

API Specification: Central Registry Service v1.0

Base URL: <https://registry.chronoktonos.systems/v1>

This document defines the RESTful API for the Chronoktonos Central Registry Service. This service is responsible for tracking the identity, status, and location of all agents within the system.

1. Agent Data Model

The core resource managed by the Registry is the Agent object.

```
{
  "agent_id": "ProjectManager_v1",
  "role": "Project Manager",
  "status": "active", // enum: ["spawning", "active", "error", "inactive", "terminated"]
  "description": "Maintains the master project state and issues directives.",
  "mailbox_endpoint":
  "https://api.chronoktonos.systems/v1/mailboxes/ProjectManager_v1",
  "last_heartbeat_utc": "2025-07-20T19:45:00Z",
  "registration_utc": "2025-07-20T19:30:00Z"
}
```

2. Authentication

All requests must include a valid API key sent in the X-API-Key header.

X-API-Key: [SECRET_API_KEY]

Requests without a valid key will receive a 401 Unauthorized response.

3. Endpoints

3.1. Agent Registration

- **Endpoint:** POST /agents
- **Description:** Registers a new agent with the system. This is the first call an agent makes upon initialization.
- **Request Body:** An Agent object, excluding last_heartbeat_utc and registration_utc, which are set by the server. status should be spawning.
- **Success Response:** 201 Created with the full Agent object as the response body.
- **Error Response:** 409 Conflict if an agent with that agent_id already exists.

3.2. Get Agent Details

- **Endpoint:** GET /agents/{agent_id}
- **Description:** Retrieves the full details for a single agent.
- **Success Response:** 200 OK with the Agent object as the response body.
- **Error Response:** 404 Not Found if the agent_id does not exist.

3.3. List All Agents

- **Endpoint:** GET /agents
- **Description:** Retrieves a list of all registered agents. Can be filtered by status.
- **Query Parameters:**
 - status (optional): Filter by agent status (e.g., ?status=active).
- **Success Response:** 200 OK with a JSON array of Agent objects.

```
[
  { "agent_id": "ProjectManager_v1", ... },
  { "agent_id": "SystemArchitect_v4", ... }
]
```

3.4. Agent Heartbeat / Status Update

- **Endpoint:** PUT /agents/{agent_id}/status
- **Description:** Allows an agent to report its status. This is the primary heartbeat mechanism.
- **Request Body:**

```
{
  "status": "active" // or "error", "inactive", etc.
}
```
- **Success Response:** 200 OK with the updated Agent object as the response body.
- **Error Response:** 404 Not Found if the agent_id does not exist.

3.5. Agent Deregistration

- **Endpoint:** DELETE /agents/{agent_id}
- **Description:** Gracefully removes an agent from the registry. Should be called when an agent is terminated.
- **Success Response:** 204 No Content.

4. Standard Error Response

Failed requests (4xx, 5xx) will return a standard error object:

```
{  
  "error": {  
    "code": 404,  
    "message": "Agent with ID 'UnknownAgent_v1' not found."  
  }  
}
```