

# 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:

- 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.

# LLM endpoints

---

## List models endpoint

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

### Code Examples:

#### Python

```
import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def list_models(self):
        url = f"{self.base_url}/api/v1/llm/models"
        headers = {
            "X-Client-ID": self.api_key,
            "X-Client-Secret": self.api_secret,
        }

        response = requests.get(url, headers=headers, timeout=30)
        response.raise_for_status()
        return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
models = api.list_models()
for model in models["data"]:
    print(f"Model: {model['modelName']}, Context: {model['contextWindow']}")
```

#### JavaScript

```
class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async listModels() {
    const url = `${this.baseUrl}/api/v1/llm/models`;

    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
    };

    const res = await fetch(url, { method: "GET", headers });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
api
  .listModels()
  .then((response) => {
```

```

response.data.forEach((model) => {
  console.log(`Model: ${model.modelName}, Context: ${model.contextWindow}`);
});
})
.catch(console.error);

```

## cURL

```

# Retrieve list of available models

curl -X GET "https://api.example.com/api/v1/llm/models" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret"

```

## Chat endpoint

**POST** <https://api.example.com/api/v1/llm/chat>

### Code Examples:

#### Python

```

import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def create_chat_completion(self, model, messages, **kwargs):
        url = f"{self.base_url}/api/v1/llm/chat"
        headers = {
            "X-Client-ID": self.api_key,
            "X-Client-Secret": self.api_secret,
            "Content-Type": "application/json",
        }
        payload = {
            "model": model,
            "messages": messages,
            **kwargs,
        }

        response = requests.post(url, headers=headers, json=payload, timeout=30)
        response.raise_for_status()
        return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
response = api.create_chat_completion(
    model="gpt-4o",
    messages=[{"role": "user", "content": "Let me know more about the libraries of the Austrian engineering universities."}],
    temperature=0.7,
    maxTokens=1000,
)
print(f"Response: {response['data']['content']}")
print(f"Tokens used: {response['data']['usage']['totalTokens']}")

```

#### JavaScript

```

class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async createChatCompletion(model, messages, options = {}) {
    const url = `${this.baseUrl}/api/v1/llm/chat`;

    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
      "Content-Type": "application/json",
    };

    const payload = {

```

```

    model,
    messages,
    ...options,
  };

  const res = await fetch(url, {
    method: "POST",
    headers,
    body: JSON.stringify(payload),
  });
  if (!res.ok) throw new Error(`Request failed: ${res.status}`);

  return res.json();
}
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
api
  .createChatCompletion("gpt-4o", [
    { role: "user", content: "Let me know more about the libraries of the Austrian engineering universities." },
  ], {
    temperature: 0.7,
    maxTokens: 1000,
  })
  .then((response) => {
    console.log(`Response: ${response.data.content}`);
    console.log(`Tokens used: ${response.data.usage.totalTokens}`);
  })
  .catch(console.error);

```

## cURL

```

# Create a chat completion

curl -X POST "https://api.example.com/api/v1/llm/chat" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret" \
-H "Content-Type: application/json" \
-d '{ \
  "model": "gpt-4o", \
  "messages": [ \
    { "role": "user", "content": "Let me know more about the libraries of the Austrian engineering universities." } \
  ], \
  "temperature": 0.7, \
  "maxTokens": 1000 \
}'

```

# Knowledge Base endpoints

---

## List all Knowledge Bases for the client

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

### Code Examples:

#### Python

```

import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def list_knowledge_bases(self):
        url = f"{self.base_url}/api/v1/knowledge-base/"
        headers = {
            "X-Client-ID": self.api_key,
            "X-Client-Secret": self.api_secret,
        }

        response = requests.get(url, headers=headers, timeout=30)

```

```

    response.raise_for_status()
    return response.json()

```

```

# Usage example
api = APIClient("your-api-key", "your-api-secret")
kbs = api.list_knowledge_bases()
print(f"Found {len(kbs['data'])} Knowledge Bases")
for kb in kbs['data']:
    print(f"- {kb['name']}: {kb['state']}")

