

# Agent CEO API Reference

---

## Quick Start

---

Base URL: `https://your-api-domain.com/api`

Authentication: Bearer Token (JWT)

```
# Example API call
curl -X GET "https://your-api-domain.com/api/agents" \
  -H "Authorization: Bearer YOUR_JWT_TOKEN" \
  -H "Content-Type: application/json"
```

## Authentication

---

### POST /auth/register

Register a new user account.

**Request Body:**

```
{
  "email": "user@example.com",
  "password": "secure_password",
  "first_name": "John",
  "last_name": "Doe",
  "company": "Example Corp"
}
```

**Response:**

```
{
  "user": {
    "id": "user_123",
    "email": "user@example.com",
    "first_name": "John",
    "last_name": "Doe"
  },
  "access_token": "jwt_token_here",
  "refresh_token": "refresh_token_here"
}
```

## POST /auth/login

Authenticate user and get access tokens.

### Request Body:

```
{
  "email": "user@example.com",
  "password": "secure_password"
}
```

## POST /auth/token/refresh

Refresh access token using refresh token.

### Headers:

```
Authorization: Bearer REFRESH_TOKEN
```

# Agents

---

## GET /agents

List all agents for the authenticated user.

**Query Parameters:** - `page` (optional): Page number (default: 1) - `per_page` (optional): Items per page (default: 10) - `type` (optional): Filter by agent type - `status` (optional): Filter by agent status

### Response:

```
{
  "agents": [
    {
      "id": "agent_123",
      "name": "CEO Agent",
      "type": "ceo",
      "status": "active",
      "capabilities": ["strategic_planning", "decision_making"],
      "created_at": "2024-01-01T00:00:00Z",
      "last_active": "2024-01-01T12:00:00Z"
    }
  ],
  "total": 1,
  "page": 1,
  "per_page": 10
}
```

## POST /agents

Create a new agent.

### Request Body:

```
{
  "name": "Marketing Agent",
  "type": "marketing",
  "description": "Handles marketing campaigns and content creation",
  "capabilities": ["content_creation", "social_media", "campaign_management"],
  "configuration": {
    "tone": "professional",
    "target_audience": "B2B"
  }
}
```

## GET /agents/{agent\_id}

Get details of a specific agent.

## PUT /agents/{agent\_id}

Update agent configuration.

## DELETE /agents/{agent\_id}

Delete an agent.

## GET /agents/{agent\_id}/metrics

Get performance metrics for an agent.

### Response:

```
{
  "performance_metrics": {
    "tasks_completed": 150,
    "success_rate": 0.95,
    "average_completion_time": 300,
    "efficiency_score": 0.88
  },
  "activity_summary": {
    "last_24_hours": 12,
    "last_7_days": 89,
    "last_30_days": 340
  }
}
```

## Tasks

---

### GET /tasks

List tasks with filtering and pagination.

**Query Parameters:** - `status` (optional): Filter by task status - `agent_id` (optional): Filter by agent - `priority` (optional): Filter by priority - `page`, `per_page`: Pagination

### POST /tasks

Create a new task.

### Request Body:

```
{
  "title": "Market Analysis Report",
  "description": "Analyze Q4 market trends",
  "agent_id": "agent_123",
  "priority": "high",
  "deadline": "2024-01-15T17:00:00Z",
  "input_data": {
    "market_segment": "enterprise_software"
  }
}
```

## GET /tasks/{task\_id}

Get task details and progress.

## PUT /tasks/{task\_id}

Update task information.

## DELETE /tasks/{task\_id}

Cancel/delete a task.

## POST /tasks/batch

Perform batch operations on multiple tasks.

# Strategic Intelligence

---

## POST /strategic/business-analysis

Generate comprehensive business analysis.

### Request Body:

```
{
  "analysis_type": "comprehensive",
  "business_context": {
    "industry": "technology",
    "company_size": "startup"
  },
  "focus_areas": ["growth_opportunities", "competitive_threats"]
}
```

## POST /strategic/competitive-analysis

Analyze competitive landscape.

## POST /strategic/growth-strategy

Generate growth strategy recommendations.

## POST /strategic/decision-making

Get decision support analysis.

## POST /strategic/quick-insights

Get rapid strategic insights.

### Request Body:

```
{
  "query": "Should we expand to European markets?",
  "context": "SaaS startup with $2M ARR",
  "urgency": "high"
}
```

## Email Automation

---

### GET /email/campaigns

List email campaigns.

### POST /email/campaigns

Create new email campaign.

### Request Body:

```
{
  "name": "Q1 Product Launch",
  "subject": "New Product Announcement",
  "template_id": "template_123",
  "recipient_lists": ["list_456"],
  "schedule": {
    "send_time": "2024-01-15T09:00:00Z"
  }
}
```

### GET /email/templates

List email templates.

## POST /email/templates

Create email template.

