API.md — Sentinel

Table of Contents

Overview

Sentinel provides a RESTful API for real-time infrastructure monitoring and alerting across the TheForge ecosystem. The API enables monitoring of VMs, Docker containers, services, backup status, and system healing operations.

Base URL: `http://localhost:3000/api` **Authentication:** None (inferred - internal infrastructure service) **Response Format:** JSON

Endpoints

Health & Status

GET /health

Health check endpoint for service monitoring.

Description: Returns the health status of the Sentinel service itself. **Response:**

```
{
  "status": "healthy",
  "timestamp": "2026-02-10T09:08:21.299Z"
}
```

Example Request:

```
curl http://localhost:3000/health
```

Status & Hosts

GET /api/status

Get overall system status.

Description: Returns aggregated status information across all monitored hosts. **Response:**

```
{
  "hosts": [],
  "containers": {},
  "services": {}
}
```

Example Request:

```
curl http://localhost:3000/api/status
```

GET /api/hosts

List all monitored hosts.

Description: Returns a list of all hosts being monitored by Sentinel. **Response:**

```
[
  {
    "id": "string",
    "name": "string",
    "status": "online|offline",
    "lastCheck": "timestamp"
  }
]
```

Example Request:

```
curl http://localhost:3000/api/hosts
```

GET /api/hosts/:id

Get specific host details.

Description: Returns detailed information about a specific host. **Path Parameters:**

Parameter	Type	Required	Description
id	string	Yes	Unique host identifier

Response:

```
{
  "id": "string",
  "name": "string",
  "status": "online|offline",
```

```
  "metrics": {},
  "containers": []
}
```

Example Request:

```
curl http://localhost:3000/api/hosts/claudinator
```

GET /api/hosts/:id/metrics

Get host metrics.

Description: Returns performance and health metrics for a specific host. **Path Parameters:**

Parameter	Type	Required	Description
----------	-------	----------	----------------------
id	string	Yes	Unique host identifier

Response:

```
{
  "cpu": "number",
  "memory": "number",
  "disk": "number",
  "uptime": "number",
  "timestamp": "timestamp"
}
```

Example Request:

```
curl http://localhost:3000/api/hosts/claudinator/metrics
```

POST /api/hosts

Register a new host.

Description: Adds a new host to the monitoring system. **Request Body:**

```
{
  "name": "string",
  "address": "string",
  "sshPort": "number",
  "credentials": {}
}
```

Response:

```
{
  "id": "string",
  "name": "string",
  "status": "registered"
}
```

Example Request:

```
curl -X POST http://localhost:3000/api/hosts \
  -H "Content-Type: application/json" \
  -d '{
    "name": "new-vm",
    "address": "192.168.1.100",
    "sshPort": 22
  }'
```

Containers

GET /api/containers

List all containers across all hosts.

Description: Returns a list of all Docker containers being monitored. **Response:**

```
[
  {
    "hostId": "string",
    "name": "string",
    "status": "running|stopped|restarting",
    "autoRestart": "boolean"
  }
]
```

Example Request:

```
curl http://localhost:3000/api/containers
```

GET /api/containers/:hostId

Get containers for a specific host.

Description: Returns all containers running on a specific host. **Path Parameters:**

Parameter	Type	Required	Description
hostId	string	Yes	Host identifier

Response:

```
[
  {
    "name": "string",
    "status": "running|stopped|restarting",
    "image": "string",
    "ports": [],
    "autoRestart": "boolean"
  }
]
```

Example Request:

```
curl http://localhost:3000/api/containers/claudinator
```

PUT /api/containers/:hostId/:name/auto-restart

Enable/disable container auto-restart.

Description: Updates the auto-restart configuration for a specific container. **Path Parameters:**

Parameter	Type	Required	Description
hostId	string	Yes	Host identifier
name	string	Yes	Container name

Request Body:

```
{
  "enabled": "boolean"
}
```

Response:

```
{
  "hostId": "string",
  "containerName": "string",
  "autoRestart": "boolean"
}
```

Example Request:

```
curl -X PUT http://localhost:3000/api/containers/claudinator/nginx/auto-rest
  -H "Content-Type: application/json" \
  -d '{"enabled": true}'
```

POST /api/containers/:hostId/:name/restart

Restart a container.

Description: Triggers a restart of the specified container. **Path Parameters:**

Parameter	Type	Required	Description
---	---	---	---
hostId	string	Yes	Host identifier
name	string	Yes	Container name

Response:

```
{
  "status": "restarting",
  "message": "Container restart initiated"
}
```

Example Request:

```
curl -X POST http://localhost:3000/api/containers/claudinator/nginx/restart
```

Alerts

GET /api/alerts

List all active alerts.

