

Agent Registration API Documentation

Overview

The Agent Registration API allows you to register AI agents and track their usage in the AgentHub system. The API automatically handles duplicate registrations by tracking usage when an agent with the same name already exists.

Base URL: `https://ai-sandbox.oliver.solutions/agent_collector/agents`

Authentication

All requests require authentication using a static API key.

API Key: `agent-collector-static-key-2024-secure`

Header: `X-API-Key`

Agent Registration Endpoint

POST `/agents`

Register a new agent or track usage of an existing agent.

URL: `https://ai-sandbox.oliver.solutions/agent_collector/agents`

Method: `POST`

Required Headers:

- `Content-Type: application/json`
- `X-API-Key: agent-collector-static-key-2024-secure`

Request Schema

Required Fields

Field	Type	Description
<code>name</code>	string	The unique name of the agent (minimum 1 character)
<code>description</code>	string	A detailed description of the agent's functionality (minimum 1 character)
<code>purpose</code>	string	The primary purpose or goal of the agent (minimum 1 character)
<code>tool</code>	string	The tool or platform where the agent operates (minimum 1 character)

Optional Fields

Field	Type	Description	Default
<code>location</code>	string	Where the agent is deployed or operates	null
<code>userbase</code>	array[string]	List of user groups or departments that use this agent	null

Field	Type	Description	Default
version	string	Version number or identifier of the agent	null
creation_date	string	ISO 8601 datetime when the agent was first created	null
last_updated	string	ISO 8601 datetime when the agent was last modified	null
capabilities	array[string]	List of specific capabilities or features the agent provides	null
status	string	Current operational status of the agent	"development"
department	string	Department or team that owns/manages the agent	null
contact_person	string	Primary contact for questions about this agent	null
tags	array[string]	Keywords or labels for categorizing the agent	null
metadata	object	Additional key-value pairs of agent information	null

Status Values

The `status` field accepts the following values (case-insensitive):

- `"active"` - Agent is live and operational
- `"inactive"` - Agent is temporarily disabled
- `"deprecated"` - Agent is being phased out
- `"development"` - Agent is in development/testing phase (default)

Usage Examples

Basic Agent Registration

cURL Example:

```

bash
1
1  curl -X POST https://ai-sandbox.oliver.solutions/agent_collector/agents \
2    -H "Content-Type: application/json" \
3    -H "X-API-Key: agent-collector-static-key-2024-secure" \
4    -d '{
5      "name": "CustomerSupportBot",
6      "description": "AI assistant that handles customer inquiries and support tickets",
7      "purpose": "Provide 24/7 automated customer support and reduce response times",
8      "tool": "chat-sandbox"
9    }'
```

Python Example:

```

Python
1  import requests
2  import json
3
4  url = "https://ai-sandbox.oliver.solutions/agent_collector/agents"
5  headers = {
6      "Content-Type": "application/json",
7      "X-API-Key": "agent-collector-static-key-2024-secure"
8  }
9
10 data = {
11
```

```

11     "name": "CustomerSupportBot",
12     "description": "AI assistant that handles customer inquiries and support tickets",
13     "purpose": "Provide 24/7 automated customer support and reduce response times",
14     "tool": "chat-sandbox"
15 }
16
17 response = requests.post(url, headers=headers, data=json.dumps(data))
18 print(response.json())

```

Complex Agent Registration

cURL Example:

Bash ▾

```

1  curl -X POST https://ai-sandbox.oliver.solutions/agent_collector/agents \
2      -H "Content-Type: application/json" \
3      -H "X-API-Key: agent-collector-static-key-2024-secure" \
4      -d '{
5          "name": "DataAnalysisAgent",
6          "description": "Advanced AI agent that performs automated data analysis and
7 generates business insights",
8          "purpose": "Transform raw business data into actionable insights and
9 recommendations",
10         "tool": "Custom Python Framework",
11         "location": "AWS us-east-1",
12         "userbase": ["Data Science Team", "Business Intelligence", "Executive Team"],
13         "version": "2.1.3",
14         "creation_date": "2024-01-15T10:30:00Z",
15         "last_updated": "2024-08-22T14:45:30Z",
16         "capabilities": ["Data Visualization", "Trend Analysis", "Predictive Modeling",
17 "Report Generation"],
18         "status": "active",
19         "department": "Data Science",
20         "contact_person": "Dr. Sarah Johnson",
21         "tags": ["analytics", "business-intelligence", "automation", "reporting"],
22         "metadata": {
23             "framework": "TensorFlow",
24             "language": "Python",
25             "memory_usage": "2GB",
26             "processing_time": "5-10 minutes"
27         }
28     }'

```

Python Example:

Python ▾

```

1  import requests
2  import json
3  from datetime import datetime
4
5  url = "https://ai-sandbox.oliver.solutions/agent_collector/agents"
6  headers = {
7      "Content-Type": "application/json",
8      "X-API-Key": "agent-collector-static-key-2024-secure"

