

Obsidian and the LLM Wiki You Already Have

Reading, navigating, and improving a real wiki — 2026 DCI Summer Workshop,
Day 2 Session 3

Kwok-leong Tang

2026-05-19

Table of contents

Today's session	2
Why Obsidian for an LLM Wiki?	2
Clone the wiki and open it	3
A tour of the vault	3
Governance files	4
The five content folders	4
Wikilinks — how the vault becomes a graph	5
Try it in this vault	5
Backlinks — discovery in reverse	5
The graph view	5
Codex on the wiki — read, improve, query, lint	6
Read — summarise a page	6
Improve — expand a thin page	6
Query — ask questions over the wiki	7
Lint — find broken or orphan links	7
Looking ahead to Session 08	8
Summary — what you should take away	8
References	9

Today's session

In Session 06 you wrote a *single* wiki page with Codex. This afternoon we work at scale: open a **real, published LLM Wiki** with about a thousand pages, read it, navigate it, and use Codex to improve it.

The wiki is [kltng/digital-china-wiki](#) — a public wiki on **digital resources for China studies** built and maintained over the last year — an LLM Wiki before the pattern had a name. It is the worked example for everything you will do later today in Session 08 when you build your own.

Note

Teaching goal: leave this session knowing (a) how Obsidian's UI maps to a folder of plain markdown, (b) what `[[wikilinks]]` give you that flat folders do not, and (c) the four canonical Codex-on-wiki operations — **read, improve, query, lint**.

Session materials

[Download session_07_materials.zip](#) — or grab the individual file: [README.md](#).

Prerequisite

Obsidian should already be installed on your laptop from the pre-workshop setup email. If it is not: download from <https://obsidian.md/>, drag into Applications (macOS) or run the installer (Windows), and launch once. ~3 minutes. Raise your hand if you get stuck.

Why Obsidian for an LLM Wiki?

A wiki, as we have been using the term, is a **folder of plain markdown files**. Any text editor opens it. You could run a wiki out of VS Code or even TextEdit if you wanted.

Obsidian is opinionated software that gives you four things on top of “a folder of markdown”:

Feature	What it does
Wikilink resolution	<code>[[Some page]]</code> becomes a clickable link to <code>Some page.md</code> . Hover for preview.
Backlinks panel	For every page, see <i>which other pages link to this one</i> . Inbound discovery for free.
Graph view	A 2D visualisation of the whole vault as a graph (nodes = pages, edges = wikilinks).

Feature	What it does
Search + tags	Full-text search, tag explorer, frontmatter queries. Fast, local, no index pre-build needed.

Crucially, **Obsidian does not own your data**. Your vault is just a folder. Close Obsidian and the files are still there. Switch to a different editor next year and your wiki survives unchanged. This is the *plain-text portability* pattern in action — exactly the right shape for a long-term research artefact.

Clone the wiki and open it

Open Codex Desktop’s terminal pane (⌘ J on macOS, Ctrl+J on Windows) and clone the wiki into your workshop folder:

```
cd ~/genai-workshop
git clone https://github.com/kltng/digital-china-wiki
```

The clone is large (~thousands of small files); give it a moment.

Now switch to Obsidian:

1. Launch Obsidian (Applications / Start menu).
2. If you have never opened a vault, click **Open folder as vault**. If you already have a vault open, **File → Open vault → Open folder as vault**.
3. Navigate to ~/genai-workshop/digital-china-wiki/ and click **Open**.

Obsidian will index the folder (~30 seconds for 986 pages). The left sidebar fills with the file tree.

Tip

On first open, Obsidian may ask whether to “trust” this vault (because it has its own config files). Click **Trust author and enable plugins**. The wiki has no malicious plugins; trusting is safe and lets the templates and snippets load.

A tour of the vault

What you see in the left sidebar is the same folder structure you would see in Finder — just with .md files rendered as readable pages when you click them. Open each of these in turn:

Governance files

File	What it does
index.md	The master catalog — links to every page in the wiki, organised by type. Start here if you do not know where to look.
log.md	The change log — every substantive edit recorded in a paragraph, newest at the top.
AGENTS.md	This is how the wiki remembers how it grew. Rules for AI agents working on this wiki. Tone, citation style, what not to touch, what to log. Read it once.
SCHEMA.md	What goes in the frontmatter of each page type. Conventions over rules.
TAG_TAXONOMY.md	The list of tags that are valid for this wiki, with definitions.

