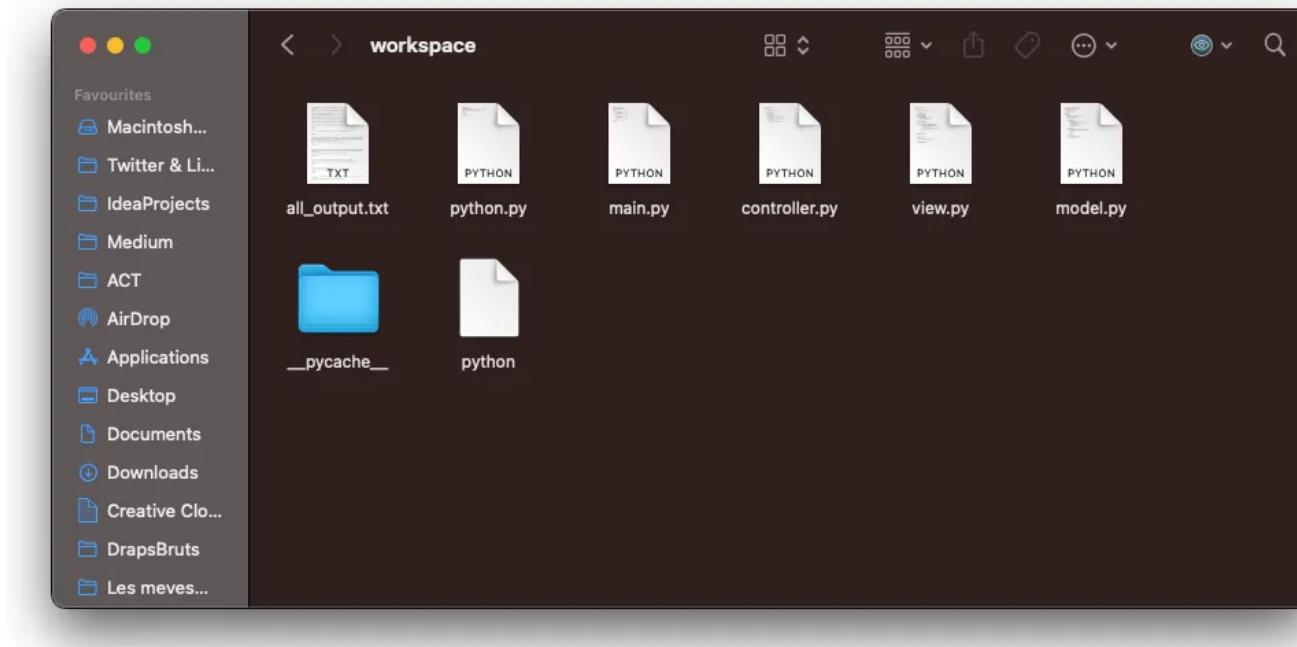
This command tells GPT Engineer to process the main prompt in the `new-game` folder and create code. This prompt serves as the foundation for developing the Snake game.



The screenshot shows a terminal window on a Mac OS X desktop. The title bar says "gpt-engineer — python -m gpt\_engineer.main new-game — 116x19". The command entered is "python -m gpt\_engineer.main new-game". A message follows: "Model gpt-4 not available for provided api key reverting to gpt-3.5.turbo. Sign up for the gpt-4 wait list here: https://openai.com/waitlist/gpt-4-api". Below this, "Core Classes and Functions:" is listed, followed by "1. Model:". The terminal window has a dark background with light-colored text.

⚠️ *GPT-4 might have a better performance compared to GPT3.5. To get access to GPT-4 via API, you need to get into the waitlist [here](#).*

We will get the results within the workspace folder of the new-game folder.