```

## JavaScript

```

class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async listKnowledgeBases() {
    const url = `${this.baseUrl}/api/v1/knowledge-base/`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
    };

    const res = await fetch(url, { method: "GET", headers });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
api
  .listKnowledgeBases()
  .then((kbs) => {
    console.log(`Found ${kbs.data.length} Knowledge Bases`);
    kbs.data.forEach(kb => {
      console.log(`- ${kb.name}: ${kb.state}`);
    });
  })
  .catch(console.error);

```

## cURL

```

# List all Knowledge Bases

curl -X GET "https://api.example.com/api/v1/knowledge-base/" \
  -H "X-Client-ID: your-api-key" \
  -H "X-Client-Secret: your-api-secret"

```

## Create a new, empty Knowledge Base

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

### Code Examples:

#### Python

```

import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def create_knowledge_base(self, name):
        url = f"{self.base_url}/api/v1/knowledge-base/"
        headers = {
            "X-Client-ID": self.api_key,
            "X-Client-Secret": self.api_secret,
            "Content-Type": "application/json",
        }

```

```

    data = {"name": name}

    response = requests.post(url, headers=headers, json=data, timeout=30)
    response.raise_for_status()
    return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
kb = api.create_knowledge_base("New Product Documentation")
print(f"Created Knowledge Base with ID: {kb['data']['id']}")

```

## JavaScript

```

class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async createKnowledgeBase(name) {
    const url = `${this.baseUrl}/api/v1/knowledge-base/`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
      "Content-Type": "application/json",
    };

    const data = { name };

    const res = await fetch(url, {
      method: "POST",
      headers,
      body: JSON.stringify(data),
    });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
api
  .createKnowledgeBase("New Product Documentation")
  .then((kb) => console.log(`Created Knowledge Base with ID: ${kb.data.id}`))
  .catch(console.error);

```

## cURL

```

# Create a new Knowledge Base

curl -X POST "https://api.example.com/api/v1/knowledge-base/" \
  -H "X-Client-ID: your-api-key" \
  -H "X-Client-Secret: your-api-secret" \
  -H "Content-Type: application/json" \
  -d '{ \
    "name": "New Product Documentation" \
  }'

```

## Get a specific Knowledge Base

*GET* [https://api.example.com/api/v1/knowledge-base/{knowledge\\_base\\_id}](https://api.example.com/api/v1/knowledge-base/{knowledge_base_id})

### Code Examples:

#### Python

```

import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def get_knowledge_base(self, kb_id):

```

```

url = f"{self.base_url}/api/v1/knowledge-base/{kb_id}"
headers = {
    "X-Client-ID": self.api_key,
    "X-Client-Secret": self.api_secret,
}

response = requests.get(url, headers=headers, timeout=30)
response.raise_for_status()
return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
kb_id = "your-kb-id-uuid-here"
kb = api.get_knowledge_base(kb_id)
print(f"Retrieved Knowledge Base: {kb['data']['name']}")
print(f"Documents: {len(kb['data']['documents'])}")

```

## JavaScript

```

class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async getKnowledgeBase(kbId) {
    const url = `${this.baseUrl}/api/v1/knowledge-base/${kbId}`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
    };

    const res = await fetch(url, { method: "GET", headers });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
const kbId = "your-kb-id-uuid-here";
api
  .getKnowledgeBase(kbId)
  .then((kb) => {
    console.log(`Retrieved Knowledge Base: ${kb.data.name}`);
    console.log(`Documents: ${kb.data.documents.length}`);
  })
  .catch(console.error);

```

## cURL

# Get a specific Knowledge Base

```

curl -X GET "https://api.example.com/api/v1/knowledge-base/your-kb-id-uuid-here" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret"

