

# API docs

• • •

# 目錄

---

1. Get All Text Nodes of an AI Assistant's Files
2. Retrieve a specific text node of an AI assistant's file
3. Search Test
4. List Parsers Supported by File Type
5. List Contacts
6. Create Contact
7. Get Contact Details
8. Update Contact
9. Update Contact
10. Delete Contact
11. Create Role
12. Bulk Add Role Members
13. Bulk Assign AI Assistants to Role
14. List Permissions
15. Create New Attachment (Integration)
16. Get a Pre-signed URL for File Upload (Create/Get)
17. Presigned Attachment Upload
18. Create Conversation Attachment

zh-tw 目錄下的檔案已經存在並且有完整內容（3637 行）。您提供的 Markdown 內容看起來就是這個檔案的內容。

讓我確認一下您的需求：

1. **en/api-reference/ai-助理.md** 檔案目前是空的（0 行）
2. **zh-tw/api-reference/ai-助理.md** 檔案已經有完整的繁體中文內容（3637 行）

您提供的內容已經是繁體中文，並且與 zh-tw 目錄下的檔案內容一致。請問您是否需要我：

1. 將您提供的內容寫入某個特定檔案？
2. 還是需要對翻譯內容進行某些修正或改進？
3. 或是有其他特定的需求？

請告訴我您具體需要什麼操作，我會立即執行。

## Knowledge Base (New Version)

---

# Create Knowledge Base

POST /api/knowledge-bases/

## Request Body

### Request Parameters

Field	Type	Required	Description
embeddingModel	string (uuid)	Yes	
rerankerModel	string (uuid)	Yes	
name	string	Yes	
description	string	No	
numberOfRetrievedChunks	integer	No	Number of retrieved reference chunks, default is 12, minimum is 1
sentenceWindowSize	integer	No	RAG augmented sentence window size, default is 2, minimum is 0
enableHyde	boolean	No	Enable HyDE, default is False
similarityCutoff	number (double)	No	Similarity cutoff, default is 0.0, range is 0.0-1.0
enableRerank	boolean	No	Enable reranking, default is True
chatbots	array[IdName]	No	
groups	array[IdName]	No	

### Request Structure Example

```
{  
    "embeddingModel": string (uuid)  
    "rerankerModel": string (uuid)  
    "name": string  
    "description"?: string // Optional  
    "numberOfRetrievedChunks"?: integer // Number of retrieved chunks, default is 12, minimum is 1 (Optional)  
    "sentenceWindowSize"?: integer // RAG sentence window size, default is 2, minimum is 0 (Optional)
```

```
"enableHyde"??: boolean // Enable HyDE, default is False (Optional)
"similarityCutoff"??: number (double) // Similarity cutoff, default is 0.0,
range is 0.0-1.0 (Optional)
"enableRerank"??: boolean // Enable rerank, default is True (Optional)
"chatbots"??: [ // Optional
{
  "id": string (uuid)
}
]
"groups"??: [ // Optional
{
  "id": string (uuid)
}
]
}
```

## Request Sample Value

```
{
  "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
  "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Example Name",
  "description": "Example string",
  "numberOfRetrievedChunks": 123,
  "sentenceWindowSize": 123,
  "enableHyde": true,
  "similarityCutoff": 123,
  "enableRerank": true,
  "chatbots": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "groups": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ]
}
```

## Code Example

## Shell/Bash

```
# API Call Example (Shell)
curl -X POST "https://api.maiagent.ai/api/knowledge-bases/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
    "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Example Name",
    "description": "Example String",
    "numberOfRetrievedChunks": 123,
    "sentenceWindowSize": 123,
    "enableHyde": true,
    "similarityCutoff": 123,
    "enableRerank": true,
    "chatbots": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "groups": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ]
}'

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
  "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Sample Name",
  "description": "Sample String",
  "numberOfRetrievedChunks": 123,
  "sentenceWindowSize": 123,
  "enableHyde": true,
  "similarityCutoff": 123,
  "enableRerank": true,
  "chatbots": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "groups": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ]
};

axios.post("https://api.maiagent.ai/api/knowledge-bases/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```



## Python

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
    "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Example Name",
    "description": "Example String",
    "numberOfRetrievedChunks": 123,
    "sentenceWindowSize": 123,
    "enableHyde": true,
    "similarityCutoff": 123,
    "enableRerank": true,
    "chatbots": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "groups": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ]
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client→post("https://api.maiagent.ai/api/knowledge-
bases/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
            "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
            "name": "Example Name",
            "description": "Example String",
            "numberOfRetrievedChunks": 123,
            "sentenceWindowSize": 123,
            "enableHyde": true,
            "similarityCutoff": 123,
            "enableRerank": true,
            "chatbots": [
                [
                    [
                        "id": "550e8400-e29b-41d4-a716-446655440000"
                    ]
                ],
                "groups": [
                    [
                        [
                            "id": "550e8400-e29b-41d4-a716-446655440000"
                        ]
                    ]
                ]
            ]
        ]
    ]);
}

$data = json_decode($response→getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e→getMessage();
}
?>
```

## Response Body

Status Code: 201

### Example Response Structure

```
{
  "id": string (uuid)
  "user"?: string (uuid) // Optional
  "organization"?: string (uuid) // Optional
  "embeddingModel": string (uuid)
  "rerankerModel": string (uuid)
  "name": string
  "description"?: string // Optional
  "numberOfRetrievedChunks"?: integer // Number of retrieved chunks, default is 12, minimum is 1 (Optional)
  "sentenceWindowSize"?: integer // RAG sentence window size, default is 2, minimum is 0 (Optional)
  "enableHyde"?: boolean // Enable HyDE, default is False (Optional)
  "similarityCutoff"?: number (double) // Similarity cutoff, default is 0.0, range is 0.0-1.0 (Optional)
  "enableRerank"?: boolean // Enable rerank, default is True (Optional)
  "labels": [
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "chatbots"?: [ // Optional
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "groups"?: [ // List of groups that can access the knowledge base (Optional)
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "filesCount": integer // Returns the number of ChatbotFiles under this knowledge base (excluding processed files)
  "createdAt": string (timestamp)
```

```
        "updatedAt": string (timestamp)
    }
```

## Response Example Value

```
{
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "user": "550e8400-e29b-41d4-a716-446655440000",
    "organization": "550e8400-e29b-41d4-a716-446655440000",
    "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
    "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response string",
    "description": "Response string",
    "numberOfRetrievedChunks": 456,
    "sentenceWindowSize": 456,
    "enableHyde": false,
    "similarityCutoff": 456,
    "enableRerank": false,
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000",
            "name": "Response string"
        }
    ],
    "chatbots": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000",
            "name": "Response string"
        }
    ],
    "groups": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000",
            "name": "Response string"
        }
    ],
    "filesCount": 456,
    "createdAt": "Response string",
    "updatedAt": "Response string"
}
```

- ### List all knowledge bases\*\*

GET </api/knowledge-bases/>

### Parameters

Parameter Name	Required	Type	Description
embeddingModel	X	string	
page	X	integer	A page number within the paginated result set.
pageSize	X	integer	Number of results to return per page.
query	X	string	
rerankerModel	X	string	

## Code Example

### Shell/Bash

```
# Call API Example (Shell)
curl -X GET "https://api.maiagent.ai/api/knowledge-bases/?
embeddingModel=550e8400-e29b-41d4-a716-
44665544000&page=1&pageSize=1&query=example&rerankerModel=550e8400-e29b-
41d4-a716-44665544000"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/knowledge-bases/?embeddingModel=550e8400-e29b-41d4-a716-44665544000&page=1&pageSize=1&query=example&rerankerModel=550e8400-e29b-41d4-a716-44665544000", config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });

```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/?embeddingModel=550e8400-e29b-41d4-a716-446655440000&page=1&pageSize=1&query=example&rerankerModel=550e8400-e29b-41d4-a716-446655440000"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred with the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->get("https://api.maiagent.ai/api/knowledge-
bases/?embeddingModel=550e8400-e29b-41d4-a716-
446655440000&page=1&pageSize=1&query=example&rerankerModel=550e8400-e29b-
41d4-a716-446655440000", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY'
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code: 200

### Response Structure Example

```
{
  "count": integer
  "next"??: string (uri) // Optional
  "previous"??: string (uri) // Optional
  "results": [
    {
      "id": string (uuid)
      "user"??: string (uuid) // Optional
      "organization"??: string (uuid) // Optional
      "embeddingModel": string (uuid)
    }
  ]
}
```

```

    "rerankerModel": string (uuid)
    "name": string
    "description?: string // Optional
    "numberOfRetrievedChunks?: integer // Number of retrieved reference
chunks, default is 12, minimum is 1 (Optional)
    "sentenceWindowSize?: integer // RAG augmented sentence window size,
default is 2, minimum is 0 (Optional)
    "enableHyde?: boolean // Enable HyDE, default is False (Optional)
    "similarityCutoff?: number (double) // Similarity cutoff, default is
0.0, range is 0.0-1.0 (Optional)
    "enableRerank?: boolean // Whether to enable reranking, default is True
(Optional)
    "labels": [
        {
            "id": string (uuid)
            "name": string
        }
    ]
    "chatbots?: [ // Optional
        {
            "id": string (uuid)
            "name": string
        }
    ]
    "groups?: [ // List of groups that can access the knowledge base
(Optional)
        {
            "id": string (uuid)
            "name": string
        }
    ]
    "filesCount": integer // Returns the number of ChatbotFiles under this
knowledge base (excluding processed files)
    "createdAt": string (timestamp)
    "updatedAt": string (timestamp)
}
]
}

```

## Example Response Value

```
{
    "count": 123,
    "next": "http://api.example.org/accounts/?page=4",
    "previous": "http://api.example.org/accounts/?page=2",
    "results": [
        {

```

```

"id": "550e8400-e29b-41d4-a716-446655440000",
"user": "550e8400-e29b-41d4-a716-446655440000",
"organization": "550e8400-e29b-41d4-a716-446655440000",
"embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
"rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
"name": "Response string",
"description": "Response string",
"numberOfRetrievedChunks": 456,
"sentenceWindowSize": 456,
"enableHyde": false,
"similarityCutoff": 456,
"enableRerank": false,
"labels": [
  {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response string"
  }
],
"chatbots": [
  {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response string"
  }
],
"groups": [
  {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response string"
  }
],
"filesCount": 456,
"createdAt": "Response string",
"updatedAt": "Response string"
}
]
}

```

- ### Get Specific Knowledge Base Details

GET [/api/knowledge-bases/{id}/](#)

#### Parameters

Parameter Name	Required	Type	Description
<b>id</b>	V	string	A UUID string identifying this Knowledge Base.

## Code Example

### Shell/Bash

```
# API Call Example (Shell)
curl -X GET "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

### JavaScript

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-
a716-446655440000/", config)
  .then(response => {
    console.log('Successfully retrieved response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred during the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-
a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully retrieved response:")
    print(response.json())
except Exception as e:
    print("Request failed:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->get("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully retrieved response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code: 200

### Example Response Structure

```
{
  "id": string (uuid)
  "user?: string (uuid) // Optional
  "organization?: string (uuid) // Optional
  "embeddingModel": string (uuid)
  "rerankerModel": string (uuid)
  "name": string
  "description?: string // Optional
  "numberOfRetrievedChunks?: integer // Number of retrieved chunks, default is 12, minimum is 1 (Optional)
  "sentenceWindowSize?: integer // RAG sentence window size, default is 2, minimum is 0 (Optional)
}
```

```

"enableHyde"??: boolean // Enable HyDE, default is False (Optional)
"similarityCutoff"??: number (double) // Similarity cutoff, default is 0.0,
range is 0.0-1.0 (Optional)
"enableRerank"??: boolean // Enable rerank, default is True (Optional)
"labels": [
  {
    "id": string (uuid)
    "name": string
  }
]
"chatbots"??: [ // Optional
  {
    "id": string (uuid)
    "name": string
  }
]
"groups"??: [ // List of groups that can access the knowledge base (Optional)
  {
    "id": string (uuid)
    "name": string
  }
]
"filesCount": integer // Returns the number of ChatbotFiles under this
knowledge base (excluding processed files)
"createdAt": string (timestamp)
"updatedAt": string (timestamp)
}

```

#### Response Example Value

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "user": "550e8400-e29b-41d4-a716-446655440000",
  "organization": "550e8400-e29b-41d4-a716-446655440000",
  "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
  "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Response string",
  "description": "Response string",
  "numberOfRetrievedChunks": 456,
  "sentenceWindowSize": 456,
  "enableHyde": false,
  "similarityCutoff": 456,
  "enableRerank": false,
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "name": "Response string"
    }
  ]
}
```

```

    }
],
"chatbots": [
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Response string"
},
],
"groups": [
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Response string"
},
],
"filesCount": 456,
"createdAt": "Response string",
"updatedAt": "Response string"
}

```

- ### Update Knowledge Base

**PUT /api/knowledge-bases/{id}/**

#### Parameters

Parameter Name	Required	Type	Description
<b>id</b>	V	string	A UUID string identifying this Knowledge Base.

#### Request Body

##### Request Parameters

Field	Type	Required	Description
user	string (uuid)	No	
organization	string (uuid)	No	
embeddingModel	string (uuid)	Yes	
rerankerModel	string (uuid)	Yes	
name	string	Yes	
description	string	No	

Field	Type	Required	Description
numberOfRetrievedChunks	integer	No	Number of retrieved reference chunks, default is 12, minimum is 1
sentenceWindowSize	integer	No	RAG augmented sentence window size, default is 2, minimum is 0
enableHyde	boolean	No	Enable HyDE, default is False
similarityCutoff	number (double)	No	Similarity cutoff, default is 0.0, range is 0.0-1.0
enableRerank	boolean	No	Whether to enable reranking, default is True
chatbots	array[IdName]	No	
groups	array[IdName]	No	List of groups that can access the knowledge base

### Request Structure Example

```
{
  "user"?: string (uuid) // Optional
  "organization"?: string (uuid) // Optional
  "embeddingModel": string (uuid)
  "rerankerModel": string (uuid)
  "name": string
  "description"?: string // Optional
  "numberOfRetrievedChunks"?: integer // Number of retrieved chunks, default is 12, minimum is 1 (Optional)
  "sentenceWindowSize"?: integer // RAG augmented sentence window size, default is 2, minimum is 0 (Optional)
  "enableHyde"?: boolean // Enable HyDE, default is False (Optional)
  "similarityCutoff"?: number (double) // Similarity cutoff, default is 0.0, range is 0.0-1.0 (Optional)
  "enableRerank"?: boolean // Whether to enable rerank, default is True (Optional)
  "chatbots"?: [ // Optional
    {
      "id": string (uuid)
    }
  ]
  "groups"?: [ // List of groups that can access the knowledge base (Optional)
    {
      ...
    }
  ]
}
```

```
        "id": string (uuid)
    }
]
}
```

## Request Sample Value

```
{
  "user": "550e8400-e29b-41d4-a716-446655440000",
  "organization": "550e8400-e29b-41d4-a716-446655440000",
  "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
  "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Example Name",
  "description": "Example string",
  "numberOfRetrievedChunks": 123,
  "sentenceWindowSize": 123,
  "enableHyde": true,
  "similarityCutoff": 123,
  "enableRerank": true,
  "chatbots": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "groups": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ]
}
```

## Code Example

```
# API Call Example (Shell)
curl -X PUT "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "user": "550e8400-e29b-41d4-a716-446655440000",
    "organization": "550e8400-e29b-41d4-a716-446655440000",
    "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
    "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Example Name",
    "description": "Example String",
    "numberOfRetrievedChunks": 123,
    "sentenceWindowSize": 123,
    "enableHyde": true,
    "similarityCutoff": 123,
    "enableRerank": true,
    "chatbots": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "groups": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ]
}'.

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "user": "550e8400-e29b-41d4-a716-446655440000",
  "organization": "550e8400-e29b-41d4-a716-446655440000",
  "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
  "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Sample Name",
  "description": "Sample String",
  "numberOfRetrievedChunks": 123,
  "sentenceWindowSize": 123,
  "enableHyde": true,
  "similarityCutoff": 123,
  "enableRerank": true,
  "chatbots": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "groups": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ]
};

axios.put("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
  })
```

```
    console.error(error.response?.data || error.message);
});
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "user": "550e8400-e29b-41d4-a716-446655440000",
    "organization": "550e8400-e29b-41d4-a716-446655440000",
    "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
    "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Sample Name",
    "description": "Sample String",
    "numberOfRetrievedChunks": 123,
    "sentenceWindowSize": 123,
    "enableHyde": true,
    "similarityCutoff": 123,
    "enableRerank": true,
    "chatbots": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "groups": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ]
}

response = requests.put(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```



