Audit du dˆ'pˆ·t NoXoZ_job

PDF companion: ‘repo_audit.pdf‘.

1) Inventaire complet des fichiers (par domaine/fonction)

Backend
- ‘2_Sources/2.1_Python/api/__init__.py‘
- ‘2_Sources/2.1_Python/api/dependencies.py‘
- ‘2_Sources/2.1_Python/api/endpoints/fastapi_full_monitor.py‘
- ‘2_Sources/2.1_Python/api/endpoints/generate.py‘
- ‘2_Sources/2.1_Python/api/endpoints/status.ORI.py‘
- ‘2_Sources/2.1_Python/api/endpoints/status.py‘
- ‘2_Sources/2.1_Python/api/endpoints/status_web.py‘
- ‘2_Sources/2.1_Python/api/endpoints/upload.py‘
- ‘2_Sources/2.1_Python/api/monitor.py‘
- ‘2_Sources/2.1_Python/api/router.py‘
- ‘2_Sources/2.1_Python/chroma_integration.py‘
- ‘2_Sources/2.1_Python/main_agent.py‘
- ‘2_Sources/2.1_Python/services/generation.py‘
- ‘2_Sources/2.1_Python/services/ingestion.py‘
- ‘2_Sources/2.1_Python/services/vector_store.py‘
- ‘2_Sources/2.1_Python/temp.py‘

Data
- ‘10_Runs/ollama.pid‘
- ‘10_Runs/service.noxoz_job.ollama.pid‘
- ‘4_Logs/init.txt‘
- ‘5_Outputs/init.txt‘
- ‘5_Outputs/out.txt‘
- ‘5_Outputs/outputs.md‘
- ‘6_Results/init.txt‘
- ‘7_Infos/PERMANENT_MEMORY.md‘
- ‘7_Infos/tree.txt‘
- ‘9_Templates/init.txt‘
- ‘fastapi.pid‘

Scripts
- ‘2_Sources/2.2_Bash/create_structure.sh‘
- ‘8_Scripts/8.1_Init/config_paths.sh‘
- ‘8_Scripts/8.1_Init/create-repo.sh‘
- ‘8_Scripts/8.1_Init/fix_ollama_group_permissions.sh‘
- ‘8_Scripts/8.1_Init/init_fastapi.sh‘
- ‘8_Scripts/8.1_Init/init_project.sh‘
- ‘8_Scripts/8.1_Init/init_python.sh‘
- ‘8_Scripts/8.1_Init/init_sqlite.sh‘
- ‘8_Scripts/8.1_Init/old_versions/all_my_scripts.old_versions.txt‘
- ‘8_Scripts/8.1_Init/ollama-batch-download.sh‘
- ‘8_Scripts/8.1_Init/reset_ollama_service.sh‘
- ‘8_Scripts/8.1_Init/service.fastapi/install_service.sh‘
- ‘8_Scripts/8.1_Init/service.fastapi/noxoz_job.fastapi.service‘
- ‘8_Scripts/8.1_Init/service.fastapi/start_fastapi.sh‘
- ‘8_Scripts/8.1_Init/service.fastapi/uninstall_service.sh‘
- ‘8_Scripts/8.1_Init/service.ollama/install_service.sh‘
- ‘8_Scripts/8.1_Init/service.ollama/noxoz_job.ollama.service‘
- ‘8_Scripts/8.1_Init/service.ollama/start_ollama.sh‘
- ‘8_Scripts/8.1_Init/service.ollama/uninstall_service.sh‘
- ‘8_Scripts/8.2_Utils/argparse‘
- ‘8_Scripts/8.2_Utils/count_tokens.sh‘

- ‘8_Scripts/8.2_Utils/datetime‘
- ‘8_Scripts/8.2_Utils/json‘
- ‘8_Scripts/8.2_Utils/md_compressor.py‘
- ‘8_Scripts/8.2_Utils/requests‘
- ‘8_Scripts/8.2_Utils/sys‘
- ‘8_Scripts/out.txt‘
- ‘check_all_web_python.sh‘

