Setting up Auto-GPT

📋 Requirements

Choose an environment to run Auto-GPT in (pick one):

- Docker (*recommended*)
- Python 3.10 or later (instructions: for Windows)
- VSCode + devcontainer

🗝️ Getting an API key

Get your OpenAI API key from: https://platform.openai.com/account/api-keys.

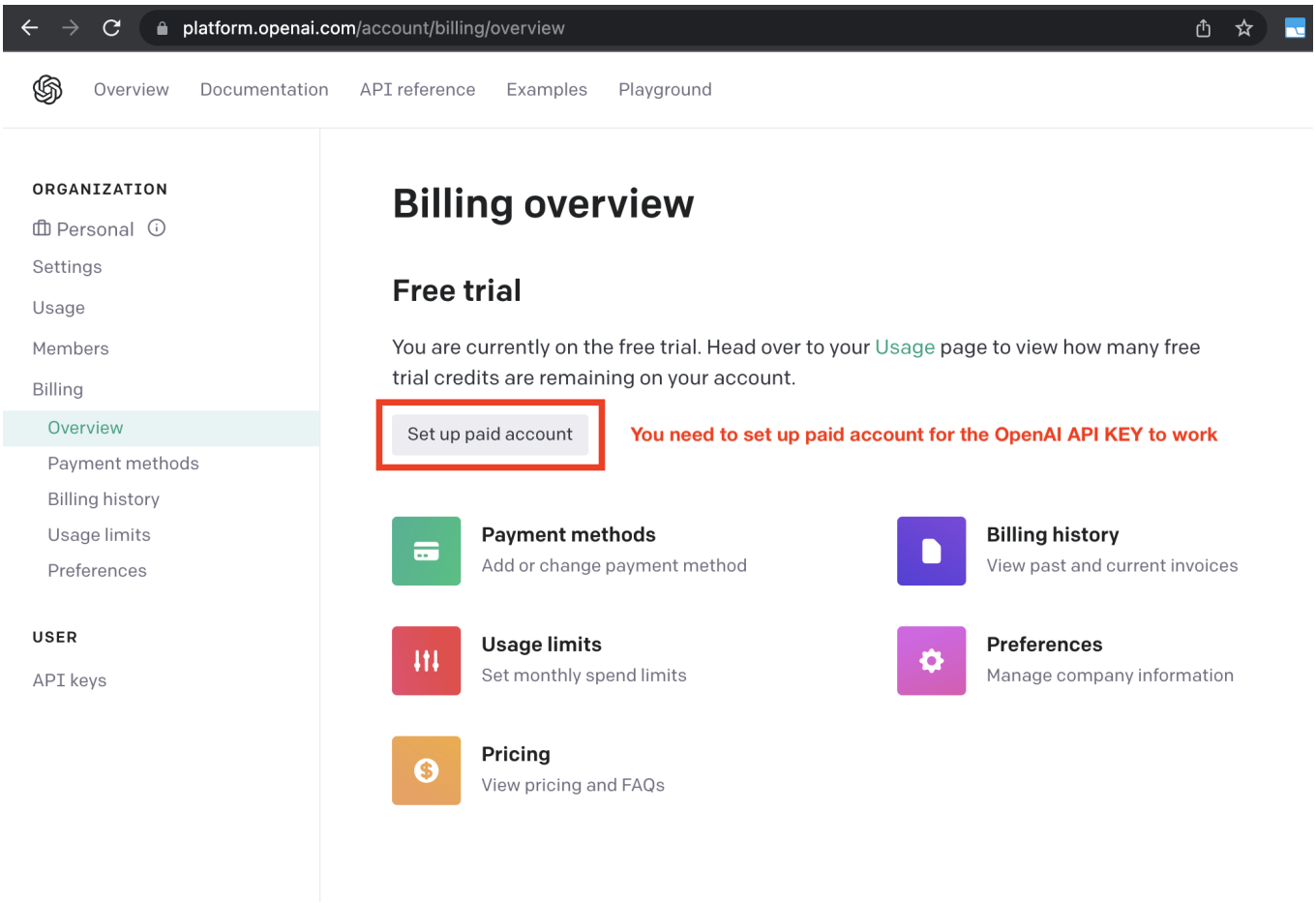
Attention

To use the OpenAI API with Auto-GPT, we strongly recommend **setting up billing** (AKA paid account). Free accounts are limited to 3 API calls per minute, which can cause the application to crash.

You can set up a paid account at Manage account > Billing > Overview.

Important

It's highly recommended that you keep track of your API costs on the Usage page. You can also set limits on how much you spend on the Usage limits page.



Setting up Auto-GPT

Set up with Docker

1. Make sure you have Docker installed, see requirements
2. Create a project directory for Auto-GPT

```
mkdir Auto-GPT
cd Auto-GPT
```

3. In the project directory, create a file called `docker-compose.yml` with the following contents:

```yaml
version: "3.9"
services:
  auto-gpt:
    image: significantgravitas/auto-gpt
    depends_on:
      - redis
    env_file:
      - .env
    environment:
      MEMORY_BACKEND: ${MEMORY_BACKEND:-redis}
      REDIS_HOST: ${REDIS_HOST:-redis}
    profiles: ["exclude-from-up"]
    volumes:
      - ./auto_gpt_workspace:/app/autogpt/auto_gpt_workspace
      - ./data:/app/data
      ## allow auto-gpt to write logs to disk
      - ./logs:/app/logs
      ## uncomment following lines if you want to make use of these files
      ## you must have them existing in the same folder as this docker-compose.yml
      #- type: bind
      #  source: ./azure.yaml
      #  target: /app/azure.yaml
      #- type: bind
      #  source: ./ai_settings.yaml
      #  target: /app/ai_settings.yaml
  redis:
    image: "redis/redis-stack-server:latest"
```

4. Create the necessary configuration files. If needed, you can find templates in the repository.
5. Pull the latest image from Docker Hub

```
docker pull significantgravitas/auto-gpt
```

6. Continue to Run with Docker

Docker only supports headless browsing

Auto-GPT uses a browser in headless mode by default: `HEADLESS_BROWSER=True`. Please do not change this setting in combination with Docker, or Auto-GPT will crash.

