Gateway Service API Documentation

Comprehensive API reference for the Mission Engadi Gateway Service

Base URL

```
http://localhost:8000/api/v1
```

Authentication

Most endpoints require JWT authentication. Include the token in the Authorization header:

```
Authorization: Bearer <your_jwt_token>
```

Table of Contents

Health & Monitoring

GET /health

Description: Basic health check without dependency verification

Authentication: Not required

Response: 200 OK

```json
{
  "status": "healthy",
  "service": "Gateway Service",
  "version": "0.1.0",
  "timestamp": "2024-12-25T00:00:00Z"
}
```

GET /ready

Description: Readiness check including database and Redis

Authentication: Not required

Response: 200 OK

```json
{
  "ready": true,
  "service": "Gateway Service",
  "checks": {
    "database": "healthy",
    "redis": "healthy"
  }
}
```

GET /live

Description: Liveness probe for container orchestration

Authentication: Not required

Response: 200 OK

```json
{
  "alive": true
}
```

Route Management

GET /routes

Description: List all configured routes

Authentication: Required

Response: 200 OK

```json
[
  {
    "id": 1,
    "path": "/api/v1/auth/*",
    "target_service": "auth-service",
    "target_url": "http://auth-service:8001",
    "methods": ["GET", "POST"],
    "auth_required": false,
    "rate_limit": 100,
    "is_active": true,
    "created_at": "2024-12-25T00:00:00Z",
    "updated_at": "2024-12-25T00:00:00Z"
  }
]
```

GET /routes/{id}

Description: Get a specific route by ID

Authentication: Required

Path Parameters:
- `id` (integer): Route ID

Response: 200 OK

```json
{
  "id": 1,
  "path": "/api/v1/auth/*",
  "target_service": "auth-service",
  "target_url": "http://auth-service:8001",
  "methods": ["GET", "POST"],
  "auth_required": false,
  "rate_limit": 100,
  "is_active": true
}
```

POST /routes

Description: Create a new route configuration

Authentication: Required (Admin role)

Request Body:

```json
{
  "path": "/api/v1/content/*",
  "target_service": "content-service",
  "target_url": "http://content-service:8002",
  "methods": ["GET", "POST", "PUT", "DELETE"],
  "auth_required": true,
  "rate_limit": 50,
  "is_active": true
}
```

Response: 201 Created

```json
{
  "id": 2,
  "path": "/api/v1/content/*",
  "target_service": "content-service",
  "target_url": "http://content-service:8002",
  "methods": ["GET", "POST", "PUT", "DELETE"],
  "auth_required": true,
  "rate_limit": 50,
  "is_active": true,
  "created_at": "2024-12-25T00:00:00Z"
}
```

PUT /routes/{id}

Description: Update an existing route

Authentication: Required (Admin role)

Path Parameters:
- `id` (integer): Route ID

Request Body:

```
{
  "rate_limit": 100,
  "is_active": false
}
```

Response: 200 OK

```
{
  "id": 2,
  "path": "/api/v1/content/*",
  "rate_limit": 100,
  "is_active": false,
  "updated_at": "2024-12-25T00:00:00Z"
}
```

DELETE /routes/{id}

Description: Delete a route configuration

Authentication: Required (Admin role)

Path Parameters:
- `id` (integer): Route ID

Response: 204 No Content

Rate Limiting

GET /rate-limits

Description: List all rate limit rules

Authentication: Required

Response: 200 OK

```json
[
  {
    "id": 1,
    "path": "/api/v1/auth/login",
    "limit": 5,
    "window": 60,
    "is_active": true,
    "created_at": "2024-12-25T00:00:00Z"
  }
]
```

POST /rate-limits

Description: Create a new rate limit rule

Authentication: Required (Admin role)

Request Body:

```json
{
  "path": "/api/v1/public/*",
  "limit": 100,
  "window": 60,
  "is_active": true
}
```

Response: 201 Created

PUT /rate-limits/{id}

Description: Update a rate limit rule

Authentication: Required (Admin role)

Path Parameters:
- `id` (integer): Rule ID

Request Body:

```json
{
  "limit": 200,
  "window": 60
}
```

Response: 200 OK

DELETE /rate-limits/{id}

Description: Delete a rate limit rule

Authentication: Required (Admin role)

Response: 204 No Content

GET /rate-limits/check

Description: Check rate limit status for a client

Authentication: Required

Query Parameters:
- `path` (string): Endpoint path
- `client_id` (string): Client identifier

Response: 200 OK

```json
{
  "allowed": true,
  "limit": 100,
  "remaining": 87,
  "reset_time": "2024-12-25T00:01:00Z"
}
```