```

## Upload a file to a Knowledge Base

POST [https://api.example.com/api/v1/knowledge-base/{knowledge\\_base\\_id}/files](https://api.example.com/api/v1/knowledge-base/{knowledge_base_id}/files)

### Code Examples:

#### Python

```

import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def upload_file_to_knowledge_base(self, kb_id, file_path):
        url = f"{self.base_url}/api/v1/knowledge-base/{kb_id}/files"

```

```

headers = {
    "X-Client-ID": self.api_key,
    "X-Client-Secret": self.api_secret,
}

with open(file_path, 'rb') as file:
    files = {'file': file}
    response = requests.post(url, headers=headers, files=files, timeout=30)
    response.raise_for_status()
    return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
kb_id = "your-kb-id-uuid-here"
result = api.upload_file_to_knowledge_base(kb_id, "document.pdf")
print(f"File uploaded with ID: {result['data']['id']}")

```

## JavaScript

```

class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async uploadFileToKnowledgeBase(kbId, file) {
    const url = `${this.baseUrl}/api/v1/knowledge-base/${kbId}/files`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
    };

    const formData = new FormData();
    formData.append('file', file);

    const res = await fetch(url, {
      method: "POST",
      headers,
      body: formData,
    });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
const kbId = "your-kb-id-uuid-here";
const fileInput = document.getElementById('fileInput');
const file = fileInput.files[0];

api
  .uploadFileToKnowledgeBase(kbId, file)
  .then((result) => console.log(`File uploaded with ID: ${result.data.id}`))
  .catch(console.error);

```

## cURL

```

# Upload a file to a Knowledge Base

curl -X POST "https://api.example.com/api/v1/knowledge-base/your-kb-id-uuid-here/files" \
  -H "X-Client-ID: your-api-key" \
  -H "X-Client-Secret: your-api-secret" \
  -F "file=@document.pdf"

```

## Delete a file from a Knowledge Base

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

## Code Examples:

### Python

```
import requests
```

```

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def delete_file_from_knowledge_base(self, kb_id, doc_id):
        url = f"{self.base_url}/api/v1/knowledge-base/{kb_id}/files/{doc_id}"
        headers = {
            "X-Client-ID": self.api_key,
            "X-Client-Secret": self.api_secret,
        }

        response = requests.delete(url, headers=headers, timeout=30)
        response.raise_for_status()
        return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
kb_id = "your-kb-id-uuid-here"
doc_id = "doc-uuid-67890"
result = api.delete_file_from_knowledge_base(kb_id, doc_id)
print(result["message"]) # "File deleted successfully"

```

### JavaScript

```

class APIClient {
    constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
        this.apiKey = apiKey;
        this.apiSecret = apiSecret;
        this.baseUrl = baseUrl;
    }

    async deleteFileFromKnowledgeBase(kbId, docId) {
        const url = `${this.baseUrl}/api/v1/knowledge-base/${kbId}/files/${docId}`;
        const headers = {
            "X-Client-ID": this.apiKey,
            "X-Client-Secret": this.apiSecret,
        };

        const res = await fetch(url, { method: "DELETE", headers });
        if (!res.ok) throw new Error(`Request failed: ${res.status}`);

        return res.json();
    }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
const kbId = "your-kb-id-uuid-here";
const docId = "doc-uuid-67890";
api
    .deleteFileFromKnowledgeBase(kbId, docId)
    .then((result) => {
        console.log(result.message); // "File deleted successfully"
    })
    .catch(console.error);

```

### cURL

```

# Delete a file from a Knowledge Base \

curl -X DELETE "https://api.example.com/api/v1/knowledge-base/your-kb-id-uuid-here/files/doc-uuid-67890" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret"

```

## Trigger the ingestion process for a Knowledge Base

*POST* [https://api.example.com/api/v1/knowledge-base/{knowledge\\_base\\_id}/ingest](https://api.example.com/api/v1/knowledge-base/{knowledge_base_id}/ingest)