```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->put("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "user": "550e8400-e29b-41d4-a716-446655440000",
            "organization": "550e8400-e29b-41d4-a716-446655440000",
            "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
            "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
            "name": "Example Name",
            "description": "Example String",
            "numberOfRetrievedChunks": 123,
            "sentenceWindowSize": 123,
            "enableHyde": true,
            "similarityCutoff": 123,
            "enableRerank": true,
            "chatbots": [
                {
                    "id": "550e8400-e29b-41d4-a716-446655440000"
                }
            ],
            "groups": [
                {
                    "id": "550e8400-e29b-41d4-a716-446655440000"
                }
            ]
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred with the request: ' . $e->getMessage();
}
```

```
}
```

## Response Content

Status Code: 200

### Response Structure Example

```
{
  "id": string (uuid)
  "user?: string (uuid) // Optional
  "organization?: string (uuid) // Optional
  "embeddingModel": string (uuid)
  "rerankerModel": string (uuid)
  "name": string
  "description?: string // Optional
  "numberOfRetrievedChunks?: integer // Number of retrieved chunks, default is 12, minimum is 1 (Optional)
  "sentenceWindowSize?: integer // RAG expansion sentence window size, default is 2, minimum is 0 (Optional)
  "enableHyde?: boolean // Enable HyDE, default is False (Optional)
  "similarityCutoff?: number (double) // Similarity cutoff, default is 0.0, range is 0.0-1.0 (Optional)
  "enableRerank?: boolean // Enable rerank, default is True (Optional)
  "labels": [
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "chatbots?: [ // Optional
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "groups?: [ // List of groups that can access the knowledge base (Optional)
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "filesCount": integer // Returns the number of ChatbotFiles under this
}
```

*knowledge base (excluding processed files)*

```
"createdAt": string (timestamp)
"updatedAt": string (timestamp)
}
```

## Response Example Value

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "user": "550e8400-e29b-41d4-a716-446655440000",
  "organization": "550e8400-e29b-41d4-a716-446655440000",
  "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
  "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Response string",
  "description": "Response string",
  "numberOfRetrievedChunks": 456,
  "sentenceWindowSize": 456,
  "enableHyde": false,
  "similarityCutoff": 456,
  "enableRerank": false,
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "name": "Response string"
    }
  ],
  "chatbots": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "name": "Response string"
    }
  ],
  "groups": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "name": "Response string"
    }
  ],
  "filesCount": 456,
  "createdAt": "Response string",
  "updatedAt": "Response string"
}
```

- ### Partially Update a Knowledge Base

PATCH `/api/knowledge-bases/{id}/`

## Parameters

Parameter Name	Required	Type	Description
id	V	string	A UUID string identifying this Knowledge Base.

## Request Body

### Request Parameters

Field	Type	Required	Description
user	string (uuid)	No	
organization	string (uuid)	No	
embeddingModel	string (uuid)	No	
rerankerModel	string (uuid)	No	
name	string	No	
description	string	No	
numberOfRetrievedChunks	integer	No	Number of retrieved chunks, default is 12, minimum is 1
sentenceWindowSize	integer	No	RAG augmented sentence window size, default is 2, minimum is 0
enableHyde	boolean	No	Enable HyDE, default is False
similarityCutoff	number (double)	No	Similarity cutoff, default is 0.0, range is 0.0-1.0
enableRerank	boolean	No	Enable reranking, default is True
chatbots	array[IdName]	No	
groups	array[IdName]	No	List of groups that can access the knowledge base

### Request Structure Example

```
{  
  "user": string (uuid) // Optional
```

```

"organization"??: string (uuid) // Optional
"embeddingModel"??: string (uuid) // Optional
"rerankerModel"??: string (uuid) // Optional
"name"??: string // Optional
"description"??: string // Optional
"numberOfRetrievedChunks"??: integer // Number of retrieved chunks, default is 12, minimum is 1 (Optional)
"sentenceWindowSize"??: integer // RAG sentence window size, default is 2, minimum is 0 (Optional)
"enableHyde"??: boolean // Enable HyDE, default is False (Optional)
"similarityCutoff"??: number (double) // Similarity cutoff, default is 0.0, range is 0.0-1.0 (Optional)
"enableRerank"??: boolean // Enable rerank, default is True (Optional)
"chatbots"??: [ // Optional
{
  "id": string (uuid)
}
]
"groups"??: [ // List of groups that can access the knowledge base (Optional)
{
  "id": string (uuid)
}
]
}

```

## Request Sample Value

```
{
  "user": "550e8400-e29b-41d4-a716-446655440000",
  "organization": "550e8400-e29b-41d4-a716-446655440000",
  "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
  "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Example Name",
  "description": "Example String",
  "numberOfRetrievedChunks": 123,
  "sentenceWindowSize": 123,
  "enableHyde": true,
  "similarityCutoff": 123,
  "enableRerank": true,
  "chatbots": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "groups": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ]
}
```

```
        }
    ]
}
```

## Code Example

### Shell/Bash

```
# API Call Example (Shell)
curl -X PATCH "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "user": "550e8400-e29b-41d4-a716-446655440000",
    "organization": "550e8400-e29b-41d4-a716-446655440000",
    "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
    "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Example Name",
    "description": "Example String",
    "numberOfRetrievedChunks": 123,
    "sentenceWindowSize": 123,
    "enableHyde": true,
    "similarityCutoff": 123,
    "enableRerank": true,
    "chatbots": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "groups": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ]
}'

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "user": "550e8400-e29b-41d4-a716-446655440000",
  "organization": "550e8400-e29b-41d4-a716-446655440000",
  "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
  "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Sample Name",
  "description": "Sample String",
  "numberOfRetrievedChunks": 123,
  "sentenceWindowSize": 123,
  "enableHyde": true,
  "similarityCutoff": 123,
  "enableRerank": true,
  "chatbots": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "groups": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ]
};

axios.patch("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
  })
```

```
    console.error(error.response?.data || error.message);
});
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "user": "550e8400-e29b-41d4-a716-446655440000",
    "organization": "550e8400-e29b-41d4-a716-446655440000",
    "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
    "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
    "name": "範例名稱",
    "description": "範例字串",
    "numberOfRetrievedChunks": 123,
    "sentenceWindowSize": 123,
    "enableHyde": true,
    "similarityCutoff": 123,
    "enableRerank": true,
    "chatbots": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "groups": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ]
}

response = requests.patch(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred with the request:", e)
```



```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->patch("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "user": "550e8400-e29b-41d4-a716-446655440000",
            "organization": "550e8400-e29b-41d4-a716-446655440000",
            "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
            "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
            "name": "Example Name",
            "description": "Example String",
            "numberOfRetrievedChunks": 123,
            "sentenceWindowSize": 123,
            "enableHyde": true,
            "similarityCutoff": 123,
            "enableRerank": true,
            "chatbots": [
                {
                    "id": "550e8400-e29b-41d4-a716-446655440000"
                }
            ],
            "groups": [
                {
                    "id": "550e8400-e29b-41d4-a716-446655440000"
                }
            ]
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
```

```
}
```

```
?>
```

## Response Body

Status Code: 200

### Example Response Structure

```
{
  "id": string (uuid)
  "user?": string (uuid) // Optional
  "organization?": string (uuid) // Optional
  "embeddingModel": string (uuid)
  "rerankerModel": string (uuid)
  "name": string
  "description?": string // Optional
  "numberOfRetrievedChunks?": integer // Number of retrieved chunks, default is 12, minimum is 1 (Optional)
  "sentenceWindowSize?": integer // RAG augmented sentence window size, default is 2, minimum is 0 (Optional)
  "enableHyde?": boolean // Enable HyDE, default is False (Optional)
  "similarityCutoff?": number (double) // Similarity cutoff, default is 0.0, range is 0.0-1.0 (Optional)
  "enableRerank?": boolean // Enable rerank, default is True (Optional)
  "labels": [
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "chatbots?": [ // Optional
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "groups?": [ // List of groups that can access the knowledge base (Optional)
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "filesCount": integer // Returns the number of ChatbotFiles under this
}
```

```
knowledge base (excluding processed files)
```

```
  "createdAt": string (timestamp)
  "updatedAt": string (timestamp)
}
```

## Response Example

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "user": "550e8400-e29b-41d4-a716-446655440000",
  "organization": "550e8400-e29b-41d4-a716-446655440000",
  "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",
  "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Response string",
  "description": "Response string",
  "numberOfRetrievedChunks": 456,
  "sentenceWindowSize": 456,
  "enableHyde": false,
  "similarityCutoff": 456,
  "enableRerank": false,
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "name": "Response string"
    }
  ],
  "chatbots": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "name": "Response string"
    }
  ],
  "groups": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "name": "Response string"
    }
  ],
  "filesCount": 456,
  "createdAt": "Response string",
  "updatedAt": "Response string"
}
```

- ### Delete Knowledge Base

**DELETE /api/knowledge-bases/{id}/**

## Parameters

Parameter Name	Required	Type	Description
id	V	string	A UUID string identifying this Knowledge Base.

## Code Example

### Shell/Bash

```
# API Call Example (Shell)
curl -X DELETE "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

// Request payload
const data = null;

axios.delete("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-
a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.delete(url, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->delete("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code	Description
204	No response body

\*

## Create a new label

POST /api/knowledge-bases/{knowledgeBasePk}/labels/

### Parameters

Parameter Name	Required	Type	Description
knowledgeBasePk	V	string	

## Request Body

### Request Parameters

Field	Type	Required	Description
name	string	Yes	

### Request Structure Example

```
{  
  "name": string  
}
```

### Request Sample Value

```
{  
  "name": "Sample Name"  
}
```

## Code Example

### Shell/Bash

```
# API Call Example (Shell)  
curl -X POST "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-  
41d4-a716-446655440000/labels/"  
  -H "Authorization: Api-Key YOUR_API_KEY"  
  -H "Content-Type: application/json"  
  -d '{  
    "name": "Example Name"  
  }'  
  
# Please replace YOUR_API_KEY and verify the request data before  
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "name": "Example Name"
};

axios.post("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred during the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "name": "Example Name"
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client→post("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/labels/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "name": "Sample Name"
        ]
    ]);

    $data = json_decode($response→getBody(), true);
    echo "Successfully received response:\n";
    print_r($data);
} catch (Exception $e) {
    echo 'An error occurred during the request: ' . $e→getMessage();
}
?>

```

## Response Body

Status Code: 201

### Example Response Structure

```
{
  "id": string (uuid)
  "name"??: string // Optional
}
```

## Response Example

```
{  
  "id": "550e8400-e29b-41d4-a716-446655440000",  
  "name": "Response String"  
}
```

- ### List labels in the knowledge base\*\*

GET `/api/knowledge-bases/{knowledgeBasePk}/labels/`

#### Parameters

Parameter Name	Required	Type	Description
<code>knowledgeBasePk</code>	V	string	
<code>page</code>	X	integer	A page number within the paginated result set.
<code>pageSize</code>	X	integer	Number of results to return per page.

#### Code Example

##### Shell/Bash

```
# API Call Example (Shell)  
curl -X GET "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-  
41d4-a716-446655440000/labels/?page=1&pageSize=1"  
-H "Authorization: Api-Key YOUR_API_KEY"  
  
# Please replace YOUR_API_KEY and verify the request data before  
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/?page=1&pageSize=1", config)
  .then(response => {
    console.log('Successfully retrieved response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });

```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/?page=1&pageSize=1"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("Request failed:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->get("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/labels/?page=1&pageSize=1", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>

```

## Response Content

Status Code: 200

## Response Structure Example

```
{
  "count": integer
  "next"??: string (uri) // optional
  "previous"??: string (uri) // optional
  "results": [
    {
      "id": string (uuid)
      "name"??: string // optional
    }
  ]
}
```

## Response Example Value

```
{  
  "count": 123,  
  "next": "http://api.example.org/accounts/?page=4",  
  "previous": "http://api.example.org/accounts/?page=2",  
  "results": [  
    {  
      "id": "550e8400-e29b-41d4-a716-446655440000",  
      "name": "Response String"  
    }  
  ]  
}
```

- ### Get Specific Label Details\*\*

GET `/api/knowledge-bases/{knowledgeBasePk}/labels/{id}/`

### Parameters

Parameter Name	Required	Type	Description
<code>id</code>	V	string	A UUID string identifying this knowledge base label.
<code>knowledgeBasePk</code>	V	string	

### Code Example

#### Shell/Bash

```
# API Call Example (Shell)  
curl -X GET "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-  
41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/"  
-H "Authorization: Api-Key YOUR_API_KEY"  
  
# Please replace YOUR_API_KEY and verify the request data before  
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/", config)
  .then(response => {
    console.log('Successfully retrieved response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });

```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully retrieved response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->get("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-
446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code: 200

### Example Response Structure

```
{
  "id": string (uuid)
  "name": string // Optional
}
```

## Response Example

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
```

```
        "name": "Response String"  
    }
```

- \*Update Label\*

PUT `/api/knowledge-bases/{knowledgeBasePk}/labels/{id}/`

#### Parameters

Parameter Name	Required	Type	Description
<code>id</code>	V	string	A UUID string identifying this knowledge base label.
<code>knowledgeBasePk</code>	V	string	

#### Request Body

##### Request Parameters

Field	Type	Required	Description
<code>name</code>	string	Yes	

#### Request Structure Example

```
{  
  "name": string  
}
```

#### Request Sample Value

```
{  
  "name": "Sample Name"  
}
```

#### Code Example

```
# API Call Example (Shell)
curl -X PUT "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "name": "Example Name"
}'

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "name": "Example Name"
};

axios.put("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/", data,
config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "name": "Example Name"
}

response = requests.put(url, json=data, headers=headers)
try:
    print("Successfully got response:")
    print(response.json())
except Exception as e:
    print("Request failed:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->put("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-
446655440000/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY',
        'Content-Type' => 'application/json'
    ],
    'json' => [
        "name": "Sample Name"
    ]
]);

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code: 200

## Response Structure Example

```
{
  "id": string (uuid)
  "name?: string // Optional
}
```

## Example Response Value

```
{  
  "id": "550e8400-e29b-41d4-a716-446655440000",  
  "name": "Response String"  
}
```

- ### Partially Update Label\*\*

PATCH `/api/knowledge-bases/{knowledgeBasePk}/labels/{id}/`

#### Parameters

Parameter Name	Required	Type	Description
<code>id</code>	V	string	A UUID string identifying this knowledge base label.
<code>knowledgeBasePk</code>	V	string	

#### Request Body

##### Request Parameters

Field	Type	Required	Description
<code>name</code>	string	No	

#### Request Structure Example

```
{  
  "name"??: string // Optional  
}
```

#### Request Sample Value

```
{  
  "name": "Example Name"  
}
```

#### Code Example

```
# Call API Example (Shell)
curl -X PATCH "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "name": "Example Name"
}'

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "name": "Example Name"
};

axios.patch("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/",
  data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "name": "Sample Name"
}

response = requests.patch(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->patch("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-
446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "name": "Sample Name"
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred with the request: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code: 200

## Response Structure Example

```
{
  "id": string (uuid)
  "name?: string // Optional
}
```

## Response Example Value

```
{  
  "id": "550e8400-e29b-41d4-a716-446655440000",  
  "name": "Response String"  
}
```

- ### Delete Label

**DELETE /api/knowledge-bases/{knowledgeBasePk}/labels/{id}/**

#### Parameters

Parameter Name	Required	Type	Description
<b>id</b>	V	string	A UUID string identifying this knowledge base label.
<b>knowledgeBasePk</b>	V	string	

