

GHI Sandbox Developer Guide – v0.2.0

Global HashCost Index Initiative

Version 0.2.0 — Sandbox API Developer Guide

Standard : GHI Technical Standard v1.0

Style : Apple / Minimal Tech

1. Couverture

Global HashCost Index Initiative

GHI Sandbox Developer Guide

Version 0.2.0

Technical Documentation — Phase 2 (Pre-Production)

Modèle minimal, clair, design premium (style Apple).

2. Table des matières

1. Introduction
2. Objectifs du Sandbox Engine
3. Architecture générale
4. Structure du dépôt ghi-engine
5. Installation & Environnement
6. Lancer le moteur sandbox

7. Documentation interactive (OpenAPI)
 8. Modèles de données (Data Model v1.0)
 9. Endpoints détaillés
 - /snapshot
 - /history
 - /regions
 - /regions/{id}
 - /stats
 10. Exemples d'appels API
 11. Intégration Python (client minimal)
 12. Intégration Dashboard / BI
 13. Tests & validation automatisée
 14. Limites de la sandbox
 15. Roadmap vers le moteur réel (Phase 3)
 16. Gouvernance & versioning
 17. Annexes techniques
-

3. Introduction

Le Global HashCost Index (GHI) est un standard ouvert visant à mesurer, de manière transparente et reproductible, **le coût global de production d'un bitcoin**.

Cette documentation présente l'**API Sandbox** : une version pré-production de l'API officielle, qui renvoie des données **synthétiques**, structurées selon le **GHI Data Model v1.0**, afin de permettre :

- des tests d'intégration,
- des preuves de concept,
- des analyses préliminaires,
- des démonstrations auprès d'équipes institutionnelles (Risk, ESG, Quant, Research).

Ce guide couvre l'intégralité du fonctionnement du sandbox engine, ses modèles de données, ses endpoints et son écosystème.

4. Objectifs du Sandbox Engine

La sandbox du GHI poursuit 4 objectifs distincts :

1. Présenter l'architecture API complète

Endpoints, modèles, structure, conventions, versioning.

2. Permettre les intégrations techniques préalables

Dashboards, notebooks, pipelines de données, tests internes.

3. Valider le modèle conceptuel

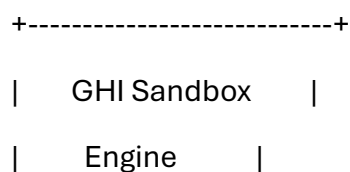
Avant l'ouverture de l'API réelle.

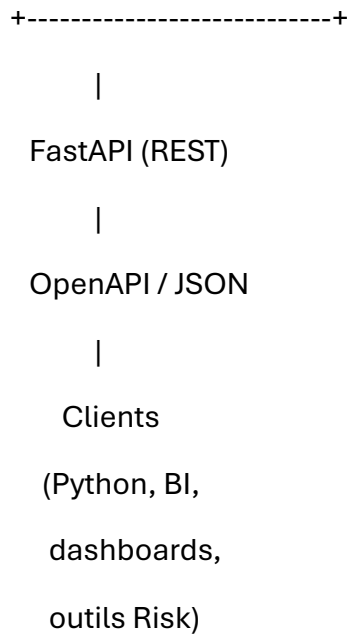
4. Faciliter l'audit institutionnel

Le sandbox fournit un environnement neutre et safe pour l'analyse préliminaire.

5. Architecture générale

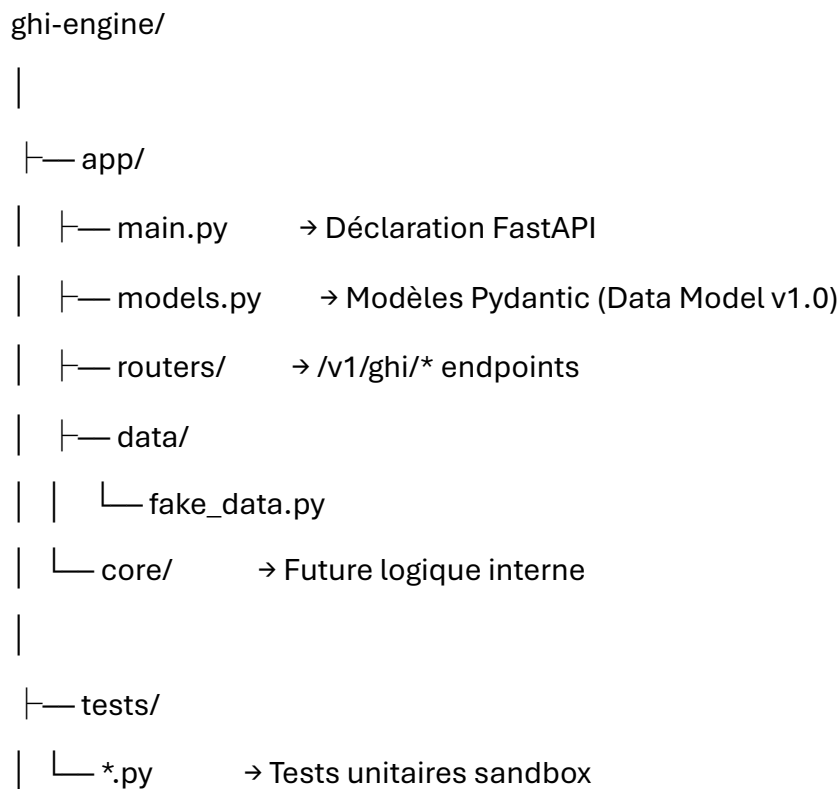
Système minimal, clair, orienté standards :





Le moteur réel de Phase 3 viendra remplacer le sandbox sans casser la compatibilité.