Service Health

GET /health/services

Description: Get health status of all backend services

Authentication: Required

Response: 200 OK

```json
[
  {
    "service_name": "auth-service",
    "status": "healthy",
    "response_time": 0.05,
    "last_check": "2024-12-25T00:00:00Z"
  },
  {
    "service_name": "content-service",
    "status": "unhealthy",
    "response_time": null,
    "error_message": "Connection timeout",
    "last_check": "2024-12-25T00:00:00Z"
  }
]
```

POST /health/services/{service_name}/check

Description: Manually trigger health check for a service

Authentication: Required (Admin role)

Path Parameters:
- `service_name` (string): Service identifier

Response: 200 OK

```json
{
  "service_name": "auth-service",
  "status": "healthy",
  "response_time": 0.05
}
```

Proxy

ANY /{path:path}

Description: Proxy requests to backend services based on route configuration

Authentication: Depends on route configuration

Path Parameters:
- `path` (string): Full request path

Headers:
- All headers are forwarded to the target service
- `X-Forwarded-For`: Client IP address
- `X-Gateway-Request-ID`: Unique request ID for tracing

Response: Varies based on backend service

Example:

```
# Request to gateway
GET /api/v1/auth/users/me
Authorization: Bearer <token>

# Proxied to auth-service
GET http://auth-service:8001/api/v1/users/me
Authorization: Bearer <token>
X-Forwarded-For: 192.168.1.100
X-Gateway-Request-ID: abc123
```

Logging & Audit

GET /logs

Description: Retrieve gateway logs

Authentication: Required (Admin role)

Query Parameters:
- `start_date` (string, ISO 8601): Start date filter
- `end_date` (string, ISO 8601): End date filter
- `path` (string, optional): Filter by path
- `status_code` (integer, optional): Filter by status code
- `limit` (integer, default: 100): Number of records
- `offset` (integer, default: 0): Pagination offset

Response: 200 OK

```json
{
  "total": 1234,
  "items": [
    {
      "id": 1,
      "request_id": "abc123",
      "method": "GET",
      "path": "/api/v1/auth/users/me",
      "status_code": 200,
      "response_time": 0.15,
      "client_ip": "192.168.1.100",
      "user_id": "user123",
      "target_service": "auth-service",
      "timestamp": "2024-12-25T00:00:00Z"
    }
  ]
}
```

GET /logs/{request_id}

Description: Get detailed log for a specific request

Authentication: Required (Admin role)

Path Parameters:
- `request_id` (string): Unique request identifier

Response: 200 OK

```json
{
  "request_id": "abc123",
  "method": "POST",
  "path": "/api/v1/content/items",
  "status_code": 201,
  "request_headers": {
    "Authorization": "Bearer ***",
    "Content-Type": "application/json"
  },
  "request_body": {"title": "New Item"},
  "response_body": {"id": 123, "title": "New Item"},
  "response_time": 0.25,
  "timestamp": "2024-12-25T00:00:00Z"
}
```

Error Responses

All error responses follow this format:

```json
{
  "error": {
    "code": "ERROR_CODE",
    "message": "Human-readable error message",
    "details": {
      "field": "Additional context"
    }
  }
}
```

Common Error Codes

- `400 Bad Request` : Invalid request parameters
- `401 Unauthorized` : Missing or invalid authentication
- `403 Forbidden` : Insufficient permissions
- `404 Not Found` : Resource not found
- `429 Too Many Requests` : Rate limit exceeded
- `500 Internal Server Error` : Server error
- `502 Bad Gateway` : Backend service error
- `503 Service Unavailable` : Service temporarily unavailable
- `504 Gateway Timeout` : Backend service timeout

Rate Limiting Headers

All responses include rate limiting headers:

```
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 87
X-RateLimit-Reset: 1640390400
```

Request Tracing

Each request is assigned a unique ID for tracing:

```
X-Gateway-Request-ID: abc123-def456-ghi789
```

Use this ID when:
- Debugging issues
- Correlating logs across services
- Contacting support

Pagination

List endpoints support pagination:

Query Parameters:
- `limit` (integer, default: 100, max: 1000): Items per page
- `offset` (integer, default: 0): Number of items to skip

Response:

```
{
  "total": 250,
  "limit": 100,
  "offset": 0,
  "items": []
}
```

Versioning

API is versioned via URL path: `/api/v1/`

Breaking changes will be introduced in new versions (v2, v3, etc.)

Interactive Documentation

Explore the API interactively:

- **Swagger UI**: http://localhost:8000/docs
- **ReDoc**: http://localhost:8000/redoc
- **OpenAPI JSON**: http://localhost:8000/openapi.json

Support

- **Issues**: GitHub Issues (https://github.com/mission-engadi/gateway_service/issues)
- **Email**: support@engadi.org
- **Documentation**: docs.engadi.org (https://docs.engadi.org)