Together these five files are the **constitution** of the wiki. A future contributor (human or agent) reads them first and writes within their rules. This is what makes a wiki *governable* at 986 pages rather than collapsing into chaos.

The five content folders

Folder	What it contains
topics/	Themes and subjects (dynasties, schools of thought, historical events)
regions/	Geographic areas (provinces, cities, drainage basins)
institutions/	Organisations (universities, museums, archives, presses)
glossary/	Term definitions
resources/	Digital tools, databases, catalogues, open-access projects

Open **one page from each folder**. Notice:

- Every page starts with YAML **frontmatter** (--- fenced) carrying **title**, **type**, **status**, **tags**, **dates**.
- The body is plain Markdown.
- **[[wikilinks]]** are scattered through the prose, connecting one page to another.

- A **Related** section near the bottom lists explicit cross-references.
- A **Sources** section attributes the page's content.

Same shape, repeated 986 times. The shape is the wiki's productivity multiplier.

Wikilinks — how the vault becomes a graph

In any markdown editor you can type:

See `[[Song dynasty]]` for context.

In Obsidian, the text in double brackets becomes a **clickable link** to `Song dynasty.md` (anywhere in the vault — Obsidian finds it by filename, not by full path). Hover over it without clicking → a preview pops up. Click it → you jump to that page. Hit `⌘⇧⬅` (Ctrl+Alt+← on Windows) or the back arrow to return.

Try it in this vault

Find a wikilink on a page you opened above (almost every page has several). Hover. Click. Walk a chain of three or four links. You should feel the vault as a *connected* artefact, not a set of files.

Backlinks — discovery in reverse

Obsidian's right-hand sidebar has a **Backlinks** panel. For every page you open, it shows you every OTHER page that links TO this page. **This is the most powerful feature in any wiki tool.** It lets you discover what *else* in your knowledge base references the thing you are reading right now, without you having to remember.

Open `topics/song-dynasty.md` (or any well-connected topic page). Watch the backlinks panel fill with every glossary entry, every institution page, every resource that mentions Song dynasty.

The graph view

Press `⌘ G` (macOS) or the **graph icon** in the left ribbon. A 2D graph of the entire vault appears.

- Each **node** is a page; each **edge** is a `[[wikilink]]`.
- Drag nodes, zoom, pan.
- The dense clusters are the wiki's *centres of gravity* — the topics with many connections.

- The orphaned nodes (no edges) are pages no one has linked to yet — usually a sign of *something missing*.

Spend two minutes just *looking*. The graph is a research instrument as much as a navigation aid.

Question

Find a sparsely-connected page in the graph and click it. Read the page. Why do you think no one has linked to it? Is the page weak (thin content), poorly tagged, or simply on a topic the rest of the wiki hasn't reached yet?

Codex on the wiki — read, improve, query, lint

Obsidian is for *reading and editing by hand*. Codex is for *delegating work to the agent*. We want both. Open the **same folder** as a Codex Desktop project:

1. In Codex Desktop: **Projects** sidebar → + → **Use an existing folder** → pick ~/genai-workshop/digital-china-wiki/.
2. The folder is now open in *both* Obsidian and Codex. Edits the agent makes appear in Obsidian automatically — Obsidian watches the files and reloads on its own.

The four canonical operations on a wiki are **read, improve, query, lint**. We will do one prompt for each.

Read — summarise a page

Pick any topic page in the wiki — for example `topics/song-dynasty.md`. In Codex, ask:

```
Read topics/song-dynasty.md and give me a three-bullet summary I could use to
↪ introduce this topic to a colleague.
```

Codex reads the file (you should see the read in the chat), produces a tight summary. Useful for orientation, useful before you decide to deepen a page.

Improve — expand a thin page

Find a glossary entry that is short — open `glossary/` in Obsidian and click around for a page that is under 20 lines. Ask Codex:

```
Read glossary/[your-chosen-page].md. The page is a stub. Following the
↪ structure used in other glossary entries (open two or three to see the
↪ shape), expand it to a draft of 50–100 lines: a clear definition,
↪ why-it-matters paragraph, related pages, sources. Don't invent facts —
↪ flag anything you would need a source for, but write what you can
↪ confidently from general knowledge. Update the frontmatter status from
↪ "stub" to "draft".
```

Approve. Open the page in Obsidian — if a pane looks stale, close and reopen the note — and read what Codex wrote. **Check for accuracy.** This is the central skill of agentic wiki work: not writing pages yourself, but *recognising whether the agent's page is what you wanted*.

