**📚 OFFICIAL Xavigate RAG Ingestion Procedure (v1)**

**📂 1. Save the New Source Doc**

* Drop any **new DOCX**, **JSONL**, or **MD** files into the correct folder:
  + /data/methods/ → New practitioner methods
  + /data/problems/ → New diagnostic problems
  + /data/programs/ → New solution modules
  + /data/alignment\_dynamics/ → New frameworks or theory
  + /data/glossary/ → New glossary files

✅ **Naming Tip:** Keep filenames simple and lowercase (e.g., burnout\_recovery.docx).

**🛠️ 2. Update Local**

**bulk\_chunks\_all\_cleaned.jsonl**

**(if needed)**

* Chunk new .docx or .md files **locally** first with your chunking.py.
* Validate and clean them into a new **bulk\_chunks\_all\_cleaned.jsonl** (or split if you prefer).

**🛠️ 3. Upload the New Files to the Server**

* SCP/SFTP the updated files or bulk\_chunks\_all\_cleaned.jsonl to the EC2 server
* Always keep /data/ organized and clean

**🚀 4. Re-Ingest the New Content**

✅ If it’s small (single glossary or module):

python backend/ingest\_glossary.py

✅ If it’s a full new batch (bulk file):

python backend/ingest\_chunks.py

✅ If you want to rebuild Chroma clean:

* Wipe chroma\_db/
* Re-run full ingestion of bulk + glossary

**🛡 5. Always Confirm Ingestion Output**

* Check the ingestion count
* Watch for JSONDecode errors (bad chunking)
* Check Chroma size if needed (e.g., ls -lh chroma\_db)

**🧠 Full Brain Maintenance**

* Keep ChromaDB backed up once stable
* Keep ingestion scripts (ingest\_chunks.py, ingest\_glossary.py) tight and modular
* Document new knowledge expansions (what/where/why)

**🧹 Final Tip for Future**

Soon we can even:

* Build a “Watch and Auto-Ingest” script to auto-detect new files
* Build a simple dashboard for monitoring ingestion stats

But today?

✅ Clean.

✅ Controlled.

✅ Professional.