

AI Service - API Documentation

Complete API reference for all 49 endpoints across 8 routers.

Base URL

```
http://localhost:8010/api/v1
```

Authentication

All endpoints require JWT authentication via Bearer token:

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

1. Content Generation Endpoints (8 endpoints)

1.1 Generate Social Media Post

POST /content/generate/social

Generate AI-powered social media posts.

Request Body:

```
{
  "topic": "Mission Update: New School in Uganda",
  "platform": "twitter",
  "tone": "professional",
  "language": "en",
  "include_hashtags": true,
  "include_emoji": false,
  "target_audience": "donors",
  "max_length": 280
}
```

Response:

```
{
  "id": "gen_123abc",
  "content": "Exciting news! We've opened a new school in Uganda...",
  "platform": "twitter",
  "language": "en",
  "metadata": {
    "hashtags": ["MissionUpdate", "Uganda", "Education"],
    "character_count": 245
  },
  "created_at": "2024-12-24T10:00:00Z"
}
```

1.2 Generate Article

POST /content/generate/article

Generate long-form articles.

Request Body:

```
{
  "topic": "Digital Ministry in Modern Missions",
  "tone": "informative",
  "length": "medium",
  "language": "en",
  "keywords": ["digital", "ministry", "missions"],
  "include_introduction": true,
  "include_conclusion": true,
  "target_word_count": 1000
}
```

Response:

```
{
  "id": "art_456def",
  "title": "Digital Ministry in Modern Missions: A Comprehensive Guide",
  "content": "In the evolving landscape of global missions...",
  "summary": "An exploration of digital tools in modern ministry",
  "word_count": 1024,
  "reading_time": 5,
  "metadata": {
    "keywords": ["digital", "ministry", "missions"],
    "sections": ["introduction", "body", "conclusion"]
  },
  "created_at": "2024-12-24T10:00:00Z"
}
```

1.3 Generate Story

POST /content/generate/story

Generate mission stories and testimonials.

Request Body:

```
{
  "story_type": "testimony",
  "topic": "Life transformation through education",
  "tone": "inspiring",
  "length": "medium",
  "language": "en",
  "include_call_to_action": true
}
```

1.4 Generate Donor Communication

POST /content/generate/donor-communication

Generate personalized donor communications.

Request Body:

```
{
  "communication_type": "thank_you",
  "donor_name": "John Smith",
  "donation_amount": 100.0,
  "project_name": "School Building Project",
  "tone": "warm",
  "language": "en",
  "include_impact_story": true
}
```

1.5 Generate Newsletter**POST** /content/generate/newsletter

Generate newsletter content.

Request Body:

```
{
  "title": "Monthly Mission Update - December 2024",
  "sections": ["updates", "stories", "prayer_requests"],
  "tone": "informative",
  "language": "en",
  "include_donation_appeal": true
}
```

1.6 Generate Batch Content**POST** /content/generate/batch

Generate multiple content pieces in one request.

Request Body:

```
{
  "requests": [
    {
      "content_type": "social_post",
      "topic": "Update 1",
      "platform": "twitter"
    },
    {
      "content_type": "social_post",
      "topic": "Update 2",
      "platform": "facebook"
    }
  ],
  "language": "en"
}
```

1.7 Get Generated Content**GET** /content/generate/{content_id}

Retrieve a specific generated content.

1.8 List Generated Content

GET /content/generate

Query Parameters:

- `content_type` (optional): Filter by type
 - `language` (optional): Filter by language
 - `skip` (default: 0): Pagination offset
 - `limit` (default: 10): Items per page
-

2. Translation Endpoints (5 endpoints)

2.1 Translate Text

POST /translation/translate

Translate text to target language.

Request Body:

```
{
  "text": "Hello, welcome to our mission",
  "source_language": "en",
  "target_language": "es",
  "preserve_formatting": true,
  "formal_tone": false
}
```

Response:

```
{
  "translated_text": "Hola, bienvenido a nuestra misión",
  "source_language": "en",
  "target_language": "es",
  "confidence_score": 0.98,
  "character_count": 35,
  "created_at": "2024-12-24T10:00:00Z"
}
```

2.2 Batch Translation

POST /translation/batch

Translate multiple texts to multiple languages.

Request Body:

```
{
  "texts": ["Hello", "Welcome", "Thank you"],
  "source_language": "en",
  "target_languages": ["es", "fr", "pt"]
}
```

2.3 Detect Language

POST /translation/detect

Detect the language of a text.