6. Structure du dépôt ghi-engine



```
|  
├── requirements.txt  
└── README.md
```

Structure épurée, conforme aux bonnes pratiques.

7. Installation & environnement

1. Cloner le dépôt

```
git clone <repo-url> ghi-engine  
cd ghi-engine
```

2. Environnement virtuel

```
python3 -m venv venv  
source venv/bin/activate
```

3. Installation

```
pip install --upgrade pip  
pip install -r requirements.txt
```

8. Lancer le moteur sandbox

```
uvicorn app.main:app --reload
```

API disponible sur :

<http://127.0.0.1:8000>

<http://127.0.0.1:8000/docs>

9. Documentation interactive

Automatiquement générée via OpenAPI 3 :

- /docs (Swagger UI)
- /redoc (Redoc)

Ces pages permettent d'exécuter les appels API en direct.

10. Modèles de données (Data Model v1.0)

Le sandbox implémente intégralement :

- Cost
- Region
- Snapshot
- HistoryPoint
- Stats

Exemple

Snapshot

```
{  
  
  "timestamp": "2025-01-01T12:00:00Z",  
  
  "ghi": {  
  
    "min_cost_btc": 23000.0,  
  
    "avg_cost_btc": 26000.0,  
  
    "max_cost_btc": 29000.0
```

```
},  
"difficulty": 800000000000000,  
"hashrate_total_th": 6000000000,  
"block_reward_btc": 3.125,  
"regions": [...],  
"version": "1.0"  
}
```

11. Endpoints détaillés

1. GET /v1/ghi/snapshot

Retourne l'état global du GHI.

2. GET /v1/ghi/history

Historique synthétique (journaliers).

3. GET /v1/ghi/regions

Liste des régions.

4. GET /v1/ghi/regions/{region_id}

Détails d'une région.

5. GET /v1/ghi/stats

Statistiques globales.

Chaque endpoint est conforme à GHI Data Model v1.0.

12. Exemples d'appels API

cURL

curl <http://127.0.0.1:8000/v1/ghi/snapshot>

Python (requests)

```
import requests
```

```
r = requests.get("http://127.0.0.1:8000/v1/ghi/snapshot")
```

```
print(r.json())
```

13. Intégration Python (client minimal)

```
import requests
```

```
class GHISandboxClient:
```

```
    BASE = "http://127.0.0.1:8000/v1/ghi"
```



```
def snapshot(self):  
    return requests.get(f"{self.BASE}/snapshot").json()  
  
def history(self):  
    return requests.get(f"{self.BASE}/history").json()  
  
def regions(self):  
    return requests.get(f"{self.BASE}/regions").json()  
  
def region(self, region_id):  
    return requests.get(f"{self.BASE}/regions/{region_id}").json()  
  
def stats(self):  
    return requests.get(f"{self.BASE}/stats").json()
```

14. Intégration Dashboard / BI

Compatible avec :

- Power BI
- Tableau
- Grafana
- Google Looker
- Python Jupyter
- Excel (PowerQuery)

Endpoints 100% JSON, timestamps ISO.

15. Tests & validation

pytest -q

Sortie attendue :

5 passed, 0 failed

Garantit la non-régression de la sandbox.

16. Limites de la Sandbox

- pas de données réelles
 - pas de moteur de coût réel
 - pas de granularité sous-régionale
 - données déterministes (pas de marché réel)
 - conçu exclusivement pour la Phase 2
-

17. Roadmap vers Phase 3

Phase 3 livrera :

- moteur réel de coût (min/avg/max)
 - données régionales réelles
 - snapshots quotidiens/intraday
 - API production versionnée
 - exports CSV/Parquet institutionnels
 - cohérence stricte avec Technical Standard v1.x
-

18. Gouvernance & versioning

Modèle SemVer :

- **MAJOR** : changements incompatibles
- **MINOR** : nouveaux endpoints / fonctionnalités
- **PATCH** : corrections & docs

Chaque release est accompagnée d'un changelog public.

19. Annexes techniques

- Notes sur le hashrate (TH/s)
 - Notes sur le block reward (3.125 BTC)
 - Définitions du trio min/avg/max
 - Contraintes institutionnelles
 - Justification du design API
-

FIN DU DOCUMENT

Global HashCost Index Initiative

Sandbox Developer Guide – v0.2.0