STORYFORGE

Connecting Your AI

A Gentle Guide to Getting Started

No technical knowledge required.

Version 1.0

Screen Edition

> *Think of it this way: if a cloud AI assistant is like going to a restaurant, Ollama is like having a private chef in your own kitchen. The ingredients never leave your house.*

Before We Begin

If you have ever installed an app on your phone, you can do this. What follows might look technical at first glance — there are commands to type and programs to download — but every step is straightforward, and none of them require you to understand what is happening behind the scenes. You just need to follow the steps in order.

This guide covers four ways to connect Storyforge to an AI brain. You only need one of them. Most people should start with Ollama, which is the first option. If you get stuck, skip ahead to the troubleshooting section at the end — most problems have simple fixes.

> *Take your time. There is no test at the end. If something does not work, nothing breaks — Storyforge itself always works, with or without the AI. The AI just makes it smarter.*

What the AI Actually Does

Storyforge uses AI for two things. Understanding those two things will help the rest of this guide make sense.

Thinking — When you ask Storyforge to analyse your story, a language model reads your question, looks at your source material, and writes an analysis. This is the same kind of AI you might have used in ChatGPT or Claude, except it is running on your own computer (or connecting to a server you choose).

Understanding — When you upload documents and click "Index All Sources," a smaller AI model reads each chunk of text and converts it into a mathematical fingerprint. Later, when you ask a question, Storyforge compares your question's fingerprint against all the document fingerprints to find the most relevant passages. This is how the AI knows which parts of your writing to look at.

The thinking model is called `llama3.2`. The understanding model is called `nomic-embed-text`. You do not need to remember these names — Storyforge knows them already.

> **GOOD TO KNOW**
> You can use Storyforge without any AI at all. Source management, lenses, notes, and export all work offline. The AI just adds the analysis and search capabilities.

Contents

> *If you just want to get started as quickly as possible, go directly to Chapter 1 (Ollama) and follow the steps. You can read everything else later.*

1. Ollama — Your Private AI

Ollama is a free program that runs AI models directly on your computer. Once installed, everything happens locally — your writing never leaves your machine, no internet connection is needed after setup, and there is no ongoing cost.

This is the recommended option for most writers. It works on Mac, Windows, and Linux.

What You Will Need

	Minimum	Recommended
Computer	Any modern laptop or desktop	Mac with Apple Silicon (M1–M4)
Memory (RAM)	8 GB	16 GB or more
Free disk space	5 GB	10 GB
Internet	For initial download only	Not needed after setup

> *If your computer is less than five years old, it almost certainly meets these requirements. Do not worry about the specifics — if it runs a web browser smoothly, it can run Ollama.*

Installation

Mac

Step 1: Download Ollama

Go to **ollama.com** in your web browser. The website will detect that you are on a Mac and show you a download button. Click it.

A file called something like "Ollama-darwin.zip" will download to your Downloads folder.

Step 2: Install it

Open the downloaded file. Drag the Ollama icon into your Applications folder, just like any other Mac app.

If your Mac asks whether you want to open an app downloaded from the internet, click "Open." This is normal — macOS asks this for all new apps.

Step 3: Launch Ollama

Open Ollama from your Applications folder, or search for it with Spotlight (press **Cmd + Space** and type "Ollama").

A small llama icon will appear in your menu bar at the top of the screen. That means Ollama is running. It sits quietly in the background and waits for Storyforge to talk to it.

WHAT YOU SHOULD SEE

A llama icon in your menu bar. That is all you need to see.

Step 4: Download the AI models

Open the **Terminal** app. You can find it in Applications > Utilities, or search for "Terminal" with Spotlight.

Terminal is a text-based window where you can type commands. It looks intimidating, but you only need to type two things and then you can close it forever.

> *The Terminal is not dangerous. You cannot break your computer by typing these commands. They just tell Ollama to download two small files.*

Type this command and press Enter:

```
ollama pull llama3.2
```

This downloads the thinking model. It is about 2 GB — roughly the size of a movie. On a typical internet connection, it takes 2–5 minutes. You will see a progress bar.

When that finishes, type this and press Enter:

```
ollama pull nomic-embed-text
```

This downloads the understanding model. It is much smaller — about 275 MB. It should take under a minute.

WHAT YOU SHOULD SEE

Both commands finish without errors, and you see the word "success" after each one.