Docs
- ‘1_Documentation/1.1_General/ARCHITECTURE.md‘
- ‘1_Documentation/1.1_General/INSTALL.docx‘
- ‘1_Documentation/1.1_General/WHY.docx‘
- ‘1_Documentation/1.1_General/ollama-models-guide.md‘
- ‘1_Documentation/1.2_Technical/API_SPECIFICATIONS.docx‘
- ‘1_Documentation/1.2_Technical/COMPOSANTS.csv‘
- ‘1_Documentation/1.2_Technical/OLLAMA_all_models_with_token_limits.md‘
- ‘1_Documentation/1.2_Technical/OLLAMA_commandes.md‘
- ‘1_Documentation/1.2_Technical/pipenv_guide.md‘
- ‘1_Documentation/1.2_Technical/structure.md‘
- ‘1_Documentation/1.2_Technical/table.csv‘
- ‘1_Documentation/1.3_Status/COMPOSANTS_mis_a_jour.csv‘
- ‘1_Documentation/1.3_Status/plan_detaille.md‘
- ‘1_Documentation/out.txt‘
- ‘README.md‘

Config
- ‘Pipfile‘
- ‘Pipfile.lock‘
- ‘requirements.txt‘

Infra
- ‘certs/cert.pem‘
- ‘certs/key.pem‘
- ‘docker-compose.yml‘

Tests
- ‘2_Sources/2.1_Python/test_db_huffing.py‘
- ‘2_Sources/2.1_Python/test_sentence_transfomers.py‘
- ‘8_Scripts/8.1_Init/test_fastapi.sh‘
- ‘8_Scripts/8.1_Init/test_sqlite.sh‘

Other
- ‘2_Sources/out.txt‘
- ‘temp.py‘

2) Fichiers clˆ's par domaine et usage

Backend (Python)
- ‘2_Sources/2.1_Python/main_agent.py‘ : point dâ entrˆ'e FastAPI, CORS, inclut le router API.
- ‘2_Sources/2.1_Python/api/router.py‘ : agrˆ¤ge les routers (generate/status/upload/monitor).
- ‘2_Sources/2.1_Python/api/endpoints/*.py‘ : endpoints REST (generate/status/upload/monitoring).
- ‘2_Sources/2.1_Python/api/monitor.py‘ : monitoring complet (Chroma/SQLite/Ollama/logs).
- ‘2_Sources/2.1_Python/services/‘ : logique dâ ingestion, gˆ'nˆ'ration et stockage vectoriel.
- ‘2_Sources/2.1_Python/chroma_integration.py‘ : exemple dâ intˆ'gration Chroma (script autonome).

Frontend
- Aucun front dˆ'diˆ'. Le HTML est gˆ'nˆ'rˆ' cˆ·tˆ' backend via ‘status_web.py‘ (endpoint

‘/api/monitor/web_status‘).

Data / Outputs / Logs
- ‘3_Data/‘ : stockage local, symlinks vers volumes externes, DB SQLite (symlink).
- ‘4_Logs/‘, ‘5_Outputs/‘, ‘6_Results/‘, ‘7_Infos/‘, ‘10_Runs/‘ : artefacts runtime, logs, rˆ'sultats, mˆ'moires.

Scripts & Automation
- ‘8_Scripts/8.1_Init/‘ : scripts dâ init (FastAPI, SQLite, Ollama, config paths, services systemd).
- ‘8_Scripts/8.2_Utils/‘ : utilitaires (ex. ‘md_compressor.py‘, ‘count_tokens.sh‘, artefacts PS).
- ‘check_all_web_python.sh‘ : note/trace de code, non-exˆ'cutable.

Docs
- ‘README.md‘ + ‘1_Documentation/‘ : documentation projet, architecture, specs, guides.

Config & Infra
- ‘requirements.txt‘, ‘Pipfile‘, ‘Pipfile.lock‘ : dˆ'pendances Python.
- ‘docker-compose.yml‘ : services Ollama & Chroma.
- ‘certs/‘ : certificats TLS locaux.
- ‘8_Scripts/8.1_Init/service.*/*.service‘ : services systemd FastAPI/Ollama.

Tests
- ‘2_Sources/2.1_Python/test_*.py‘ : scripts de test embeddings/Chroma.
- ‘8_Scripts/8.1_Init/test_*.sh‘ : tests dâ API et SQLite.

3) Structure globale du projet

Points dâ entrˆ'e
- **API** : ‘2_Sources/2.1_Python/main_agent.py‘ (FastAPI, CORS, inclut ‘/api/*‘).
- **Services** : ‘2_Sources/2.1_Python/services/‘ (ingestion/gˆ'nˆ'ration/vector store).
- **Monitoring** : ‘2_Sources/2.1_Python/api/monitor.py‘ + ‘status_web.py‘.

