

Github: <https://github.com/danielmiessler/fabric>

Tutorial Video by Prakash Joshi Pax: <https://youtu.be/2fj2R6vK6O0?si=8klbUHvXTGXGdjJO>

[Go Installation](#)

[Fabric Installation](#)

[Migration to Go Version](#)

[Setup](#)

[Patterns](#)

[Using Patterns](#)

[Custom Patterns \(Optional\)](#)

[User Interface \(Optional\)](#)

## Go Installation

You must have [Go](#) installed prior to installing Fabric. To check if you already have Go installed, run the following command:

```
go version
```

If you do not have Go installed, go to this website to install: <https://go.dev/doc/install>

## Fabric Installation

If you have installed Fabric previously and need to migrate, go to [Migration to Go Version](#), otherwise, run this command to install Fabric directly from the repo:

```
go install github.com/danielmiessler/fabric@latest
```

Go to [Setup](#) for further instructions.

## Migration to Go Version

If you have the Legacy (Python) version installed and want to migrate to the Go version follow these steps.

Run the following command to uninstall Fabric:

```
pipx uninstall fabric
```

Clear any old Fabric aliases by checking your **.bashrc**, **.zshrc**, etc. On Mac, run the following command:

```
nano ~/.zshrc
```

In this file, delete any references to Fabric and save. Then, run the following command to reload your shell configuration:

```
source ~/.zshrc
```

To install the Go version of Fabric run the following command:

```
go install github.com/danielmiessler/fabric@latest
```

Try to run the following command:

```
fabric -h
```

If you receive an error message, please follow the instructions below. If you do not receive an error message, go to [Setup](#) for further instructions.

If you receive an error message such as “command not found: fabric”, then follow these steps:

- 1) Verify fabric was installed by running the following command:

```
ls $HOME/go/bin
```

- 2) If “fabric” is listed, then add the following line to your shell configuration file:

```
export PATH=$PATH:$HOME/go/bin
```

- 3) Then, run the following to access the shell configuration file:

```
nano ~/.zshrc
```

## Setup

After installation, run the following command to begin setting up Fabric:

```
fabric --setup
```

Running this command will allow you to input API keys for the following:

- OpenAI
- Azure
- Ollama
- Grocq
- Gemini
- Anthropic

You will need to choose your base model when prompted. A good one to pick is GPT-4o from OpenAI, which is index number 39.

Next, you can input your YouTube API key if you want. This will allow you to access and interact with YouTube videos through Fabric.

It will then ask for the default Git repository URL for the patterns you want to use. Leave this blank and press enter.

Then, it will ask you to enter the default folder in the Git repository where patterns are stored. Leave this blank as well and press enter.

## Patterns

To see the list of available patterns run either of the following commands:

```
fabric -l
fabric --listpatterns
```

To access the patterns, run the following command:

```
cd ~/.config/fabric/patterns/
```

## Using Patterns

You can use patterns by running the following command:

```
echo "blah" | fabric -sp [pattern_name]
```

For example:

```
echo "hello everyone I am cool" | fabric -sp tweet
```

Response:

```
Hey everyone! 🙌 I'm feeling cool today 😎✨ #FeelingGood
#HelloWorld
```

If you provided your YouTube API key, you can run a command like the following:

```
yt --transcript
https://www.youtube.com/watch?v=2fj2R6vK6O0 | fabric -sp
extract_wisdom
```

This will summarize the YouTube video for you in markdown format.

You can also specify the model you would like to use by including the following in your command:

```
--model gpt-4
```

With this included, the new command would look like the following:

```
echo "hello everyone I am cool" | fabric --model gpt-4 -sp
tweet
```

You can find a list of available models by running either of the following commands:

```
fabric --listmodels
```

```
fabric -L
```

You can also find a list of available commands with either of the following commands:

```
fabric -h
Fabric --help
```

## Custom Patterns (Optional)

To create your own custom patterns follow the instructions below.

Navigate to the desired directory by running the following command:

```
cd ~/.config/fabric
```

Create a new directory to store your custom patterns in by running the following command:

```
mkdir portx_patterns
```

Navigate into the new directory by running the command:

```
cd portx_patterns
```

Then, create a new directory that will be the name of your custom pattern.

```
mkdir map_prompt
```

Navigate into this new directory.

```
cd map_prompt
```

Create a new file called **system.md** and open it by running the following command:

```
nano system.md
```

Here you can put whatever prompt you would like. For example, this is what I put for the map prompt:

```
Banking API Attribute Mapping Task
Instructions:
You are tasked with mapping an attribute from an external
core banking API to the most suitable fi$
Please adhere to the structured format outlined below for
each attribute mapping request:
Attribute: [Insert the given attribute here]
  1. Primary Choice: [Insert path.in.json.schema1 here]
  2. Secondary Choice: [Insert path.in.json.schema2 here]
  3. Tertiary Choice: [Insert path.in.json.schema3 here]
Guidelines:
- Provide exactly three mapping options for each request,
ordered by your confidence level from hi$
```

- Stick to the specified format to ensure clarity and consistency across all mapping tasks.
  - Do not include any additional comments, notes, or expressions of uncertainty outside the structure.
- By adhering to these instructions, you will contribute to a streamlined and effective attribute mapping process.

Save the file.

Run the following command to update fabric and all its patterns:

```
fabric --update
```

Run this command to copy the contents of your custom pattern directory to the main pattern directory:

```
cp -r ~/.config/fabric/portx_patterns/*  
~/.config/fabric/patterns/
```

Replace `portx_patterns` with the name of your custom pattern directory.

Verify success by running the following command and locating your custom pattern:

```
fabric --l
```

Now, you can use this custom pattern. Repeat any number of times for other custom patterns you desire.

## User Interface (Optional)

A user interface to use Fabric has been developed if you would like to use it. You will receive a zipped folder called **fabric-app.zip**.

Unzip the folder and put it in your desired location.

If you do not have Fabric installed, please go to [Installation](#).

You will also need to add the custom `map_prompt` pattern to Fabric by following the directions in the [Custom Patterns](#).

Navigate to the directory where the **app.py** file is located.

Then, run the following command:

```
pyinstaller --onefile --windowed app.py
```

Locate the **app.app** file in the **dist** folder and add it to your desktop by right clicking on the file and selecting "Make Alias". Name it whatever you want and then drag it to your desktop.

Now, you can open and use the application by double clicking this new icon.

You can also add more JSON schemas by adding them to the **schemas** folder. Rerun the following command each time you make a change and it will update the application:

```
pyinstaller --onefile --windowed app.py
```