Step 5: Open Storyforge

Open **storyforge.html** in your web browser. Look at the top-right corner. You should see a small green dot and the word "connected" or a version number like "ollama v0.5.4".

If you see a red dot and the word "offline," do not panic. Check that Ollama is still running (llama icon in the menu bar). If it is not, open it again from Applications.

WHAT YOU SHOULD SEE

Green dot in the top-right corner of Storyforge. You are ready to write.

Windows

Step 1: Download Ollama

Go to **ollama.com** in your browser. Click the download button for Windows.

A file called "OllamaSetup.exe" will download.

Step 2: Install it

Run OllamaSetup.exe and follow the prompts. Click "Next" a few times, then "Install."

When it finishes, Ollama will appear in your system tray (the small icons in the bottom-right corner of your screen, near the clock).

Step 3: Download the AI models

Open **Command Prompt**. The quickest way: press **Windows key + R**, type **cmd**, and press Enter.

A black window with white text will appear. This is where you type the commands.

Type this and press Enter:

```
ollama pull llama3.2
```

Wait for the download to finish. You will see a progress bar.

Then type this and press Enter:

```
ollama pull nomic-embed-text
```

Wait for this one too. It is faster.

Step 4: Open Storyforge

Open storyforge.html in your browser. Check for the green dot in the top-right corner.

WHAT YOU SHOULD SEE

Green dot. Connected. Done.

Linux

Step 1: Install Ollama

Open a terminal and run:

```
curl -fsSL https://ollama.com/install.sh | sh
```

This downloads and installs Ollama in one step.

Step 2: Start Ollama

Run:

```
ollama serve
```

Leave this terminal window open. Ollama runs in the foreground here.

Step 3: Download the models

Open a second terminal window and run:

```
ollama pull llama3.2

ollama pull nomic-embed-text
```

Step 4: Open Storyforge

Open storyforge.html in your browser. Green dot means connected.

A Note on Speed

Local AI is slower than cloud AI. This is normal and expected. When you ask Storyforge a question, the answer will take between 5 and 30 seconds to start appearing, depending on your hardware.

The response appears word by word, streaming in real time, so you can start reading before it finishes. This is exactly how it is supposed to work.

If responses feel very slow (more than a minute), try closing other memory-intensive applications. Browsers with many tabs, video editing software, and games all compete for the same resources Ollama needs.

Your Computer	Expected Speed	Experience
Older laptop (8 GB RAM)	15–30 seconds	Usable but slow on long analyses
Modern laptop (16 GB)	5–15 seconds	Smooth for most work
Mac with Apple Silicon	3–10 seconds	Excellent — Apple chips are very good at this
Desktop with GPU	3–8 seconds	Fast, especially with NVIDIA graphics

> **FOR THE CURIOUS**
>
> You can try larger, more capable models if you have the hardware. Type `ollama pull llama3.1:8b` or `ollama pull gemma2:9b` for better quality at the cost of more RAM and slower responses. Change the model name in Storyforge's Settings tab.

2. Anthropic — Claude in the Cloud

Anthropic makes Claude, one of the most capable AI assistants available. Instead of running AI on your own computer, you send your questions to Anthropic's servers over the internet, and they send back the answers.

This is faster and often produces higher-quality analysis than local models. The trade-off is that your prompts leave your computer, and there is a small cost per use.

When to Choose Anthropic

Choose Anthropic if:

- Your computer is older or has limited RAM
- You want the highest-quality analysis
- You do not mind your prompts being sent to the cloud
- Speed matters to you

Choose Ollama instead if:

- Privacy is paramount — sensitive material, personal stories, autobiographical work
- You want to work offline (on a plane, in a cabin, at 3am when the internet is down)
- You do not want any ongoing cost

What Gets Sent to the Cloud

When you use the Anthropic backend, Storyforge sends two things:

Your question — whatever you type into the Synthesis or Deep Dive input.

Relevant source chunks — the 6 most relevant passages from your indexed documents, as found by the RAG pipeline.

Storyforge does *not* send your full documents. It only sends the small pieces that are relevant to your specific question. If you have a 300-page novel uploaded, Anthropic

might see six paragraphs from it, not the whole book.

PRIVACY

Anthropic’s data policy states that API inputs are not used to train their models. But if your material is extremely sensitive, use Ollama.