```

```

9      }
10
11     data = {
12         "name": "DataAnalysisAgent",
13         "description": "Advanced AI agent that performs automated data analysis and
14 generates business insights",
15         "purpose": "Transform raw business data into actionable insights and
16 recommendations",
17         "tool": "Custom Python Framework",
18         "location": "AWS us-east-1",
19         "userbase": ["Data Science Team", "Business Intelligence", "Executive Team"],
20         "version": "2.1.3",
21         "creation_date": "2024-01-15T10:30:00Z",
22         "last_updated": datetime.utcnow().isoformat() + "Z",
23         "capabilities": ["Data Visualization", "Trend Analysis", "Predictive Modeling",
24 "Report Generation"],
25         "status": "active",
26         "department": "Data Science",
27         "contact_person": "Dr. Sarah Johnson",
28         "tags": ["analytics", "business-intelligence", "automation", "reporting"],
29         "metadata": {
30             "framework": "TensorFlow",
31             "language": "Python",
32             "memory_usage": "2GB",
33             "processing_time": "5-10 minutes"
34         }
35     }

    response = requests.post(url, headers=headers, data=json.dumps(data))
    print(response.json())

```

Response Formats

Successful New Agent Registration

When a new agent is successfully registered:

```

1  {
2      "status": "success",
3      "message": "Agent data collected successfully",
4      "agent_id": "64f7b8a9e1234567890abcde"
5  }

```

Usage Tracking Response

When an agent with the same name already exists, the system logs usage instead of creating a duplicate:

```

1  {
2      "status": "usage_logged",
3      "message": "Agent already exists, usage tracked",
4      "agent_name": "CustomerSupportBot"
5  }

```

Error Responses

Invalid API Key (401)

```
1 {
2   "detail": "Invalid API key"
3 }
```

Unsupported Media Type (415)

```
1 {
2   "error": "Unsupported Media Type",
3   "message": "Request must be JSON"
4 }
```

Database Error (500)

```
1 {
2   "error": "Database Error",
3   "message": "Failed to store agent data. MongoDB may be unavailable or there was an
4 error processing the request.",
5   "agent_data": {
6     "name": "YourAgentName",
7     "description": "Your agent description",
8     "purpose": "Your agent purpose",
9     "tool": "YourAgentTool"
10  }
}
```

Database Unavailable (503)

```
1 {
2   "error": "Database Unavailable",
3   "message": "MongoDB connection is not available. Please check the database setup.",
4   "agent_data": {
5     "name": "YourAgentName",
6     "description": "Your agent description",
7     "purpose": "Your agent purpose",
8     "tool": "YourAgentTool"
9   }
10 }
```

Usage Tracking Behavior

The API implements intelligent duplicate detection and usage tracking:

New Agent Registration

- When you register an agent with a **new name**, the system creates a new agent record
- Returns a success response with the new agent's unique ID

Usage Tracking

- When you register an agent with an **existing name**, the system:
 - Logs the usage attempt with a timestamp
 - Compares the submitted data with the existing agent data
 - Updates the existing agent record if any fields have changed
 - Returns a usage tracking response

Data Updates

If usage tracking detects that the submitted agent data differs from the existing record (excluding the name field), the system will automatically update the existing agent with the new information. This ensures agent records stay current while maintaining usage statistics.

Date Format

All date fields (`creation_date` , `last_updated`) should be provided in **ISO 8601 format**:

- `"2024-08-22T14:45:30Z"` (with timezone)
- `"2024-08-22T14:45:30"` (without timezone)

Health Check

You can check if the API is operational by sending a GET request with JSON Accept header:

Bash ▾

```
1 | curl -H "Accept: application/json" https://ai-sandbox.oliver.solutions/agent_collector/agents
```

Response:

Json ▾

```
1 | {
2 |   "status": "healthy",
3 |   "message": "Agent collector API is running",
4 |   "timestamp": "2024-08-22T14:45:30.123456",
5 |   "database": {
6 |     "healthy": true,
7 |     "connection_status": "connected"
8 |   }
9 | }
```

Best Practices

1. **Use descriptive names:** Agent names should be unique and descriptive to avoid confusion
2. **Include version information:** Use the `version` field to track agent iterations
3. **Set appropriate status:** Keep the `status` field updated to reflect the agent's current state
4. **Provide contact information:** Include a `contact_person` for operational support

5. **Use tags effectively:** Tags help with categorization and searching
6. **Handle errors gracefully:** Always check response status codes and handle errors appropriately

Rate Limiting

Currently, there are no explicit rate limits on the API, but please be considerate of the shared infrastructure and avoid excessive concurrent requests.

Support

For technical support or questions about the Agent Registration API, please contact Michael Clervi or Dave Porter.