# MasumiRanker REST API Specification

\*\*File:\*\* `api\_spec.md`

\*\*Version:\*\* 1.0 — 18 Apr 2025

\*\*Author:\*\* Thomas Wang (Backend)

---

## 1 Overview

MasumiRanker provides a REST interface for listing AI agents, submitting and retrieving ratings, and obtaining category‑based rankings.

\* All endpoints are read‑only except \*\*POST /ratings\*\*.

\* Authentication is optional for the hackathon prototype; each rating must include a `user\_id` to support future auditing.

---

## 2 Base URL

<http://localhost:3000>

\*(adjust host/port as needed)\*

---

## 3 Authentication

| Header | Type | Description |

|--------|---------|-----------------------------------------------|

| token | API key | Reserved for future use – not enforced yet |

---

## 4 HTTP Status Codes

| Code | Meaning |

|------|-----------------------------------|

| 200 | Successful request |

| 201 | Resource created |

| 400 | Validation or formatting error |

| 404 | Resource not found |

| 500 | Internal server error |

---

## 5 Endpoints

### 5.1 GET `/agents`

Returns a paginated list of all agents.

| Query Parameter | Type | Required | Description |

|-----------------|--------|----------|-------------------------------------------------------------|

| `category` | string | no | Filter by agent category |

| `sort\_by` | enum | no | `avg\_score` \| `num\_ratings` (default `avg\_score`) |

| `page` | int | no | Page index, starting at 1 (default 1) |

| `page\_size` | int | no | Items per page (default 20, max 100) |

\*\*Response 200\*\*

```json

{

"items": [

{

"id": "agent001",

"name": "Coding Mentor",

"category": "Education",

"avg\_score": 4.6,

"num\_ratings": 12

}

],

"page": 1,

"page\_size": 20,

"total\_items": 42

}

### 5.2 GET /agents/{id}

Returns full metadata for a specific agent.

| **Path Parameter** | **Type** | **Description** |
| --- | --- | --- |
| id | string | Agent identifier |

**Response 200**

{

"id": "agent001",

"name": "Coding Mentor",

"category": "Education",

"description": "Helps you learn to code.",

"did": "did:masumi:abc123",

"url": "https://example.com/codingmentor",

"avg\_score": 4.6,

"num\_ratings": 12

}

### 5.3 POST /ratings

Creates a new rating and comment.

| **Field** | **Type** | **Required** | **Validation / Notes** |
| --- | --- | --- | --- |
| agent\_id | string | yes | Must reference an existing agent |
| user\_id | string | yes | Arbitrary reviewer identifier |
| score | int | yes | Integer 1 – 5 |
| comment | string | no | 1 – 1 000 characters |

The server appends:

* timestamp — ISO‑8601 (UTC)
* hash — SHA‑256(agent\_id + user\_id + score + comment + timestamp)

**Request Example**

{

"agent\_id": "agent001",

"user\_id": "user123",

"score": 5,

"comment": "Outstanding agent for beginners."

}

**Response 201**

{

"agent\_id": "agent001",

"user\_id": "user123",

"score": 5,

"comment": "Outstanding agent for beginners.",

"timestamp": "2025-04-18T19:12:30Z",

"hash": "b8c6b2b275b3e4e2e2b41c7ce34e89d7c5b10a1bf5e6fe3d1a5b0c4f2e917a8f"

}

### 5.4 GET /ratings/{agent\_id}

Retrieves all ratings for the specified agent.

| **Path Parameter** | **Type** | **Description** |
| --- | --- | --- |
| agent\_id | string | Target agent |

| **Query Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| page | int | no | Page index (start 1) |
| page\_size | int | no | Items per page (max 100) |

**Response 200**

{

"items": [

{

"user\_id": "user123",

"score": 4,

"comment": "Helpful tool.",

"timestamp": "2025-04-18T12:00:00Z",

"hash": "a22b8f...90ff"

}

],

"page": 1,

"page\_size": 20,

"total\_items": 7

}

### 5.5 GET /rankings

Returns the top‑rated agents, optionally scoped to a category.

| **Query Parameter** | **Type** | **Required** | **Description** |
| --- | --- | --- | --- |
| category | string | no | If supplied, restrict to that category |
| limit | int | no | Number of agents to return (default 5) |

**Response 200**

[

{

"id": "agent007",

"name": "SEO Optimizer",

"category": "Marketing",

"avg\_score": 4.9,

"num\_ratings": 20

}

]

## 6 Data Integrity and Audit Trail

* Each rating response includes a server‑side timestamp and hash.
* Clients or third‑party verifiers can recompute the hash locally to detect any tampering.
* Future iterations may anchor the hash on‑chain via Masumi Decision Logging for immutable proof.

## 7 Rate Limiting (optional)

* **Unauthenticated quota:** 60 requests / minute per IP
* **Authenticated quota:** 1 000 requests / minute per API key

## 8 Pagination Conventions

* Index origin: 1
* Default page\_size: 20
* Maximum page\_size: 100

## 9 Example cURL Usage

# List agents in Education category, sorted by average score

curl -G "http://localhost:3000/agents" \

--data-urlencode "category=Education" \

--data-urlencode "sort\_by=avg\_score"

# Retrieve agent detail

curl "http://localhost:3000/agents/agent001"

# Submit a rating

curl -X POST "http://localhost:3000/ratings" \

-H "Content-Type: application/json" \

-d '{ "agent\_id": "agent001", "user\_id": "user123", "score": 5, "comment": "Great!" }'

# Fetch rankings for Marketing agents (top 10)

curl -G "http://localhost:3000/rankings" \

--data-urlencode "category=Marketing" \

--data-urlencode "limit=10"

*End of file*