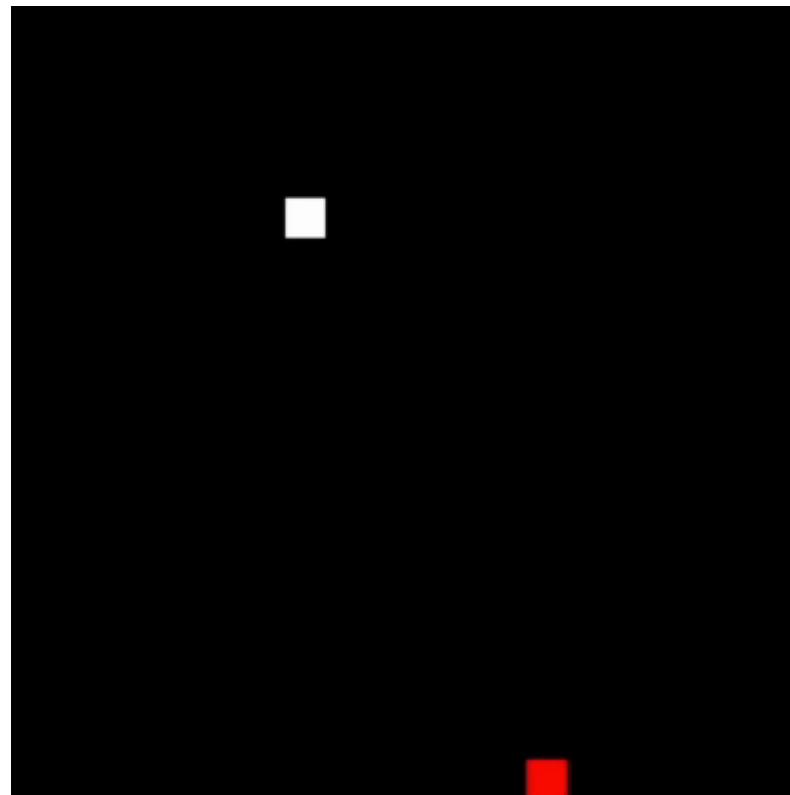


Self-made screenshot, Workspace folder where the output can be found.

And, executing with the simple command:

```
python main.py
```

The snake game will start directly!



And that's all!

*So... what have we learned from this?*

## The Impact of GPT-Engineer

AI new developments — and especially GPT models — are still being the main key players to develop solutions applied in different fields. In this case, GPT-engineer is an amazing GPT powered tool that allows:

- Generation of code in just few seconds — and using just a few words!

It allows a further customization by:

- Choosing the identity of the AI agent by editing or adding the files in the identity folder.
- Shaping the behavior and memory of the agent between different projects.
- Knowing that each step in the code generation process is stored in the logs folder, enabling rerunning of specific steps for iterative refinement and debugging of the generated code.

## Main Conclusions

The impact of GPT-Engineer on the world of software development cannot be overstated. It represents the next generation of AI-powered development tools.

With its ability to generate entire codebases based on prompts and its flexible and adaptable nature, it simplifies the process of code generation and customization.

**From high-level prompting to seamless human-AI interaction, GPT-Engineer empowers developers to efficiently build and extend their projects, opening up new possibilities in the realm of AI-driven software development.**

Despite its power, GPT-Engineer is not without its limitations. One of these is the lack of chain of thought prompting, which can improve reliability and ensure that requested functionality is not missed in the main prompt. However, the project welcomes contributions, and ideas for improvement can be found in the Projects tab of the GitHub repository.

GPT-Engineer is more than just a tool — it's a game-changer.

*Hope you found this GPT-engineer tutorial useful! :)*

*If you want to discover how to browse the internet with ChatGPT plugins, go check the following link* 

**7 new use cases now that ChatGPT is plugged into the internet.**

Unleashing the Potential of ChatGPT's Latest Update

medium.com

You can subscribe to my Medium Newsletter to stay tuned and receive my content. *I promise it will be unique!*

If you are not a full Medium member yet, just check it out here to support me and many other writers. *It really helps :D*

You can find me on Twitter and LinkedIn as well!

Some other nice medium related articles you should go check out! :D

**AutoGPT Unleashed — The Autonomous AI Agent Revolution**

Discover the Next-Generation AI System Designed to Perform Tasks Without Human Intervention

medium.com

## **Using AI won't make you a cheater — but not using it will leave you behind**

And why people using AI will be the one replacing you.

medium.com

## **Supercharge Your Productivity with 8 OpenAI's ChatGPT Plugins**

And how does the Future of Productivity look like

medium.com

Artificial Intelligence

Data Science

Python

Technology

Writing



## Written by Josep Ferrer

9.6K Followers · Writer for Geek Culture

Follow



1M+ views on Medium. || Outstand using data. ■ Data Science, Design and Tech  
 ■ Not an expert, but still trying! :D 🤪 Inquiries in [rfeers@gmail.com](mailto:rfeers@gmail.com)

## More from Josep Ferrer and Geek Culture

The screenshot shows the "New ChatGPT Plugins" section of the Plugin store. It displays several cards for different plugins:

