About this API

What is the platform

This is an AI software platform, which exposes some of the functionalities to API clients. The platform consists of a web application with various AI modules. The API enables regular LLM chatting with several available AI models, Tailored AI and Knowledge base management, and use for Retrieval-Augmented Generation over documents, and management endpoints.

WARNING: This API is stateless and does not store conversation history; your application is solely responsible for managing all state and context. You must store each user's conversation history separately within your application. It is your critical responsibility to implement strict data isolation to prevent users from seeing each other's conversations.

How API authentication works

API clients are managed through the web application by administrators with the MANAGE_API_CLIENTS permission. This process involves configuring the client's properties and generating secrets for authentication. For each API client, an administrator can configure the following attributes:

- **Name:** A unique identifier for the client.
- **Cost Limit:** The maximum cost allowed for a given period.
- **Rate Limit:** The maximum number of requests allowed per minute.
- **Role:** The permission level assigned to the client.
- **Responsible Entity:** The team or individual responsible for the client.
- **Comment:** An optional field for notes.

To authenticate, your application must include the following headers in every API request:

- ***X-Client-ID:*** The public identifier for the API client.
- ***X-Client-Secret:*** The secret credential used to prove ownership.

You can generate up to **two** client secrets per API client to enable seamless, zero-downtime secret rotation. This allows you to introduce a new secret, deploy it to your application, and then safely deactivate the old one without any service interruption. This practice is highly recommended for maintaining security without scheduling downtime.

Notes on backwards compatibility

The current version API endpoints are prefixed with /api/v1. Future API versions will be prefixed similarly, following the /api/v2 version. To achieve backwards compatibility, when a new version of the API is introduced, the previous version will remain available until the clients have switched.

API endpoint clusters

The API endpoints are grouped into 4 clusters:

- Cost: with the /api/v1/cost prefix, requires ACCESS_API_MONITOR_CREDIT permission
- LLM: with the /api/v1/llm prefix, requires USE_API_LLM permission
- Tailored AI: with the /api/v1/tailored-ai prefix, requires USE_API_TAILORED_AI permission
- Knowledge Base: with the /api/v1/knowledge-base prefix, requires USE_API_TAILORED_AI permission

The API role system works similarly. Regular LLM role access enables only the basic AI model usage, and not the RAG feature. Tailored AI use enables both LLM use, managing Tailored AIs, and Knowledge Bases for Retrieval-Augmented Generation AI chatting. The cost endpoint returns a list of user and API client cost usage.

Cost endpoint

Retrieve aggregated API cost data for users and API clients

GET https://api.example.com/api/v1/cost

This endpoint returns a unified export of cost information for all users and API clients associated with the authenticated API client. The data returned includes individual cost summaries as well as aggregated totals for the entire result set. Authentication via API client credentials is required, and the caller must have the ACCESS_API_MONITOR_CREDIT permission. Date filtering is performed using query parameters.

Query Parameters:

- fromDate(date, optional): Start date of the cost range (inclusive), formatted as 'yyyy-mm-dd'. If not provided, the backend may return all available historical cost entries.
- toDate(date, optional): End date of the cost range (inclusive of the full day), formatted as 'yyyy-mm-dd'. Must not be earlier than fromDate. If omitted, defaults to the current date.

Returns

Response

JSON response containing:

- costs: List of user or API client objects with cost details
- totalClients: Number of returned entities
- totalCost: Sum of visible totalCost values across all entities

Response Structure:

```
{
   "costs": [
       {
           "id": "149c6036-e2dc-40c2-ae88-f5593f21a855",
           "name": "John Doe",
           "email": "john.doe@example.com",
           "freeCredit": 100.0,
           "totalCost": 57.25,
           "clientType": "USER"
       },
       {
           "id": "bbbbefdc-7a1c-49d7-b0f8-3e6007a811d9",
           "name": "Integration Client",
           "email": null,
           "freeCredit": 5000.0,
           "totalCost": 0.0,
           "clientType": "API_CLIENT"
       },
       ...
   ],
   "totalClients": 18,
   "totalCost": 3421.75
}
```

Notes on Cost Visibility:

- If the environment variable `ADMIN_CAN_VIEW_USER_COSTS` is disabled, the `totalCost` field will be returned as 0.0 for all entries.
- Users or clients with no recorded usage in the specified range will also show totalCost: 0.0.
- Cost values are aggregated and formatted via the APICostService.