Request Body:

```
{
  "text": "Bonjour, bienvenue à notre mission"
}
```

Response:

```
{
  "language": "fr",
  "confidence": 0.99,
  "alternatives": [
    {"language": "en", "confidence": 0.01}
  ]
}
```

2.4 Auto-Translate Workflow

POST /translation/auto-translate

Create automatic translation workflow.

Request Body:

```
{
  "content_id": "content_123",
  "target_languages": ["es", "fr", "pt"],
  "auto_publish": false,
  "quality_threshold": 0.9
}
```

2.5 Get Translation History

GET /translation/history

Query Parameters:

- source_language (optional)
- target_language (optional)
- skip (default: 0)
- limit (default: 10)

3. Image Generation Endpoints (4 endpoints)

3.1 Generate Image

POST /images/generate

Generate AI images from text prompts.

Request Body:

```
{
  "prompt": "Beautiful sunset over African village with children playing",
  "size": "1024x1024",
  "style": "photorealistic",
  "num_images": 1,
  "quality": "hd"
}
```

Response:

```
{
  "id": "img_789ghi",
  "images": [
    {
      "url": "https://example.com/generated_image.png",
      "size": "1024x1024",
      "format": "png"
    }
  ],
  "prompt": "Beautiful sunset over African village...",
  "created_at": "2024-12-24T10:00:00Z"
}
```

3.2 Generate Image Variation

POST /images/variation

Create variations of an existing image.

3.3 Get Generated Image

GET /images/{image_id}

3.4 List Generated Images

GET /images

4. Content Enhancement Endpoints (6 endpoints)

4.1 Improve Grammar

POST /enhancement/grammar

Correct grammar and spelling.

Request Body:

```
{
  "text": "This are a test sentence with mistakes",
  "language": "en",
  "preserve_style": true
}
```

Response:

```
{
  "original_text": "This are a test sentence with mistakes",
  "improved_text": "This is a test sentence with mistakes",
  "corrections": [
    {
      "type": "grammar",
      "original": "This are",
      "corrected": "This is",
      "explanation": "Subject-verb agreement"
    }
  ],
  "confidence": 0.99
}
```

4.2 Adjust Tone

POST /enhancement/tone

Adjust content tone.

Request Body:

```
{
  "text": "We need money for our project",
  "target_tone": "professional",
  "language": "en"
}
```

4.3 Optimize for SEO

POST /enhancement/seo

Optimize content for search engines.

Request Body:

```
{
  "text": "Our mission work in Africa...",
  "target_keywords": ["mission", "Africa", "education"],
  "language": "en"
}
```

4.4 Summarize Content

POST /enhancement/summarize

Create content summaries.

Request Body:

```
{
  "text": "Long article content...",
  "summary_length": "short",
  "language": "en",
  "bullet_points": false
}
```

4.5 Batch Enhancement

POST /enhancement/batch

Process multiple enhancement requests.

4.6 Enhancement History

GET /enhancement/history

5. Automation Endpoints (7 endpoints)

5.1 Create Workflow

POST /automation/workflows

Create automation workflow.

Request Body:

```
{
  "name": "Daily Social Media Posts",
  "workflow_type": "content_generation",
  "schedule": "0 9 * * *",
  "configuration": {
    "content_type": "social_post",
    "platforms": ["twitter", "facebook"],
    "topics": ["mission_updates"]
  },
  "enabled": true
}
```

Response:

```
{
  "id": "wf_123",
  "name": "Daily Social Media Posts",
  "workflow_type": "content_generation",
  "status": "active",
  "schedule": "0 9 * * *",
  "created_at": "2024-12-24T10:00:00Z"
}
```

5.2 Get Workflow

GET /automation/workflows/{workflow_id}

5.3 List Workflows

GET /automation/workflows

5.4 Update Workflow

PUT /automation/workflows/{workflow_id}

5.5 Delete Workflow

DELETE /automation/workflows/{workflow_id}

5.6 Execute Workflow

POST /automation/workflows/{workflow_id}/execute

5.7 Get Workflow History

GET /automation/workflows/{workflow_id}/history

6. AI Tasks Endpoints (6 endpoints)

6.1 Create Task

POST /tasks

Create new AI task.