### Code Examples:

#### Python

```

import requests

class APIClient:

```

```

def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
    self.api_key = api_key
    self.api_secret = api_secret
    self.base_url = base_url

def trigger_knowledge_base_ingestion(self, kb_id):
    url = f"{self.base_url}/api/v1/knowledge-base/{kb_id}/ingest"
    headers = {
        "X-Client-ID": self.api_key,
        "X-Client-Secret": self.api_secret,
    }

    response = requests.post(url, headers=headers, timeout=30)
    response.raise_for_status()
    return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
kb_id = "your-kb-id-uuid-here"
result = api.trigger_knowledge_base_ingestion(kb_id)
print("Ingestion process started")

```

## JavaScript

```

class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async triggerKnowledgeBaseIngestion(kbId) {
    const url = `${this.baseUrl}/api/v1/knowledge-base/${kbId}/ingest`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
    };

    const res = await fetch(url, { method: "POST", headers });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
const kbId = "your-kb-id-uuid-here";
api
  .triggerKnowledgeBaseIngestion(kbId)
  .then(() => console.log("Ingestion process started"))
  .catch(console.error);

```

## cURL

```

# Trigger Knowledge Base ingestion

curl -X POST "https://api.example.com/api/v1/knowledge-base/your-kb-id-uuid-here/ingest" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret"

```

## Get the ingestion status of a Knowledge Base

GET [https://api.example.com/api/v1/knowledge-base/{knowledge\\_base\\_id}/status](https://api.example.com/api/v1/knowledge-base/{knowledge_base_id}/status)

### Code Examples:

#### Python

```

import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

```

```

def get_knowledge_base_status(self, kb_id):
    url = f"{self.base_url}/api/v1/knowledge-base/{kb_id}/status"
    headers = {
        "X-Client-ID": self.api_key,
        "X-Client-Secret": self.api_secret,
    }

    response = requests.get(url, headers=headers, timeout=30)
    response.raise_for_status()
    return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
kb_id = "your-kb-id-uuid-here"
status = api.get_knowledge_base_status(kb_id)
print(f"Knowledge Base status: {status['data']['state']}")

```

## JavaScript

```

class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async getKnowledgeBaseStatus(kbId) {
    const url = `${this.baseUrl}/api/v1/knowledge-base/${kbId}/status`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
    };

    const res = await fetch(url, { method: "GET", headers });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
const kbId = "your-kb-id-uuid-here";
api
  .getKnowledgeBaseStatus(kbId)
  .then((status) => {
    console.log(`Knowledge Base status: ${status.data.state}`);
  })
  .catch(console.error);

```

## cURL

```

# Get Knowledge Base status

curl -X GET "https://api.example.com/api/v1/knowledge-base/your-kb-id-uuid-here/status" \
  -H "X-Client-ID: your-api-key" \
  -H "X-Client-Secret: your-api-secret"

```

## Delete a specific Knowledge Base

**DELETE** [https://api.example.com/api/v1/knowledge-base/{knowledge\\_base\\_id}](https://api.example.com/api/v1/knowledge-base/{knowledge_base_id})

### Code Examples:

#### Python

```

import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def delete_knowledge_base(self, kb_id):
        url = f"{self.base_url}/api/v1/knowledge-base/{kb_id}"
        headers = {

```

```

        "X-Client-ID": self.api_key,
        "X-Client-Secret": self.api_secret,
    }

    response = requests.delete(url, headers=headers, timeout=30)
    response.raise_for_status()
    return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
kb_id = "your-kb-id-uuid-here"
result = api.delete_knowledge_base(kb_id)
print(result["message"]) # "Knowledge Base deleted successfully"

```

## JavaScript

```

class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async deleteKnowledgeBase(kbId) {
    const url = `${this.baseUrl}/api/v1/knowledge-base/${kbId}`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
    };

    const res = await fetch(url, { method: "DELETE", headers });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
const kbId = "your-kb-id-uuid-here";
api
  .deleteKnowledgeBase(kbId)
  .then((result) => {
    console.log(result.message); // "Knowledge Base deleted successfully"
  })
  .catch(console.error);

