TECH TIP — BEGINNER

What an API Key Is and How to Store It Safely

From Nolan Law Firm — nemolegal.com/tools

What It Is

An API key is a password for a service. When your code calls OpenAI, Stripe, Google, or any other API, it sends this key with every request to prove it's authorized. The service uses it to identify you, track your usage, and bill your account.

If someone else gets your API key, they can use the service as you — running up charges on your account or accessing your data. This is why how you store it matters.

> **Real cost:** Exposed OpenAI API keys have resulted in bills of thousands of dollars within hours. Crypto miners scan GitHub continuously for leaked keys.

The Wrong Ways to Store It

```
# WRONG — hardcoded in your script

api_key = "sk-abc123yourrealkey"


# WRONG — in a config file you commit to Git

# config.py with your key in it, pushed to GitHub


# WRONG — in a .env file you accidentally commit

# (more on this in the .env guide)
```

The Right Way — Environment Variables

Store the key outside your code. Your code reads it from the environment at runtime. The key never appears in your script.

```
# In your .env file (never commit this file):

OPENAI_API_KEY=sk-abc123yourrealkey
```

```
# In your Python script:

import os

from dotenv import load_dotenv

load_dotenv()

api_key = os.getenv("OPENAI_API_KEY")
```

On a Server — System Environment

For production, set the variable at the system level rather than using a file:

```
# Add to /etc/environment or your service's systemd unit file:

OPENAI_API_KEY="sk-abc123yourrealkey"


# Or export temporarily in your shell session:

export OPENAI_API_KEY=sk-abc123yourrealkey
```

The .gitignore Rule

Add `.env` to your `.gitignore` file so Git never tracks it. One line prevents the most common key leak.

```
# .gitignore

.env

*.env

secrets.py

config_local.py
```

If You Accidentally Expose a Key

1. Revoke it immediately from the service's dashboard — don't wait. 2. Generate a new key. 3. Rotate it into your environment. Assume the old key was compromised the moment it was exposed.

> **Using AI to Help With This** You don't have to fully understand this to use it. Here are prompts that work:

```
"I need to use an OpenAI API key in my Python script without hardcoding it. Show me the correct way using dotenv."
```

```
"I accidentally committed my API key to GitHub. What exactly do I need to do right now, in order?"
```

```
"Set up my [language/framework] project to read API keys from environment variables. Here is my current code: [paste]."
```