Errors:

- 400 Bad Request: If toDate is before fromDate, or if date parameters are invalid or improperly formatted.
- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the API client lacks the ACCESS_API_MONITOR_CREDIT permission.
- 500 Internal Server Error: If a database error occurs during cost data retrieval or processing.

Notes:

- This is a GET endpoint that accepts query parameters

- Dates should be in ISO 8601 format (yyyy-mm-dd)
- The endpoint includes the entire day of the toDate (up to 23:59:59)
- Users with no usage in the date range will show totalCost: 0.0
- This endpoint returns both user cost entries and API client cost entries in a unified, normalized format.

LLM endpoints

List models endpoint

GET https://api.example.com/api/v1/llm/models

List all available language models for API clients.

This endpoint returns a detailed list of supported models that can be used for chat completions. Each entry contains metadata such as model name, token limits, and pricing. Requires API client authentication and the `USE_API_LLM` permission.

Returns

Response

JSON response containing:

- `data`: List of supported models with metadata and pricing
- `message`: Success message
- `method`: HTTP method used
- `path`: Request path
- `success`: Boolean indicating operation success
- `timestamp`: UTC timestamp of the response

Response Structure:

```
{
   "data": [
       {
           "contextWindow": 200000,
           "costs": [
               {"cost": 0.00000125, "costType": "input_tokens"},
               {"cost": 0.00000125, "costType": "output_tokens"}
           ],
           "modelName": "gpt-5-large",
           "outputTokenLimit": 100000
       },
       ...
   ],
   "message": "Models list retrieved successfully",
   "method": "GET",
   "path": "/api/v1/llm/models",
   "success": true,
   "timestamp": "2025-12-10T09:09:58.672269+00:00Z"
}
```

Field Descriptions:

- **contextWindow** *(int)*: Maximum number of tokens the model can process in a single request.
- **costs** *(list)*: Pricing information for input and output tokens.
- `cost.costType`: Either `"input_tokens"` or `"output_tokens"`.
- `cost.cost`: Price per token in USD; Usual pricing is 1.25 USD per 1 Million Tokens.
- **modelName** *(str)*: Canonical identifier of the model.
- **outputTokenLimit** *(int)*: Maximum number of tokens that can be generated in a response.
- **timestamp** *(str, ISO8601)*: Server timestamp of the response.

Errors:

- 401 Unauthorized: API client authentication fails
- 403 Forbidden: Client lacks required permissions
- 500 Internal Server Error: Internal service error

Notes:

- The model list is dynamically generated based on available providers and configurations.
- Costs are expressed per token in USD and may change over time.
- Some models may support extended token contexts and specialized capabilities.
- Use this endpoint to populate model selectors or validate model IDs before making requests.
- Some models may be restricted depending on the client's permissions or subscription.

Chat endpoint

POST https://api.example.com/api/v1/llm/chat

Create a chat completion with optional Retrieval-Augmented Generation (RAG) support. Streaming is not supported and there is no option to enable streaming.

This unified endpoint generates chat completions using supported large language models (LLMs) such as OpenAI (gpt-4o, gpt-5, o3-mini) or Mistral (mistral-large-2411). It automatically detects when to use Retrieval-Augmented Generation (RAG) — if a `tailored_ai_id` is provided and the tailored AI has a valid knowledge base — to enrich model responses with contextual knowledge from your organization's indexed data sources (e.g., Azure Cognitive Search). When RAG is enabled, the system prompt defined for the tailored AI is always used to guide the model's behavior and tone. Even if no knowledge base is connected to the tailored AI, the configured system prompt will still be applied.