Getting an API Key

An API key is like a password that lets Storyforge talk to Anthropic on your behalf. Getting one takes about five minutes and requires a credit card, but you will not be charged anything until you actually use the service.

Step 1: Create an Anthropic account

Go to **console.anthropic.com** in your browser. Click "Sign Up." Enter your email, create a password, and verify your email address.

Step 2: Add payment information

Anthropic requires a credit card on file to issue API keys. You will only be charged for what you use. A typical Storyforge session costs less than a few cents.

Go to "Billing" in the console sidebar and add your payment method.

Step 3: Create an API key

Go to "API Keys" in the console sidebar. Click "Create Key." Give it a name like "Storyforge" (the name is just for your reference). Click "Create."

You will see a long string of letters and numbers starting with "sk-ant-". This is your API key. **Copy it now** — you will not be able to see it again after you close this dialog.

IF SOMETHING GOES WRONG

If you lose the API key, you can always create a new one. You do not need to recover the old one. Just delete the old key and make a fresh one.

Step 4: Enter the key in Storyforge

Open Storyforge. Click the **Settings** tab. Under "LLM Backend," click **Anthropic**. Paste your API key into the "API Key" field.

The status indicator in the top-right corner should change to a green dot with the word "anthropic."

WHAT YOU SHOULD SEE

Green dot, "anthropic." You are connected.

Cost

Anthropic charges per token (roughly per word). A typical synthesis with 5 lenses and 6 source chunks costs about $0.005–$0.02. That is half a cent to two cents.

In practical terms: you could run 100 syntheses for about a dollar. Most writers will spend less per month on Storyforge API calls than they spend on coffee.

Action	Approximate Cost
One synthesis (5 lenses)	$0.005 – $0.02
One deep dive	$0.003 – $0.01
100 analyses in a month	$0.50 – $2.00
Heavy daily use for a month	$5 – $15

3. WebAI & StudioLM

These are alternative AI servers that work the same way as Ollama but are made by different teams. If you already have one of these running — because you use them for other projects, or because someone recommended them — Storyforge can connect to them directly.

If you have never heard of these, you can safely skip this chapter. Ollama or Anthropic will serve you well.

How to Connect

Both WebAI and StudioLM use the same communication protocol as Ollama. Storyforge just needs to know the server address.

Step 1: Open Storyforge Settings

Click the **Settings** tab.

Step 2: Select your backend

Under "LLM Backend," click **WebAI** or **StudioLM**.

Step 3: Enter the server URL

In the "Server URL" field, enter the address where your server is running. The default for Ollama is `http://localhost:11434`. Your server may use a different port number.

> If you do not know the URL, check the documentation for your specific server, or ask whoever set it up for you.

Step 4: Check the connection

Look at the status indicator in the top-right corner. A green dot means connected.

4. Which One Should I Choose?

Here is the honest answer, in plain language.

	`Ollama`	`Anthropic`
Cost	Free forever	A few cents per analysis
Speed	Moderate (5–30 sec)	Fast (2–8 sec)
Quality	Good for brainstorming	Excellent — more nuanced, longer
Privacy	Total. Nothing leaves your computer.	Prompts + source chunks sent to cloud.
Internet needed?	Only for initial download	Yes, always
Works on old computers?	Slowly	Yes — the heavy work happens on their servers
Setup difficulty	Download app + 2 terminal commands	Create account + get API key

If you are new to all of this — start with Ollama. It is the simplest to understand, it costs nothing, and your writing stays private. You can always switch to Anthropic later.

If speed and quality matter most — and you are comfortable with your prompts going to the cloud — use Anthropic. The analysis is noticeably better, especially for complex, multi-lens synthesis.

If you work with sensitive material — diary-based fiction, autobiographical writing, stories about real people, themes you are not ready to share with anyone — use Ollama. The privacy is absolute. No one sees your work but you.

> *You are not making a permanent decision. Storyforge lets you switch backends at any time. Many writers use Ollama for private brainstorming and switch to Anthropic when they want a sharper analysis of material they are less protective of.*

5. Checking Your Connection

Storyforge tells you whether the AI is connected through a small indicator in the top-right corner of the screen. Here is what each state means:

Indicator	What It Means	What to Do
Green dot + "ollama v..."	Connected to Ollama. Everything works.	Nothing — you are good.
Green dot + "anthropic"	Connected to Anthropic. Everything works.	Nothing — you are good.
Green dot + "connected"	Connected to WebAI or StudioLM.	Nothing — you are good.
Yellow dot + "checking"	Storyforge is trying to connect.	Wait a moment. It usually resolves.
Red dot + "offline"	Cannot reach the backend server.	See troubleshooting below.
Red dot + "no api key"	Anthropic is selected but no key entered.	Go to Settings and paste your API key.