Modules principaux
- **API Layer** : ‘api/router.py‘ + ‘api/endpoints/*.py‘.
- **Ingestion** : ‘services/ingestion.py‘ + ‘services/vector_store.py‘.
- **Gˆ'nˆ'ration** : ‘services/generation.py‘ (Ollama + DOCX).
- **Vector store** : ‘services/vector_store.py‘ (Chroma + SQLite).

Flux de donnˆ'es (simplifiˆ')
1. Upload â ‘upload‘ endpoint â sauvegarde fichier â ‘ingest_file()‘ (Chroma + SQLite).
2. Generate â ‘generate‘ endpoint â ‘search_similar()‘ â prompt Ollama â docx output.
3. Monitoring â ‘monitor/full‘ â checks Chroma/SQLite/Ollama/logs.

Dˆ'pendances majeures
- **FastAPI / Starlette** : API REST.
- **ChromaDB** : vector store.
- **LangChain + Sentence-Transformers** : embeddings.
- **Ollama** : exˆ'cution LLM local.
- **SQLite** : mˆ'tadonnˆ'es.

4) API (endpoints)

Mˆ'thode	Chemin	Description	Source
POST	‘/api/generate/‘	Gˆ'nˆ¤re un document via Ollama ˆ partir dâ un prompt et template.	‘2_Sources/2.1_Python/api/endpoints/generate.py‘
GET	‘/api/generate/health‘	Health check de lâ endpoint generate.

‘2_Sources/2.1_Python/api/endpoints/generate.py‘ |
| GET | ‘/api/status/‘ | Statut global + mˆ'triques dâ ingestion SQLite. |
‘2_Sources/2.1_Python/api/endpoints/status.py‘ |
| GET | ‘/api/status/health‘ | Health check de lâ endpoint status. |
‘2_Sources/2.1_Python/api/endpoints/status.py‘ |
| POST | ‘/api/upload/‘ | Upload de fichier + ingestion Chroma/SQLite. |
‘2_Sources/2.1_Python/api/endpoints/upload.py‘ |
| GET | ‘/api/upload/health‘ | Health check de lâ endpoint upload. |
‘2_Sources/2.1_Python/api/endpoints/upload.py‘ |
| GET | ‘/api/monitor/web_status‘ | Page HTML de monitoring (health + fichiers). |
‘2_Sources/2.1_Python/api/endpoints/status_web.py‘ |
| GET | ‘/api/monitor/read_file‘ | Lecture HTML dâ un fichier Python. |
‘2_Sources/2.1_Python/api/endpoints/status_web.py‘ |
| GET | ‘/api/monitor/health‘ | Health check (status_web). |
‘2_Sources/2.1_Python/api/endpoints/status_web.py‘ |
| GET | ‘/api/monitor/full‘ | Monitoring complet (FastAPI/Chroma/SQLite/Ollama/logs). |
‘2_Sources/2.1_Python/api/monitor.py‘ |
| GET | ‘/api/monitor/health‘ | Health check (monitor). | ‘2_Sources/2.1_Python/api/monitor.py‘
|

Remarque : deux endpoints ‘/api/monitor/health‘ sont dˆ'finis (dans ‘status_web.py‘ et ‘monitor.py‘), ce qui peut crˆ'er un conflit de routing.

5) Scripts ‘.sh‘ (chemin, usage, dˆ'tails)