Request Body:

```
{
   // required
   "model": "gpt-4o",

   // required
   "messages": [
     {"role": "user", "content": "Let me know more about the libraries of the Austrian engineering universities."}
   ],

   // The temperature is scaled proportionally from OpenAI's range (0.0-2.0) to Mistral's range (0.0-0.7)
   // optional, float between the 0.0 and 2.0 range, supported by all models
   "temperature": 0.7,

   // Specify the maximum length of the model's generated response, measured in tokens
   // optional, Used for: Standard OpenAI models (gpt-4o, gpt-4o-mini) and Mistral models
   "maxTokens": 1000,

   // Set a hard limit on the number of tokens the model will generate in its response
   // optional, Used for: OpenAI reasoning models (o1, o3, o3-mini, gpt-5, gpt-5-mini, gpt-5-nano)
   "maxCompletionTokens": 4000,

   // optional, triggers RAG mode
   "tailoredAiId": "uuid-here",

   // Penalizes the repetition of words in the generated text
   // optional, float between the -2.0 and 2.0 range, supported by all models
   "frequencyPenalty": 0.5,

   // A higher presence_penalty encourages the model to use a wider variety of words and topics
   // optional, float between the -2.0 and 2.0 range, supported by all models
   "presencePenalty": 0.3,

   // Stop sequences which force the model to stop generating text
   // optional, string or array of strings, Mistral only
   "stop": ["###"],

   // optional, integer
   // Used for: Standard OpenAI models (gpt-4o, gpt-4o-mini) and Mistral models
   "seed": 42,

   // optional, {"type": "text"} or {"type": "json_object"}
```

```
   // Used for: Standard OpenAI models (gpt-4o, gpt-4o-mini)
   "responseFormat": {"type": "text"},

   // Controls how much "thinking" (hidden reasoning tokens) the AI does before giving a final answer
   // optional, "low", "medium", or "high"
   // (reasoning models only)
   "reasoningEffort": "medium",

   // Allows specifying the desired length and detail of the model's response without rewriting the prompt
   // optional, "low", "medium", or "high"
   // (reasoning models only)
   "verbosity": "high"
}
```

Behavior:

- RAG mode is only enabled if tailoredAiId is provided and the tailored AI has a valid knowledge base
- If tailoredAiId is provided but there's no knowledge base, RAG is not enabled (system prompt is still applied, but no citations)

Returns:

```
{
 "data": {
   "model": "gpt-4o",
   "role": "assistant",
   "content": "Here's a summary of your Q2 report...",
   "finishReason": "stop",
   "usage": {
     "promptTokens": 152,
     "completionTokens": 74,
     "totalTokens": 226
   },
   // RAG mode only
   "citations": [
     {
         "index": 1,
         "source": {
             "documentId": "bff94a7d-c41b-4db6-8471-70e91db19a52",
             "documentName": "Document Name.pdf",
             "knowledgeBaseId": "8083de83-ce52-43e0-bbe6-234ef7d7dbcf",
             "pageContent": "Document Content ...",
             "pageNumber": 2
         },
         "sourceType": "document"
     },
     ...
   ]
 },
 "message": "Chat completion created successfully",
 ...
}
```

Errors

- **400 Bad Request** — Invalid model or unsupported parameters
- **401 Unauthorized** — Missing or invalid API credentials
- **403 Forbidden** — Insufficient permissions (e.g., using Tailored AI without permission)
- **404 Not Found** — Tailored AI knowledge base not found or unavailable
- **422 Unprocessable Entity** — Validation error
- **500 Internal Server Error** — Internal processing failure

Knowledge Base endpoints

List all Knowledge Bases for the client

GET https://api.example.com/api/v1/knowledge-base/

Fetches a list of all Knowledge Base instances owned by the authenticated API client. Requires the `USE_API_TAILORED_AI` permission.

Returns

Response

A JSON response containing a list of Knowledge Base objects.

Response Structure (200 OK):

```
{
   "data": [
       {
           "id": "kb-uuid-1",
           "name": "Product Documentation",
           "state": "ready",
           ...
       },
       {
           "id": "kb-uuid-2",
           "name": "Internal Policies",
           "state": "processing",
           ...
       }
   ]
}
```

Errors:

- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks the `USE_API_TAILORED_AI` permission.

Notes:

- This is a GET endpoint with no parameters required
- Returns all Knowledge Bases owned by the authenticated API client (Knowledge Bases with state:TO_DELETE are excluded)
- The response includes complete details for each Knowledge Base including documents

Create a new, empty Knowledge Base

POST https://api.example.com/api/v1/knowledge-base/

This endpoint creates a new Knowledge Base for the authenticated API client. The Knowledge Base is initially empty. Files must be uploaded to it and then the ingestion process must be triggered to make the content available.

Request Body:

```
{
   "name": "New Product Documentation"
}
```

- The `name` field is required and must be between 3 and 50 characters in length.
- The name can contain any Unicode characters.