Set up with Git

Important

Make sure you have Git installed for your OS.

Executing commands

To execute the given commands, open a CMD, Bash, or Powershell window.
On Windows: press `Win+X` and pick *Terminal*, or `Win+R` and enter `cmd`

1. Clone the repository

```
git clone -b stable https://github.com/Significant-Gravitas/Auto-GPT.git
```

2. Navigate to the directory where you downloaded the repository

```
cd Auto-GPT
```

Set up without Git/Docker

Warning

We recommend to use Git or Docker, to make updating easier. Also note that some features such as Python execution will only work inside docker for security reasons.

1. Download `Source code (zip)` from the latest stable release
2. Extract the zip-file into a folder

Configuration

1. Find the file named `.env.template` in the main `Auto-GPT` folder. This file may be hidden by default in some operating systems due to the dot prefix. To reveal hidden files, follow the instructions for your specific operating system: Windows, macOS.
2. Create a copy of `.env.template` and call it `.env`; if you're already in a command prompt/terminal window: `cp .env.template .env`.
3. Open the `.env` file in a text editor.
4. Find the line that says `OPENAI_API_KEY=`.
5. After the `=`, enter your unique OpenAI API Key *without any quotes or spaces*.
6. Enter any other API keys or tokens for services you would like to use.

> **Note**
>
> To activate and adjust a setting, remove the `#` prefix.

7. Save and close the `.env` file.

> **Using a GPT Azure-instance**

If you want to use GPT on an Azure instance, set `USE_AZURE` to `True` and make an Azure configuration file:

- Rename `azure.yaml.template` to `azure.yaml` and provide the relevant `azure_api_base`, `azure_api_version` and all the deployment IDs for the relevant models in the `azure_model_map` section:
  - `fast_llm_model_deployment_id`: your gpt-3.5-turbo or gpt-4 deployment ID
  - `smart_llm_model_deployment_id`: your gpt-4 deployment ID
  - `embedding_model_deployment_id`: your text-embedding-ada-002 v2 deployment ID

Example:

```
# Please specify all of these values as double-quoted strings
# Replace string in angled brackets (<>) to your own deployment Name
azure_model_map:
    fast_llm_model_deployment_id: "<auto-gpt-deployment>"
        ...
```

Details can be found in the openai-python docs, and in the Azure OpenAI docs for the embedding model. If you're on Windows you may need to install an MSVC library.

Running Auto-GPT

Run with Docker

Easiest is to use `docker-compose`.

Important: Docker Compose version 1.29.0 or later is required to use version 3.9 of the Compose file format. You can check the version of Docker Compose installed on your system by running the following command:

```
docker-compose version
```

This will display the version of Docker Compose that is currently installed on your system.

If you need to upgrade Docker Compose to a newer version, you can follow the installation instructions in the Docker documentation: https://docs.docker.com/compose/install/

Once you have a recent version of docker-compose, run the commands below in your Auto-GPT folder.

1. Build the image. If you have pulled the image from Docker Hub, skip this step (NOTE: You *will* need to do this if you are modifying requirements.txt to add/remove depedencies like Python libs/frameworks)

   ```
   docker-compose build auto-gpt
   ```

2. Run Auto-GPT

   ```
   docker-compose run --rm auto-gpt
   ```

   By default, this will also start and attach a Redis memory backend. If you do not want this, comment or remove the `depends: - redis` and `redis:` sections from `docker-compose.yml`.

   For related settings, see Memory > Redis setup.

You can pass extra arguments, e.g. running with `--gpt3only` and `--continuous`:

```
docker-compose run --rm auto-gpt --gpt3only --continuous
```

If you dare, you can also build and run it with "vanilla" docker commands:

```
docker build -t auto-gpt .
docker run -it --env-file=.env -v $PWD:/app auto-gpt
docker run -it --env-file=.env -v $PWD:/app --rm auto-gpt --gpt3only --continuous
```

Run with Dev Container

1. Install the Remote - Containers extension in VS Code.
2. Open command palette with F1 and type `Dev Containers: Open Folder in Container`.
3. Run `./run.sh`.

Run without Docker

Create a Virtual Environment

Create a virtual environment to run in.

```
python -m venv venvAutoGPT
source venvAutoGPT/bin/activate
pip3 install --upgrade pip
```

> **Warning**
>
> Due to security reasons, certain features (like Python execution) will by default be disabled when running without docker. So, even if you want to run the program outside a docker container, you currently still need docker to actually run scripts.

Simply run the startup script in your terminal. This will install any necessary Python packages and launch Auto-GPT.

- On Linux/MacOS:

  ```
  ./run.sh
  ```

- On Windows:

  ```
  .\run.bat
  ```

If this gives errors, make sure you have a compatible Python version installed. See also the requirements.