- ‘check_all_web_python.sh‘ : dump texte de code Python/monitoring, pas un vrai script exˆ'cutable.
- ‘2_Sources/2.2_Bash/create_structure.sh‘ : fichier vide (placeholder).
- ‘8_Scripts/8.1_Init/init_project.sh‘ : crˆ'e lâ arborescence, docx, logs, gitignore.
- ‘8_Scripts/8.1_Init/init_fastapi.sh‘ : initialise FastAPI, ˆ'crit fichiers sources, installe deps.
- ‘8_Scripts/8.1_Init/init_python.sh‘ : gˆ'nˆ¤re requirements.txt et installe deps via pip.
- ‘8_Scripts/8.1_Init/init_sqlite.sh‘ : crˆ'e DB SQLite sur volume externe + symlink + tables.
- ‘8_Scripts/8.1_Init/test_fastapi.sh‘ : dˆ'marre FastAPI, teste ‘/status‘, ‘/generate‘, ‘/upload‘.
- ‘8_Scripts/8.1_Init/test_sqlite.sh‘ : tests SQLite (standard + perf + perfbig).
- ‘8_Scripts/8.1_Init/config_paths.sh‘ : symlinks vers volumes externes + .env + deps pipenv.
- ‘8_Scripts/8.1_Init/reset_ollama_service.sh‘ : rˆ'initialise le service systemd Ollama.
- ‘8_Scripts/8.1_Init/fix_ollama_group_permissions.sh‘ : corrections permissions/groupe pour Ollama.
- ‘8_Scripts/8.1_Init/ollama-batch-download.sh‘ : batch download de modˆ¤les Ollama.
- ‘8_Scripts/8.1_Init/service.fastapi/*.sh‘ : install/start/uninstall service systemd FastAPI.
- ‘8_Scripts/8.1_Init/service.ollama/*.sh‘ : install/start/uninstall service systemd Ollama.
- ‘8_Scripts/8.2_Utils/count_tokens.sh‘ : compte tokens via tiktoken dans VENV Ollama.

6) Scripts ‘.py‘ (chemin, usage, imports)

- ‘2_Sources/2.1_Python/main_agent.py‘ : FastAPI app + CORS + router. Imports: ‘fastapi‘, ‘CORSMiddleware‘, ‘api.router‘.
- ‘2_Sources/2.1_Python/api/router.py‘ : enregistre routers. Imports: ‘fastapi.APIRouter‘, endpoints, ‘api.monitor‘.
- ‘2_Sources/2.1_Python/api/dependencies.py‘ : placeholder (pas dâ imports).
- ‘2_Sources/2.1_Python/api/endpoints/generate.py‘ : gˆ'nˆ'ration doc. Imports: ‘fastapi‘, ‘JSONResponse‘, ‘services.generation‘.
- ‘2_Sources/2.1_Python/api/endpoints/status.py‘ : statut ingestion. Imports: ‘fastapi‘, ‘sqlite3‘, ‘datetime‘, ‘services.vector_store‘.
- ‘2_Sources/2.1_Python/api/endpoints/status.ORI.py‘ : duplicat de ‘status.py‘.
- ‘2_Sources/2.1_Python/api/endpoints/upload.py‘ : upload + ingestion. Imports: ‘fastapi‘,

‘JSONResponse‘, ‘services.ingestion‘.
- ‘2_Sources/2.1_Python/api/endpoints/status_web.py‘ : HTML status + lecture fichiers. Imports: ‘fastapi‘, ‘Path‘, ‘httpx‘.
- ‘2_Sources/2.1_Python/api/endpoints/fastapi_full_monitor.py‘ : FastAPI app autonome (monitoring complet). Imports: ‘fastapi‘, ‘JSONResponse‘, ‘chromadb‘, ‘sqlite3‘, ‘subprocess‘, ‘json‘.
- ‘2_Sources/2.1_Python/api/monitor.py‘ : monitoring complet. Imports: ‘fastapi‘, ‘JSONResponse‘, ‘chromadb‘, ‘sqlite3‘, ‘os‘, ‘subprocess‘, ‘Path‘.
- ‘2_Sources/2.1_Python/services/ingestion.py‘ : ingestion upload. Imports: ‘fastapi.UploadFile‘, ‘Path‘, ‘shutil‘, ‘vector_store‘.
- ‘2_Sources/2.1_Python/services/generation.py‘ : gˆ'nˆ'ration docx via Ollama. Imports: ‘Path‘, ‘datetime‘, ‘subprocess‘, ‘vector_store‘, ‘docx‘.
- ‘2_Sources/2.1_Python/services/vector_store.py‘ : Chroma + SQLite + loaders. Imports: ‘sqlite3‘, ‘Path‘, ‘typing‘, ‘chromadb‘, ‘langchain_community‘, ‘pypdf‘, ‘docx‘.
- ‘2_Sources/2.1_Python/chroma_integration.py‘ : script autonome dâ intˆ'gration Chroma. Imports: ‘os‘, ‘Path‘, ‘typing‘, ‘chromadb‘, ‘embedding_functions‘, ‘pypdf‘, ‘docx‘.
- ‘2_Sources/2.1_Python/test_db_huffing.py‘ : tests Chroma/embeddings. Imports: ‘os‘, ‘chromadb‘, ‘Settings‘, ‘SentenceTransformer‘, ‘datetime‘.
- ‘2_Sources/2.1_Python/test_sentence_transfomers.py‘ : test SentenceTransformer. Imports: ‘SentenceTransformer‘.
- ‘2_Sources/2.1_Python/temp.py‘ : test torch minimal. Imports: ‘torch‘.
- ‘temp.py‘ (racine) : sandbox embeddings/torch. Imports: ‘sentence_transformers‘, ‘torch‘, ‘os‘.
- ‘8_Scripts/8.2_Utils/md_compressor.py‘ : compresse Markdown via Ollama. Imports: ‘argparse‘, ‘sys‘, ‘os‘, ‘json‘, ‘requests‘, ‘datetime‘.