## GET /email/lists

List email lists and segments.

## POST /email/lists

Create new email list.

## GET /email/campaigns/{campaign\_id}/analytics

Get campaign performance analytics.

# Data Analysis

---

## POST /data-analysis/upload

Upload data file for analysis.

**Form Data:** - `file`: Data file (CSV, Excel, JSON, PDF) - `analysis_type`: Type of analysis to perform - `data_format`: File format

## POST /data-analysis/analyze

Analyze uploaded data.

### Request Body:

```
{
  "data_source_id": "data_123",
  "analysis_type": "trend_analysis",
  "parameters": {
    "time_column": "date",
    "value_columns": ["revenue", "customers"]
  }
}
```

## GET /data-analysis/results/{analysis\_id}

Get analysis results.

## POST /data-analysis/reports

Create custom report.

# Social Media

---

## GET /social-media/posts

List social media posts.

## POST /social-media/posts

Create and schedule social media post.

### Request Body:

```
{
  "content": "Exciting news about our new product!",
  "platforms": ["twitter", "linkedin"],
  "schedule": {
    "publish_at": "2024-01-15T10:00:00Z"
  },
  "media_urls": ["https://example.com/image.jpg"]
}
```

## GET /social-media/analytics

Get social media performance analytics.

# n8n Workflows

---

## GET /n8n/workflows

List available workflows.



## POST /n8n/workflows/{workflow\_id}/execute

Execute a workflow.

### Request Body:

```
{
  "input_data": {
    "customer_email": "customer@example.com",
    "order_id": "order_123"
  }
}
```

## GET /n8n/executions

List workflow executions.

## GET /n8n/executions/{execution\_id}

Get execution details.

# System

---

## GET /health

System health check.

### Response:

```
{
  "status": "healthy",
  "timestamp": "2024-01-01T12:00:00Z",
  "services": {
    "database": "healthy",
    "redis": "healthy",
    "ai_services": "healthy"
  }
}
```

## GET /dashboard/stats

Get dashboard statistics.

## Response:

```
{
  "agents": {
    "total": 5,
    "active": 4
  },
  "tasks": {
    "total": 150,
    "completed": 140,
    "in_progress": 8,
    "failed": 2
  },
  "performance": {
    "success_rate": 0.93,
    "average_completion_time": 450
  }
}
```

## Error Handling

---

All API endpoints return consistent error responses:

```
{
  "error": {
    "code": "VALIDATION_ERROR",
    "message": "Invalid input data",
    "details": {
      "field": "email",
      "issue": "Invalid email format"
    }
  }
}
```

## HTTP Status Codes

- 200 : Success
- 201 : Created
- 400 : Bad Request
- 401 : Unauthorized
- 403 : Forbidden
- 404 : Not Found
- 429 : Rate Limited

- `500` : Internal Server Error

## Rate Limits

---

- **Free Tier:** 100 requests/hour
- **Pro Tier:** 1000 requests/hour
- **Enterprise:** Custom limits

Rate limit headers:

```
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 95
X-RateLimit-Reset: 1640995200
```

## Webhooks

---

Configure webhooks to receive real-time notifications:

### POST /webhooks

Create webhook endpoint.

**Request Body:**

```
{
  "url": "https://your-app.com/webhook",
  "events": ["task.completed", "agent.status_changed"],
  "secret": "webhook_secret_key"
}
```

### Webhook Events

- `task.created`
- `task.completed`
- `task.failed`
- `agent.status_changed`
- `campaign.sent`

- `workflow.executed`

## SDKs and Examples

---

### JavaScript/Node.js

```
const AgentCEO = require('@agent-ceo/sdk');

const client = new AgentCEO({
  apiKey: 'your-api-key',
  baseUrl: 'https://your-api-domain.com/api'
});

// Create an agent
const agent = await client.agents.create({
  name: 'Sales Agent',
  type: 'sales',
  capabilities: ['lead_generation', 'email_outreach']
});

// Create a task
const task = await client.tasks.create({
  title: 'Generate leads for Q1',
  agent_id: agent.id,
  priority: 'high'
});
```

### Python

```
from agent_ceo import AgentCEOClient

client = AgentCEOClient(
    api_key='your-api-key',
    base_url='https://your-api-domain.com/api'
)

# Create strategic analysis
analysis = client.strategic.business_analysis({
    'analysis_type': 'comprehensive',
    'business_context': {
        'industry': 'technology',
        'company_size': 'startup'
    }
})
```

## cURL Examples

```
# Get all agents
curl -X GET "https://your-api-domain.com/api/agents" \
  -H "Authorization: Bearer YOUR_TOKEN"

# Create a task
curl -X POST "https://your-api-domain.com/api/tasks" \
  -H "Authorization: Bearer YOUR_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "title": "Market Research",
    "agent_id": "agent_123",
    "priority": "high"
  }'
```