

AI Service - API Summary

Overview

The AI Service provides AI-powered content generation, translation, enhancement, and automation capabilities for the Mission Engadi platform. Built with FastAPI, SQLAlchemy, and Abacus.AI integration.

Port: 8010

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

Architecture

Core Components

1. **Abacus.AI Integration Client** (`app/core/abacus_client.py`)
 - LLM-powered text generation
 - Multi-language translation
 - Content enhancement
 - Image generation (placeholder)
 2. **Service Layers** (8 services in `app/services/`)
 - AIUserService - Task management
 - ContentGenerationService - Content creation
 - TranslationService - Multi-language support
 - ImageGenerationService - Image creation
 - ContentEnhancementService - Content improvement
 - AutomationService - Workflow automation
 - ContentTemplateService - Template management
 - ServiceIntegration - External service calls
 3. **API Endpoints** (8 routers with 49 total endpoints)
-

API Endpoints Summary

1. Content Generation (8 endpoints)

Tag: `content-generation`

Method	Endpoint	Description	Auth Required
POST	/generate/social-post	Generate social media post	✓
POST	/generate/article	Generate article	✓
POST	/generate/story	Generate impact story	✓
POST	/generate/donor-letter	Generate donor thank you letter	✓
POST	/generate/newsletter	Generate newsletter	✓
POST	/generate/prayer-request	Generate prayer request	✓
POST	/generate/campaign-copy	Generate campaign copy	✓
POST	/generate/batch	Batch generate content	✓

Example Request:

```
POST /api/v1/generate/social-post
{
  "platform": "facebook",
  "topic": "New water well project in Kenya",
  "tone": "inspirational",
  "max_length": 500,
  "include_hashtags": true,
  "template_id": "optional-uuid"
}
```

Example Response:

```
{
  "task_id": "uuid",
  "content_id": "uuid",
  "content": "Generated social media post text...",
  "status": "completed"
}
```

2. Translation (5 endpoints)

Tag: `translation`

Method	Endpoint	Description	Auth Required
POST	/translate	Translate text	✓
POST	/translate/batch	Batch translate texts	✓
POST	/translate/auto	Auto-translate content	✓
GET	/translate/{id}	Get translation job	✓
GET	/translate	List translation jobs	✓

Supported Languages: en , es , fr , pt

Example Request:

```
POST /api/v1/translate
{
  "text": "Welcome to our mission",
  "source_lang": "en",
  "target_lang": "es"
}
```

Example Response:

```
{
  "task_id": "uuid",
  "translation_id": "uuid",
  "translated_text": "Bienvenido a nuestra misión",
  "quality_score": 0.85,
  "status": "completed"
}
```

3. Image Generation (4 endpoints)

Tag: image-generation

Method	Endpoint	Description	Auth Required
POST	/images/generate	Generate image from prompt	✓
POST	/images/variations	Generate image variations	✓
GET	/images/{id}	Get generated image	✓
GET	/images	List generated images	✓

Supported Sizes: 256x256 , 512x512 , 1024x1024 , 1024x1792 , 1792x1024

Example Request:

```
POST /api/v1/images/generate
{
  "prompt": "Children playing in a new school playground",
  "size": "1024x1024",
  "style": "realistic"
}
```

4. Content Enhancement (6 endpoints)

Tag: content-enhancement

Method	Endpoint	Description	Auth Required
POST	/enhance/grammar	Fix grammar and spelling	✓
POST	/enhance/tone	Adjust content tone	✓
POST	/enhance/seo	Optimize for SEO	✓
POST	/enhance/summarize	Summarize content	✓
POST	/enhance/improve	General improvement	✓
POST	/enhance/batch	Batch enhance content	✓

Example Request:

```
POST /api/v1/enhance/grammar
{
  "text": "This text has some grammatical errors."
}
```

Example Response:

```
{
  "task_id": "uuid",
  "original_text": "This text has some grammatical errors.",
  "enhanced_text": "This text has some grammatical errors.",
  "enhancement_type": "grammar",
  "changes_made": ["Fixed spelling: grammatical -> grammatical"],
  "status": "completed"
}
```

5. Automation (7 endpoints)

Tag: automation

Method	Endpoint	Description	Auth Required
POST	/automation/work-flows	Create workflow	✓
GET	/automation/work-flows/{id}	Get workflow	✓
GET	/automation/work-flows	List workflows	✓
PUT	/automation/work-flows/{id}	Update workflow	✓
DELETE	/automation/work-flows/{id}	Delete workflow	✓
POST	/automation/work-flows/{id}/trigger	Trigger workflow	✓
GET	/automation/work-flows/{id}/history	Get workflow history	✓

Workflow Types:

- auto_translate - Automatically translate content
- scheduled_post - Schedule social media posts
- Custom workflows