Query — ask questions over the wiki

The wiki is itself a knowledge corpus. You can ask questions of it.

```
Across topics/, which pages discuss the relationship between religion and the
↪ state? List them with the one-line description from each page's
↪ frontmatter.
```

Or:

```
The user is researching the Ming dynasty's relationship with maritime trade.
↪ Walk the wiki and list every page (across topics/, regions/,
↪ institutions/, resources/) that bears on this question. Sort the list by
↪ how directly the page addresses the question.
```

Codex reads many pages, synthesises an answer grounded in the wiki's actual contents. This is the **LLM Wiki version of RAG** — the retrieval is over a corpus *you* curated, not arbitrary web content. The answers are auditable: every page Codex named, you can open and read.

Lint — find broken or orphan links

```
Scan the whole vault. List every [[wikilink]] that points to a page that does
↪ not exist (a "broken link"). For each, show me the source page and the
↪ missing target. Don't try to fix anything yet — just report.
```

Codex walks the vault, returns a list. **Look at it.** The broken links are often the most useful list in any wiki: they tell you what pages *should exist* but don't yet — a curated to-do list generated by your past self.

Same operation in the opposite direction:

Find pages that have no incoming or outgoing wikilinks (orphans). List them.

Orphans are pages disconnected from the rest of the wiki — usually because they were created and then forgotten. Each one is either a candidate for new cross-references or a candidate for deletion.

💡 The four-verb mental model

Read · Improve · Query · Lint is the entire surface of Codex-on-wiki work. Every more elaborate ask is a composition: “*read pages A and B, query for related work in folder C, improve the resulting summary in page D*” is just a sequenced version of the same four verbs. Memorise the verbs; the prompts follow.

Looking ahead to Session 08

You have spent this session **inside someone else's wiki**. Later today — Session 08 — you build **your own**, from a real research corpus: **29 sources on Tang–Song fiscal history**, centered on the Two-Tax Law (兩稅法):

1. **Read Karpathy's gist** for 15 minutes — the source that named the LLM Wiki pattern.
2. Set up a scaffold for your new wiki — folder structure, governance files, page templates.
3. Drop the **liangsui** corpus (29 PDFs + a few Markdown sources) into the **raw/** layer.
4. Send Codex one prompt — it generates **AGENTS.md**, a build script, source stubs, and Obsidian Bases views.
5. Send Codex one more prompt — it bulk-ingests the entire corpus into source / people / concept / institution / debate pages.
6. Enrich one page with a **Skill** that reaches outside the corpus (CBDB, Wikidata, calendar).

The wiki you build later today is the artefact you take home from this workshop. The remaining two days of teaching all add capability to that artefact.

Summary — what you should take away

1. **A wiki is a folder of markdown.** Any editor reads it. Obsidian adds graph view, wikilinks, backlinks, search — but the data is plain text and yours.

2. **Governance files** (index, log, AGENTS, SCHEMA, TAG_TAXONOMY) are what let a wiki stay coherent at scale. Read them first when you open someone else's wiki.
3. **[[Wikilinks]]** are the structural backbone. The graph view makes the structure visible.
4. **Backlinks** are the single most useful navigation feature. They turn “what did I write about X?” from a problem into a glance.
5. **Codex on the wiki has four canonical operations: read, improve, query, lint.** Everything else is a sequence of these.
6. **The wiki is your durable artefact.** Codex is the agent that writes into it. Obsidian is the reader. Git is the time machine. Each layer is replaceable; the markdown files are what survive.

References

- [kltng/digital-china-wiki](#) — the worked example wiki used today.
- [Obsidian](#) — the editor.
- [Obsidian Help — Wikilinks](#) — the canonical reference for `[[link]]` syntax.
- [Obsidian Help — Graph view](#) — interactive vault visualisation.
- Wiki: `patterns/llm-wiki-pattern.md`, `patterns/three-layer-architecture.md`, `patterns/ingest-query-lint.md` — the architectural patterns this session enacts.