7) Duplications / conflits / obsolescence

- ‘status.ORI.py‘ duplique ‘status.py‘ â risque de divergence.
- ‘api/endpoints/fastapi_full_monitor.py‘ propose un app FastAPI autonome alors que ‘api/monitor.py‘ fournit le mˆ“me rˆ·le â redondance.
- Deux routes ‘/api/monitor/health‘ (status_web + monitor) â conflit potentiel au runtime.
- Chemins SQLite incohˆ'rents : ‘services/vector_store.py‘ utilise ‘metadata.db‘, ‘init_sqlite.sh‘ crˆ'e ‘noxoz_metadata.db‘, ‘api/monitor.py‘ pointe un autre chemin.
- Plusieurs ‘temp.py‘ (racine + ‘2_Sources/2.1_Python/temp.py‘) â scripts locaux de test sans intˆ'gration.
- Fichiers vides/placeholder : ‘create-repo.sh‘, ‘create_structure.sh‘.
- Fichiers runtime/artefacts : ‘fastapi.pid‘, ‘10_Runs/*.pid‘, logs â devraient ˆ“tre ignorˆ's via ‘.gitignore‘.
- ‘8_Scripts/8.2_Utils/argparse|datetime|json|requests|sys‘ : artefacts PostScript/ImageMagick dans un dossier utils (probable dˆ'pˆ·t historique ou export dâ images).

8) Tableau synthˆ'tique des fichiers

Chemin	Usage
‘10_Runs/ollama.pid‘	Artefacts runtime (PID).
‘10_Runs/service.noxoz_job.ollama.pid‘	Artefacts runtime (PID).
‘1_Documentation/1.1_General/ARCHITECTURE.md‘	Documentation projet (md/docx/csv).
‘1_Documentation/1.1_General/INSTALL.docx‘	Documentation projet (md/docx/csv).
‘1_Documentation/1.1_General/WHY.docx‘	Documentation projet (md/docx/csv).
‘1_Documentation/1.1_General/ollama-models-guide.md‘	Documentation projet (md/docx/csv).
‘1_Documentation/1.2_Technical/API_SPECIFICATIONS.docx‘	Documentation projet (md/docx/csv).
‘1_Documentation/1.2_Technical/COMPOSANTS.csv‘	Documentation projet (md/docx/csv).
‘1_Documentation/1.2_Technical/OLLAMA_all_models_with_token_limits.md‘	Documentation projet (md/docx/csv).
‘1_Documentation/1.2_Technical/OLLAMA_commandes.md‘	Documentation projet (md/docx/csv).
‘1_Documentation/1.2_Technical/pipenv_guide.md‘	Documentation projet (md/docx/csv).

