AI BENCH LAB

REST API Reference

Trigger benchmarks, retrieve results, and manage the queue programmatically.

Version 1.0 | March 2026

aibenchlab.com

Reference · Professional AI Model Benchmarking for Windows

Contents

Trigger benchmarks, retrieve results, and manage the queue programmatically.

1. Overview

The `aibenchlab-api` binary starts an HTTP server that exposes a REST API for benchmark automation. It shares the same backend as the desktop app and CLI.

Requirements: Consultant tier or higher license.

Default address: `http://127.0.0.1:8787`

2. Configuration

Configure the API server with environment variables:

Variable	Default	Description
`AIBL_API_ADDR`	`127.0.0.1:8787`	Listen address and port
`AIBL_API_TOKEN`	*(none)*	API authentication token. If not set, requests are accepted without authentication.
`AIBL_API_TOKEN_HEADER`	`x-aibl-token`	Custom header name for the token
`RUST_LOG`	`info`	Log verbosity level

Start the server:

BASH

```
AIBL_API_TOKEN=my-secret-token aibenchlab-api
```

3. Authentication

If `AIBL_API_TOKEN` is set, all endpoints (except `/health`) require the token in the request header.

BASH

```
curl -H "x-aibl-token: my-secret-token" http://localhost:8787/models
```

If the token is missing or incorrect, the server returns `401 Unauthorized`.

If `AIBL_API_TOKEN` is not set, no authentication is required. This is suitable for local development but not recommended for shared or networked environments.

---

4. Endpoints

GET /health

Health check. No authentication required.

Response (200):

JSON

```
{
  "status": "ok",
  "version": "0.1.0"
}
```

GET /models

List all available AI models across configured providers.

Response (200):

JSON

```
[
  {
    "id": "openai:gpt-4o",
    "name": "GPT-4 Optimized",
    "provider": "openai"
  },
  {
    "id": "ollama:llama3.2",
    "name": "llama3.2",
    "provider": "ollama"
  }
]
```

POST /benchmark

Start a benchmark run. Returns immediately with a session ID. Use `GET /models` to verify model IDs before submitting.

Request body:

JSON

```
{
  "model_ids": ["openai:gpt-4o", "ollama:llama3.2"],
  "session_name": "api-benchmark-2026-03",
  "domains": ["reasoning", "coding"],
  "tier": "standard",
  "deterministic": true,
  "seed": 42,
  "temperature": 0.7
}
```

Field	Type	Required	Description
`model_ids`	string[]	Yes	Model IDs in `provider:model` format
`session_name`	string	No	Custom name (defaults to `API-Run-{timestamp}`)
`domains`	string[]	No	Domain filter (see valid domains)
`languages`	string[]	No	Language filter
`tier`	string	No	`quick`, `standard`, or `comprehensive`
`suite_id`	string	No	Run a specific test suite
`suite_run_id`	string	No	Associate with a suite run
`deterministic`	boolean	No	Enable deterministic execution
`seed`	integer	No	Global seed for deterministic mode
`temperature`	float	No	Temperature override

Response (200):

JSON

```
{
  "session_id": "abc123def456"
}
```

Error responses:

- `400 Bad Request` — Invalid input (e.g., empty `model_ids`, invalid domain name)
- `500 Internal Server Error` — Processing error

POST /estimate-cost

Estimate the cost of a benchmark run without executing it.

Request body:

JSON

```
{
  "model_name": "gpt-4o",
  "provider_name": "openai",
  "provider_type": "cloud",
  "suite_id": "standard"
}
```

Field	Type	Required	Description
`model_name`	string	Yes	Model name
`provider_name`	string	Yes	Provider name
`provider_type`	string	Yes	`cloud` or `local`
`suite_id`	string	No	Suite to estimate (default: `standard`)

Response (200):

JSON

```
{
  "model_name": "gpt-4o",
  "provider_name": "openai",
  "provider_type": "cloud",
  "suite_id": "standard",
  "token_estimate": {
    "total_input_tokens": 50000,
    "total_output_tokens": 25000,
    "output_buffer_pct": 15.0,
    "estimation_source": "historical"
  },
  "cost_breakdown": {
    "input_token_cost": 0.125,
    "output_token_cost": 0.25,
    "total_api_cost": 0.375,
    "pricing_source": "OpenAI API pricing",
    "pricing_cached_at": "2026-03-29T12:00:00Z"
  },
  "confidence": "high",
  "projections": {
    "daily_light": 0.375,
    "daily_moderate": 3.75,
    "monthly_light": 11.25,
    "monthly_moderate": 112.50
  }
}
```


POST /export

Export a completed benchmark session to a file.