Returns

Response

A JSON response containing the details of the newly created Knowledge Base.

Response Structure (201 Created):

```
{
   "data": {
       "id": "new-kb-uuid",
       "name": "New Product Documentation",
       "state": "created",
       ...
   }
}
```

Errors:

- 400 Bad Request: If the 'name' field is missing or invalid.
- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks the `USE_API_TAILORED_AI` permission.

Notes:

- This is a POST endpoint that requires a JSON request body
- The `name` field is required
- The Knowledge Base is created in an empty state
- Files must be uploaded and ingestion triggered to make content available

Get a specific Knowledge Base

GET https://api.example.com/api/v1/knowledge-base/{knowledge_base_id}

Retrieve the details of a single Knowledge Base instance, identified by its UUID. The endpoint verifies that the requested Knowledge Base belongs to the authenticated API client.

Returns

Response

A JSON response with the requested Knowledge Base's data including its documents.

Response Structure (200 OK):

```
{
   "data": {
       "id": "kb-uuid-1",
       "name": "Product Documentation",
       "state": "ready",
       "documents": [
           {
               "id": "doc-uuid-1",
               "name": "manual.pdf",
               "sizeBytes": 1024000,
               "pageCount": 50,
               "lastUpdated": 1700000000.0,
               "indexed": true
           }
       ],
       ...
   },
   "message": "Knowledge Base retrieved successfully"
}
```

Note on `lastUpdated` field:

The `lastUpdated` field is a Unix timestamp (seconds since January 1, 1970 UTC) represented as a floating-point number.

Errors:

- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks the `USE_API_TAILORED_AI` permission.
- 404 Not Found: If no Knowledge Base with the given ID exists or it does not belong to the client.

Notes:

- This is a GET endpoint that requires a valid UUID in the URL path
- Only returns Knowledge Bases owned by the authenticated API client
- Returns 404 if the Knowledge Base doesn't exist or doesn't belong to the client
- The response includes the list of documents in the Knowledge Base

Upload a file to a Knowledge Base

POST https://api.example.com/api/v1/knowledge-base/{knowledge_base_id}/files

This endpoint accepts a file upload (`multipart/form-data`) and associates it with the specified Knowledge Base. After uploading, the ingestion process must be triggered to make the file's content searchable. Files are not ingested automatically after an upload. After the files are uploaded to a Knowledge Base, manually start the ingestion process by using the ingest endpoint.

Request Body:

- Content-Type: multipart/form-data
- A form field named "file" containing the binary data of the file.

Returns

Response

A JSON response with details of the created document record.

Response Structure (201 Created):

```
{
   "data": {
       "id": "new-document-uuid",
       "name": "uploaded_file_name.pdf"
   },
   "message": "File uploaded successfully.",
   ...
}
```

Errors:

- 400 Bad Request: If the 'file' part is missing, no file is selected, or a file exceeding 50 MB is selected (last error is raised for the /ingest call).
- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks `USE_API_TAILORED_AI` permission.
- 404 Not Found: If no Knowledge Base with the given ID exists for the client.

Notes:

- This is a POST endpoint that accepts multipart/form-data
- The file must be sent as a form field named "file"
- After uploading, trigger ingestion to make the content searchable
- Supported file types include PDF, DOC, DOCX, TXT, and other text formats

Delete a file from a Knowledge Base

DELETE https://api.example.com/api/v1/knowledge-base/{knowledge_base_id}/files/{document_id}

Permanently removes a file (Document record) from a Knowledge Base. The endpoint verifies ownership of the Knowledge Base before deletion.

Returns

Response

A JSON response with a success message and 200 OK status on successful deletion.

Errors:

- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks `USE_API_TAILORED_AI` permission.
- 404 Not Found: If the Knowledge Base or Document does not exist for the client.

After Deletion:

Re-ingestion Required:

After deleting a document, you must call the `/ingest` endpoint to remove it from the search index. The DELETE operation only:

- Removes the file from blob storage
- Marks the document as deleted in the database

The search index cleanup happens during the ingestion process. Until ingestion is triggered and completed, the deleted document may still appear in search results when using the Knowledge Base in chat completions.

