

What Never Goes Into AI (Or Your Code)

The list of what stays out of a chat box, the 10 second redaction trick that lets you use AI on real work anyway, and what to do if a key already leaked.

There are 2 ways people get burned with AI, and they're the same mistake wearing different clothes.

1. They paste something into a chat that should never have left their computer.

2. They build something, and a password or key ends up inside the code where anybody can read it.

Both happen because it *feels* private. A chat feels like a conversation. Your own project feels like your own folder. Neither is as private as it feels.

Here's the list, the fix, and what to do if it's already happened.

The never-paste list

Don't paste these into any AI chat, any of them, ever:

- **Customer or client names** tied to anything about them
- **Card numbers, bank details, account numbers**
- **Anything medical** about a real person
- **Anything under an NDA** or a confidentiality clause
- **Employee records:** pay, reviews, disciplinary stuff
- **Passwords and API keys**, including the ones hiding inside a config file you pasted in to get help with 1 line

Read that list again. None of it is about Claude or ChatGPT being dodgy. It's the same list you'd get for any cloud tool, any email, any shared drive. A chat makes people drop their guard.

The wrong lesson

Here's where people go wrong with this advice. They hear "don't paste sensitive stuff," so they stop using AI for real work. They use it for captions and birthday messages and never get any actual leverage out of it.

You don't avoid the task. You redact and keep going.

The 10 second redaction trick

Swap the identifying parts for placeholders before you paste:

- **Sarah Mitchell** becomes **Client A**
- **14 Oak Lane, Delaware OH** becomes **a homeowner in zip 43015**
- **Invoice #20481** becomes **the invoice**
- **Dr. Patel at Riverside** becomes **the doctor**
- **\$12,480.00** becomes **about \$12k**

Then ask exactly what you were going to ask:

Here's a redacted version. Client A, a homeowner in zip 43015, is upset the job ran 2 days over. Write me a reply that apologizes without admitting fault, and offers a 5% credit.

Identical quality answer. Zero exposure. It costs you about 10 extra seconds of typing.

Claude doesn't need to know *who* they are to write the reply.

The test I use, and it's the only one you need: if you'd be uncomfortable with that paste showing up in a screenshot 6 months from now, redact it first.

And while you're at it: open your settings and check whether your chats are used to train the model. That's a real control. Know where yours is set.

Now the expensive one: keys in your code

The moment you start building things (tools, automations, anything that talks to another service), you'll get **API keys**. A key is a password that lets your project use a paid service on your account.

Paste that key straight into your code, push the code to GitHub, and if that repo is public or ever becomes public, **bots scan GitHub for leaked keys around the clock**. People lose real money within hours. Somebody else runs up your bill.

The fix is a file called `.env`

A `.env` file holds your keys. It sits next to your code on your computer, and it **never travels with it**.

Your code doesn't say "here's my key." It says "go get the key from the environment," and the key only exists on the machine that needs it.

It looks like this, plain text:

```
OPENAI_API_KEY=sk-your-key-goes-here
STRIPE_SECRET_KEY=sk_live_your-key-here
```

Make sure it never gets uploaded

That's what `.gitignore` is for: the list of files GitHub must never take.

When you create a repo, pick a **.gitignore template** from the dropdown. GitHub maintains these themselves, 1 per language, and the Node one already lists `.env`, `.env.local`, `.env.production` and the rest.

That line exists because this mistake is so common they wrote it down for you. Pick it when you create the repo and you never have to remember later.

If you're building with Claude, say it plainly:

```
Keep every API key in a .env file, make sure .env is in .gitignore, and never write a key
into the code itself.
```

If it already happened

People will have done this. Here's the real answer:

1. **Rotate the key.** Go to the service that issued it (OpenAI, Stripe, whoever), revoke the old key, make a new one. Do this first.
2. **Put the new key in your .env**, not the code.
3. **Don't trust deleting the commit.** Anything that was ever pushed, assume it's public forever. Deleting it from GitHub doesn't un-copy it from whoever already grabbed it.
4. **Check the bill** on that service for anything you didn't run.

Rotating takes 2 minutes. Not rotating can cost you hundreds.

The whole thing on 1 card

- Chat box: **no names, no numbers, no medical, no NDA, no passwords.** Redact and keep going.
- Code: **keys live in .env, .env lives in .gitignore.**
- Leaked: **rotate first**, then clean up.

Screenshot that. It's the cheapest insurance you'll ever buy.

There's a lot more where this came from

Everything I give away is free and lives at **jordybaze.com/free**. Start with whichever of these fits what you're stuck on:

- **Find everything the internet still has on you:** jordybaze.com/free/footprint
- **The scam that uses your kid's voice:** jordybaze.com/free/scams
- **Build your own tool, and never lose it:** jordybaze.com/free/build

Free from jordybaze.com/free/keys · updated 2026-10-27.