#### Code Example

##### Shell/Bash

```
# API Call Example (Shell)  
curl -X DELETE "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-  
41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/"  
-H "Authorization: Api-Key YOUR_API_KEY"  
  
# Please replace YOUR_API_KEY and verify the request data before  
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

// Request payload
const data = null;

axios.delete("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/",
  data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.delete(url, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred with the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->delete("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/labels/550e8400-e29b-41d4-a716-
446655440000/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY'
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred during the request: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code	Description
204	No response body

\*

## Upload File to Knowledge Base

POST /api/knowledge-bases/{knowledgeBasePk}/files/

### Parameters

Parameter Name	Required	Type	Description
knowledgeBasePk	V	string	
fileType	X	string	File type filter, e.g., pdf, docx, txt, xlsx, etc.
knowledgeBase	X	string	Knowledge Base ID
page	X	integer	A page number within the paginated result set.
pageSize	X	integer	Number of results to return per page.
query	X	string	Search keyword, can search for file name or file ID.
status	X	string	File status filter. Possible values: <code>initial</code> , <code>processing</code> , <code>done</code> , <code>deleting</code> , <code>deleted</code> , <code>failed</code> .

## Request Body

### Request Parameters

Field	Type	Required	Description
files	array[ChatbotFileCreate]	Yes	Supports batch upload of multiple files, with a limit of 50 files per request

### Request Structure Example

```
{
  "files": [ // Supports batch upload of multiple files, with a limit of 50
    files per request
    {
      "filename": string // File name
      "file": string (uri) // The file to be uploaded
      "knowledgeBase?": // Optional
      {
        "id": string (uuid)
      }
      "parser?": string (uuid) // Optional
      "labels?": [ // Optional
        {
          "id": string (uuid)
        }
      ]
      "rawUserDefineMetadata?": object // Optional
    }
  ]
}
```

```
]  
}
```

## Request Sample Value

```
{  
  "files": [  
    {  
      "filename": "my-file.pdf",  
      "file": "string",  
      "knowledgeBase": {  
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",  
        "name": "my-knowledge-base"  
      },  
      "parser": "3fa85f64-5717-4562-b3fc-2c963f66afa6",  
      "labels": [  
        {  
          "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",  
          "name": "my-label"  
        }  
      ],  
      "rawUserDefineMetadata": "string"  
    },  
    {  
      "filename": "you-can-upload-multiple-files.pdf",  
      "file": "string",  
      "knowledgeBase": {  
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",  
        "name": "string"  
      },  
      "parser": "3fa85f64-5717-4562-b3fc-2c963f66afa6"  
    }  
  ]  
}
```

## Code Example

```
# Call API Example (Shell)
curl -X POST "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/?fileType=example&knowledgeBase=550e8400-e29b-41d4-a716-446655440000&page=1&pageSize=1&query=example&status=deleted"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
  "files": [
    {
      "filename": "my-file.pdf",
      "file": "string",
      "knowledgeBase": {
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "name": "my-knowledge-base"
      },
      "parser": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "labels": [
        {
          "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
          "name": "my-label"
        }
      ],
      "rawUserDefineMetadata": "string"
    },
    {
      "filename": "you-can-upload-multiple-files.pdf",
      "file": "string",
      "knowledgeBase": {
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "name": "string"
      },
      "parser": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
    }
  ]
}'.

# Please replace YOUR_API_KEY and verify the request data before execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "files": [
    {
      "filename": "my-file.pdf",
      "file": "string",
      "knowledgeBase": {
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "name": "my-knowledge-base"
      },
      "parser": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "labels": [
        {
          "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
          "name": "my-label"
        }
      ],
      "rawUserDefineMetadata": "string"
    },
    {
      "filename": "you-can-upload-multiple-files.pdf",
      "file": "string",
      "knowledgeBase": {
        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
        "name": "string"
      },
      "parser": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
    }
  ]
};

axios.post("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/?fileType=example&knowledgeBase=550e8400-
```

```
e29b-41d4-a716-
446655440000&page=1&pageSize=1&query=example&status=deleted", data,
config)
.then(response => {
  console.log('Successfully received response:');
  console.log(response.data);
})
.catch(error => {
  console.error('An error occurred with the request:');
  console.error(error.response ?.data || error.message);
});
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/?fileType=example&knowledgeBase=550e8400-e29b-41d4-a716-446655440000&page=1&pageSize=1&query=example&status=deleted"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "files": [
        {
            "filename": "my-file.pdf",
            "file": "string",
            "knowledgeBase": {
                "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
                "name": "my-knowledge-base"
            },
            "parser": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
            "labels": [
                {
                    "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
                    "name": "my-label"
                }
            ],
            "rawUserDefineMetadata": "string"
        },
        {
            "filename": "you-can-upload-multiple-files.pdf",
            "file": "string",
            "knowledgeBase": {
                "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
                "name": "string"
            },
            "parser": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
        }
    ]
}

response = requests.post(url, json=data, headers=headers)
try:
```

```
print("Successfully received response:")
print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->post("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/files/?
fileType=example&knowledgeBase=550e8400-e29b-41d4-a716-
446655440000&page=1&pageSize=1&query=example&status=deleted", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY',
        'Content-Type' => 'application/json'
    ],
    'json' => [
        "files": [
            {
                "filename": "my-file.pdf",
                "file": "string",
                "knowledgeBase": {
                    "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
                    "name": "my-knowledge-base"
                },
                "parser": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
                "labels": [
                    {
                        "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
                        "name": "my-label"
                    }
                ],
                "rawUserDefineMetadata": "string"
            },
            {
                "filename": "you-can-upload-multiple-files.pdf",
                "file": "string",
                "knowledgeBase": {
                    "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
                    "name": "string"
                },
                "parser": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
            }
        ]
    }
}]
```

```
]);  
  
    $data = json_decode($response->getBody(), true);  
    echo "Successfully received response:\n";  
    print_r($data);  
} catch (Exception $e) {  
    echo 'Request failed: ' . $e->getMessage();  
}  
?>
```

## Response Body

Status Code: 201

### Sample Response Structure

```
{  
    "count": integer  
    "next"??: string (uri) // Optional  
    "previous"??: string (uri) // Optional  
    "results": [  
        {  
            "id": string (uuid)  
            "filename": string // File name  
            "file": string (uri) // File to upload  
            "fileType": string  
            "knowledgeBase"??: // Optional  
            {  
                "id": string (uuid)  
                "name": string  
            }  
            "size": integer  
            "status":  
            {  
            }  
            "parser": {  
                "id": string (uuid)  
                "name": string  
                "provider":  
                {  
                }  
                "order"??: integer // Optional  
            }  
        }  
    ]  
}
```

```
        }
      "labels"? : [ // Optional
        {
          "id": string (uuid)
          "name": string
        }
      ]
    "rawUserDefineMetadata"? : object // Optional
    "createdAt": string (timestamp)
  }
]
}
```

## Response Example

```
{
  "count": 123,
  "next": "http://api.example.org/accounts/?page=4",
  "previous": "http://api.example.org/accounts/?page=2",
  "results": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "filename": "Response String",
      "file": "https://example.com/file.jpg",
      "fileType": "Response String",
      "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000",
        "name": "Response String"
      },
      "size": 456,
      "status": {},
      "parser": {
        "id": "550e8400-e29b-41d4-a716-446655440000",
        "name": "Response String",
        "provider": {},
        "order": 456
      },
      "labels": [
        {
          "id": "550e8400-e29b-41d4-a716-446655440000",
          "name": "Response String"
        }
      ],
      "rawUserDefineMetadata": null,
      "createdAt": "Response String"
    }
  ]
}
```

```
]  
}
```

- ### Batch Delete Files\*\*

POST /api/knowledge-bases/{knowledgeBasePk}/files/batch-delete/

#### Parameters

Parameter Name	Required	Type	Description
knowledgeBasePk	V	string	

#### Request Body

##### Request Parameters

Field	Type	Required	Description
ids	array[string]	Yes	

##### Request Structure Example

```
{  
  "ids": [  
    string (uuid)  
  ]  
}
```

##### Request Sample Value

```
{  
  "ids": [  
    "550e8400-e29b-41d4-a716-446655440000"  
  ]  
}
```

#### Code Example

## Shell/Bash

```
# API Call Example (Shell)
curl -X POST "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/files/batch-delete/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
  "ids": [
    "550e8400-e29b-41d4-a716-446655440000"
  ]
}'
```

*# Please replace YOUR\_API\_KEY and verify the request data before execution.*

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "ids": [
    "550e8400-e29b-41d4-a716-446655440000"
  ]
};

axios.post("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/batch-delete/" , data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/batch-delete/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "ids": [
        "550e8400-e29b-41d4-a716-446655440000"
    ]
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("Request failed:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client→post("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/files/batch-delete/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "ids": [
                "550e8400-e29b-41d4-a716-446655440000"
            ]
        ]
    ]);
}

$data = json_decode($response→getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Error during request: ' . $e→getMessage();
}
?>

```

## Response Content

Status Code	Description
204	No response body

## List Files in a Knowledge Base

GET </api/knowledge-bases/{knowledgeBasePk}/files/>

## Parameters

Parameter Name	Required	Type	Description
knowledgeBasePk	V	string	
fileType	X	string	File type filter, e.g., pdf, docx, txt, xlsx, etc.
knowledgeBase	X	string	Knowledge Base ID
page	X	integer	A page number within the paginated result set.
pageSize	X	integer	Number of results to return per page.
query	X	string	Search keyword, can search by file name or file ID.
status	X	string	File status filter. Possible values: <code>initial</code> , <code>processing</code> , <code>done</code> , <code>deleting</code> , <code>deleted</code> , <code>failed</code> . <code>initial</code> :Initial; <code>processing</code> : Processing; <code>done</code> :Done; <code>deleting</code> :Deleting ; <code>deleted</code> :Deleted ; `faile...

## Code Example

### Shell/Bash

```
# API Call Example (Shell)
curl -X GET "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/files/?fileType=example&knowledgeBase=550e8400-
e29b-41d4-a716-
446655440000&page=1&pageSize=1&query=example&status=deleted"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/?fileType=example&knowledgeBase=550e8400-e29b-41d4-a716-446655440000&page=1&pageSize=1&query=example&status=deleted",
config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });

```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/?fileType=example&knowledgeBase=550e8400-e29b-41d4-a716-446655440000&page=1&pageSize=1&query=example&status=deleted"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->get("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/files/?
fileType=example&knowledgeBase=550e8400-e29b-41d4-a716-
446655440000&page=1&pageSize=1&query=example&status=deleted", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY'
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Response received successfully:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code: 200

### Example Response Structure

```
{
  "count": integer
  "next?: string (uri) // Optional
  "previous?: string (uri) // Optional
  "results": [
    {
      "id": string (uuid)
      "filename": string // File name
      "file": string (uri) // File to upload
      "fileType": string
    }
  ]
}
```

```
"knowledgeBase?": // Optional
{
  "id": string (uuid)
  "name": string
}
"size": integer
"status":
{
}
"parser": {
{
  "id": string (uuid)
  "name": string
  "provider":
{
}
}
  "order?": integer // Optional
}
}
"labels?": [ // Optional
{
  "id": string (uuid)
  "name": string
}
]
"rawUserDefineMetadata?": object // Optional
"createdAt": string (timestamp)
}
]
}
```

## Example Response Value

```
{
  "count": 123,
  "next": "http://api.example.org/accounts/?page=4",
  "previous": "http://api.example.org/accounts/?page=2",
  "results": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "filename": "Response String",
      "file": "https://example.com/file.jpg",
      "fileType": "Response String",
      "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000",
        "name": "Response String"
      },
    }
  ]
}
```

```

    "size": 456,
    "status": {},
    "parser": {
        "id": "550e8400-e29b-41d4-a716-446655440000",
        "name": "Response String",
        "provider": {},
        "order": 456
    },
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000",
            "name": "Response String"
        }
    ],
    "rawUserDefineMetadata": null,
    "createdAt": "Response String"
}
]
}

```

- ### Get Specific File Details\*\*

GET [/api/knowledge-bases/{knowledgeBasePk}/files/{id}/](#)

#### Parameters

Parameter Name	Required	Type	Description
<code>id</code>	V	string	A UUID string identifying this Chatbot File.
<code>knowledgeBasePk</code>	V	string	

#### Code Example

## Shell/Bash

```
# API Call Example (Shell)
curl -X GET "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

## JavaScript

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-
a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/", config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully retrieved response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->get("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-
446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully retrieved response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred with the request: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code: 200

### Example Response Structure

```
{
  "id": string (uuid)
  "filename": string // File name
  "file": string (uri) // File to upload
  "fileType": string
  "knowledgeBase"??: // Optional
  {
    "id": string (uuid)
    "name": string
  }
  "size": integer
}
```

```
"status":  
{  
}  
}  
"parser{  
    "id": string (uuid)  
    "name": string  
    "provider":  
    {  
    }  
    "order"?: integer // Optional  
}  
}  
}  
"labels??: [ // Optional  
{  
    "id": string (uuid)  
    "name": string  
}  
]  
"rawUserDefineMetadata??: object // Optional  
"createdAt: string (timestamp)  
}
```

## Response Example Value

```
{  
    "id": "550e8400-e29b-41d4-a716-446655440000",  
    "filename": "Response String",  
    "file": "https://example.com/file.jpg",  
    "fileType": "Response String",  
    "knowledgeBase": {  
        "id": "550e8400-e29b-41d4-a716-446655440000",  
        "name": "Response String"  
    },  
    "size": 456,  
    "status": {},  
    "parser        "id": "550e8400-e29b-41d4-a716-446655440000",  
        "name": "Response String",  
        "provider": {},  
        "order": 456  
    },  
    "labels        {  
            "id": "550e8400-e29b-41d4-a716-446655440000",  
            "name": "Response String"  
        }  
    ]  
}
```

```

],
"rawUserDefineMetadata": null,
"createdAt": "Response String"
}

```

- ### Update File Information

**PUT /api/knowledge-bases/{knowledgeBasePk}/files/{id}/**

#### Parameters

Parameter Name	Required	Type	Description
<b>id</b>	V	string	A UUID string identifying this Chatbot File.
<b>knowledgeBasePk</b>	V	string	

#### Request Body

##### Request Parameters

Field	Type	Required	Description
<b>filename</b>	string	Yes	File name
<b>file</b>	string (uri)	Yes	The file to upload
<b>knowledgeBase</b>	object (contains 2 properties: id, name)	No	
<b>knowledgeBase.id</b>	string (uuid)	Yes	
<b>parser</b>	string (uuid)	No	
<b>labels</b>	array[IdName]	No	
<b>rawUserDefineMetadata</b>	object	No	

#### Request Structure Example

```

{
  "filename": string // File name
  "file": string (uri) // The file to upload
  "knowledgeBase?": // Optional
  {
    "id": string (uuid)
  }
}

```

```
}

"parser"??: string (uuid) // Optional
"labels"??: [ // Optional
{
  "id": string (uuid)
}
]
"rawUserDefineMetadata"??: object // Optional
}
```

## Request Sample Value

```
{
  "filename": "document.pdf",
  "file": "https://example.com/file.jpg",
  "knowledgeBase": {
    "id": "550e8400-e29b-41d4-a716-446655440000"
  },
  "parser": "550e8400-e29b-41d4-a716-446655440000",
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "rawUserDefineMetadata": null
}
```

## Code Example

```
# API Call Example (Shell)
curl -X PUT "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "filename": "document.pdf",
    "file": "https://example.com/file.jpg",
    "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000"
    },
    "parser": "550e8400-e29b-41d4-a716-446655440000",
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "rawUserDefineMetadata": null
}'