Description: Returns all currently active alerts in the system. **Response:**

```
[
  {
    "id": "string",
    "type": "string",
    "severity": "critical|warning|info",
    "message": "string",
    "timestamp": "timestamp",
    "acknowledged": "boolean"
  }
]
```

Example Request:

```
curl http://localhost:3000/api/alerts
```

POST /api/alerts

Create a new alert.

Description: Manually creates a new alert in the system. **Request Body:**

```
{
  "type": "string",
  "severity": "critical|warning|info",
  "message": "string",
  "metadata": {}
}
```

Response:

```
{
  "id": "string",
  "type": "string",
  "severity": "string",
  "message": "string",
  "timestamp": "timestamp"
}
```

Example Request:

```
curl -X POST http://localhost:3000/api/alerts \
  -H "Content-Type: application/json" \
  -d '{
    "type": "manual",
    "severity": "warning",
    "message": "Scheduled maintenance starting"
  }'
```

PUT /api/alerts/:id

Update an alert.

Description: Updates an existing alert's properties. **Path Parameters:**

Parameter	Type	Required	Description
id	string	Yes	Alert identifier

Request Body:

```
{
  "severity": "critical|warning|info",
  "message": "string",
  "acknowledged": "boolean"
}
```

Response:

```
{
  "id": "string",
  "updated": "boolean"
}
```

Example Request:

```
curl -X PUT http://localhost:3000/api/alerts/alert-123 \
  -H "Content-Type: application/json" \
  -d '{"severity": "info"}'
```

DELETE /api/alerts/:id

Delete an alert.

Description: Removes an alert from the system. **Path Parameters:**

Parameter	Type	Required	Description
id	string	Yes	Alert identifier

Response:

```
{
  "deleted": "boolean"
}
```

Example Request:

```
curl -X DELETE http://localhost:3000/api/alerts/alert-123
```

GET /api/alerts/history

Get alert history.

Description: Returns historical alert data. **Query Parameters:**

Parameter	Type	Required	Description
limit	number	No	Maximum number of records (inferred)
offset	number	No	Pagination offset (inferred)

Response:

```
[
  {
    "id": "string",
    "type": "string",
    "severity": "string",
    "message": "string",
    "timestamp": "timestamp",
    "acknowledged": "boolean",
    "acknowledgedAt": "timestamp",
    "resolvedAt": "timestamp"
  }
]
```

Example Request:

```
curl http://localhost:3000/api/alerts/history?limit=50
```

POST /api/alerts/history/:id/acknowledge

Acknowledge a historical alert.

Description: Marks a historical alert as acknowledged. **Path Parameters:**

Parameter	Type	Required	Description
id	string	Yes	Alert identifier

Response:

```
{
  "id": "string",
  "acknowledged": "boolean",
  "acknowledgedAt": "timestamp"
}
```

Example Request:

```
curl -X POST http://localhost:3000/api/alerts/history/alert-123/acknowledge
```

Backups

GET /api/backups

List all backup configurations and status.

Description: Returns information about configured backups across all hosts. **Response:**

```
[
  {
    "hostId": "string",
    "backupName": "string",
    "lastRun": "timestamp",
    "status": "success|failed|pending",
    "size": "number"
  }
]
```

Example Request:

```
curl http://localhost:3000/api/backups
```

GET /api/backups/history

Get backup execution history.

Description: Returns historical backup execution data. **Response:**

```
[
  {
    "id": "string",
    "hostId": "string",
    "backupName": "string",
    "timestamp": "timestamp",
    "status": "success|failed",
    "duration": "number",
    "size": "number"
  }
]
```

Example Request:

```
curl http://localhost:3000/api/backups/history
```

POST /api/backups/check

Trigger a backup check.

Description: Manually triggers a backup verification check across all hosts. **Request Body:**

```
{
  "hostId": "string"
}
```

Response:

```
{
  "status": "checking",
  "timestamp": "timestamp"
}
```

Example Request:

```
curl -X POST http://localhost:3000/api/backups/check \
  -H "Content-Type: application/json" \
  -d '{"hostId": "claudinator"}'
```

Configuration

GET /api/config

Get system configuration.