‘1_Documentation/1.2_Technical/structure.md‘	Documentation projet (md/docx/csv).
‘1_Documentation/1.2_Technical/table.csv‘	Documentation projet (md/docx/csv).
‘1_Documentation/1.3_Status/COMPOSANTS_mis_a_jour.csv‘	Documentation projet (md/docx/csv).
‘1_Documentation/1.3_Status/plan_detaille.md‘	Documentation projet (md/docx/csv).
‘1_Documentation/out.txt‘	Documentation projet (md/docx/csv).
‘2_Sources/2.1_Python/api/__init__.py‘	Composant API FastAPI (router/dˆ'pendances/monitoring).
‘2_Sources/2.1_Python/api/dependencies.py‘	Composant API FastAPI (router/dˆ'pendances/monitoring).
‘2_Sources/2.1_Python/api/endpoints/fastapi_full_monitor.py‘	Endpoint FastAPI (routes API).
‘2_Sources/2.1_Python/api/endpoints/generate.py‘	Endpoint FastAPI (routes API).
‘2_Sources/2.1_Python/api/endpoints/status.ORI.py‘	Endpoint FastAPI (routes API).
‘2_Sources/2.1_Python/api/endpoints/status.py‘	Endpoint FastAPI (routes API).
‘2_Sources/2.1_Python/api/endpoints/status_web.py‘	Endpoint FastAPI (routes API).
‘2_Sources/2.1_Python/api/endpoints/upload.py‘	Endpoint FastAPI (routes API).
‘2_Sources/2.1_Python/api/monitor.py‘	Composant API FastAPI (router/dˆ'pendances/monitoring).
‘2_Sources/2.1_Python/api/router.py‘	Composant API FastAPI (router/dˆ'pendances/monitoring).
‘2_Sources/2.1_Python/chroma_integration.py‘	Module backend Python.
‘2_Sources/2.1_Python/main_agent.py‘	Module backend Python.
‘2_Sources/2.1_Python/services/generation.py‘	Service backend (ingestion/gˆ'nˆ'ration/vector store).
‘2_Sources/2.1_Python/services/ingestion.py‘	Service backend (ingestion/gˆ'nˆ'ration/vector store).
‘2_Sources/2.1_Python/services/vector_store.py‘	Service backend (ingestion/gˆ'nˆ'ration/vector store).
‘2_Sources/2.1_Python/temp.py‘	Module backend Python.
‘2_Sources/2.1_Python/test_db_huffing.py‘	Module backend Python.
‘2_Sources/2.1_Python/test_sentence_transfomers.py‘	Module backend Python.
‘2_Sources/2.2_Bash/create_structure.sh‘	Script vide/placeholder pour structure bash.
‘2_Sources/out.txt‘	Fichier projet.
‘4_Logs/init.txt‘	Logs runtime/projet.
‘5_Outputs/init.txt‘	Sorties gˆ'nˆ'rˆ'es ou placeholders outputs.
‘5_Outputs/out.txt‘	Sorties gˆ'nˆ'rˆ'es ou placeholders outputs.
‘5_Outputs/outputs.md‘	Sorties gˆ'nˆ'rˆ'es ou placeholders outputs.
‘6_Results/init.txt‘	Rˆ'sultats/analyses (placeholders).
‘7_Infos/PERMANENT_MEMORY.md‘	Informations de suivi/mˆ'moire (ex. PERMANENT_MEMORY).
‘7_Infos/tree.txt‘	Informations de suivi/mˆ'moire (ex. PERMANENT_MEMORY).
‘8_Scripts/8.1_Init/config_paths.sh‘	Scripts dâ initialisation/configuration.
‘8_Scripts/8.1_Init/create-repo.sh‘	Script vide/placeholder.
‘8_Scripts/8.1_Init/fix_ollama_group_permissions.sh‘	Scripts dâ initialisation/configuration.
‘8_Scripts/8.1_Init/init_fastapi.sh‘	Scripts dâ initialisation/configuration.
‘8_Scripts/8.1_Init/init_project.sh‘	Scripts dâ initialisation/configuration.
‘8_Scripts/8.1_Init/init_python.sh‘	Scripts dâ initialisation/configuration.
‘8_Scripts/8.1_Init/init_sqlite.sh‘	Scripts dâ initialisation/configuration.
‘8_Scripts/8.1_Init/old_versions/all_my_scripts.old_versions.txt‘	Scripts dâ initialisation/configuration.
‘8_Scripts/8.1_Init/ollama-batch-download.sh‘	Scripts dâ initialisation/configuration.
‘8_Scripts/8.1_Init/reset_ollama_service.sh‘	Scripts dâ initialisation/configuration.
‘8_Scripts/8.1_Init/service.fastapi/install_service.sh‘	Scripts et service systemd pour FastAPI.
‘8_Scripts/8.1_Init/service.fastapi/noxoz_job.fastapi.service‘	Scripts et service systemd pour FastAPI.
‘8_Scripts/8.1_Init/service.fastapi/start_fastapi.sh‘	Scripts et service systemd pour FastAPI.

