

LLM Mentor Backend

API Documentation

FastAPI Application - 16 Endpoints

Framework: FastAPI (Async)

Default LLM: DeepSeek (configurable per request)

CORS: Enabled for all origins

RAG Integration: SearchRagManager for enhanced responses

Common Request Parameters:

- model_provider (string, default: "deepseek")
- model_name (string, default: "deepseek-chat")

1. [GET] /list-llm-models

Lists all available LLM models configured in the application.

Inputs:

None

Outputs:

```
{"models": [{"model_name": "string", "model_provider": "string"}]}
```

2. [POST] /chat-with-tutor

Chat with AI tutor to answer learner questions and provide personalized tutoring using RAG-enhanced responses.

Inputs:

- messages (string, required): JSON array of chat messages
- learner_profile (string, optional): Learner profile info
- model_provider (string): LLM provider (default: deepseek)
- model_name (string): Model name (default: deepseek-chat)

Outputs:

```
{"response": "tutor response string"}
```

3. [POST] /refine-learning-goal

Refine and improve a learning goal based on learner information to make it more specific and actionable.

Inputs:

learning_goal (string, required): Initial learning goal
learner_information (string, optional): Info about the learner
model_provider, model_name (strings): LLM config

Outputs:

Refined learning goal object with improved specificity

4. [POST] /identify-skill-gap-with-info

Identify skill gaps based on learning goal and learner information provided as text.

Inputs:

learning_goal (string, required): Target learning goal
learner_information (string, required): Learner background
skill_requirements (string, optional): Pre-identified requirements
model_provider, model_name (strings): LLM config

Outputs:

Combined skill gaps and skill requirements object

5. [POST] /identify-skill-gap

Identify skill gaps by uploading a CV/resume PDF file. Extracts text from PDF and analyzes against the learning goal.

Inputs:

- goal (string, form): Learning goal
- cv (file, form): PDF file of learner's CV
- model_provider, model_name (strings, form): LLM config

Outputs:

```
Combined skill gaps and skill requirements with cv_path
```

6. [POST] /create-learner-profile-with-info

Initialize a learner profile from text information including background, goals, and identified skill gaps.

Inputs:

- learning_goal (string, required): Learner's goal
- learner_information (string, required): Background info
- skill_gaps (string, required): Identified skill gaps
- model_provider, model_name (strings): LLM config

Outputs:

```
{"learner_profile": {...profile object}}
```

7. [POST] /create-learner-profile

Initialize learner profile from an uploaded CV file path. Extracts text from PDF at specified path.

Inputs:

- learning_goal (string, required): Learner's goal
- skill_requirements (string, required): Required skills
- skill_gaps (string, required): Identified gaps
- cv_path (string, required): Path to stored CV
- model_provider, model_name (strings): LLM config

Outputs:

```
{"learner_profile": {...profile object}}
```

8. [POST] /update-learner-profile

Update learner profile based on interactions, session information, and behavioral data.

Inputs:

- learner_profile (string, required): Current profile
- learner_interactions (string, required): Interaction data
- learner_information (string, optional): Updated info
- session_information (string, optional): Session details
- model_provider, model_name (strings): LLM config

Outputs:

```
{"learner_profile": {...updated profile}}
```

9. [POST] /schedule-learning-path

Generate a personalized learning path schedule with specified number of sessions.

Inputs:

- learner_profile (string, required): Learner profile
- session_count (integer, required): Number of sessions
- model_provider, model_name (strings): LLM config

Outputs:

Learning path object with scheduled sessions

10. [POST] /reschedule-learning-path

Adjust existing learning path based on feedback, progress, and new requirements.

Inputs:

- learner_profile (string, required): Learner profile
- learning_path (string, required): Current path
- session_count (integer, optional): New count (-1 = unchanged)
- other_feedback (string, optional): Additional feedback
- model_provider, model_name (strings): LLM config

Outputs:

Updated learning path object

11. [POST] /explore-knowledge-points

Explore and identify knowledge points for a specific learning session.

Inputs:

- learner_profile (string, required): Learner profile
- learning_path (string, required): Learning path
- learning_session (string, required): Current session

Outputs:

List of knowledge points to explore

12. [POST] /draft-knowledge-point

Draft learning content for a single knowledge point with optional RAG search enhancement.

Inputs:

- learner_profile (string, required): Learner profile
- learning_path (string, required): Learning path
- learning_session (string, required): Session info
- knowledge_points (string, required): All points
- knowledge_point (string, required): Point to draft
- use_search (boolean, required): Use RAG search

Outputs:

{"knowledge_draft": {...draft content}}

13. [POST] /draft-knowledge-points

Draft learning content for multiple knowledge points with optional parallelization.

Inputs:

- learner_profile (string, required): Learner profile
- learning_path (string, required): Learning path
- learning_session (string, required): Session info
- knowledge_points (string, required): Points to draft
- use_search (boolean, required): Use RAG search
- allow_parallel (boolean, required): Parallelize drafting

Outputs:

```
{"knowledge_drafts": [...draft contents]}
```

14. [POST] /integrate-learning-document

Integrate drafted knowledge content into a cohesive learning document.

Inputs:

- learner_profile (string, required): Learner profile
- learning_path (string, required): Learning path
- learning_session (string, required): Session info
- knowledge_points (string, required): Knowledge points
- knowledge_drafts (string, required): Drafted content
- output_markdown (boolean, optional): Markdown output

Outputs:

```
{"learning_document": {...integrated doc}}
```

15. [POST] /generate-document-quizzes

Generate quizzes based on learning document content with configurable question types.

Inputs:

- learner_profile (string, required): Learner profile
- learning_document (string, required): Content for quizzes
- single_choice_count (int, default=3)
- multiple_choice_count (int, default=0)
- true_false_count (int, default=0)
- short_answer_count (int, default=0)

Outputs:

```
{"document_quiz": {...quiz object}}
```

16. [POST] /tailor-knowledge-content

Create comprehensive personalized learning content with optional quizzes. Combines multiple steps into one endpoint.

Inputs:

- learner_profile (string, required): Learner profile
- learning_path (string, required): Learning path

learning_session (string, required): Session info
use_search (boolean, default=true): RAG enhancement
allow_parallel (boolean, default=true): Parallelize
with_quiz (boolean, default=true): Include quiz

Outputs:

```
{"tailored_content": {...complete content}}
```

Common Response Codes

Code	Description
200 OK	Successful response with JSON data
400 Bad Request	Validation error (e.g., invalid message format)
500 Internal Server Error	{"detail": "Error message describing what went wrong"}