

Contents

Cato Admin API	1
Overview	1
Status & Health	2
GET /status	2
GET /health	2
Budget Management	3
GET /budget/status	3
GET /budget/config	3
PUT /budget/config	3
GET /budget/history	4
Cache Management	4
GET /cache/stats	4
POST /cache/invalidate	5
Memory Management	5
GET /memory/stats	5
GET /memory/facts	5
POST /memory/facts	6
GET /memory/goals	6
PUT /memory/goals	6
GET /memory/meta-state	7
Shadow Self	7
GET /shadow-self/status	7
POST /shadow-self/test	7
NLI Testing	8
POST /nli/test	8
Error Responses	8
Rate Limits	8
Related Documentation	9

Cato Admin API

Admin endpoints for managing the Cato global consciousness service.

Base Path: /api/admin/cato

Authentication: Requires admin role with `consciousness_admin` permission.

Overview

The Cato Admin API provides endpoints for: - **Status & Health:** Monitor Cato's operational state - **Budget Management:** Configure and monitor spending limits - **Cache Management:** View and invalidate semantic cache - **Memory Management:** Access and modify Cato's memory systems - **Shadow Self:** Test and monitor the Shadow Self endpoint - **NLI:** Test the NLI entailment classifier

Status & Health

GET /status

Get comprehensive Cato status including budget, cache, memory, and health.

Response:

```
{
  "mode": "day",
  "canExplore": true,
  "budget": {
    "dailySpend": 3.45,
    "monthlySpend": 127.89,
    "dailyRemaining": 11.55,
    "monthlyRemaining": 372.11
  },
  "cache": {
    "hitRate": 0.86,
    "size": 1234567
  },
  "memory": {
    "semanticFactCount": 50000,
    "workingMemoryEntries": 1000,
    "domainsCount": 800
  },
  "shadowSelf": {
    "healthy": true
  },
  "timestamp": "2024-01-15T12:00:00Z"
}
```

GET /health

Simple health check for monitoring systems.

Response:

```
{
  "healthy": true,
  "components": {
    "shadowSelf": true,
    "budget": true,
    "mode": "day"
  },
  "timestamp": "2024-01-15T12:00:00Z"
}
```

Budget Management

GET /budget/status

Get current budget status.

Response:

```
{
  "mode": "day",
  "dailySpend": 3.45,
  "monthlySpend": 127.89,
  "dailyRemaining": 11.55,
  "monthlyRemaining": 372.11,
  "canExplore": true,
  "nextModeChange": "2024-01-16T02:00:00Z",
  "config": {
    "monthlyLimit": 500,
    "dailyExplorationLimit": 15,
    "explorationRatio": 0.2,
    "nightStartHour": 2,
    "nightEndHour": 6,
    "emergencyThreshold": 0.9
  }
}
```

GET /budget/config

Get budget configuration.

Response:

```
{
  "monthlyLimit": 500,
  "dailyExplorationLimit": 15,
  "explorationRatio": 0.2,
  "nightStartHour": 2,
  "nightEndHour": 6,
  "emergencyThreshold": 0.9
}
```

PUT /budget/config

Update budget configuration.

Request:

```
{
  "monthlyLimit": 1000,
  "dailyExplorationLimit": 30,
  "nightStartHour": 3,
```

```
    "nightEndHour": 5
}
```

Validation: - monthlyLimit: 0-100000 - dailyExplorationLimit: 0-1000 - nightStartHour: 0-23 - nightEndHour: 0-23 - emergencyThreshold: 0.5-0.99

Response:

```
{
  "message": "Budget config updated",
  "updates": {
    "monthlyLimit": 1000,
    "dailyExplorationLimit": 30
  }
}
```

GET /budget/history

Get cost history for the current month.

Response:

```
{
  "dailyHistory": [
    { "date": "2024-01-01", "amount": 12.34 },
    { "date": "2024-01-02", "amount": 14.56 }
  ],
  "monthlyBreakdown": {
    "inference": 80.00,
    "curiosity": 30.00,
    "grounding": 10.00,
    "consolidation": 5.00,
    "total": 125.00
  }
}
```

Cache Management

GET /cache/stats

Get semantic cache statistics.