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "filename": "document.pdf",
  "file": "https://example.com/file.jpg",
  "knowledgeBase": {
    "id": "550e8400-e29b-41d4-a716-446655440000"
  },
  "parser": "550e8400-e29b-41d4-a716-446655440000",
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "rawUserDefineMetadata": null
};

axios.put("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/", data,
config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "filename": "document.pdf",
    "file": "https://example.com/file.jpg",
    "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000"
    },
    "parser": "550e8400-e29b-41d4-a716-446655440000",
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "rawUserDefineMetadata": null
}

response = requests.put(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->put("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-
446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "filename": "document.pdf",
            "file": "https://example.com/file.jpg",
            "knowledgeBase": [
                "id": "550e8400-e29b-41d4-a716-446655440000"
            ],
            "parser": "550e8400-e29b-41d4-a716-446655440000",
            "labels": [
                [
                    {
                        "id": "550e8400-e29b-41d4-a716-446655440000"
                    }
                ],
                "rawUserDefineMetadata": null
            ]
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Response received successfully:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

Status Code: 200

## Response Structure Example

```
{  
  "id": string (uuid)  
  "filename": string // File name  
  "file": string (uri) // File to upload  
  "fileType": string  
  "knowledgeBase"??: // Optional  
  {  
    "id": string (uuid)  
    "name": string  
  }  
  "size": integer  
  "status":  
  {  
  }  
  "parser": {  
    "id": string (uuid)  
    "name": string  
    "provider":  
    {  
    }  
    "order"??: integer // Optional  
  }  
  }  
  "labels"??: [ // Optional  
  {  
    "id": string (uuid)  
    "name": string  
  }  
]  
  "rawUserDefineMetadata"??: object // Optional  
  "createdAt": string (timestamp)  
}
```

## Example Response Value

```
{  
  "id": "550e8400-e29b-41d4-a716-446655440000",  
  "filename": "Response String",  
  "file": "https://example.com/file.jpg",  
  "fileType": "Response String",  
  "knowledgeBase": {
```

```

    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response String"
},
"size": 456,
"status": {},
"parser": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response String",
    "provider": {},
    "order": 456
},
"labels": [
{
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response String"
}
],
"rawUserDefineMetadata": null,
"createdAt": "Response String"
}

```

- ### Partially Update a File

**PATCH [/api/knowledge-bases/{knowledgeBasePk}/files/{id}/](#)**

#### Parameters

Parameter Name	Required	Type	Description
<b>id</b>	V	string	A UUID string identifying this Chatbot File.
<b>knowledgeBasePk</b>	V	string	

#### Request Body

##### Request Parameters

Field	Type	Required	Description
<b>filename</b>	string	No	File name
<b>file</b>	string (uri)	No	The file to be uploaded
<b>knowledgeBase</b>	object (contains 2 properties: id, name)	No	
<b>knowledgeBase.id</b>	string (uuid)	Yes	

Field	Type	Required	Description
parser	string (uuid)	No	
labels	array[IdName]	No	
rawUserDefineMetadata	object	No	

### Request Structure Example

```
{
  "filename"??: string // File name (Optional)
  "file"??: string (uri) // File to upload (Optional)
  "knowledgeBase"??: // Optional
  {
    "id": string (uuid)
  }
  "parser"??: string (uuid) // Optional
  "labels"??: [ // Optional
  {
    "id": string (uuid)
  }
]
  "rawUserDefineMetadata"??: object // Optional
}
```

### Request Example Value

```
{
  "filename": "document.pdf",
  "file": "https://example.com/file.jpg",
  "knowledgeBase": {
    "id": "550e8400-e29b-41d4-a716-446655440000"
  },
  "parser": "550e8400-e29b-41d4-a716-446655440000",
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "rawUserDefineMetadata": null
}
```

### Code Example

```
# API Call Example (Shell)
curl -X PATCH "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "filename": "document.pdf",
    "file": "https://example.com/file.jpg",
    "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000"
    },
    "parser": "550e8400-e29b-41d4-a716-446655440000",
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "rawUserDefineMetadata": null
}'

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "filename": "document.pdf",
  "file": "https://example.com/file.jpg",
  "knowledgeBase": {
    "id": "550e8400-e29b-41d4-a716-446655440000"
  },
  "parser": "550e8400-e29b-41d4-a716-446655440000",
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "rawUserDefineMetadata": null
};

axios.patch("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/", data,
config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "filename": "document.pdf",
    "file": "https://example.com/file.jpg",
    "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000"
    },
    "parser": "550e8400-e29b-41d4-a716-446655440000",
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "rawUserDefineMetadata": None
}

response = requests.patch(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->patch("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-
446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "filename": "document.pdf",
            "file": "https://example.com/file.jpg",
            "knowledgeBase": [
                "id": "550e8400-e29b-41d4-a716-446655440000"
            ],
            "parser": "550e8400-e29b-41d4-a716-446655440000",
            "labels": [
                [
                    {
                        "id": "550e8400-e29b-41d4-a716-446655440000"
                    }
                ],
                "rawUserDefineMetadata": null
            ]
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

Status Code: 200

## Response Structure Example

```
{  
  "id": string (uuid)  
  "filename": string // File name  
  "file": string (uri) // File to upload  
  "fileType": string  
  "knowledgeBase"??: // Optional  
  {  
    "id": string (uuid)  
    "name": string  
  }  
  "size": integer  
  "status":  
  {  
  }  
  "parser": {  
    "id": string (uuid)  
    "name": string  
    "provider":  
    {  
    }  
    "order"??: integer // Optional  
  }  
  }  
  "labels"??: [ // Optional  
  {  
    "id": string (uuid)  
    "name": string  
  }  
]  
  "rawUserDefineMetadata"??: object // Optional  
  "createdAt": string (timestamp)  
}
```

## Response Example Value

```
{  
  "id": "550e8400-e29b-41d4-a716-446655440000",  
  "filename": "Response String",  
  "file": "https://example.com/file.jpg",  
  "fileType": "Response String",  
  "knowledgeBase": {
```

```

    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response String"
},
"size": 456,
"status": {},
"parser": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response String",
    "provider": {},
    "order": 456
},
"labels": [
{
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response String"
}
],
"rawUserDefineMetadata": null,
"createdAt": "Response String"
}

```

- ### Batch Reparse Files

PATCH [/api/knowledge-bases/{knowledgeBasePk}/files/batch-reparse/](#)

#### Parameters

Parameter Name	Required	Type	Description
<b>knowledgeBasePk</b>	V	string	
<b>fileType</b>	X	string	File type filter, e.g., pdf, docx, txt, xlsx, etc.
<b>knowledgeBase</b>	X	string	Knowledge Base ID
<b>page</b>	X	integer	A page number within the paginated result set.
<b>pageSize</b>	X	integer	Number of results to return per page.
<b>query</b>	X	string	Search keyword, can search by file name or file ID.
<b>status</b>	X	string	File status filter. Possible values: <b>initial</b> , <b>processing</b> , <b>done</b> , <b>deleting</b> , <b>deleted</b> , <b>failed</b> . <b>initial</b> :Initial; <b>processing</b> : Processing; <b>done</b> :Done; <b>deleting</b> :Deleting ; <b>deleted</b> :Deleted ; `faile...

## Code Example

### Shell/Bash

```
# API Call Example (Shell)
curl -X PATCH "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
  "filename": "document.pdf",
  "file": "https://example.com/file.jpg",
  "knowledgeBase": {
    "id": "550e8400-e29b-41d4-a716-446655440000"
  },
  "parser": "550e8400-e29b-41d4-a716-446655440000",
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "rawUserDefineMetadata": null
}'

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "filename": "document.pdf",
  "file": "https://example.com/file.jpg",
  "knowledgeBase": {
    "id": "550e8400-e29b-41d4-a716-446655440000"
  },
  "parser": "550e8400-e29b-41d4-a716-446655440000",
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "rawUserDefineMetadata": null
};

axios.patch("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/", data,
config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "filename": "document.pdf",
    "file": "https://example.com/file.jpg",
    "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000"
    },
    "parser": "550e8400-e29b-41d4-a716-446655440000",
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "rawUserDefineMetadata": null
}

response = requests.patch(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->patch("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-
446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "filename": "document.pdf",
            "file": "https://example.com/file.jpg",
            "knowledgeBase": [
                "id": "550e8400-e29b-41d4-a716-446655440000"
            ],
            "parser": "550e8400-e29b-41d4-a716-446655440000",
            "labels": [
                [
                    {
                        "id": "550e8400-e29b-41d4-a716-446655440000"
                    }
                ],
                "rawUserDefineMetadata": null
            ]
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

Status Code: 200

## Response Structure Example

```
{  
  "count": integer  
  "next"??: string (uri) // Optional  
  "previous"??: string (uri) // Optional  
  "results": [  
    {  
      "id": string (uuid)  
      "filename": string // File name  
      "file": string (uri) // File to upload  
      "fileType": string  
      "knowledgeBase"??: // Optional  
      {  
        "id": string (uuid)  
        "name": string  
      }  
      "size": integer  
      "status":  
      {  
      }  
      "parser": {  
        {  
          "id": string (uuid)  
          "name": string  
          "provider":  
          {  
          }  
          "order"??: integer // Optional  
        }  
      }  
      "labels"??: [ // Optional  
        {  
          "id": string (uuid)  
          "name": string  
        }  
      ]  
      "rawUserDefineMetadata"??: object // Optional  
      "createdAt": string (timestamp)  
    }  
  ]  
}
```

## Response Example Value

```
{
  "count": 123,
  "next": "http://api.example.org/accounts/?page=4",
  "previous": "http://api.example.org/accounts/?page=2",
  "results": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "filename": "Response String",
      "file": "https://example.com/file.jpg",
      "fileType": "Response String",
      "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000",
        "name": "Response String"
      },
      "size": 456,
      "status": {},
      "parser": {
        "id": "550e8400-e29b-41d4-a716-446655440000",
        "name": "Response String",
        "provider": {},
        "order": 456
      },
      "labels": [
        {
          "id": "550e8400-e29b-41d4-a716-446655440000",
          "name": "Response String"
        }
      ],
      "rawUserDefineMetadata": null,
      "createdAt": "Response String"
    }
  ]
}
```

- ### Delete File

**DELETE** `/api/knowledge-bases/{knowledgeBasePk}/files/{id}/`

#### Parameters

Parameter Name	Required	Type	Description
<code>id</code>	V	string	A UUID string identifying this Chatbot File.
<code>knowledgeBasePk</code>	V	string	

## Code Example

### Shell/Bash

```
# API Call Example (Shell)
curl -X DELETE "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

// Request payload
const data = null;

axios.delete("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/", data,
config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.delete(url, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred with the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->delete("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/files/550e8400-e29b-41d4-a716-
446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred during the request: ' . $e->getMessage();
}
?>

```

## Response Content

Status Code	Description
204	No response body

\*

## Create a new FAQ

POST /api/knowledge-bases/{knowledgeBasePk}/faqs/

### Parameters

Parameter Name	Required	Type	Description
knowledgeBasePk	V	string	

## Request Body

### Request Parameters

Field	Type	Required	Description
question	string	Yes	
answer	string	Yes	
answerMediaUrls	object	No	URLs for media files such as images and videos in the answer.
labels	array[IdName]	No	
rawUserDefineMetadata	object	No	
knowledgeBase	object (with 2 properties: id, name)	No	
knowledgeBase.id	string (uuid)	Yes	

### Request Structure Example

```
{
  "question": string
  "answer": string
  "answerMediaUrls"??: object // Stores the URLs of media files such as images, videos, etc., in the answer (optional)
  "labels"??: [ // (optional)
    {
      "id": string (uuid)
    }
  ]
  "rawUserDefineMetadata"??: object // (optional)
  "knowledgeBase"??: // (optional)
  {
    "id": string (uuid)
  }
}
```

## Request Sample Value

```
{  
    "question": "Sample string",  
    "answer": "Sample string",  
    "answerMediaUrls": null,  
    "labels": [  
        {  
            "id": "550e8400-e29b-41d4-a716-446655440000"  
        }  
    ],  
    "rawUserDefineMetadata": null,  
    "knowledgeBase": {  
        "id": "550e8400-e29b-41d4-a716-446655440000"  
    }  
}
```

## Code Example

```
# API Call Example (Shell)
curl -X POST "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/faqs/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "question": "Example string",
    "answer": "Example string",
    "answerMediaUrls": null,
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "rawUserDefineMetadata": null,
    "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000"
    }
}'
```

*# Please replace YOUR\_API\_KEY and verify the request data before execution.*

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "question": "Sample string",
  "answer": "Sample string",
  "answerMediaUrls": null,
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "rawUserDefineMetadata": null,
  "knowledgeBase": {
    "id": "550e8400-e29b-41d4-a716-446655440000"
  }
};

axios.post("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "question": "Example string",
    "answer": "Example string",
    "answerMediaUrls": null,
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "rawUserDefineMetadata": null,
    "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000"
    }
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->post("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/faqs/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "question": "Sample string",
            "answer": "Sample string",
            "answerMediaUrls": null,
            "labels": [
                {
                    "id": "550e8400-e29b-41d4-a716-446655440000"
                }
            ],
            "rawUserDefineMetadata": null,
            "knowledgeBase": [
                "id": "550e8400-e29b-41d4-a716-446655440000"
            ]
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 201

## Example Response Structure

```
{  
  "id": string (uuid)  
  "question": string  
  "answer": string  
  "answerMediaUrls"??: object // Stores the URLs of media files such as images,  
videos, etc. in the answer (optional)  
  "hitsCount": integer  
  "embeddingTokensCount": integer // The number of embedding tokens used when  
creating this FAQ  
  "labels"??: [ // optional  
    {  
      "id": string (uuid)  
      "name": string  
    }  
  ]  
  "rawUserDefineMetadata"??: object // optional  
  "knowledgeBase"??: // optional  
  {  
    "id": "string (uuid)",  
    "name": "string"  
  }  
}
```

## Response Example

```
{  
  "id": "550e8400-e29b-41d4-a716-446655440000",  
  "question": "Response string",  
  "answer": "Response string",  
  "answerMediaUrls": null,  
  "hitsCount": 456,  
  "embeddingTokensCount": 456,  
  "labels": [  
    {  
      "id": "550e8400-e29b-41d4-a716-446655440000",  
      "name": "Response string"  
    }  
  ],  
  "rawUserDefineMetadata": null,  
  "knowledgeBase": {  
    "id": "550e8400-e29b-41d4-a716-446655440000",  
    "name": "Response string"
```

```
    }  
}
```

- ### Batch Delete FAQs\*\*

POST `/api/knowledge-bases/{knowledgeBasePk}/faqs/batch-delete/`

#### Parameters

Parameter Name	Required	Type	Description
<code>knowledgeBasePk</code>	V	string	

#### Request Body

#### Request Parameters

Field	Type	Required	Description
<code>ids</code>	array[string]	No	

#### Request Structure Example

```
{  
  "ids"??: [ // Optional  
    string (uuid)  
  ]  
}
```

#### Request Sample Value

```
{  
  "ids": [  
    "550e8400-e29b-41d4-a716-446655440000"  
  ]  
}
```

#### Code Example

## Shell/Bash

```
# API Call Example (Shell)
curl -X POST "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/faqs/batch-delete/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
  "ids": [
    "550e8400-e29b-41d4-a716-446655440000"
  ]
}'
```

*# Please replace YOUR\_API\_KEY and verify the request data before execution.*

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "ids": [
    "550e8400-e29b-41d4-a716-446655440000"
  ]
};

axios.post("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/batch-delete/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/batch-delete/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "ids": [
        "550e8400-e29b-41d4-a716-446655440000"
    ]
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client→post("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/faqs/batch-delete/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "ids": [
                "550e8400-e29b-41d4-a716-446655440000"
            ]
        ]
    ]);
}

$data = json_decode($response→getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e→getMessage();
}
?>

```

## Response Content

Status Code	Description
204	No response body

\*

## List all FAQs for a specific knowledge base

GET </api/knowledge-bases/{knowledgeBasePk}/faqs/>

## Parameters

Parameter Name	Required	Type	Description
knowledgeBasePk	V	string	
page	X	integer	A page number within the paginated result set.
pageSize	X	integer	Number of results to return per page.
query	X	string	

## Code Example

### Shell/Bash

```
# API Call Example (Shell)
curl -X GET "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/faqs/?page=1&pageSize=1&query=example"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/?page=1&pageSize=1&query=example", config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });

```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/?page=1&pageSize=1&query=example"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully got response:")
    print(response.json())
except Exception as e:
    print("Request failed:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->get("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/faqs/?
page=1&pageSize=1&query=example", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred during the request: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code: 200

### Response Structure Example