```

## cURL

```

# Delete a specific Knowledge Base

curl -X DELETE "https://api.example.com/api/v1/knowledge-base/your-kb-id-uuid-here" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret"

```

# Tailored AI endpoints

---

## Create Tailored AI endpoint

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

### Code Examples:

#### Python

```

import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def create_tailored_ai(self, name, summary, system_prompt):

```

```

url = f"{self.base_url}/api/v1/tailored-ai/"
headers = {
    "X-Client-ID": self.api_key,
    "X-Client-Secret": self.api_secret,
    "Content-Type": "application/json",
}

data = {
    "name": name,
    "summary": summary,
    "systemPrompt": system_prompt
}

response = requests.post(url, headers=headers, json=data, timeout=30)
response.raise_for_status()
return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
ai = api.create_tailored_ai(
    name="Customer Support Assistant",
    summary="AI to help with common customer questions.",
    system_prompt="You are a friendly and helpful customer support assistant."
)
print(f"Created AI with ID: {ai['data']['id']}")

```

## JavaScript

```

class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async createTailoredAI(name, summary, systemPrompt) {
    const url = `${this.baseUrl}/api/v1/tailored-ai/`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
      "Content-Type": "application/json",
    };

    const data = {
      name,
      summary,
      systemPrompt
    };

    const res = await fetch(url, {
      method: "POST",
      headers,
      body: JSON.stringify(data),
    });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
api
  .createTailoredAI(
    "Customer Support Assistant",
    "AI to help with common customer questions.",
    "You are a friendly and helpful customer support assistant."
  )
  .then((ai) => console.log(`Created AI with ID: ${ai.data.id}`))
  .catch(console.error);

```

## cURL

```

# Create a basic Tailored AI
curl -X POST "https://api.example.com/api/v1/tailored-ai/" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret" \
-H "Content-Type: application/json" \
-d '{ \

```

```

"name": "Customer Support Assistant", \
"summary": "AI to help with common customer questions.", \
"systemPrompt": "You are a friendly and helpful customer support assistant." \
}'

```

## List Tailored AI endpoint

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

### Code Examples:

#### Python

```

import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def list_tailored_ais(self):
        url = f"{self.base_url}/api/v1/tailored-ai/"
        headers = {
            "X-Client-ID": self.api_key,
            "X-Client-Secret": self.api_secret,
        }

        response = requests.get(url, headers=headers, timeout=30)
        response.raise_for_status()
        return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
ais = api.list_tailored_ais()
print(f"Found {len(ais['data'])} Tailored AIs")
for ai in ais['data']:
    print(f"- {ai['name']}: {ai['summary']}")

```

#### JavaScript

```

class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async listTailoredAIs() {
    const url = `${this.baseUrl}/api/v1/tailored-ai/`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
    };

    const res = await fetch(url, { method: "GET", headers });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
api
  .listTailoredAIs()
  .then((ais) => {
    console.log(`Found ${ais.data.length} Tailored AIs`);
    ais.data.forEach(ai => {
      console.log(`- ${ai.name}: ${ai.summary}`);
    });
  })
  .catch(console.error);

```

#### cURL

```
# List all Tailored AIs
curl -X GET "https://api.example.com/api/v1/tailored-ai/"
-H "X-Client-ID: your-api-key"
-H "X-Client-Secret: your-api-secret"
```

## Get Tailored AI endpoint

*GET* https://api.example.com/api/v1/tailored-ai/{tailored\_ai\_id}

### Code Examples:

#### Python

```
import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def get_tailored_ai(self, tailored_ai_id):
        url = f"{self.base_url}/api/v1/tailored-ai/{tailored_ai_id}"
        headers = {
            "X-Client-ID": self.api_key,
            "X-Client-Secret": self.api_secret,
        }

        response = requests.get(url, headers=headers, timeout=30)
        response.raise_for_status()
        return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
tailored_ai_id = "your-tailored-ai-id-uuid-here"
tailored_ai = api.get_tailored_ai(tailored_ai_id)
print(f"Retrieved Tailored AI: {tailored_ai['data']['name']}")
```