Even when the indicator shows "offline," Storyforge itself works fine. You can upload sources, configure lenses, write notes, and manage your state. You just cannot run syntheses or index sources until the AI connects.

> **REFRESH**
> The status checks automatically when Storyforge loads. If you start Ollama after opening Storyforge, just refresh the page (Ctrl+R or Cmd+R) and it will re-check.

6. Changing Backends Later

Switching backends takes about ten seconds.

Step 1: Open Settings

Click the **Settings** tab in Storyforge.

Step 2: Click the backend you want

Under "LLM Backend," click the option you want to switch to: Ollama, Anthropic, WebAI, or StudioLM.

Step 3: Done

The status indicator updates immediately. Your sources, lenses, notes, and history are all preserved. Nothing is lost when you switch.

A few things to know about switching:

- Your **source embeddings** (the searchable fingerprints) are created by the embedding model, not the backend. If you change the embedding model itself, you will need to re-index your sources. If you just change the thinking backend, your embeddings are fine.
- The **model name** may change. Ollama uses names like "llama3.2"; Anthropic uses names like "claude-sonnet-4-20250514." Storyforge sets a sensible default when you switch, but you can change it in the model field.
- Your **analysis history** records which model produced each result, so you can compare outputs from different backends over time.

7. Troubleshooting

Most problems have simple causes. Work through the relevant section below.

"Offline" — Cannot Connect to Ollama

Is Ollama running?

Mac: Look for the llama icon in your menu bar (top of the screen). If it is not there, open Ollama from your Applications folder.

Windows: Look for the Ollama icon in your system tray (bottom-right corner, near the clock). If it is not there, search for "Ollama" in the Start menu and open it.

Linux: Check if the terminal window where you ran `ollama serve` is still open. If you closed it, open a new terminal and run the command again.

Did you open the file directly?

If your browser's address bar shows something starting with `file://`, some browsers (particularly Chrome) may block the connection to Ollama for security reasons.

The fix: serve the file through a local web server. Open a terminal in the folder containing storyforge.html and run:

```
python3 -m http.server 8090
```

Then open `http://localhost:8090/storyforge.html` in your browser.

> *This is not a bug. Browsers are cautious about web pages talking to other programs on your computer, and they are more cautious about files opened from your hard drive than about files served through a web server. The web server makes the browser trust the connection.*

Is Ollama actually listening?

You can test this directly. Open your browser and go to:

```
http://localhost:11434
```

If Ollama is running and listening, you will see a message that says "Ollama is running."

If you see an error page, Ollama is not running or is listening on a different port.

"Model Not Found" Error

This means Ollama is running, but the AI model has not been downloaded yet. Open a terminal and run:

```
ollama pull llama3.2
```

Wait for it to finish, then try your synthesis again.

If you get this error about the embedding model when trying to index sources, run:

```
ollama pull nomic-embed-text
```

Indexing Fails

If clicking "Index All Sources" produces an error, the most likely cause is that the embedding model (`nomic-embed-text`) is not installed. Pull it:

```
ollama pull nomic-embed-text
```

If you have changed the embedding model name in Settings, make sure the name matches exactly what Ollama reports. Capitalisation and hyphens matter.

Responses Are Very Slow

Local AI is always slower than cloud AI. This is normal. But if responses take more than a minute, try:

- Close other memory-intensive applications (video editors, games, browsers with dozens of tabs)
- Stick to the default model (`llama3.2`) — larger models need more resources
- If you have an older computer, consider switching to the Anthropic backend for heavy analysis sessions

Anthropic Returns an Error

Common causes:

- **No API key** — Go to Settings and paste your key in the API Key field.

- **Invalid API key** — Go to console.anthropic.com, delete the old key, create a new one, and paste it in.
- **No payment method** — Add a credit card in Anthropic's billing settings.
- **Rate limit** — You have sent too many requests too quickly. Wait a minute and try again.
- **No internet** — Anthropic requires an internet connection. Check your Wi-Fi.