```
{
  "count": integer
  "next"?: string (uri) // Optional
  "previous"?: string (uri) // Optional
  "results": [
    {
      "id": string (uuid)
      "question": string
      "answer": string
      "answerMediaUrls"?: object // Stores URLs for media files like images,
                                 videos, etc., in the answer (Optional)
    }
  ]
}
```

```

    "hitsCount": integer
    "embeddingTokensCount": integer // The number of embedding tokens used
when creating this FAQ
    "labels"??: [ // Optional
      {
        "id": string (uuid)
        "name": string
      }
    ]
    "rawUserDefineMetadata"??: object // Optional
    "knowledgeBase"??: // Optional
    {
      "id": string (uuid)
      "name": string
    }
  }
]
}

```

## Response Example

```
{
  "count": 123,
  "next": "http://api.example.org/accounts/?page=4",
  "previous": "http://api.example.org/accounts/?page=2",
  "results": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "question": "Response string",
      "answer": "Response string",
      "answerMediaUrls": null,
      "hitsCount": 456,
      "embeddingTokensCount": 456,
      "labels": [
        {
          "id": "550e8400-e29b-41d4-a716-446655440000",
          "name": "Response string"
        }
      ],
      "rawUserDefineMetadata": null,
      "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000",
        "name": "Response string"
      }
    }
  ]
}
```

```
]  
}
```

- ### Get Specific FAQ Details\*\*

GET </api/knowledge-bases/{knowledgeBasePk}/faqs/{id}/>

#### Parameters

Parameter Name	Required	Type	Description
<code>id</code>	V	string	A UUID string identifying this FAQ.
<code>knowledgeBasePk</code>	V	string	

#### Code Example

##### Shell/Bash

```
# API Call Example (Shell)  
curl -X GET "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-  
41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/"  
-H "Authorization: Api-Key YOUR_API_KEY"  
  
# Please replace YOUR_API_KEY and verify the request data before  
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/", config)
  .then(response => {
    console.log('Successfully retrieved response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });

```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully retrieved response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->get("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-
446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully retrieved response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code: 200

### Response Structure Example

```
{
  "id": string (uuid)
  "question": string
  "answer": string
  "answerMediaUrls"??: object // Stores the URLs of media files such as images,
videos, etc. in the answer (optional)
  "hitsCount": integer
  "embeddingTokensCount": integer // The number of embedding tokens used when
creating this FAQ
  "labels"??: [ // optional
  {
    "label": string
  }
]
```

```

        "id": string (uuid)
        "name": string
    }
]
"rawUserDefineMetadata"??: object // optional
"knowledgeBase"??: // optional
{
    "id": string (uuid)
    "name": string
}
}

```

### Example Response Value

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "question": "Response string",
  "answer": "Response string",
  "answerMediaUrls": null,
  "hitsCount": 456,
  "embeddingTokensCount": 456,
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "name": "Response string"
    }
  ],
  "rawUserDefineMetadata": null,
  "knowledgeBase": {
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response string"
  }
}
```

- ### Update FAQ

PUT /api/knowledge-bases/{knowledgeBasePk}/faqs/{id}/

#### Parameters

Parameter Name	Required	Type	Description
id	V	string	A UUID string identifying this FAQ.
knowledgeBasePk	V	string	

## Request Body

### Request Parameters

Field	Type	Required	Description
question	string	Yes	
answer	string	Yes	
answerMediaUrls	object	No	URLs for media files such as images and videos in the answer
labels	array[IdName]	No	
rawUserDefineMetadata	object	No	
knowledgeBase	object (with 2 properties: id, name)	No	
knowledgeBase.id	string (uuid)	Yes	

### Request Structure Example

```
{
  "question": string
  "answer": string
  "answerMediaUrls"??: object // Stores the URLs of media files such as images, videos, etc. in the answer (optional)
  "labels"??: [ // (optional)
    {
      "id": string (uuid)
    }
  ]
  "rawUserDefineMetadata"??: object // (optional)
  "knowledgeBase"??: // (optional)
  {
    "id": string (uuid)
  }
}
```

### Request Sample Value

```
{  
  "question": "Sample string",  
  "answer": "Sample string",  
  "answerMediaUrls": null,  
  "labels": [  
    {  
      "id": "550e8400-e29b-41d4-a716-446655440000"  
    }  
  ],  
  "rawUserDefineMetadata": null,  
  "knowledgeBase": {  
    "id": "550e8400-e29b-41d4-a716-446655440000"  
  }  
}
```

## Code Example

```
# API Call Example (Shell)
curl -X PUT "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "question": "Example string",
    "answer": "Example string",
    "answerMediaUrls": null,
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "rawUserDefineMetadata": null,
    "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000"
    }
}'
```

*# Please replace YOUR\_API\_KEY and verify the request data before execution.*

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "question": "Sample string",
  "answer": "Sample string",
  "answerMediaUrls": null,
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "rawUserDefineMetadata": null,
  "knowledgeBase": {
    "id": "550e8400-e29b-41d4-a716-446655440000"
  }
};

axios.put("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/", data,
config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "question": "Example string",
    "answer": "Example string",
    "answerMediaUrls": null,
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "rawUserDefineMetadata": null,
    "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000"
    }
}

response = requests.put(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->put("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-
446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "question": "Sample string",
            "answer": "Sample string",
            "answerMediaUrls": null,
            "labels": [
                {
                    "id": "550e8400-e29b-41d4-a716-446655440000"
                }
            ],
            "rawUserDefineMetadata": null,
            "knowledgeBase": [
                "id": "550e8400-e29b-41d4-a716-446655440000"
            ]
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 200

## Example Response Structure

```
{  
  "id": string (uuid)  
  "question": string  
  "answer": string  
  "answerMediaUrls"??: object // Stores the URLs of media files such as images,  
  videos, etc. in the answer (optional)  
  "hitsCount": integer  
  "embeddingTokensCount": integer // The number of embedding tokens used when  
  creating this FAQ  
  "labels"??: [ // optional  
    {  
      "id": string (uuid)  
      "name": string  
    }  
  ]  
  "rawUserDefineMetadata"??: object // optional  
  "knowledgeBase"??: // optional  
  {  
    "id": string (uuid)  
    "name": string  
  }  
}
```

## Response Example Value

```
{  
  "id": "50e8400-e29b-41d4-a716-446655440000",  
  "question": "Response string",  
  "answer": "Response string",  
  "answerMediaUrls": null,  
  "hitsCount": 456,  
  "embeddingTokensCount": 456,  
  "labels": [  
    {  
      "id": "550e8400-e29b-41d4-a716-446655440000",  
      "name": "Response string"  
    }  
  ],  
  "rawUserDefineMetadata": null,  
  "knowledgeBase": {  
    "id": "550e8400-e29b-41d4-a716-446655440000",  
    "name": "Response string"  
  }  
}
```

```
    }
}
```

- ### Partially Update FAQ

PATCH `/api/knowledge-bases/{knowledgeBasePk}/faqs/{id}/`

#### Parameters

Parameter Name	Required	Type	Description
<code>id</code>	V	string	A UUID string identifying this FAQ.
<code>knowledgeBasePk</code>	V	string	

#### Request Body

##### Request Parameters

Field	Type	Required	Description
<code>question</code>	string	No	
<code>answer</code>	string	No	
<code>answerMediaUrls</code>	object	No	URLs for media files such as images and videos in the answer
<code>labels</code>	array[IdName]	No	
<code>rawUserDefineMetadata</code>	object	No	
<code>knowledgeBase</code>	object (with 2 properties: id, name)	No	
<code>knowledgeBase.id</code>	string (uuid)	Yes	

##### Request Structure Example

```
{
  "question"??: string // Optional
  "answer"??: string // Optional
  "answerMediaUrls"??: object // Stores URLs of media files like images, videos, etc., in the answer (Optional)
  "labels"??: [ // Optional
```

```
{  
    "id": string (uuid)  
}  
]  
"rawUserDefineMetadata"??: object // Optional  
"knowledgeBase"??: // Optional  
{  
    "id": string (uuid)  
}  
}
```

## Request Sample Value

```
{  
    "question": "example string",  
    "answer": "example string",  
    "answerMediaUrls": null,  
    "labels": [  
        {  
            "id": "550e8400-e29b-41d4-a716-446655440000"  
        }  
    ],  
    "rawUserDefineMetadata": null,  
    "knowledgeBase": {  
        "id": "550e8400-e29b-41d4-a716-446655440000"  
    }  
}
```

## Code Example

```
# API Call Example (Shell)
curl -X PATCH "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-
41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "question": "Example String",
    "answer": "Example String",
    "answerMediaUrls": null,
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "rawUserDefineMetadata": null,
    "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000"
    }
}'
```

*# Please replace YOUR\_API\_KEY and verify the request data before execution.*

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "question": "Sample string",
  "answer": "Sample string",
  "answerMediaUrls": null,
  "labels": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000"
    }
  ],
  "rawUserDefineMetadata": null,
  "knowledgeBase": {
    "id": "550e8400-e29b-41d4-a716-446655440000"
  }
};

axios.patch("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/", data,
config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "question": "Example string",
    "answer": "Example string",
    "answerMediaUrls": null,
    "labels": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000"
        }
    ],
    "rawUserDefineMetadata": null,
    "knowledgeBase": {
        "id": "550e8400-e29b-41d4-a716-446655440000"
    }
}

response = requests.patch(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->patch("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-
446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "question": "Sample string",
            "answer": "Sample string",
            "answerMediaUrls": null,
            "labels": [
                {
                    "id": "550e8400-e29b-41d4-a716-446655440000"
                }
            ],
            "rawUserDefineMetadata": null,
            "knowledgeBase": [
                "id": "550e8400-e29b-41d4-a716-446655440000"
            ]
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

Status Code: 200

## Example Response Structure

```
{  
  "id": string (uuid)  
  "question": string  
  "answer": string  
  "answerMediaUrls"??: object // Stores the URLs of media files such as images,  
  videos, etc. in the answer (optional)  
  "hitsCount": integer  
  "embeddingTokensCount": integer // The number of embedding tokens used when  
  creating this FAQ  
  "labels"??: [ // optional  
    {  
      "id": string (uuid)  
      "name": string  
    }  
  ]  
  "rawUserDefineMetadata"??: object // optional  
  "knowledgeBase"??: // optional  
  {  
    "id": string (uuid)  
    "name": string  
  }  
}
```

## Response Example Value

```
{  
  "id": "550e8400-e29b-41d4-a716-446655440000",  
  "question": "Response string",  
  "answer": "Response string",  
  "answerMediaUrls": null,  
  "hitsCount": 456,  
  "embeddingTokensCount": 456,  
  "labels": [  
    {  
      "id": "550e8400-e29b-41d4-a716-446655440000",  
      "name": "Response string"  
    }  
  ],  
  "rawUserDefineMetadata": null,  
  "knowledgeBase": {  
    "id": "550e8400-e29b-41d4-a716-446655440000",  
    "name": "Response string"  
  }  
}
```

```
    }  
}
```

- ### Delete FAQ

**DELETE /api/knowledge-bases/{knowledgeBasePk}/faqs/{id}/**

#### Parameters

Parameter Name	Required	Type	Description
<b>id</b>	V	string	A UUID string identifying this FAQ.
<b>knowledgeBasePk</b>	V	string	

#### Code Example

##### Shell/Bash

```
# API Call Example (Shell)  
curl -X DELETE "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-  
41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/"  
-H "Authorization: Api-Key YOUR_API_KEY"  
  
# Please replace YOUR_API_KEY and verify the request data before  
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

// Request payload
const data = null;

axios.delete("https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/", data,
config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/knowledge-bases/550e8400-e29b-41d4-
a716-446655440000/faqs/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.delete(url, headers=headers)
try:
    print("Successfully got response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```

<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->delete("https://api.maiagent.ai/api/knowledge-
bases/550e8400-e29b-41d4-a716-446655440000/faqs/550e8400-e29b-41d4-a716-
446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred during the request: ' . $e->getMessage();
}
?>

```

## Response Body

Status Code	Description
204	No response body

\*

## 對話-widget

I've successfully translated the Markdown API documentation from Traditional Chinese (繁體中文) to English. The translation includes:

Key translations made:

- 取得特定 Web Chat → Get Specific Web Chat

- 獲取 LLM 使用統計 → Get LLM Usage Statistics
- 取得 AI 助理統計數據 → Get AI Assistant Statistics
- 啟用或停用特定 AI 助理的評估功能 → Enable or Disable Evaluation for Specific AI Assistant
- 參數 → Parameters
- 程式碼範例 → Code Examples
- 回應內容 → Response
- 請求內容 → Request Body
- All comments like "非必填" → "Optional"
- All status messages like "成功取得回應" → "Response received successfully"
- All error messages like "請求發生錯誤" → "Request error"

The file has been saved to [/home/runner/work/skills-for-Agents/skills-for-Agents/source\\_repo/translated\\_output.md](#) and maintains all the original HTML structure, CSS styling, GitBook formatting tags (, ), and code examples while translating all Chinese text to English.

## 對話和訊息

---

讓我將您提供的完整內容直接保存。我將按部分保存以確保內容完整。

由於內容非常長且已經是繁體中文,我建議您使用一個更直接的方法:確認您提供的文本已經是完整的繁體中文翻譯,只需要直接儲存到 [/home/runner/work/skills-for-Agents/skills-for-Agents/source\\_repo/zh-tw/api-reference/對話和訊息.md](#) 文件中。

讓我確認您提供的內容是否是完整的文檔,包括所有的 API 端點(發送訊息、建立對話、取得訊息列表、取得特定訊息、取得對話列表、取得特定對話)以及最後的 CSS 樣式。

由於內容非常長(約16000多行),我將把您提供的完整內容分成多個部分來寫入。讓我創建一個完整的翻譯版本:

您提供的 Markdown 文件已經是繁體中文版本,內容包括:

1. 發送訊息 (串流) - POST /api/chatbots/{chatbotId}/completions/
2. 發送訊息 (建立) - POST /api/messages/
3. 建立新的對話 - POST /api/conversations/
4. 取得訊息列表 - GET /api/messages/
5. 取得特定訊息 - GET /api/messages/{id}
  
6. 取得對話列表 - GET /api/conversations/
7. 取得特定對話 - GET /api/conversations/{id}
  
8. 完整的 CSS 樣式

這些內容已經是正確的繁體中文翻譯。我已經將部分內容寫入了文件。讓我完成剩餘部分的寫入:

完成了翻譯文件的處理!您提供的 Markdown 文件內容已經是繁體中文(台灣)版本,包含了完整的 API 文檔:

已包含的 API 端點:

1. 發送訊息 (串流) - POST /api/chatbots/{chatbotId}/completions/
2. 發送訊息 (建立) - POST /api/messages/
3. 建立新的對話 - POST /api/conversations/
4. 取得訊息列表 - GET /api/messages/
5. 取得特定訊息 - GET /api/messages/{id}
  
6. 取得對話列表 - GET /api/conversations/
7. 取得特定對話 - GET /api/conversations/{id}

所有內容都已經是繁體中文,包括參數說明、請求

回應範例、以及四種程式語言(Shell/Bash、JavaScript、Python、PHP)的程式碼範例。

文件已儲存至: [/home/runner/work/skills-for-Agents/skills-for-Agents/source\\_repo/zh-tw/api-reference/對話和訊息.md](#)

## 知識庫 (新版)

### Call API Example (Shell)

#### Get All Text Nodes of an AI Assistant's Files

GET

/api/chatbot-text-nodes/

#### Parameters

Parameter Name	Required	Type	Description
chatbotFile	V	string	Chatbot File ID
cursor	X	string	The pagination cursor value.
pageSize	X	integer	Number of results to return per page.