Request Body:

```
{
  "task_type": "content_generation",
  "description": "Generate social media posts for campaign",
  "priority": "high",
  "metadata": {
    "campaign_id": "camp_123",
    "platform": "twitter"
  }
}
```

6.2 Get Task

GET /tasks/{task_id}

6.3 List Tasks

GET /tasks

Query Parameters:

- `status` (optional): Filter by status
- `task_type` (optional): Filter by type
- `skip` (default: 0)
- `limit` (default: 10)

6.4 Approve Task

POST /tasks/{task_id}/approve

Request Body:

```
{
  "comments": "Looks good, approved",
  "metadata": {}
}
```

6.5 Reject Task**POST** /tasks/{task_id}/reject**Request Body:**

```
{
  "reason": "Needs revision - tone is too casual",
  "metadata": {}
}
```

6.6 Get Task Statistics**GET** /tasks/statistics**Response:**

```
{
  "total": 150,
  "by_status": {
    "pending": 25,
    "in_progress": 10,
    "approved": 100,
    "rejected": 15
  },
  "by_type": {
    "content_generation": 80,
    "translation": 45,
    "image_generation": 25
  },
  "average_completion_time": 3600
}
```

7. Content Templates Endpoints (7 endpoints)**7.1 Create Template****POST** /templates

Create reusable content template.

Request Body:

```
{
  "name": "Social Media Update Template",
  "content_type": "social_post",
  "template_text": "Exciting news! {{announcement}}. Join us in {{action}}. #{{hashtag}}",
  "variables": ["announcement", "action", "hashtag"],
  "category": "social_media",
  "metadata": {
    "platform": "twitter"
  }
}
```

7.2 Get Template

GET /templates/{template_id}

7.3 List Templates

GET /templates

Query Parameters:

- content_type (optional)
- category (optional)
- skip (default: 0)
- limit (default: 10)

7.4 Update Template

PUT /templates/{template_id}

7.5 Delete Template

DELETE /templates/{template_id}

7.6 Apply Template

POST /templates/{template_id}/apply

Request Body:

```
{
  "variables": {
    "announcement": "New school opening",
    "action": "celebrating this milestone",
    "hashtag": "Education"
  }
}
```

7.7 Suggest Templates

POST /templates/suggest

Get AI-suggested templates.

Request Body:

```
{
  "content_type": "social_post",
  "context": "Mission updates for social media"
}
```

8. Generated Content Endpoints (6 endpoints)

8.1 Create Generated Content

POST /generated

Save generated content.

8.2 Get Generated Content

GET /generated/{content_id}

8.3 List Generated Content

GET /generated

8.4 Update Generated Content

PUT /generated/{content_id}

8.5 Delete Generated Content

DELETE /generated/{content_id}

8.6 Publish Generated Content

POST /generated/{content_id}/publish

Publish content to external services.

Request Body:

```
{
  "target_service": "content_service",
  "publish_immediately": true,
  "schedule_time": null,
  "metadata": {
    "category": "mission_updates"
  }
}
```

Error Responses

All endpoints return standard error responses:

```
{
  "detail": "Error message",
  "error_code": "VALIDATION_ERROR",
  "timestamp": "2024-12-24T10:00:00Z"
}
```

HTTP Status Codes

- 200 OK : Successful request
- 201 Created : Resource created
- 400 Bad Request : Invalid input
- 401 Unauthorized : Missing or invalid authentication
- 403 Forbidden : Insufficient permissions
- 404 Not Found : Resource not found
- 422 Unprocessable Entity : Validation error
- 500 Internal Server Error : Server error

Rate Limiting

- Standard tier:** 100 requests/minute
- Premium tier:** 1000 requests/minute

Rate limit headers:

```
X-RateLimit-Limit: 100
X-RateLimit-Remaining: 95
X-RateLimit-Reset: 1640347200
```

Pagination

List endpoints support pagination:

Query Parameters:

- `skip` : Number of items to skip (default: 0)
- `limit` : Number of items to return (default: 10, max: 100)

Response:

```
{
  "items": [...],
  "total": 150,
  "skip": 0,
  "limit": 10,
  "has_more": true
}
```

Webhooks

Configure webhooks to receive notifications:

Events:

- content.generated
 - translation.completed
 - image.generated
 - task.approved
 - task.rejected
 - workflow.completed
-

SDK Support

Official SDKs available:

- Python: `pip install mission-engadi-ai-client`
 - JavaScript: `npm install @mission-engadi/ai-client`
-

Last Updated: December 2024

API Version: 1.0.0