Binoculars API User Guide

This guide covers how to run the local Binoculars API, use the GUI demo harness, and send example requests.

1) What You Get

The API mode (`--api`) exposes local HTTP endpoints for Binoculars scoring:

- `GET /` , `GET /health` , `GET /healthz`
- `POST /score`

`POST /score` returns the same core scoring object used by CLI JSON output (`input` , `observer` , `performer` , `cross` , `binoculars` , `cache`).

2) Fastest Start (Harness + API Together)

Use the launcher script:

```
./run_api_demo_harness.sh
```

What it does:

1. Checks whether API is already healthy on the configured port.
2. Starts `binoculars.sh --config fast --api <port>` if needed.
3. Waits for API health.
4. Opens the GUI harness (`api_demo_harness.py`).

Port configuration:

- The port is controlled by the top-level constant `API_PORT` in `run_api_demo_harness.sh` .

Notes:

- If the script started the API process, it will stop it when the harness exits.
- If an API server is already running on the same port, the script reuses it.

3) Start API Manually

Default port:

```
./binoculars.sh --config fast --api
```

Custom port:

```
./binoculars.sh --config fast --api 8787
```

Health check:

```
curl -sS http://127.0.0.1:8765/health
```

Expected shape:

```
{
  "ok": true,
  "service": "binoculars",
  "route": "/health",
  "ts_utc": "2026-02-22T21:06:47Z"
}
```

4) Launch Harness Manually

If API is already running:

```
venv/bin/python api_demo_harness.py
```

Optional: override API URL for this session:

```
BINOCULARS_API_BASE_URL=http://127.0.0.1:8787 venv/bin/python api_demo_harnes
```

Harness controls:

- `Base URL` : API endpoint root (for example `http://127.0.0.1:8765`)
- `Timeout (s)` : request timeout
- `Health Check` : calls `GET /health`
- `Score Segment` : calls `POST /score`
- `input_label` , `diagnose_paragraphs` , `diagnose_top_k` , `need_paragraph_profile` : optional request flags
- Response panes:
  - summary metrics
  - raw JSON response

5) API Request Examples

Minimal score request:

```
curl -sS \
  -H 'Content-Type: application/json' \
  -d '{"text":"This is a short test segment for binocular scoring."}' \
  http://127.0.0.1:8765/score
```

Request with optional flags:

```
curl -sS \
  -H 'Content-Type: application/json' \
  -d '{
    "text": "A longer text segment you want to evaluate...",
    "input_label": "draft-1",
    "diagnose_paragraphs": true,
    "diagnose_top_k": 12,
    "need_paragraph_profile": true
  }' \
  http://127.0.0.1:8765/score
```

The response includes:

- `ok` (boolean)
- `result` (standard Binoculars scoring object)
- `paragraph_profile` (present when `need_paragraph_profile=true`)

6) API Request Contract

`POST /score` body:

- Required:
    - `text` (string, non-empty)
- Optional:
    - `input_label` (string)
    - `diagnose_paragraphs` (boolean, default `false`)
    - `diagnose_top_k` (integer, default `10`, min `1`)
    - `need_paragraph_profile` (boolean, default `false`)

Error behavior:

- Invalid JSON or missing required fields returns an error JSON payload:
    - `{ "ok": false, "error": "<message>" }`

7) Common Issues

Connection refused:

- API is not running, wrong port, or wrong base URL.
- Verify with:

```
curl -sS http://127.0.0.1:8765/health
```

Long first response time:

- First score request may include model load time.
- Subsequent requests are typically faster while process is warm.

`text` validation error:

- Ensure request JSON includes non-empty string field `text` .

Port conflict:

- Change API port via:
  - `./binoculars.sh --config fast --api <port>` for manual mode, or
  - `API_PORT` constant in `run_api_demo_harness.sh` for launcher mode.