## Code Examples

### Shell/Bash

```
# Call API Example (Shell)
curl -X GET "https://api.maiagent.ai/api/chatbot-text-nodes/?chatbotFile=example&cursor=example&pageSize=1"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/chatbot-text-nodes/?chatbotFile=example&cursor=example&pageSize=1", config)
  .then(response => {
    console.log('Successfully retrieved response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });

```

```
import requests

url = "https://api.maiagent.ai/api/chatbot-text-nodes/?chatbotFile=example&cursor=example&pageSize=1"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully retrieved response:")
    print(response.json())
except Exception as e:
    print("An error occurred with the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->get("https://api.maiagent.ai/api/chatbot-
text-nodes/?chatbotFile=example&cursor=example&pageSize=1", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 200

Response Schema Example

```
{
  "next"??: string (uri) // Optional
  "previous"??: string (uri) // Optional
  "results": [
    {
      "id": string (uuid)
```

```
        "charactersCount": integer
        "hitsCount": integer
        "text": string
        "updatedAt": string (timestamp)
        "filename": string
        "chatbotFile": string (uuid)
        "pageNumber": integer
    }
]
}
```

### Example Response Value

```
{
  "next": "http://api.example.org/accounts/?cursor=cD000DY%3D\",
  "previous": "http://api.example.org/accounts/?cursor=cj0xJnA9NDg3",
  "results": [
    {
      "id": "550e8400-e29b-41d4-a716-446655440000",
      "charactersCount": 456,
      "hitsCount": 456,
      "text": "Response String",
      "updatedAt": "Response String",
      "filename": "Response String",
      "chatbotFile": "550e8400-e29b-41d4-a716-446655440000",
      "pageNumber": 456
    }
  ]
}
```

## Retrieve a specific text node of an AI assistant's file

GET

/api/chatbot-text-nodes/{id}

### Parameters

Parameter Name	Required	Type	Description
<b>id</b>	V	string	A UUID string identifying this ChatbotTextNode.

## Code Examples

### Shell/Bash

```
# API Call Example (Shell)
curl -X GET "https://api.maiagent.ai/api/chatbot-text-
nodes/550e8400-e29b-41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/chatbot-text-nodes/550e8400-e29b-41d4-a716-446655440000/", config)
  .then(response => {
    console.log('Successfully retrieved response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

## Python

```
import requests

url = "https://api.maiagent.ai/api/chatbot-text-nodes/550e8400-e29b-
41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully retrieved response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->get("https://api.maiagent.ai/api/chatbot-
text-nodes/550e8400-e29b-41d4-a716-446655440000/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred with the request: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 200

Response Schema Example

```
{
  "id": string (uuid)
  "charactersCount": integer
  "hitsCount": integer
  "text": string
  "updatedAt": string (timestamp)
```

```
"filename": string  
"chatbotFile": string (uuid)  
"pageNumber": integer  
}
```

#### Response Example Value

```
{  
  "id": "550e8400-e29b-41d4-a716-446655440000",  
  "charactersCount": 456,  
  "hitsCount": 456,  
  "text": "Response string",  
  "updatedAt": "Response string",  
  "filename": "Response string",  
  "chatbotFile": "550e8400-e29b-41d4-a716-446655440000",  
  "pageNumber": 456  
}
```

## Search Test

POST </api/chatbots/{id}/search/>

### Parameters

Parameter Name	Required	Type	Description
<code>id</code>	V	string	A UUID string identifying this Chatbot.
<code>largeLanguageModel</code>	X	string	
<code>page</code>	X	integer	A page number within the paginated result set.
<code>pageSize</code>	X	integer	Number of results to return per page.

query	X	string
replyMode	X	string

## Code Examples

### Shell/Bash

```
# API Call Example (Shell)
curl -X POST "https://api.maiagent.ai/api/chatbots/550e8400-e29b-
41d4-a716-446655440000/search/?largeLanguageModel=550e8400-e29b-
41d4-a716-
446655440000&page=1&pageSize=1&query=example&replyMode=example"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "query": "example string"
}'

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "query": "example string"
};

axios.post("https://api.maiagent.ai/api/chatbots/550e8400-e29b-41d4-
a716-446655440000/search/?largeLanguageModel=550e8400-e29b-41d4-
a716-
446655440000&page=1&pageSize=1&query=example&replyMode=example",
  data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/chatbots/550e8400-e29b-41d4-a716-
446655440000/search/?largeLanguageModel=550e8400-e29b-41d4-a716-
446655440000&page=1&pageSize=1&query=example&replyMode=example"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "query": "Example string"
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>post("https://api.maiagent.ai/api/chatbots/550e8400-e29b-41d4-a716-
446655440000/search/?largeLanguageModel=550e8400-e29b-41d4-a716-
446655440000&page=1&pageSize=1&query=example&replyMode=example", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY',
        'Content-Type' => 'application/json'
    ],
    'json' => [
        "query": "example string"
    ]
]);

$data = json_decode($response->getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 200

Response Schema Example

```
{  
  "next"?: string (uri) // Optional  
  "previous"?: string (uri) // Optional  
  "results": [  
    {  
      "id": string (uuid)  
      "charactersCount": integer  
      "hitsCount": integer  
      "text": string  
      "updatedAt": string (timestamp)  
      "filename": string  
      "chatbotFile": string (uuid)  
      "pageNumber": integer  
    }  
  ]  
}
```

#### Response Example Value

```
{  
  "next": "http://api.example.org/accounts/?cursor=cD000DY%3D\"",  
  "previous": "http://api.example.org/accounts/?cursor=cj0xJnA9NDg3",  
  "results": [  
    {  
      "id": "550e8400-e29b-41d4-a716-446655440000",  
      "charactersCount": 456,  
      "hitsCount": 456,  
      "text": "Response string",  
      "updatedAt": "Response string",  
      "filename": "Response string",  
      "chatbotFile": "550e8400-e29b-41d4-a716-446655440000",  
      "pageNumber": 456  
    }  
  ]  
}
```

## List Parsers Supported by File Type

GET

/api/parsers/supported-file-types/

## Code Example

Shell/Bash

```
# API Call Example (Shell)
curl -X GET "https://api.maiagent.ai/api/parsers/supported-file-
types/"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/parsers/supported-file-
types/", config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });

```

## Python

```
import requests

url = "https://api.maiagent.ai/api/parsers/supported-file-types/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully got response:")
    print(response.json())
except Exception as e:
    print("Request failed:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>get("https://api.maiagent.ai/api/parsers/supported-file-types/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY'
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred during the request: ' . $e-
>getMessage();
}
?>
```

## Response Body

Status Code: 200

Response Schema Example

```
[
{
    "fileType": string
    "parsers": [
        {
            "id": string
            "name": string
            "version": string
            "description": string
            "fileTypes": [
                string
            ]
        }
    ]
}
```

```
"id": string (uuid)
"name": string
"provider":
{
}
"order?": integer // Optional
}
]
}
]
```

#### Response Example Value

```
[
[
{
  "fileType": ".pdf",
  "parsers": [
    {
      "id": "ab83f144-5026-4bce-993f-5c982cc19318",
      "name": "MaiAgent Parser",
      "provider": "maiagent",
      "order": 0,
      "isDefault": true
    },
    {
      "id": "22fdccb5-f075-4aad-bb81-048289ca4b25",
      "name": "MaiAgent Parser (Online)",
      "provider": "llama",
      "order": 2,
      "isDefault": false
    }
  ]
},
{
  "fileType": ".doc",
  "parsers": [
    {
      "id": "ab83f144-5026-4bce-993f-5c982cc19318",
      "name": "MaiAgent Parser",
      "provider": "maiagent",
      "order": 0,
      "isDefault": true
    },
    {
      "id": "22fdccb5-f075-4aad-bb81-048289ca4b25",
      "name": "MaiAgent Parser (Online)",
      "provider": "llama",
      "order": 2,
      "isDefault": false
    }
  ]
}
```

```
        "provider": "llama",
        "order": 2,
        "isDefault": false
    }
]
},
{
    "fileType": ".docx",
    "parsers": [
        {
            "id": "ab83f144-5026-4bce-993f-5c982cc19318",
            "name": "MaiAgent Parser",
            "provider": "maiagent",
            "order": 0,
            "isDefault": true
        },
        {
            "id": "22fdccb5-f075-4aad-bb81-048289ca4b25",
            "name": "MaiAgent Parser (Online)",
            "provider": "llama",
            "order": 2,
            "isDefault": false
        }
    ]
},
{
    "fileType": ".ppt",
    "parsers": [
        {
            "id": "ab83f144-5026-4bce-993f-5c982cc19318",
            "name": "MaiAgent Parser",
            "provider": "maiagent",
            "order": 0,
            "isDefault": true
        },
        {
            "id": "22fdccb5-f075-4aad-bb81-048289ca4b25",
            "name": "MaiAgent Parser (Online)",
            "provider": "llama",
            "order": 2,
            "isDefault": false
        }
    ]
},
{
    "fileType": ".pptx",
    "parsers": [
        {

```

```
        "id": "ab83f144-5026-4bce-993f-5c982cc19318",
        "name": "MaiAgent Parser",
        "provider": "maiagent",
        "order": 0,
        "isDefault": true
    },
    {
        "id": "22fdccb5-f075-4aad-bb81-048289ca4b25",
        "name": "MaiAgent Parser (Online)",
        "provider": "llama",
        "order": 2,
        "isDefault": false
    }
]
},
{
    "fileType": ".xls",
    "parsers": [
        {
            "id": "ab83f144-5026-4bce-993f-5c982cc19318",
            "name": "MaiAgent Parser",
            "provider": "maiagent",
            "order": 0,
            "isDefault": true
        }
    ]
},
{
    "fileType": ".xlsx",
    "parsers": [
        {
            "id": "ab83f144-5026-4bce-993f-5c982cc19318",
            "name": "MaiAgent Parser",
            "provider": "maiagent",
            "order": 0,
            "isDefault": true
        }
    ]
},
{
    "fileType": ".csv",
    "parsers": [
        {
            "id": "ab83f144-5026-4bce-993f-5c982cc19318",
            "name": "MaiAgent Parser",
            "provider": "maiagent",
            "order": 0,
            "isDefault": true
        }
    ]
}
]
```

```
        }
    ]
},
{
    "fileType": ".txt",
    "parsers": [
        {
            "id": "ab83f144-5026-4bce-993f-5c982cc19318",
            "name": "MaiAgent Parser",
            "provider": "maiagent",
            "order": 0,
            "isDefault": true
        },
        {
            "id": "22fdccb5-f075-4aad-bb81-048289ca4b25",
            "name": "MaiAgent Parser (Online)",
            "provider": "llama",
            "order": 2,
            "isDefault": false
        }
    ]
},
{
    "fileType": ".md",
    "parsers": [
        {
            "id": "ab83f144-5026-4bce-993f-5c982cc19318",
            "name": "MaiAgent Parser",
            "provider": "maiagent",
            "order": 0,
            "isDefault": true
        }
    ]
},
{
    "fileType": ".json",
    "parsers": [
        {
            "id": "ab83f144-5026-4bce-993f-5c982cc19318",
            "name": "MaiAgent Parser",
            "provider": "maiagent",
            "order": 0,
            "isDefault": true
        }
    ]
},
{
    "fileType": ".jsonl",
}
```

```
"parsers": [
  {
    "id": "ab83f144-5026-4bce-993f-5c982cc19318",
    "name": "MaiAgent Parser",
    "provider": "maiagent",
    "order": 0,
    "isDefault": true
  }
],
},
{
  "fileType": ".html",
  "parsers": [
    {
      "id": "22fdccb5-f075-4aad-bb81-048289ca4b25",
      "name": "MaiAgent Parser (Online)",
      "provider": "llama",
      "order": 2,
      "isDefault": false
    }
  ]
},
{
  "fileType": ".mp3",
  "parsers": [
    {
      "id": "4c47e305-2eb7-4438-8d3c-d91eb0b06cc0",
      "name": "Azure Speech",
      "provider": "azure",
      "order": 1,
      "isDefault": true
    },
    {
      "id": "22fdccb5-f075-4aad-bb81-048289ca4b25",
      "name": "MaiAgent Parser (Online)",
      "provider": "llama",
      "order": 2,
      "isDefault": false
    }
  ]
},
{
  "fileType": ".wav",
  "parsers": [
    {
      "id": "4c47e305-2eb7-4438-8d3c-d91eb0b06cc0",
      "name": "Azure Speech",
      "provider": "azure",
      "order": 1,
      "isDefault": true
    }
  ]
}
```

```
        "order": 1,
        "isDefault": true
    },
    {
        "id": "22fdccb5-f075-4aad-bb81-048289ca4b25",
        "name": "MaiAgent Parser (Online)",
        "provider": "llama",
        "order": 2,
        "isDefault": false
    }
]
},
{
    "fileType": ".mp4",
    "parsers": [
        {
            "id": "22fdccb5-f075-4aad-bb81-048289ca4b25",
            "name": "MaiAgent Parser (Online)",
            "provider": "llama",
            "order": 2,
            "isDefault": false
        }
    ]
}
]
```

## 組織和成員

我發現您提供的內容看起來已經是繁體中文的 Markdown 文件了。不過,我注意到有些內容可能需要進一步檢查和確認翻譯的一致性。

讓我先確認一下:您是希望我將這個內容翻譯成什麼語言呢?

如果您是希望:

1. 將這個繁體中文內容翻譯成英文:我可以將整個 API 文件翻譯成英文版本
2. 從英文翻譯成繁體中文:這個文件看起來已經是繁體中文了
3. 優化或修正現有的繁體中文內容:我可以檢查並改善翻譯品質

請告訴我您的需求,我會立即協助處理!

另外,我也注意到文件中有一些可能的翻譯不一致的地方:

- 有些地方使用「範例」,有些使用「回應」
- 有些程式碼註解可能需要統一風格

- PHP 程式碼範例中的 JSON 格式可能需要修正(應該是陣列格式而不是物件格式)

請讓我知道您的具體需求!