Response:

```
{
  "hitRate": 0.86,
  "totalHits": 1234567,
  "totalMisses": 200000,
  "cacheSize": 1434567
}
```

POST /cache/invalidate

Invalidate cache entries for a domain.

Request:

```
{
  "domain": "climate_change"
}
```

Response:

```
{
  "message": "Cache invalidated",
  "domain": "climate_change",
  "entriesRemoved": 45
}
```

Memory Management

GET /memory/stats

Get memory system statistics.

Response:

```
{
  "semanticFactCount": 50000,
  "workingMemoryEntries": 1000,
  "domainsCount": 800
}
```

GET /memory/facts

Get semantic facts by domain.

Query Parameters: - `domain` (optional, default: "general"): Domain to query - `limit` (optional, default: 50): Maximum facts to return

Response:

```
{
  "domain": "physics",
  "facts": [
    {
      "factId": "abc123",
      "subject": "light",
      "predicate": "travels_at",
      "object": "299792458 m/s",
      "domain": "physics",
      "confidence": 0.99,
      "sources": ["physics.gov"],
    }
  ]
}
```

```

        "createdAt": "2024-01-01T00:00:00Z",
        "updatedAt": "2024-01-15T00:00:00Z",
        "version": 3
    }
],
"count": 1
}

```

POST /memory/facts

Store a new semantic fact.

Request:

```

{
  "subject": "water",
  "predicate": "boils_at",
  "object": "100°C at 1 atm",
  "domain": "chemistry",
  "confidence": 0.99,
  "sources": ["chemistry.org"]
}

```

Response:

```

{
  "message": "Fact stored",
  "factId": "def456"
}

```

GET /memory/goals

Get current Cato goals.

Response:

```

{
  "goals": [
    "Learn more about quantum computing",
    "Improve understanding of climate models",
    "Consolidate knowledge about neural networks"
  ]
}

```

PUT /memory/goals

Update Cato goals.

Request:

```

{
  "goals": [
    "Explore machine learning fundamentals",

```

```
    "Understand protein folding"
  ]
}
```

Response:

```
{
  "message": "Goals updated",
  "goals": [
    "Explore machine learning fundamentals",
    "Understand protein folding"
  ]
}
```

GET /memory/meta-state

Get current meta-cognitive state.

Response:

```
{
  "state": "CONFIDENT",
  "attentionFocus": "machine learning"
}
```

Shadow Self

GET /shadow-self/status

Get Shadow Self endpoint status.

Response:

```
{
  "status": "InService",
  "instanceCount": 10,
  "pendingInstanceCount": 10
}
```

POST /shadow-self/test

Test Shadow Self with a prompt.

Request:

```
{
  "text": "Explain the theory of relativity"
}
```

Response:

```
{
  "generatedText": "The theory of relativity...",
}
```

```
"uncertainty": 0.23,  
"logitsEntropy": 1.45,  
"latencyMs": 234,  
"hiddenStateLayersExtracted": 3  
}
```

NLI Testing

POST /nli/test

Test NLI entailment classification.

Request:

```
{  
  "premise": "The sky is blue",  
  "hypothesis": "The sky has a color"  
}
```

Response:

```
{  
  "label": "entailment",  
  "scores": {  
    "entailment": 0.95,  
    "neutral": 0.04,  
    "contradiction": 0.01  
  },  
  "confidence": 0.95,  
  "surprise": 0.0,  
  "latencyMs": 45  
}
```

Error Responses

All endpoints return errors in this format:

```
{  
  "error": "Error message here"  
}
```

Common Status Codes: - 400 - Bad Request (validation error) - 401 - Unauthorized (missing tenant ID) - 404 - Not Found - 500 - Internal Server Error

Rate Limits

Endpoint	Rate Limit
GET endpoints	100/minute
POST /cache/invalidate	10/minute
POST /shadow-self/test	10/minute
POST /nli/test	100/minute
PUT /budget/config	10/minute

Related Documentation

- [Architecture Overview](#)
- [ADR-005: Circadian Budget](#)
- [ADR-007: Semantic Caching](#)
- [ADR-008: Shadow Self](#)
- [Deployment Runbook](#)