#### JavaScript

```
class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async getTailoredAI(tailoredAiId) {
    const url = `${this.baseUrl}/api/v1/tailored-ai/${tailoredAiId}`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
    };

    const res = await fetch(url, { method: "GET", headers });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
const tailoredAiId = "your-tailored-ai-id-uuid-here";
api
  .getTailoredAI(tailoredAiId)
  .then((tailoredAi) => console.log(`Retrieved Tailored AI: ${tailoredAi.data.name}`))
  .catch(console.error);
```

#### cURL

```
# Get a specific Tailored AI

curl -X GET "https://api.example.com/api/v1/tailored-ai/your-tailored-ai-id-uuid-here" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret"
```

## Update Tailored AI endpoint

*PUT* [https://api.example.com/api/v1/tailored-ai/{tailored\\_ai\\_id}](https://api.example.com/api/v1/tailored-ai/{tailored_ai_id})

### Code Examples:

#### Python

```
import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def update_tailored_ai(self, tailored_ai_id, **updates):
        url = f"{self.base_url}/api/v1/tailored-ai/{tailored_ai_id}"
        headers = {
            "X-Client-ID": self.api_key,
            "X-Client-Secret": self.api_secret,
            "Content-Type": "application/json",
        }

        response = requests.put(url, headers=headers, json=updates, timeout=30)
        response.raise_for_status()
        return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
tailored_ai_id = "your-tailored-ai-id-uuid-here"
updated_tailored_ai = api.update_tailored_ai(
    tailored_ai_id,
    name="Updated AI Name",
    summary="Updated summary for the AI.",
    system_prompt="You are an updated AI assistant with new instructions."
)
print(f"Updated Tailored AI: {updated_tailored_ai['data']['name']}")
```

#### JavaScript

```
class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async updateTailoredAI(tailoredAiId, updates) {
    const url = `${this.baseUrl}/api/v1/tailored-ai/${tailoredAiId}`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
      "Content-Type": "application/json",
    };

    const res = await fetch(url, {
      method: "PUT",
      headers,
      body: JSON.stringify(updates),
    });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
const tailoredAiId = "your-tailored-ai-id-uuid-here";
api
  .updateTailoredAI(tailoredAiId, {
    name: "Updated AI Name",
    summary: "Updated summary for the AI.",
    systemPrompt: "You are an updated AI assistant with new instructions."
  })
  .then((tailoredAi) => console.log(`Updated Tailored AI: ${tailoredAi.data.name}`))
```

```
.catch(console.error);
```

## cURL

```
# Update AI name and summary
curl -X PUT "https://api.example.com/api/v1/tailored-ai/your-tailored-ai-id-uuid-here" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret" \
-H "Content-Type: application/json" \
-d '{ \
  "name": "Updated AI Name", \
  "summary": "Updated summary for the AI." \
}'

# Update only the system prompt
curl -X PUT "https://api.example.com/api/v1/tailored-ai/your-tailored-ai-id-uuid-here" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret" \
-H "Content-Type: application/json" \
-d '{ \
  "systemPrompt": "You are an updated AI assistant with new instructions." \
}'
```

## Delete Tailored AI endpoint

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

### Code Examples:

#### Python

```
import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def delete_tailored_ai(self, tailored_ai_id):
        url = f"{self.base_url}/api/v1/tailored-ai/{tailored_ai_id}"
        headers = {
            "X-Client-ID": self.api_key,
            "X-Client-Secret": self.api_secret,
        }

        response = requests.delete(url, headers=headers, timeout=30)
        response.raise_for_status()
        return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
tailored_ai_id = "your-tailored-ai-id-uuid-here"
result = api.delete_tailored_ai(tailored_ai_id)
print(result["message"]) # "Tailored AI deleted successfully"
```