‘8_Scripts/8.1_Init/service.fastapi/uninstall_service.sh‘	Scripts et service systemd pour FastAPI.
‘8_Scripts/8.1_Init/service.ollama/install_service.sh‘	Scripts et service systemd pour Ollama.
‘8_Scripts/8.1_Init/service.ollama/noxoz_job.ollama.service‘	Scripts et service systemd pour Ollama.
‘8_Scripts/8.1_Init/service.ollama/start_ollama.sh‘	Scripts et service systemd pour Ollama.
‘8_Scripts/8.1_Init/service.ollama/uninstall_service.sh‘	Scripts et service systemd pour Ollama.
‘8_Scripts/8.1_Init/test_fastapi.sh‘	Scripts dâ initialisation/configuration.
‘8_Scripts/8.1_Init/test_sqlite.sh‘	Scripts dâ initialisation/configuration.
‘8_Scripts/8.2_Utils/argparse‘	Artefact utilitaire (fichier binaire/PS/log).
‘8_Scripts/8.2_Utils/count_tokens.sh‘	Artefact utilitaire (fichier binaire/PS/log).
‘8_Scripts/8.2_Utils/datetime‘	Artefact utilitaire (fichier binaire/PS/log).
‘8_Scripts/8.2_Utils/json‘	Artefact utilitaire (fichier binaire/PS/log).
‘8_Scripts/8.2_Utils/md_compressor.py‘	Utilitaire Python (automation).
‘8_Scripts/8.2_Utils/requests‘	Artefact utilitaire (fichier binaire/PS/log).
‘8_Scripts/8.2_Utils/sys‘	Artefact utilitaire (fichier binaire/PS/log).
‘8_Scripts/out.txt‘	Scripts dâ automatisation.
‘9_Templates/init.txt‘	Templates/placeholder.
‘Pipfile‘	Dˆ'pendances Python gˆ'rˆ'es via Pipenv.
‘Pipfile.lock‘	Lockfile Pipenv pour versions figˆ'es.
‘README.md‘	Documentation gˆ'nˆ'rale et vision du projet.
‘certs/cert.pem‘	Certificats TLS pour HTTPS local.
‘certs/key.pem‘	Certificats TLS pour HTTPS local.
‘check_all_web_python.sh‘	Note/text dump avec exemples de code (pas un script exˆ'cutable).
‘docker-compose.yml‘	Dˆ'finition des services Ollama et Chroma via Docker Compose.
‘fastapi.pid‘	PID runtime FastAPI (artefact).
‘requirements.txt‘	Liste exhaustive des dˆ'pendances Python.
‘temp.py‘	Brouillon local de tests embeddings/torch.

9) Tableau des endpoints

Mˆ'thode	Chemin	Description	Source
POST	‘/api/generate/‘	Gˆ'nˆ¤re un document via Ollama ˆ partir dâ un prompt et template.	‘2_Sources/2.1_Python/api/endpoints/generate.py‘
GET	‘/api/generate/health‘	Health check de lâ endpoint generate.	‘2_Sources/2.1_Python/api/endpoints/generate.py‘
GET	‘/api/status/‘	Statut global + mˆ'triques dâ ingestion SQLite.	‘2_Sources/2.1_Python/api/endpoints/status.py‘
GET	‘/api/status/health‘	Health check de lâ endpoint status.	‘2_Sources/2.1_Python/api/endpoints/status.py‘
POST	‘/api/upload/‘	Upload de fichier + ingestion Chroma/SQLite.	‘2_Sources/2.1_Python/api/endpoints/upload.py‘
GET	‘/api/upload/health‘	Health check de lâ endpoint upload.	‘2_Sources/2.1_Python/api/endpoints/upload.py‘
GET	‘/api/monitor/web_status‘	Page HTML de monitoring (health + fichiers).	‘2_Sources/2.1_Python/api/endpoints/status_web.py‘
GET	‘/api/monitor/read_file‘	Lecture HTML dâ un fichier Python.	‘2_Sources/2.1_Python/api/endpoints/status_web.py‘
GET	‘/api/monitor/health‘	Health check (status_web).	‘2_Sources/2.1_Python/api/endpoints/status_web.py‘
GET	‘/api/monitor/full‘	Monitoring complet (FastAPI/Chroma/SQLite/Ollama/logs).	‘2_Sources/2.1_Python/api/monitor.py‘
GET	‘/api/monitor/health‘	Health check (monitor).	‘2_Sources/2.1_Python/api/monitor.py‘