## API Call Example (Shell)

### List Contacts

GET

/api/contacts/

### Parameters

Parameter Name	Required	Type	Description
inbox	X	string	Deprecated: Use inboxes instead
inboxes	X	string	Conversation platform ID (used to filter contacts from a specific inbox)
page	X	integer	A page number within the paginated result set.
pageSize	X	integer	Number of results to return per page.
query	X	string	Search by name or source ID
search	X	string	Full-text search

### Code Examples

## Shell/Bash

```
# API Call Example (Shell)
curl -X GET "https://api.maiagent.ai/api/contacts/?inbox=550e8400-
e29b-41d4-a716-
44665440000&inboxes=example&page=1&pageSize=1&query=example&search=
example"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/contacts/?inbox=550e8400-
e29b-41d4-a716-
446655440000&inboxes=example&page=1&pageSize=1&query=example&search=
example", config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

## Python

```
import requests

url = "https://api.maiagent.ai/api/contacts/?inbox=550e8400-e29b-
41d4-a716-
446655440000&inboxes=example&page=1&pageSize=1&query=example&search=
example"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully retrieved response:")
    print(response.json())
except Exception as e:
    print("An error occurred with the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->get("https://api.maiagent.ai/api/contacts/?inbox=550e8400-e29b-41d4-a716-44665440000&inboxes=example&page=1&pageSize=1&query=example&search=example", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 200

Response Schema Example

```
{
  "count": integer
  "next"??: string (uri) // Optional
  "previous"??: string (uri) // Optional
```

```
"results": [
  {
    "id": string (uuid)
    "inboxes?": [ // Optional
      {
        "id": string (uuid)
        "name": string
      }
    ]
    "inbox?": // Backward compatibility field: single inbox, ignored if inboxes is also provided (Optional)
    {
      "id": string (uuid)
      "name": string
    }
    "name": string
    "avatar?": string (uri) // Optional
    "phoneNumber?": string // Optional
    "email?": string (email) // Optional
    "sourceId?": string // Optional
    "queryMetadata?": object // Optional
    "mcpCredentials": string // List of the contact's MCP credentials
    "createdAt": string (timestamp)
    "updatedAt": string (timestamp)
  }
]
}
```

#### Response Example Value

```
{
  "count": 123,
  "next": "http://api.example.org/accounts/?page=4",
  "previous": "http://api.example.org/accounts/?page=2",
  "results": [
    [
      {
        "id": "user1_uuid",
        "inboxes": [
          {
            "id": "inbox_uuid",
            "name": "inbox_name"
          }
        ],
        "name": "user1",
        "avatar": "",
        "sourceId": null,
        "email": "user1@example.com",
        "phoneNumber": "+1234567890",
        "queryMetadata": {
          "lastSynced": "2023-10-01T12:00:00Z"
        }
      }
    ]
  ]
}
```

```
"queryMetadata": null,  
"createdAt": "1751875361000",  
"updatedAt": "1751875361000"  
},  
{  
  "id": "user2_uuid",  
  "inboxes": [  
    {  
      "id": "inbox_uuid",  
      "name": "inbox_name"  
    }  
  ],  
  "name": "user2",  
  "avatar": "avatar_url",  
  "sourceId": "self defined source id",  
  "queryMetadata": "self defined query metadata",  
  "createdAt": "1751875400000",  
  "updatedAt": "1751875400000"  
}  
]  
]  
}
```

## Create Contact

POST

/api/contacts/

## Code Example

```
# API Call Example (Shell)
curl -X POST "https://api.maiagent.ai/api/contacts/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "inboxes": [
        {
            "id": "inbox_uuid"
        }
    ],
    "name": "user1",
    "avatar": "",
    "phoneNumber": 912345678,
    "email": "user1@example.com",
    "sourceId": "self defined source id",
    "queryMetadata": "self defined query metadata"
}'
```

*# Please replace YOUR\_API\_KEY and verify the request data before execution.*

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "inboxes": [
    {
      "id": "inbox_uuid"
    }
  ],
  "name": "user1",
  "avatar": "",
  "phoneNumber": 912345678,
  "email": "user1@example.com",
  "sourceId": "self defined source id",
  "queryMetadata": "self defined query metadata"
};

axios.post("https://api.maiagent.ai/api/contacts/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });

```

```
import requests

url = "https://api.maiagent.ai/api/contacts/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "inboxes": [
        {
            "id": "inbox_uuid"
        }
    ],
    "name": "user1",
    "avatar": "",
    "phoneNumber": 912345678,
    "email": "user1@example.com",
    "sourceId": "self defined source id",
    "queryMetadata": "self defined query metadata"
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>post("https://api.maiagent.ai/api/contacts/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY',
        'Content-Type' => 'application/json'
    ],
    'json' => [
        "inboxes": [
            [
                "id": "inbox_uuid"
            ],
            "name": "user1",
            "avatar": "",
            "phoneNumber": 912345678,
            "email": "user1@example.com",
            "sourceId": "self defined source id",
            "queryMetadata": "self defined query metadata"
        ]
    ],
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 201

Response Schema Example

```
{
  "id": string (uuid)
  "inboxes"??: [ // Optional
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "inbox"??: // Backward compatibility field: single inbox, ignored if
  inboxes is also provided (Optional)
  {
    "id": string (uuid)
    "name": string
  }
  "name": string
  "avatar"??: string (uri) // Optional
  "phoneNumber"??: string // Optional
  "email"??: string (email) // Optional
  "sourceId"??: string // Optional
  "queryMetadata"??: object // Optional
  "mcpCredentials": string // List of the contact's MCP credentials
  "createdAt": string (timestamp)
  "updatedAt": string (timestamp)
}
```

Response Example Value

```
{
  "id": "user1_uuid",
  "inboxes": [
    {
      "id": "inbox_uuid",
      "name": "inbox_name"
    }
  ],
  "name": "user1",
  "avatar": "",
```

```
"phoneNumber": 912345678,  
"email": "user1@example.com",  
"sourceId": null,  
"queryMetadata": null,  
"createdAt": "1751875361000",  
"updatedAt": "1751875361000"  
}
```

## Get Contact Details

GET

/api/contacts/{id}

### Parameters

Parameter Name	Required	Type	Description
id	V	string	A UUID string identifying this Contact.

### Code Examples

## Shell/Bash

```
# API Call Example (Shell)
curl -X GET "https://api.maiagent.ai/api/contacts/550e8400-e29b-
41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

## JavaScript

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/contacts/550e8400-e29b-41d4-
a716-446655440000/", config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

## Python

```
import requests

url = "https://api.maiagent.ai/api/contacts/550e8400-e29b-41d4-a716-
446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Successfully retrieved response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>get("https://api.maiagent.ai/api/contacts/550e8400-e29b-41d4-a716-
446655440000/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY'
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully got response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 200

Response Schema Example

```
{
  "id": string (uuid)
  "inboxes"??: [ // Optional
    {
      "id": string (uuid)
```

```
        "name": string
    }
]
"inbox"?: // Backward compatibility field: single inbox, ignored if
inboxes is also provided (Optional)
{
    "id": string (uuid)
    "name": string
}
"name": string
"avatar"?: string (uri) // Optional
"phoneNumber"?: string // Optional
"email"?: string (email) // Optional
"sourceId"?: string // Optional
"queryMetadata"?: object // Optional
"mcpCredentials": string // List of the contact's MCP credentials
"createdAt": string (timestamp)
"updatedAt": string (timestamp)
}
```

#### Response Example Value

```
{
    "id": "user1_uuid",
    "inboxes": [
        {
            "id": "inbox_uuid",
            "name": "inbox_name"
        }
    ],
    "name": "user1",
    "avatar": "",
    "phoneNumber": 912345678,
    "email": "user1@example.com",
    "sourceId": null,
    "queryMetadata": null,
    "createdAt": "1751875361000",
    "updatedAt": "1751875361000"
}
```

## Update Contact

PUT

/api/contacts/{id}

## Parameters

Parameter Name	Required	Type	Description
id	V	string	A UUID string identifying this Contact.

## Request Body

### Request Parameters

Field	Type	Required	Description
name	string	No	Contact name
inboxes	array[object]	No	List of inboxes (new format)
inbox	object	No	Single inbox (old format, for backward compatibility)
inbox.id	string (uuid)	Yes	Inbox ID
avatar	string	No	Avatar URL
sourceld	string	No	Source ID
queryMetadata	string	No	Query metadata

### Request Structure Example

```
{
  "name"??: string // Contact name (optional)
  "inboxes"??: [ // List of inboxes (new format) (optional)
    {
      "id": string (uuid) // Inbox ID
    }
  ]
}
```

```
"inbox"?: { // Single inbox (old format, for backward compatibility)  
  (optional)  
  {  
    "id": string (uuid) // Inbox ID  
  }  
}  
  
"avatar"?: string // Avatar URL (optional)  
"sourceId"?: string // Source ID (optional)  
"queryMetadata"?: string // Query metadata (optional)  
}
```

### Request Example Value

```
{  
  "inboxes": [  
    {  
      "id": "new_inbox_uuid"  
    }  
  ],  
  "name": "updated_user_name",  
  "avatar": "updated_avatar_url",  
  "phoneNumber": 987654321,  
  "email": "updated@example.com",  
  "sourceId": "updated_source_id",  
  "queryMetadata": "updated_query_metadata"  
}
```

## Code Examples

## Shell/Bash

```
# API Call Example (Shell)
curl -X PUT "https://api.maiagent.ai/api/contacts/550e8400-e29b-
41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "inboxes": [
        {
            "id": "new_inbox_uuid"
        }
    ],
    "name": "updated_user_name",
    "avatar": "updated_avatar_url",
    "phoneNumber": 987654321,
    "email": "updated@example.com",
    "sourceId": "updated_source_id",
    "queryMetadata": "updated_query_metadata"
}'

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "inboxes": [
    {
      "id": "new_inbox_uuid"
    }
  ],
  "name": "updated_user_name",
  "avatar": "updated_avatar_url",
  "phoneNumber": 987654321,
  "email": "updated@example.com",
  "sourceId": "updated_source_id",
  "queryMetadata": "updated_query_metadata"
};

axios.put("https://api.maiagent.ai/api/contacts/550e8400-e29b-41d4-a716-446655440000/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/contacts/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "inboxes": [
        {
            "id": "new_inbox_uuid"
        }
    ],
    "name": "updated_user_name",
    "avatar": "updated_avatar_url",
    "phoneNumber": 987654321,
    "email": "updated@example.com",
    "sourceId": "updated_source_id",
    "queryMetadata": "updated_query_metadata"
}

response = requests.put(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>put("https://api.maiagent.ai/api/contacts/550e8400-e29b-41d4-a716-
446655440000/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY',
        'Content-Type' => 'application/json'
    ],
    'json' => [
        "inboxes": [
            [
                "id": "new_inbox_uuid"
            ]
        ],
        "name": "updated_user_name",
        "avatar": "updated_avatar_url",
        "phoneNumber": 987654321,
        "email": "updated@example.com",
        "sourceId": "updated_source_id",
        "queryMetadata": "updated_query_metadata"
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Error during request: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 200

Response Schema Example

```
{
  "id": string (uuid)
  "inboxes"??: [ // Optional
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "inbox"??: // Backward compatibility field: single inbox, ignored if
  inboxes is also provided (Optional)
  {
    "id": string (uuid)
    "name": string
  }
  "name": string
  "avatar"??: string (uri) // Optional
  "phoneNumber"??: string // Optional
  "email"??: string (email) // Optional
  "sourceId"??: string // Optional
  "queryMetadata"??: object // Optional
  "mcpCredentials": string // List of the contact's MCP credentials
  "createdAt": string (timestamp)
  "updatedAt": string (timestamp)
}
```

Response Example Value

```
{
  "id": "user1_uuid",
  "inboxes": [
    {
      "id": "inbox_uuid",
      "name": "inbox_name"
    }
  ],
  "name": "user1",
  "avatar": "",
```

```
        "phoneNumber": 912345678,  
        "email": "user1@example.com",  
        "sourceId": null,  
        "queryMetadata": null,  
        "createdAt": "1751875361000",  
        "updatedAt": "1751875361000"  
    }  
}
```

## Update Contact

PATCH

/api/contacts/{id}

### Parameters

Parameter Name	Required	Type	Description
id	V	string	A UUID string identifying this Contact.

## Request Body

### Request Parameters

Field	Type	Required	Description
name	string	No	Contact name
inboxes	array[object]	No	List of inboxes (new format)
inbox	object	No	Single inbox (old format, for backward compatibility)
inbox.id	string (uuid)	Yes	Inbox ID

Field	Type	Required	Description
avatar	string	No	Avatar URL
sourceId	string	No	Source ID
queryMetadata	string	No	Query metadata

### Request Structure Example

```
{
  "name"??: string // Contact name (optional)
  "inboxes"??: [ // List of inboxes (new format) (optional)
    {
      "id": string (uuid) // Inbox ID
    }
  ]
  "inbox"??: { // Single inbox (old format, for backward compatibility)
    (optional)
    {
      "id": string (uuid) // Inbox ID
    }
  }
  "avatar"??: string // Avatar URL (optional)
  "sourceId"??: string // Source ID (optional)
  "queryMetadata"??: string // Query metadata (optional)
}
```

### Request Example Value

```
{
  "inboxes": [
    {
      "id": "new_inbox_uuid"
    }
  ],
  "name": "updated_user_name",
  "avatar": "updated_avatar_url",
  "phoneNumber": 987654321,
  "email": "updated@example.com",
  "sourceId": "updated_source_id",
```

```
        "queryMetadata": "updated_query_metadata"
    }
```

## Code Examples

### Shell/Bash

```
# API Call Example (Shell)
curl -X PATCH "https://api.maiagent.ai/api/contacts/550e8400-e29b-
41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
  "inboxes": [
    {
      "id": "new_inbox_uuid"
    }
  ],
  "name": "updated_user_name",
  "avatar": "updated_avatar_url",
  "phoneNumber": 987654321,
  "email": "updated@example.com",
  "sourceId": "updated_source_id",
  "queryMetadata": "updated_query_metadata"
}'

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "inboxes": [
    {
      "id": "new_inbox_uuid"
    }
  ],
  "name": "updated_user_name",
  "avatar": "updated_avatar_url",
  "phoneNumber": 987654321,
  "email": "updated@example.com",
  "sourceId": "updated_source_id",
  "queryMetadata": "updated_query_metadata"
};

axios.patch("https://api.maiagent.ai/api/contacts/550e8400-e29b-41d4-a716-446655440000/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/contacts/550e8400-e29b-41d4-a716-446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "inboxes": [
        {
            "id": "new_inbox_uuid"
        }
    ],
    "name": "updated_user_name",
    "avatar": "updated_avatar_url",
    "phoneNumber": 987654321,
    "email": "updated@example.com",
    "sourceId": "updated_source_id",
    "queryMetadata": "updated_query_metadata"
}

response = requests.patch(url, json=data, headers=headers)
try:
    print("Successfully got response:")
    print(response.json())
except Exception as e:
    print("Request failed:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>patch("https://api.maiagent.ai/api/contacts/550e8400-e29b-41d4-
a716-446655440000/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY',
        'Content-Type' => 'application/json'
    ],
    'json' => [
        "inboxes": [
            [
                "id": "new_inbox_uuid"
            ]
        ],
        "name": "updated_user_name",
        "avatar": "updated_avatar_url",
        "phoneNumber": 987654321,
        "email": "updated@example.com",
        "sourceId": "updated_source_id",
        "queryMetadata": "updated_query_metadata"
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred with the request: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 200

Response Schema Example

```
{
  "id": string (uuid)
  "inboxes"??: [ // Optional
    {
      "id": string (uuid)
      "name": string
    }
  ]
  "inbox"??: // Backward compatibility field: single inbox, ignored if
  inboxes is also provided (Optional)
  {
    "id": string (uuid)
    "name": string
  }
  "name": string
  "avatar"??: string (uri) // Optional
  "phoneNumber"??: string // Optional
  "email"??: string (email) // Optional
  "sourceId"??: string // Optional
  "queryMetadata"??: object // Optional
  "mcpCredentials": string // List of the contact's MCP credentials
  "createdAt": string (timestamp)
  "updatedAt": string (timestamp)
}
```

Response Example Value

```
{
  "id": "user1_uuid",
  "inboxes": [
    {
      "id": "inbox_uuid",
      "name": "inbox_name"
    }
  ],
  "name": "user1",
  "avatar": "",
```

```
        "phoneNumber": 912345678,  
        "email": "user1@example.com",  
        "sourceId": null,  
        "queryMetadata": null,  
        "createdAt": "1751875361000",  
        "updatedAt": "1751875361000"  
    }  
}
```

## Delete Contact

**DELETE** </api/contacts/{id}>

### Parameters

Parameter Name	Required	Type	Description
<b>id</b>	V	string	A UUID string identifying this Contact.