#### JavaScript

```
class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async deleteTailoredAI(tailoredAiId) {
    const url = `${this.baseUrl}/api/v1/tailored-ai/${tailoredAiId}`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
    };

    const res = await fetch(url, { method: "DELETE", headers });
    if (!res.ok) throw new Error(`Request failed: ${res.status}`);

    return res.json();
  }
}
```

```

}

// Usage example
const api = new APIClient("your-api-key", "your-api-secret");
const tailoredAiId = "your-tailored-ai-id-uuid-here";
api
  .deleteTailoredAI(tailoredAiId)
  .then((result) => {
    console.log(result.message); // "Tailored AI deleted successfully"
  })
  .catch(console.error);

```

## cURL

```

# Delete a specific Tailored AI

curl -X DELETE "https://api.example.com/api/v1/tailored-ai/your-tailored-ai-id-uuid-here" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret"

```

## Connect a Knowledge Base to a Tailored AI endpoint

POST [https://api.example.com/api/v1/tailored-ai/{tailored\\_ai\\_id}/knowledge-base](https://api.example.com/api/v1/tailored-ai/{tailored_ai_id}/knowledge-base)

### Code Examples:

#### Python

```

import requests

class APIClient:
    def __init__(self, api_key, api_secret, base_url="https://api.example.com"):
        self.api_key = api_key
        self.api_secret = api_secret
        self.base_url = base_url

    def connect_knowledge_base(self, tailored_ai_id, kb_id):
        url = f"{self.base_url}/api/v1/tailored-ai/{tailored_ai_id}/knowledge-base"
        headers = {
            "X-Client-ID": self.api_key,
            "X-Client-Secret": self.api_secret,
            "Content-Type": "application/json",
        }

        data = {"knowledgeBaseId": kb_id}

        response = requests.post(url, headers=headers, json=data, timeout=30)
        response.raise_for_status()
        return response.json()

# Usage example
api = APIClient("your-api-key", "your-api-secret")
tailored_ai_id = "your-tailored-ai-id-uuid-here"
kb_id = "your-kb-id-uuid-here"
result = api.connect_knowledge_base(tailored_ai_id, kb_id)
print("Knowledge Base successfully set")

```

#### JavaScript

```

class APIClient {
  constructor(apiKey, apiSecret, baseUrl = "https://api.example.com") {
    this.apiKey = apiKey;
    this.apiSecret = apiSecret;
    this.baseUrl = baseUrl;
  }

  async connectKnowledgeBase(tailoredAiId, kbId) {
    const url = `${this.baseUrl}/api/v1/tailored-ai/${tailoredAiId}/knowledge-base`;
    const headers = {
      "X-Client-ID": this.apiKey,
      "X-Client-Secret": this.apiSecret,
      "Content-Type": "application/json",
    };

    const data = { knowledgeBaseId: kbId };

    const res = await fetch(url, {

```

```
    method: "POST",
    headers,
    body: JSON.stringify(data),
  });
  if (!res.ok) throw new Error(`Request failed: ${res.status}`);

  return res.json();
}
}
```

// Usage example

```
const api = new APIClient("your-api-key", "your-api-secret");
const tailoredAiId = "your-tailored-ai-id-uuid-here";
const kbId = "your-kb-id-uuid-here";
api
  .connectKnowledgeBase(tailoredAiId, kbId)
  .then(() => console.log("Knowledge Base successfully set"))
  .catch(console.error);
```

## cURL

# Connect a Knowledge Base to a Tailored AI

```
curl -X POST "https://api.example.com/api/v1/tailored-ai/your-tailored-ai-id-uuid-here/knowledge-base" \
-H "X-Client-ID: your-api-key" \
-H "X-Client-Secret: your-api-secret" \
-H "Content-Type: application/json" \
-d '{ \
  "knowledgeBaseId": "your-kb-id-uuid-here" \
}'
```