

API Reference

Base URL

- Development: `http://localhost:8000/api/v1`
- Production: `https://your-domain.com/api/v1`

Authentication

Currently, the API does not require authentication. Future versions will implement JWT-based authentication.

Endpoints

Health Check

`GET /health`

Check the health status of the API and connected services.

Response:

```
{
  "status": "healthy",
  "version": "0.1.0",
  "database": "connected",
  "redis": "connected",
  "s3": "connected"
}
```

Ingestion

`POST /api/v1/ingest/add-work`

Submit a new work for ingestion.

Request Body:

```
{
  "repo_url": "https://github.com/nbbulk-dotcom/COSMOLOGY",
  "slug": "cosmology-hub",
  "force_regenerate": false
}
```

Response:

```
{
  "job_id": "uuid",
  "status": "queued"
}
```

GET /api/v1/ingest/job/{job_id}

Check the status of an ingestion job.

Response:

```
{
  "job_id": "uuid",
  "status": "complete",
  "chunks": 42,
  "embeddings": 42
}
```

Query

POST /api/v1/query

Submit a query for hybrid retrieval.

Request Body:

```
{
  "session_id": "uuid",
  "user_query": "What is quantum resonance gravity?",
  "constraints": {
    "tags": ["quantum"],
    "canonical": true
  }
}
```

Response:

```
{
  "answer": "Quantum resonance gravity is...",
  "claims": [
    {
      "text": "...",
      "citation_ids": ["cosmology-hub:1.0.0:chunk001"]
    }
  ],
  "retrieval_ids": ["chunk001", "chunk002"]
}
```

Session Management

POST /api/v1/session/checkpoint

Create a session checkpoint.

Request Body:

```
{
  "session_id": "uuid",
  "condensed_summary": "...",
  "accepted_claims": [...],
  "top_citation_ids": [...]
```

Response:

```
{
  "checkpoint_id": "uuid"
}
```

GET /api/v1/session/rehydrate?checkpoint_id=uuid

Rehydrate a session from a checkpoint.

Response:

```
{
  "condensed_summary": "...",
  "top_short_summaries": [...],
  "supporting_chunk_ids": [...]
}
```

Verification

POST /api/v1/verify/run

Run citation verification on model output.

Request Body:

```
{
  "run_id": "uuid",
  "model_output": "...",
  "retrieval_ids": [...]
}
```

Response:

```
{
  "verifier_decision": "pass",
  "annotated_claims": [...]
}
```

Audit

GET /api/v1/audit/logs

Retrieve audit log entries.

Query Parameters:

- start_date : ISO 8601 datetime
- end_date : ISO 8601 datetime
- event_type : Filter by event type
- limit : Number of results (default: 100)

Response:

```
{
  "logs": [
    {
      "timestamp": "2025-10-24T12:00:00Z",
      "event_type": "retrieval",
      "correlation_id": "uuid",
      "metadata": {...}
    }
  ],
  "total": 1000,
  "page": 1
}
```

Error Responses

All endpoints return standard HTTP status codes:

- 200 OK : Success
- 400 Bad Request : Invalid request parameters
- 404 Not Found : Resource not found
- 500 Internal Server Error : Server error

Error Response Format:

```
{
  "detail": "Error message describing what went wrong"
}
```