Description: Returns the current Sentinel configuration settings. **Response:**

```
{
  "checkInterval": "number",
  "alertThresholds": {},
  "monitoredHosts": [],
  "backupSettings": {}
}
```

Example Request:

```
curl http://localhost:3000/api/config
```

Events

GET /api/events

List recent system events.

Description: Returns a stream of recent infrastructure events. **Query Parameters:**

Parameter	Type	Required	Description
limit	number	No	Maximum number of events (inferred)
type	string	No	Filter by event type (inferred)

Response:

```
[
  {
    "id": "string",
    "type": "string",
    "timestamp": "timestamp",
    "hostId": "string",
    "message": "string",
    "metadata": {}
  }
]
```

Example Request:

```
curl http://localhost:3000/api/events?limit=100
```

GET /api/events/status

Get event processing status.

Description: Returns the status of the event processing system. **Response:**

```
{
  "processing": "boolean",
  "queueSize": "number",
  "lastProcessed": "timestamp"
}
```

Example Request:

```
curl http://localhost:3000/api/events/status
```

Forge Integration

GET /api/forge/tasks

List all Forge orchestrator tasks.

Description: Returns all tasks being tracked from TheForge orchestrator. **Response:**

```
[
  {
    "taskId": "string",
    "projectId": "string",
    "status": "string",
    "progress": "number",
    "startedAt": "timestamp"
  }
]
```

Example Request:

```
curl http://localhost:3000/api/forge/tasks
```

GET /api/forge/tasks/:projectId

Get tasks for a specific project.

Description: Returns all Forge tasks associated with a specific project. **Path Parameters:**

Parameter	Type	Required	Description
projectId	string	Yes	TheForge project ID

Response:

```
[
  {
    "taskId": "string",
    "projectId": "string",
    "status": "string",
    "progress": "number",
    "startedAt": "timestamp",
    "completedAt": "timestamp"
  }
]
```

Example Request:

```
curl http://localhost:3000/api/forge/tasks/26
```

GET /api/forge/activity

Get recent Forge activity.

Description: Returns recent activity from TheForge orchestrator. **Response:**

```
[
  {
    "timestamp": "timestamp",
    "projectId": "string",
    "action": "string",
    "status": "string"
  }
]
```

Example Request:

```
curl http://localhost:3000/api/forge/activity
```

GET /api/forge/projects

List monitored Forge projects.

Description: Returns all TheForge projects being monitored by Sentinel. **Response:**

```
[
  {
    "projectId": "string",
    "name": "string",
    "status": "string",
    "lastActivity": "timestamp"
  }
]
```

Example Request:

```
curl http://localhost:3000/api/forge/projects
```

Healing

GET /api/healing

Get auto-healing status.

Description: Returns the status of the auto-healing system and recent actions.
Response:

```
{
  "enabled": "boolean",
  "recentActions": [],
  "stats": {
    "totalHeals": "number",
    "successRate": "number"
  }
}
```

Example Request:

```
curl http://localhost:3000/api/healing
```

GET /api/healing/stats

Get healing statistics.

Description: Returns detailed statistics about auto-healing operations. **Response:**

```
{
  "totalAttempts": "number",
  "successful": "number",
  "failed": "number",
  "byType": {},
  "byHost": {}
}
```

Example Request:

```
curl http://localhost:3000/api/healing/stats
```

Error Handling

All endpoints follow a consistent error response format:

```
{
  "error": {
    "code": "string",
    "message": "string",
    "details": {}
  }
}
```

Common HTTP Status Codes

Status Code	Description
200	Success
201	Resource created successfully
400	Bad request - invalid parameters
404	Resource not found
500	Internal server error
503	Service unavailable (e.g., SSH connection failed)

Example Error Response

```
{
  "error": {
    "code": "HOST_UNREACHABLE",
    "message": "Unable to connect to host 'claudinator'",
```

```
    "details": {
      "hostId": "claudinator",
      "reason": "SSH connection timeout"
    }
  }
}
```

Rate Limiting

(inferred) No explicit rate limiting detected in the codebase. As an internal infrastructure service, rate limiting may not be necessary. However, consider implementing rate limiting if exposing this API beyond internal infrastructure.

Notes

- This API is designed for internal infrastructure monitoring within the TheForge ecosystem
- SSH connectivity (via ssh2 library) is used for host communication
- Data persistence is handled via better-sqlite3
- Scheduled monitoring tasks run via node-cron
- All timestamps are in ISO 8601 format
- Most response structures are inferred from route definitions and may vary based on actual implementation

Related Documentation

- Readme
- Architecture
- Deployment
- Contributing