- KAYAK**: Helps flight, travel & leisure users get recommendations for where to go on your budget.
- Notable**: A productivity tool for taking notes, SSO, and managing to-do lists. It integrates with Google Sheets, Google Calendar, and more.
- Video Insights**: Instantly extract video statistics from YouTube or Daily Motion.
- LIN Reader**: Reads the content of all kinds of links, from YouTube, PDF, EPUB, images, Word, and other documents.
- ChitoAI**: Offers the power of crowdsourcing research with peer-reviewed papers from PubMed, Arxiv, Springer, and more.
- WebScript**: Automates repetitive tasks on the web, such as extracting data from tables, interacting with forms, and more.
- Show Me**: Creates and edits diagrams directly in email.
- William**: A plugin for Microsoft Word that adds knowledge & real-time data through Microsoft Graph and Web API Language.

Josep Ferrer in Geek Culture

## Supercharge Your Productivity with 8 OpenAI's ChatGPT Plugins

And how does the Future of Productivity look like

```

parse_expenses.py
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the list of triples (date, value, currency).
5     Ignore lines starting with #.
6     Parse the date using datetime.
7     Example expenses_string:
8         2016-01-02 -34.01 USD
9         2016-01-03 2.59 DKK
10        2016-01-03 -2.72 EUR
11    """
12    expenses = []
13    for line in expenses_string.splitlines():
14        if line.startswith("#"):
15            continue
16        date, value, currency = line.split(" ")
17        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
18                         float(value),
19                         currency))
20
21    return expenses
  
```

Jacob Bennett in Geek Culture

## The 5 paid subscriptions I actually use in 2023 as a software engineer

Tools I use that are cheaper than Netflix

◆ · 6 min read · May 29



627



5



...

◆ · 4 min read · Mar 25



2.8K



30



...

## Load Balancer vs. Reverse Proxy vs. API Gateway



Arslan Ahmad in Geek Culture

### Load Balancer vs. Reverse Proxy vs. API Gateway

Understanding the Key Components for  
Efficient, Secure, and Scalable Web...

◆ · 12 min read · May 17



419



3



...

◆ · 6 min read · May 30



612



3



...



See all from Josep Ferrer

See all from Geek Culture

## Recommended from Medium



Don Giannatti  in Art, Interrupted

### So Long “Prompt Engineering,” We Hardly Knew Ya

Thinking that prompt engineering is the future is to not fully understand what machin...

5 min read · Jun 22



2.2K



41



...



The Useful Tech in Mac O'Clock

### 9 New Must-Have macOS Productivity Apps For Daily Usage

How did you manage all these days without these apps?

★ · 11 min read · Jun 29



2.2K



21



...

## Lists

**ChatGPT**

21 stories · 25 saves

**Predictive Modeling w/  
Python**

18 stories · 76 saves

**ChatGPT prompts**

17 stories · 64 saves

**AI Regulation**

6 stories · 18 saves



Sami Maameri in Better Programming

**Private LLMs on Your Local  
Machine and in the Cloud With...**

The idea of private LLMs resonates with us for sure. The appeal is that we can query and...

17 min read · Jun 15



774



...



Usman Zahid in Artificial Corner

**Don't use MidJourney, if you have  
an NVIDIA card**

Why would someone even do that

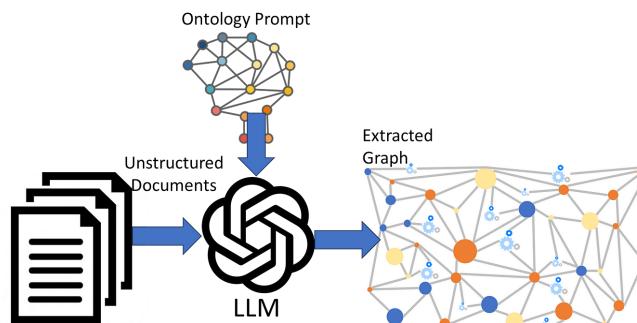
4 min read · May 4



215



...



Peter Lawrence, answering users' data ... in GoPe...

## LLM Ontology-prompting for Knowledge Graph Extraction

Prompting an LLM with an ontology to drive Knowledge Graph extraction from...

5 min read · Jun 27



347



7



[See more recommendations](#)



Julie Perilla Garcia in Level Up Coding

## To Be A Great Software Developer — You Need a System

Here's how to build one.

★ · 7 min read · Jun 23



1.8K



24