Example Request:

```
POST /api/v1/automation/workflows
{
  "name": "Auto-translate new articles",
  "workflow_type": "auto_translate",
  "config": {
    "target_languages": [ "es", "fr", "pt" ]
  },
  "schedule": "0 9 * * *"
}
```

6. AI Tasks (6 endpoints)

Tag: ai-tasks

Method	Endpoint	Description	Auth Required
GET	/tasks/{id}	Get task details	✓
GET	/tasks	List tasks	✓
POST	/tasks/{id}/approve	Approve task	✓
POST	/tasks/{id}/reject	Reject task	✓
DELETE	/tasks/{id}	Delete task	✓
GET	/tasks/statistics	Get task statistics	✓

Task Types:

- content_generation
- translation
- image_generation
- content_enhancement
- automation

Task Statuses:

- pending
- processing
- completed
- failed
- cancelled

7. Content Templates (7 endpoints)

Tag: content-templates

Method	Endpoint	Description	Auth Required
POST	/templates	Create template	✓
GET	/templates/{id}	Get template	✓
GET	/templates	List templates	✓
PUT	/templates/{id}	Update template	✓
DELETE	/templates/{id}	Delete template	✓
POST	/templates/{id}/test	Test template	✓
GET	/templates/suggestions	Get template suggestions	✓

Template Variables:

Templates support variable placeholders in the format `{variable_name}`.

Example Template:

```
POST /api/v1/templates
{
  "name": "Donor Thank You",
  "template_type": "donor_letter",
  "description": "Standard donor thank you template",
  "prompt_template": "Write a thank you letter to {donor_name} for their ${donation_amount} donation to {campaign_name}. Mention this impact: {impact_story}",
  "language": "en",
  "platform": null
}
```

8. Generated Content (6 endpoints)

Tag: generated-content

Method	Endpoint	Description	Auth Required
GET	/content/{id}	Get generated content	✓
GET	/content	List generated content	✓
PUT	/content/{id}	Update content	✓
DELETE	/content/{id}	Delete content	✓
POST	/content/{id}/publish	Publish content	✓
GET	/content/statistics	Get content statistics	✓

Content Types:

- social_post
- article
- story
- donor_letter
- newsletter
- prayer_request
- campaign_copy

Service Integration

External Services

The AI Service integrates with other Mission Engadi microservices:

- 1. Content Service (Port 8003)**
 - Publish generated content
 - Store final content
- 2. Social Media Service (Port 8007)**
 - Publish social media posts
 - Schedule posts
- 3. Notification Service (Port 8008)**
 - Send generated newsletters
 - Email notifications
- 4. Partners CRM Service (Port 8005)**
 - Personalize content with partner data
- 5. Projects Service (Port 8006)**
 - Generate project-specific content
 - Impact stories

Authentication

All endpoints (except health checks) require JWT authentication:

```
Authorization: Bearer <jwt_token>
```

The token should contain:

- `sub` : User ID
- `email` : User email
- `roles` : User roles (optional)

Configuration

Environment Variables

Key configuration variables in `.env` :

```
# Abacus.AI
ABACUS_AI_DEPLOYMENT_TOKEN="your-deployment-token"
ABACUS_AI_DEPLOYMENT_ID="your-deployment-id"

# Service URLs
CONTENT_SERVICE_URL="http://localhost:8003"
SOCIAL_MEDIA_SERVICE_URL="http://localhost:8007"
NOTIFICATION_SERVICE_URL="http://localhost:8008"
PARTNERS_CRM_SERVICE_URL="http://localhost:8005"
PROJECTS_SERVICE_URL="http://localhost:8006"

# AI Settings
AI_DEFAULT_MODEL="gpt-4"
AI_MAX_TOKENS=2000
AI_TEMPERATURE=0.7
```

Database Models

1. AITask

Tracks AI processing tasks with approval workflow.

2. GeneratedContent

Stores AI-generated content with metadata.

3. ContentTemplate

Reusable templates with variable placeholders.

4. TranslationJob

Tracks translation tasks and quality scores.

Error Handling

Standard HTTP status codes:

- 200 - Success
 - 201 - Created
 - 204 - No Content
 - 400 - Bad Request
 - 401 - Unauthorized
 - 403 - Forbidden
 - 404 - Not Found
 - 500 - Internal Server Error
 - 501 - Not Implemented
-

Development

Running the Service

```
cd ai_service
uvicorn app.main:app --reload --port 8010
```

API Documentation

- **Swagger UI:** <http://localhost:8010/docs>
- **ReDoc:** <http://localhost:8010/redoc>

Running Tests

```
pytest tests/
```

Future Enhancements

1. Complete implementation of placeholder endpoints
 2. Add image generation integration
 3. Implement batch processing
 4. Add caching layer for translations
 5. Implement webhook notifications
 6. Add content versioning
 7. Implement A/B testing for generated content
-

Support

For questions or issues, contact the development team or refer to the main project documentation.