Request body:

JSON

```
{
  "session_id": "abc123def456",
  "format": "pdf",
  "output_path": "/absolute/path/to/report.pdf"
}
```

Field	Type	Required	Description
`session_id`	string	Yes	Session to export
`format`	string	Yes	`pdf`, `mbx`, `json`, or `csv`
`output_path`	string	Yes	Absolute path. Parent directory must exist.

Response (200):

JSON

```
{
  "ok": true
}
```

Error responses:

- `400 Bad Request` — Unsupported format
- `404 Not Found` — Session not found
- `500 Internal Server Error` — Export failed

POST /export/batch

Export multiple sessions in one request.

Request body:

JSON

```
{
  "session_ids": ["id1", "id2", "id3"],
  "format": "json",
  "output_path": "/absolute/path/to/exports"
}
```

Field	Type	Required	Description
`session_ids`	string[]	Yes	Non-empty array of session IDs
`format`	string	Yes	`pdf`, `mbx`, `json`, or `csv`
`output_path`	string	Yes	Absolute path to output directory. Must exist.

Response (200):

JSON

```
{
  "ok": true
}
```

---

5. Valid Domain Names

```
reasoning        code             chat
tool_calling     adversarial_safety   deployment_risk
agentic          multimodal       multi_turn_adversarial
agentic_email    context_retention
```

6. Usage Examples

Run a benchmark and poll for completion

BASH

```
# Start the benchmark
SESSION=$(curl -s -X POST http://localhost:8787/benchmark \
  -H "x-aibl-token: $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{"model_ids": ["ollama:llama3.2"], "tier": "quick"}' \
  | jq -r '.session_id')

echo "Started session: $SESSION"

# Export results when ready
curl -X POST http://localhost:8787/export \
  -H "x-aibl-token: $TOKEN" \
  -H "Content-Type: application/json" \
  -d "{\"session_id\": \"$SESSION\", \"format\": \"json\", \"output_path\": \"$(pwd)/results.json\"}"
```

Estimate cost before running

BASH

```
curl -s -X POST http://localhost:8787/estimate-cost \
  -H "x-aibl-token: $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "model_name": "gpt-4o",
    "provider_name": "openai",
    "provider_type": "cloud",
    "suite_id": "full-benchmark"
  }' | jq '.cost_breakdown.total_api_cost'
```

List available models

BASH

```
curl -s http://localhost:8787/models \
  -H "x-aibl-token: $TOKEN" | jq '.[].id'
```

Batch export

BASH

```
curl -X POST http://localhost:8787/export/batch \
  -H "x-aibl-token: $TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "session_ids": ["session1", "session2", "session3"],
    "format": "csv",
    "output_path": "/home/user/exports"
  }'
```

7. Error Handling

All error responses follow a consistent format:

JSON

```
{
  "error": "Description of what went wrong"
}
```

Status Code	Meaning
200	Success
400	Bad Request — invalid input
401	Unauthorized — missing or invalid token
404	Not Found — session or resource not found
500	Internal Server Error — unexpected failure

8. Initialization

The API server performs the same startup sequence as the desktop app:

› Storage and database initialization

› Provider registry and auto-detection

› Model catalog loading

› Hardware scan

› Judge server startup

The judge model and llama.cpp runtime must be installed via the desktop app's Setup screen before the API server can run benchmarks.

Resources

- Documentation: **aibenchlab.com/docs**
- Support: **aibenchlab.com/contact**
- YouTube walkthroughs: **youtube.com/@aibenchlab**
- Email: **support@aibenchlab.com**

AiBenchLab is built by a solo founder with 40+ years of software engineering experience. Every feature is designed with the conviction that you deserve honest, transparent, and reproducible AI model evaluations. No hype. Just truth.