



# OptiZeus SCADA REST API

Integrate with External Systems · ERP · LIMS · Custom Applications

OptiZeus SCADA v2.5.4 · API-OZ-001 · © 2026 Zeus Automation

optizeus.org · sales@optizeus.org · 3 pages



## API OVERVIEW

The OptiZeus REST API allows Enterprise tier users to integrate with external systems, read and write tag values programmatically, retrieve historian data, manage alarms, and trigger batch operations. All endpoints require JWT Bearer token authentication.

## AUTHENTICATION

All API requests require a Bearer token in the Authorization header.

```
# 1. Get token
POST /api/auth/login
{
  "username": "api_user",
  "password": "your_password"
}

# Response
{ "token": "eyJhbGciOiJI..." }

# 2. Use token in all requests
Authorization: Bearer eyJhbGciOiJI...
```

Tokens expire after the configured session timeout. Use POST /api/auth/refresh to get a new token without re-login.

## TAG ENDPOINTS

METHOD	ENDPOINT	DESCRIPTION
GET	/api/tags	List all tags with current values
GET	/api/tags/{name}	Get single tag value + metadata
POST	/api/tags/{name}/write	Write value to tag (Supervisor+)
GET	/api/tags/{name}/history	Get historian data with time range
POST	/api/tags	Create new tag (Engineer+)
PUT	/api/tags/{name}	Update tag configuration
DELETE	/api/tags/{name}	Delete tag (Admin only)

## API BASE URL & HEADERS

```
Base URL:
https://your-server:3000/api

Required headers:
Content-Type: application/json
Authorization: Bearer {token}
Accept: application/json
```

## ALARM ENDPOINTS

METHOD	ENDPOINT	DESCRIPTION
GET	/api/alarms	List active alarms
GET	/api/alarms/history	Alarm journal with filters
POST	/api/alarms/{id}/ack	Acknowledge alarm
POST	/api/alarms/{id}/shelve	Shelve alarm with duration

## HISTORIAN QUERY EXAMPLE

```
GET /api/tags/Reactor_Temp/history
?start=2026-04-01T00:00:00Z
&end=2026-04-02T00:00:00Z
&mode=average
&interval=3600

# Response
{ "tag": "Reactor_Temp",
  "data": [
    { "ts": "2026-04-01T00:00:00Z", "v": 72.4 },
    { "ts": "2026-04-01T01:00:00Z", "v": 73.1 }
  ]
}
```

## BATCH ENDPOINTS

GET	/api/batches	List batch records
GET	/api/batches/{id}	Get full EBR for batch
POST	/api/batches/start	Start batch execution
POST	/api/batches/{id}/close	Close batch with e-sig





## USER MANAGEMENT ENDPOINTS

GET	/api/users	List all users (Admin)
POST	/api/users	Create user (Admin)
PUT	/api/users/{id}	Update user role/settings
DELETE	/api/users/{id}	Disable user account
GET	/api/users/me	Current user profile
POST	/api/auth/logout	Invalidate current token

## WEBSOCKET — REAL-TIME TAG UPDATES

Subscribe to live tag values via WebSocket for real-time dashboards:

```
const ws = new WebSocket(
  "wss://server:3000/ws?token=eyJ..."
);

// Subscribe to tags
ws.send(JSON.stringify({
  type: "subscribe",
  tags: ["Reactor_Temp", "Pump_Speed"]
}));

// Receive updates
ws.onmessage = (e) => {
  const {tag,value,quality,ts} = JSON.parse(e.data);
};
```

## ERROR CODES

CODE	STATUS	DESCRIPTION
400	Bad Request	Invalid parameters or missing required fields
401	Unauthorized	Missing or invalid JWT token
403	Forbidden	Insufficient role for this operation
404	Not Found	Tag, alarm, or batch does not exist
409	Conflict	Tag name already exists
422	Unprocessable	Value out of range or wrong data type
429	Too Many Requests	API rate limit exceeded — wait and retry
500	Server Error	Internal error — check server logs

## RATE LIMITS

Default Limit	100 requests per minute per token
Historian Query	20 requests per minute (heavy queries)
Tag Write	500 writes per minute per connection
WebSocket	10 concurrent subscriptions per client
Auth Endpoint	10 login attempts per minute per IP
Header	X-RateLimit-Remaining in response headers

## INTEGRATION EXAMPLES

```
# Python — read tag value
import requests
r = requests.get(
  "https://server:3000/api/tags/Temp_PV",
  headers={"Authorization": f"Bearer {token}"}
)
print(r.json()["value"])
```





## COMPLETE ENDPOINT REFERENCE

Method	Endpoint	Auth Required	Description
POST	/api/auth/login	Public	Authenticate — returns JWT token
POST	/api/auth/refresh	Token	Refresh expiring token
POST	/api/auth/logout	Token	Invalidate current token
GET	/api/tags	Operator+	List all tags with live values
GET	/api/tags/{name}	Operator+	Single tag value, quality, EU
POST	/api/tags/{name}/write	Supervisor+	Write value with e-sig support
GET	/api/tags/{name}/history	Operator+	Historian data with mode/interval
POST	/api/tags	Engineer+	Create tag with full config
PUT	/api/tags/{name}	Engineer+	Update tag configuration
DELETE	/api/tags/{name}	Admin	Delete tag and historian data
GET	/api/alarms	Operator+	Active alarms list
GET	/api/alarms/history	Operator+	Alarm journal — filters: date/user
POST	/api/alarms/{id}/ack	Operator+	Acknowledge with reason code
POST	/api/alarms/{id}/shelve	Supervisor+	Shelve alarm with duration
GET	/api/batches	Operator+	List batch records
GET	/api/batches/{id}	Operator+	Full EBR for batch
POST	/api/batches/start	Supervisor+	Start batch execution
POST	/api/batches/{id}/close	Supervisor+	Close batch with e-signature
GET	/api/screens	Operator+	List HMI screens
GET	/api/users	Admin	List all user accounts
POST	/api/users	Admin	Create user account
PUT	/api/users/{id}	Admin	Update user role/settings
DELETE	/api/users/{id}	Admin	Disable user account
GET	/api/system/status	Operator+	Server health and connection status
GET	/api/system/drivers	Engineer+	Driver connection status
POST	/api/system/deploy	Admin	Deploy Test→Production config
GET	/api/audit	Admin	Audit trail records with filters
GET	/api/reports	Operator+	List available reports
POST	/api/reports/{id}/generate	Operator+	Generate report — returns PDF/CSV