Recommended Workflow:

1. Delete the document using this endpoint
2. Call `/ingest` to trigger ingestion
3. Monitor the ingestion status using `/status` until the Knowledge Base reaches `ready` state

Notes:

- This is a DELETE endpoint that permanently removes a file from a Knowledge Base
- Returns 200 OK with a success message on successful deletion
- Only allows deletion of files in Knowledge Bases owned by the authenticated API client
- This action cannot be undone

Trigger the ingestion process for a Knowledge Base

POST https://api.example.com/api/v1/knowledge-base/{knowledge_base_id}/ingest

Starts the process of indexing all uploaded and unprocessed files in a Knowledge Base. This is an asynchronous operation; use the `/status` endpoint to track progress.

If you try to ingest a knowledge base without uploading any files, the ingestion process will fail.

Returns

Response

A JSON response with a confirmation message.

Response Structure (202 Accepted):

```
{
 "message": "Ingestion process for knowledge base [ID] has been started.",
 ...
}
```

Errors:

- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks `USE_API_TAILORED_AI` permission.
- 404 Not Found: If no Knowledge Base with the given ID exists for the client.

Usage After Ingestion:

Knowledge Base State Requirements:

Before a Knowledge Base can be used in chat completions (via the `/chat/completions` endpoint), it must be in the `ready` state, or in `enqueued` state with `last_synchronized` set (indicating it was previously ingested). Attempting to use a Knowledge Base that hasn't completed ingestion will result in a `KB_UNAVAILABLE` error (error code 603).

Document Availability:

- Only documents that have been successfully indexed during the ingestion process will be available for retrieval during chat completions.

- If no documents are indexed, chat completions will still execute but will generate responses without any retrieved context from the Knowledge Base.
- Unindexed documents (those that failed validation, weren't processed, or are still pending) will not appear in search results.

Monitoring Ingestion:

Use the `/status` endpoint to check the Knowledge Base state and monitor ingestion progress. The Knowledge Base state will transition from `enqueued` → `processing` → `ready` upon successful completion.

Notes:

- This is a POST endpoint that starts an asynchronous ingestion process
- Returns 202 Accepted to indicate the process has been started
- Use the `/status` endpoint to monitor progress
- Only one ingestion process can run per Knowledge Base at a time

Get the ingestion status of a Knowledge Base

GET https://api.example.com/api/v1/knowledge-base/{knowledge_base_id}/status

Checks and returns the detailed status of a Knowledge Base, including information about the ingestion progress for its associated files. This is useful for polling after triggering an ingestion.

Returns

Response

A JSON response with the detailed status information.

Response Structure (200 OK):

```
{
   "data": {
       "errors": [],
       "id": "d627cf2e-960d-4961-9f2a-5b451cfc7f8a",
       "name": "example_string",
       "progress": null,
       "state": "created"
   },
   ...
}
```

Status Values:

The `state` field indicates the current ingestion status of the Knowledge Base. Possible values are:

- **`created`**: The Knowledge Base has been created but ingestion has not been triggered yet
- **`enqueued`**: Ingestion has been queued and is waiting to be processed
- **`preparing`**: The system is preparing to start the ingestion process
- **`processing`**: Ingestion is currently in progress
- **`ready`**: Ingestion has completed successfully and the Knowledge Base is ready for use
- **`failed`**: Ingestion has failed (check the `errors` array for details)
- **`to_delete`**: The Knowledge Base is marked for deletion

Progress Field:

The `progress` field is a floating-point number between 0.0 and 1.0 indicating the completion percentage of the ingestion process. A value of 1.0 indicates completion.

Errors Field:

The `errors` array contains error objects for any documents that failed during ingestion. Each error object includes:

- `documentId`: The ID of the document that encountered an error
- `knowledgeBaseId`: The ID of the Knowledge Base
- `errorMessage`: A human-readable error message describing what went wrong

Errors:

- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks `USE_API_TAILORED_AI` permission.
- 404 Not Found: If no Knowledge Base with the given ID exists for the client.

Notes:

- This is a GET endpoint that returns detailed ingestion status
- Useful for polling after triggering an ingestion process
- Returns status for each individual file in the Knowledge Base
- Overall status indicates the completion state of the entire Knowledge Base

Delete a specific Knowledge Base

DELETE https://api.example.com/api/v1/knowledge-base/{knowledge_base_id}

Asynchronously and permanently removes a Knowledge Base and all its associated files and data. The endpoint verifies that the Knowledge Base is owned by the authenticated API client.