Import Fails

If importing a state file produces an error, the file may be malformed or may have been created by a different version of Storyforge. Check that:

- The file has a `.json` extension
- The file was exported from Storyforge (not from a different tool)
- The file is not empty or corrupted

If none of that helps, open the file in a text editor — it should start with a curly brace `{` and end with one. If it starts with something else, the file is probably not valid JSON.

The Terminal / Command Prompt Scares Me

That is completely understandable. The terminal is a text-based way to talk to your computer. You type a command, press Enter, and the computer does something. That is all it is.

You only need the terminal for two things: installing Ollama's models (two commands) and optionally starting a web server (one command). After that, you can close it and never open it again.

Nothing you type in the terminal during this setup can harm your computer. The commands are read-only (pulling models) or run a simple web server (serving files). They do not modify system settings, delete files, or change anything important.

> *The worst that can happen is an error message. Error messages are the terminal's way of saying "I did not understand that." They are not damage reports. They are just*

confusion.

8. Glossary

Every technical term in this guide, explained in the plainest English I can manage.

AI Model
A program trained on text to understand and generate language. Think of it as a very well-read assistant that can discuss anything but sometimes makes things up. Storyforge uses AI models to analyse your writing.

API
Application Programming Interface. A way for two programs to talk to each other. When Storyforge sends a question to Ollama or Anthropic, it uses an API. You do not need to understand how it works — Storyforge handles it.

API Key
A password-like code that proves Storyforge has permission to use Anthropic's service. Like a library card: it identifies you so the library knows to let you in and what to charge you.

Backend
The AI service that Storyforge connects to. Ollama is a local backend (runs on your computer). Anthropic is a cloud backend (runs on their servers).

Cloud
Someone else's computer. When people say "the cloud," they mean servers in a data centre somewhere. Using a cloud service means your data travels over the internet to that data centre.

Command Prompt / Terminal
A text-based window where you type commands to your computer. On Mac it is called Terminal. On Windows it is called Command Prompt or PowerShell. It looks old-fashioned but it is just a different way to talk to your machine.

Embedding
A mathematical fingerprint of a piece of text. Two pieces of text about similar topics will have similar fingerprints. Storyforge uses embeddings to find the parts of your writing that are relevant to your question.

GPU
Graphics Processing Unit. The chip in your computer that draws images on screen. GPUs are also very good at running AI models, which is why computers with powerful GPUs run Ollama faster.

IndexedDB
A storage area inside your web browser where Storyforge keeps the mathematical fingerprints of your documents. It persists across browser sessions — you do not lose your embeddings when you close the tab.

Inference
The process of an AI model generating a response. When you ask Storyforge a question and it starts typing an answer, that is inference happening.

llama3.2
The default thinking model. Made by Meta and freely available. It is about 2 GB in size and runs well on most modern computers.

Local
On your own computer. "Local AI" means the AI runs on your machine, not on someone else's server. Nothing goes to the internet.

localhost
Your own computer, referred to as a network address. When you see `http://localhost:11434`, it means "connect to port 11434 on this computer." Nothing is going to the internet.

nomic-embed-text
The default understanding model. It creates the mathematical fingerprints (embeddings) that make your sources searchable. About 275 MB in size.

Ollama
A free, open-source program that runs AI models on your computer. It is the recommended backend for Storyforge.

Port
A numbered channel on your computer where a program listens for connections. Ollama listens on port 11434 by default. Think of it as an apartment number in a building — the building is your computer, and the port number tells visitors which door to knock on.

Pull
Ollama's word for downloading a model. "ollama pull llama3.2" means "download the llama3.2 model to my computer."

RAM
Random Access Memory. The short-term memory your computer uses while it is working. AI models need to be loaded into RAM to function, which is why more RAM means smoother performance.

Server
A program that sits quietly in the background and waits for other programs to ask it questions. Ollama is a server — it runs in the background and waits for Storyforge to send it prompts.

Streaming
When the AI's response appears word by word, in real time, rather than all at once after a long wait. Storyforge uses streaming so you can start reading the analysis before it finishes generating.

Token

A small piece of text, roughly equivalent to a word. AI services charge per token. "The cat sat on the mat" is about 7 tokens.

STORYFORGE

Connecting Your AI

> *The terminal is not dangerous. Error messages are not damage reports. And the worst that can happen is that you learn something.*

Version 1.0

Your stories. Your studio. Your rules.