Returns

Response

A JSON response with a success message and 200 OK status on successful deletion.

Errors:

- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks `USE_API_TAILORED_AI` permission.
- 404 Not Found: If no Knowledge Base with the given ID exists for the client.

Notes:

- This is a DELETE endpoint that permanently removes the Knowledge Base
- Returns 200 OK with a success message on successful deletion
- Only allows deletion of Knowledge Bases owned by the authenticated API client
- This action cannot be undone and removes all associated files and data

Tailored AI endpoints

Create Tailored AI endpoint

POST https://api.example.com/api/v1/tailored-ai/

Create a new Tailored AI instance.

This endpoint allows an authenticated API client to create a new, fully configured Tailored AI. The client must possess the `USE_API_TAILORED_AI` permission.

Request Body:

```
{
   "name": "Customer Support Assistant",
```

```
   "summary": "AI to help with common customer questions.",
   "systemPrompt": "You are a friendly and helpful customer support assistant."
}
```

Body

- name(str, required, max_length=50): The name of the Tailored AI.
- summary(str, required, max_length=100): A brief summary of the AI's purpose.
- systemPrompt(str, required, max_length=2000): The core instructions defining the AI's persona and task.

Returns

Response

A JSON response containing the full details of the newly created Tailored AI.

Response Structure (201 Created):

```
{
   "data": {
       "createdAt": 1765184807.56,
       "id": "7c26da0a-71cc-4e8f-8773-9937dda8de15",
       "knowledgeBases": [],
       "name": "Customer Support Assistant",
       "systemPrompt": "You are a friendly and helpful customer support assistant.",
       "summary": "AI to help with common customer questions.",
       "updatedAt": 1765184807.56
   },
   ...
}
```

Errors:

- 400 Bad Request: If request body validation fails (e.g., missing required fields).
- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks the `USE_API_TAILORED_AI` permission.

Notes:

- This is a POST endpoint that requires a JSON request body
- The `name`, `summary`, and `systemPrompt` fields are required
- The response includes the complete AI configuration with generated ID

To create a Tailored AI that uses documents from a Knowledge Base, follow this workflow:

1. **Create Knowledge Base**: `POST /api/v1/knowledge-base/` - Create an empty Knowledge Base
2. **Upload Files**: `POST /api/v1/knowledge-base/{kb_id}/files` - Upload one or more files (repeat as needed)
3. **Trigger Ingestion**: `POST /api/v1/knowledge-base/{kb_id}/ingest` - Start the indexing process
4. **Wait for Ingestion**: Poll `GET /api/v1/knowledge-base/{kb_id}/status` until `state` is `ready`
5. **Create Tailored AI**: `POST /api/v1/tailored-ai/` - Create the Tailored AI (can be done in parallel with steps 1-4)
6. **Connect Knowledge Base**: `POST /api/v1/tailored-ai/{tai_id}/knowledge-base` - Link the Knowledge Base to the Tailored AI

Important: The Knowledge Base must be in `ready` state before connecting it to a Tailored AI. Attempting to use a Knowledge Base that hasn't completed ingestion will result in a `KB_UNAVAILABLE` error when using the Tailored AI in chat completions.

List Tailored AI endpoint

GET https://api.example.com/api/v1/tailored-ai/

List all Tailored AIs for the client.

Retrieves a list of all Tailored AI instances that have been created by and belong to the currently authenticated API client. Requires the `USE_API_TAILORED_AI` permission.

Returns

Response

A JSON response containing a list of Tailored AI objects.

Response Structure (200 OK):

```
{
   "data": [
       {
           "id": "your-tailored-ai-id-uuid-here",
           "name": "Customer Support Assistant",
           "summary": "AI to help with common customer questions.",
           ...
       },
       {
           "id": "b2c3d4e5-f6a7-8901-2345-67890abcdef1",
           "name": "Marketing Copy Generator",
           "summary": "Generates creative marketing copy.",
           ...
       }
   ],
   "message": "Tailored AIs retrieved successfully",
   ...
}
```

Errors:

- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks the `USE_API_TAILORED_AI` permission.

Notes:

- This is a GET endpoint with no parameters required
- Returns all Tailored AIs owned by the authenticated API client
- The response includes complete details for each AI

Get Tailored AI endpoint

GET https://api.example.com/api/v1/tailored-ai/{tailored_ai_id}

Retrieve a specific Tailored AI.

Fetches the details of a single Tailored AI instance, identified by its UUID. The endpoint verifies that the requested AI belongs to the authenticated API client.

Returns

Response

A JSON response with the requested Tailored AI's data.

Response Structure (200 OK):

```
{
   "data": {
       "id": "your-tailored-ai-id-uuid-here",
       "name": "Customer Support Assistant",
       ...
   },
   "message": "Tailored AI retrieved successfully",
   ...
}
```

Errors:

- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks the `USE_API_TAILORED_AI` permission.

- 404 Not Found: If no Tailored AI with the given ID exists or it does not belong to the client.

Notes:

- This is a GET endpoint that requires a valid UUID in the URL path
- Only returns AIs owned by the authenticated API client
- Returns 404 if the AI doesn't exist or doesn't belong to the client

Update Tailored AI endpoint

PUT https://api.example.com/api/v1/tailored-ai/{tailored_ai_id}

Update a specific Tailored AI.

This endpoint modifies an existing Tailored AI. Only the fields provided in the request body will be updated. The endpoint verifies that the AI belongs to the authenticated API client before applying changes.

Request Body:

```
{
   "name": "New AI Name",
   "summary": "Updated summary for the AI.",
   "systemPrompt": "You are an updated AI assistant with new instructions."
}
```

Body

- name(str, optional, max_length=50): The name of the Tailored AI.
- summary(str, optional, max_length=100): A brief summary of the AI's purpose.
- systemPrompt(str, optional, max_length=2000): The core instructions defining the AI's persona and task. Updates the base prompt template.

Returns

Response

A JSON response containing the full data of the updated Tailored AI.

Errors:

- 400 Bad Request: If request body validation fails.
- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks `USE_API_TAILORED_AI` permission.
- 404 Not Found: If no Tailored AI with the given ID exists for the client.

Notes:

- This is a PUT endpoint that accepts partial updates
- Only the fields provided in the request body will be updated
- All other fields remain unchanged
- Returns the complete updated AI configuration
- PUT method is used even though the REST semantics of PATCH is applied (i.e., partial update, not full update)

Delete Tailored AI endpoint

DELETE https://api.example.com/api/v1/tailored-ai/{tailored_ai_id}

Delete a specific Tailored AI.

Permanently removes a Tailored AI instance, identified by its UUID. The endpoint verifies ownership before deletion.

Returns

Response

A JSON response with a success message and 200 OK status on successful deletion.

Errors:

- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks `USE_API_TAILORED_AI` permission.
- 404 Not Found: If no Tailored AI with the given ID exists for the client.

Notes:

- This is a DELETE endpoint that permanently removes the AI
- Returns 200 OK with a success message on successful deletion
- Only allows deletion of AIs owned by the authenticated API client
- This action cannot be undone

Connect a Knowledge Base to a Tailored AI endpoint

POST https://api.example.com/api/v1/tailored-ai/{tailored_ai_id}/knowledge-base

Establishes a link between an existing Tailored AI and an existing Knowledge Base, allowing the AI to use the Knowledge Base as a source of information. Both resources must be owned by the authenticated API client.

The current endpoint sets the associated knowledge base for the tailored AI It allows setting a knowledge base, changing the assigned knowledge base to a different one, or unassigning it (setting to null)

If knowledgeBaseId is set to null, removes the knowledge base from the Tailored AI. If new knowledgeBaseId is passed, it will replace the existing association.

The Tailored AI may only have one Knowledge Base.

Request Body:

```
{
   "knowledgeBaseId": "k_b_uuid_12345"
}
```

Returns

Response

A JSON response with a confirmation message on successful connection.

Errors:

- 400 Bad Request: If the knowledgeBaseId is invalid.
- 401 Unauthorized: If API client authentication fails.
- 403 Forbidden: If the client lacks `USE_API_TAILORED_AI` permission.
- 404 Not Found: If the Tailored AI or Knowledge Base does not exist or is not owned by the client.

Notes:

- This is a POST endpoint that creates a connection between AI and Knowledge Base
- Both the AI and Knowledge Base must be owned by the authenticated API client
- Only one Knowledge Base can be connected per AI