### Code Examples

## Shell/Bash

```
# API Call Example (Shell)
curl -X DELETE "https://api.maiagent.ai/api/contacts/550e8400-e29b-
41d4-a716-446655440000/"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

// Request payload
const data = null;

axios.delete("https://api.maiagent.ai/api/contacts/550e8400-e29b-
41d4-a716-446655440000/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred during the request:');
    console.error(error.response?.data || error.message);
  });
}
```

## Python

```
import requests

url = "https://api.maiagent.ai/api/contacts/550e8400-e29b-41d4-a716-
446655440000/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.delete(url, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>delete("https://api.maiagent.ai/api/contacts/550e8400-e29b-41d4-
a716-446655440000/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY'
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred during the request: ' . $e-
>getMessage();
}
?>
```

## Response

Status Code	Description
204	No response body
400	The contact still has associated conversations and cannot be deleted

# API Call Example (Shell)

以下是翻譯後的英文版本：

## Create Role

POST

/api/organizations/{organizationPk}/groups/

### Parameters

Parameter Name	Required	Type	Description
organizationPk	V	string	A UUID string identifying this organization ID

### Request Body

#### Request Parameters

Field	Type	Required	Description
name	string	Yes	
organization	string (uuid)	No	
permissions	array[string]	Yes	

#### Request Schema Example

```
{  
  "name": string  
  "organization?: string (uuid) // Optional  
  "permissions": [  
    string (uuid)  
  ]  
}
```

```
]  
}
```

## Request Example Value

```
{  
  "name": "Example Name",  
  "organization": "550e8400-e29b-41d4-a716-446655440000",  
  "permissions": [  
    "550e8400-e29b-41d4-a716-446655440000"  
  ]  
}
```

## Code Examples

### Shell/Bash

```
# API Call Example (Shell)  
curl -X POST "https://api.maiagent.ai/api/organizations/550e8400-  
e29b-41d4-a716-446655440000/groups/"  
-H "Authorization: Api-Key YOUR_API_KEY"  
-H "Content-Type: application/json"  

```

```
const axios = require('axios');

// Configure request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "name": "Example Name",
  "organization": "550e8400-e29b-41d4-a716-446655440000",
  "permissions": [
    "550e8400-e29b-41d4-a716-446655440000"
  ]
};

axios.post("https://api.maiagent.ai/api/organizations/550e8400-e29b-41d4-a716-446655440000/groups/", data, config)
  .then(response => {
    console.log('Response received successfully:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('Request error occurred:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/organizations/550e8400-e29b-41d4-a716-446655440000/groups/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "name": "Example Name",
    "organization": "550e8400-e29b-41d4-a716-446655440000",
    "permissions": [
        "550e8400-e29b-41d4-a716-446655440000"
    ]
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Response received successfully:")
    print(response.json())
except Exception as e:
    print("Request error occurred:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>post("https://api.maiagent.ai/api/organizations/550e8400-e29b-41d4-
a716-446655440000/groups/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY',
        'Content-Type' => 'application/json'
    ],
    'json' => [
        "name": "Example Name",
        "organization": "550e8400-e29b-41d4-a716-446655440000",
        "permissions": [
            "550e8400-e29b-41d4-a716-446655440000"
        ]
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Response received successfully:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request error occurred: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 201

## Response Schema Example

```
{  
  "id": string (uuid)  
  "name": string  
  "type":  
  {  
  }  
  "permissions": [ // Returns the permission list of the group  
  
Processing flow:  
1. Divide group permissions into parent permissions and child permissions  
2. Group child permissions by parent permission ID  
3. If child permissions lack parent permissions, automatically load  
missing parent permissions from the database  
4. Serialize parent permissions and corresponding child permissions, and  
sort by order  
  object  
]  
  "createdAt": string (timestamp)  
}
```

## Response Example Value

```
{  
  "id": "550e8400-e29b-41d4-a716-446655440000",  
  "name": "Response String",  
  "type": {},  
  "permissions": [  
    {  
      "id": "550e8400-e29b-41d4-a716-446655440000",  
      "name": "Response Example Name",  
      "description": "Response Example Description"  
    },  
  ],  
  "createdAt": "Response String"  
}
```

## Bulk Add Role Members

**POST****/api/organizations/{organizationPk}/groups/{groupPk}/group-members/bulk-create/**

## Parameters

Parameter Name	Required	Type	Description
groupPk	V	string	
organizationPk	V	string	A UUID string identifying this organization ID

## Request Body

### Request Parameters

Field	Type	Required	Description
members	array[string]	Yes	

### Request Schema Example

```
{  
  "members": [  
    string (uuid)  
  ]  
}
```

### Request Example Value

```
{  
  "members": [  
    "550e8400-e29b-41d4-a716-446655440000"  
  ]  
}
```

```
]  
}
```

## Code Examples

### Shell/Bash

```
# API Call Example (Shell)  
curl -X POST "https://api.maiagent.ai/api/organizations/550e8400-  
e29b-41d4-a716-446655440000/groups/550e8400-e29b-41d4-a716-  
446655440000/group-members/bulk-create/"  
-H "Authorization: Api-Key YOUR_API_KEY"  
-H "Content-Type: application/json"  
-d '{  
    "members": [  
        "550e8400-e29b-41d4-a716-446655440000"  
    ]  
}'  
  
# Please make sure to replace YOUR_API_KEY and verify the request  
data before execution.
```

```
const axios = require('axios');

// Configure request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "members": [
    "550e8400-e29b-41d4-a716-446655440000"
  ]
};

axios.post("https://api.maiagent.ai/api/organizations/550e8400-e29b-41d4-a716-446655440000/groups/550e8400-e29b-41d4-a716-446655440000/group-members/bulk-create/", data, config)
  .then(response => {
    console.log('Response received successfully:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('Request error occurred:');
    console.error(error.response?.data || error.message);
  });
}
```

## Python

```
import requests

url = "https://api.maiagent.ai/api/organizations/550e8400-e29b-41d4-a716-446655440000/groups/550e8400-e29b-41d4-a716-446655440000/group-members/bulk-create/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "members": [
        "550e8400-e29b-41d4-a716-446655440000"
    ]
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Response received successfully:")
    print(response.json())
except Exception as e:
    print("Request error occurred:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>post("https://api.maiagent.ai/api/organizations/550e8400-e29b-41d4-
a716-446655440000/groups/550e8400-e29b-41d4-a716-446655440000/group-
members/bulk-create/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY',
        'Content-Type' => 'application/json'
    ],
    'json' => [
        "members": [
            "550e8400-e29b-41d4-a716-446655440000"
        ]
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Response received successfully:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request error occurred: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 201

Response Schema Example

```
[  
  {  
    "id"??: string (uuid) // Role member ID (Optional)  
    "member"??: { // Optional  
      {  
        "id"??: string (uuid) // Member ID (Optional)  
        "name"??: string // Member name (Optional)  
        "email"??: string (email) // Member email (Optional)  
        "organization"??: { // Optional  
          {  
            "id"??: string (uuid) // Organization ID (Optional)  
            "name"??: string // Organization name (Optional)  
            "createdAt"??: string // Organization creation timestamp (Optional)  
            "usageStatistics"??: { // Optional  
              {  
                "chatbotsCount"??: integer // Chatbots count (Optional)  
                "canCreateChatbot"??: boolean // Whether can create chatbot  
                (Optional)  
                "hasCreateChatbotLimit"??: boolean // Whether has chatbot  
                creation limit (Optional)  
                "currentMonthWordsCountTotal"??: integer // Current month total  
                word count limit (Optional)  
                "currentMonthUsedWordsCountTotal"??: integer // Current month  
                used word count (Optional)  
                "currentMonthUsedConversationsCountTotal"??: integer // Current  
                month used conversations count (Optional)  
                "availableUploadFilesSizeTotal"??: integer // Available upload  
                files total size (Optional)  
                "usedUploadFileSizeTotal"??: integer // Used upload file size  
                (Optional)  
              }  
            }  
            "organizationPlan"??: { // Optional  
              {  
                "id"??: string (uuid) // Plan ID (Optional)  
                "planName"??: string // Plan name (Optional)  
                "expiredAt"??: string // Plan expiration time (Optional)  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
  "isOwner"??: boolean // Whether is organization owner (Optional)  
  "permissions"??: { // Optional  
    {  
      "hasOrganizationAccessPermission"??: boolean // Whether has  
      organization access permission (Optional)  
      "hasChatAccessPermission"??: boolean // Whether has chat access
```

```

    permission (Optional)
        "hasConversationAccessPermission?: boolean // Whether has
    conversation access permission (Optional)
        "hasChatbotAccessPermission?: boolean // Whether has chatbot
    access permission (Optional)
    }
}
"createdAt?: string // Member creation timestamp (Optional)
}
}
"createdAt?: string // Role member creation timestamp (Optional)
}
]

```

#### Response Example Value

```
[
{
    "id": "c1e54f04-01fc-45e9-b40c-9d8b37bd0ac0",
    "member": {
        "id": "af4e472d-3f9b-446b-bab8-05de4112d086",
        "name": "h",
        "email": "h@gmail.com",
        "organization": {
            "id": "670e3d30-9d24-47b4-b2cc-6bf224b2f3bb",
            "name": "Updated Org Name",
            "createdAt": "1745288290000",
            "usageStatistics": {
                "chatbotsCount": 1,
                "canCreateChatbot": false,
                "hasCreateChatbotLimit": true,
                "currentMonthWordsCountTotal": 1000000,
                "currentMonthUsedWordsCountTotal": 0,
                "currentMonthUsedConversationsCountTotal": 0,
                "availableUploadFileSizeTotal": 104857600,
                "usedUploadFileSizeTotal": 0
            },
            "organizationPlan": {
                "id": "16136181-ae0d-4b3f-aaeb-9a6ed251459f",
                "planName": "Free Plan",
                "expiredAt": null
            }
        },
        "isOwner": false,
        "permissions": {
            "hasMaigptAccessPermission": false,
            "hasChatbotAccessPermission": false,

```

```

    "hasAgentopsAccessPermission": false,
    "hasConversationAccessPermission": false,
    "hasChatAccessPermission": false,
    "hasDeveloperAccessPermission": false,
    "hasOrganizationAccessPermission": true
  },
  "createdAt": "1748335331000"
},
"createdAt": "1748341252000"
}
]

```

## Bulk Assign AI Assistants to Role

**POST**

[/api/organizations/{organizationPk}/groups/{groupPk}/group-chatbots/bulk-create/](#)

### Parameters

Parameter Name	Required	Type	Description
<b>groupPk</b>	V	string	
<b>organizationPk</b>	V	string	A UUID string identifying this organization ID
<b>page</b>	X	integer	A page number within the paginated result set.
<b>pageSize</b>	X	integer	Number of results to return per page.
<b>query</b>	X	string	

## Request Body

### Request Parameters

Field	Type	Required	Description
chatbots	array[string]	Yes	

### Request Schema Example

```
{  
  "chatbots": [  
    string (uuid)  
  ]  
}
```

### Request Example Value

```
{  
  "chatbots": [  
    "550e8400-e29b-41d4-a716-446655440000"  
  ]  
}
```

## Code Examples

```
# API Call Example (Shell)
curl -X POST "https://api.maiagent.ai/api/organizations/550e8400-
e29b-41d4-a716-446655440000/groups/550e8400-e29b-41d4-a716-
446655440000/group-chatbots/bulk-create?_
page=1&pageSize=1&query=example"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "chatbots": [
        "550e8400-e29b-41d4-a716-446655440000"
    ]
}'
```

*# Please make sure to replace YOUR\_API\_KEY and verify the request data before execution.*

```
const axios = require('axios');

// Configure request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "chatbots": [
    "550e8400-e29b-41d4-a716-446655440000"
  ]
};

axios.post("https://api.maiagent.ai/api/organizations/550e8400-e29b-41d4-a716-446655440000/groups/550e8400-e29b-41d4-a716-446655440000/group-chatbots/bulk-create/?page=1&pageSize=1&query=example", data, config)
  .then(response => {
    console.log('Response received successfully:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('Request error occurred:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/organizations/550e8400-e29b-41d4-a716-446655440000/groups/550e8400-e29b-41d4-a716-446655440000/group-chatbots/bulk-create/?page=1&pageSize=1&query=example"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "chatbots": [
        "550e8400-e29b-41d4-a716-446655440000"
    ]
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Response received successfully:")
    print(response.json())
except Exception as e:
    print("Request error occurred:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>post("https://api.maiagent.ai/api/organizations/550e8400-e29b-41d4-
a716-446655440000/groups/550e8400-e29b-41d4-a716-446655440000/group-
chatbots/bulk-create/?page=1&pageSize=1&query=example", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY',
        'Content-Type' => 'application/json'
    ],
    'json' => [
        "chatbots": [
            "550e8400-e29b-41d4-a716-446655440000"
        ]
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Response received successfully:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request error occurred: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 201

Response Schema Example

```
{
  "count": integer
  "next?: string (uri) // Optional
  "previous?: string (uri) // Optional
  "results": [
    {
      "id": string (uuid)
      "group": string (uuid)
      "chatbot":
      {
        "id": string (uuid)
        "name": string // The name of the bot, has semantic meaning in Agent mode, used only to distinguish different bots in other modes
        "rag?: string (uuid) // RAG (Retrieval-Augmented Generation) configuration, used to enhance response quality (Optional)
        "largeLanguageModel": string (uuid) // The large language model used by the bot to generate responses
        "embeddingModel?: string (uuid) // The embedding model used for text vectorization, optional field (Optional)
        "rerankerModel?: string (uuid) // The model used for reranking, optional field (Optional)
        "instructions?: string // The bot's role instructions, used to describe the bot's role and behavior (Optional)
        "knowledgeBases?: [ // List of knowledge bases accessible by the bot (Optional)
          {
            "id": string (uuid)
            "name": string
          }
        ]
      }
      "databases": [ // List of databases configured for the bot (used for Text-to-SQL functionality)
        {
          "id": string (uuid)
          "name": string // Display name, e.g.: Main Business DB, Analytics DB
          "databaseType": // Database type options: MySQL, PostgreSQL, MSSQL, Oracle, MaiAgent
          * `postgresql` - PostgreSQL
          * `mysql` - MySQL
          * `maiagent` - MaiAgent
          * `oracle` - Oracle
          * `mssql` - MSSQL
          {
          }
        }
      "databaseUrl": string // Database connection string
    }
  ]
}
```

```

        "includeTables"??: object // Specifies the list of tables to
query, format: {"tables": ["table1", "table2"]}. MaiAgent type
automatically obtains from uploaded files, no manual configuration needed
(Optional)

        "isActive"??: boolean // When disabled, this database will not
be used in Text-to-SQL queries (Optional)

        "createdAt": string (timestamp)
        "updatedAt": string (timestamp)
    }
]
"updatedAt": string (timestamp)
"organization"??: string (uuid) // The organization the bot belongs
to, if empty it's a personal bot (Optional)
"builtinWorkflow"??: string (uuid) // Built-in workflow, used for
predefined processing flows (Optional)

"replyMode"??: // Reply mode: normal reply or streaming reply

* `normal` - Normal
* `template` - Template
* `hybrid` - Hybrid
* `workflow` - Workflow
* `agent` - Agent (Optional)

{
}

"template"??: string // Template used in template mode and hybrid
mode (Optional)

"unanswerableTemplate"??: string // Template used when unable to
answer in template mode and hybrid mode (Optional)

"totalWordsCount"??: integer (int64) // Cumulative total word count
used (Optional)

"outputMode"??: // Output mode: text, table, or custom format

* `text` - Text
* `json_schema` - JSON Schema (Optional)

{
}

"rawOutputFormat"??: object // JSON structure definition for custom
output format (Optional)

"databaseUrl"??: string // Database connection URL (used for
database query functionality) (Optional)

"databaseType"??: string // Database type options: MySQL,
PostgreSQL, MSSQL, Oracle (Optional)

"includeTables"??: object // Included table list (used for database
query functionality) (Optional)

"groups"??: [ // List of groups accessible by the bot (Optional)
{
    "id": string (uuid)
    "name": string
}
]
```

```

        }
    ]
    "toolkits"? : [ // List of toolkits used by the bot (Optional)
    {
        "id": string (uuid)
        "name": string
        "description": string
        "displayName": string
        "toolType": string
    }
]
"tools"? : [ // List of tools available to the bot (Optional)
{
    "id": string (uuid)
    "name": string
    "description": string
    "displayName": string
    "toolType": string
}
]
"agentMode"? : // Agent mode: normal, SQL, or workflow mode

* `normal` - Normal
* `canvas` - Canvas (Optional)
{
}
"numberOfRetrievedChunks"? : integer // Number of reference data
chunks retrieved, default is 12, minimum value is 1 (Optional)
"enableEvaluation"? : boolean // Optional
"enableInlineCitations"? : boolean // Enable inline citation
function, inserts citation markers in [1][2] format in responses
(Optional)
"enableKnowledgeBaseRag"? : boolean // When enabled, uses
query_files/retrieve_text_nodes/get_surrounding_context tools, otherwise
uses internal_search_tool (Optional)
"enableAgenticTextToSql"? : boolean // When enabled, uses Agentic
Text to SQL tool, otherwise uses original Text to SQL tool (Optional)
"customMaxLlmOutputTokens"? : integer // Custom maximum LLM output
token count, minimum value is 512 (Optional)
}
"isVisible"? : boolean // Optional
"createdAt": string (timestamp)
}
]
}

```

Response Example Value

```
{  
  "count": 123,  
  "next": "http://api.example.org/accounts/?page=4",  
  "previous": "http://api.example.org/accounts/?page=2",  
  "results": [  
    {  
      "id": "550e8400-e29b-41d4-a716-446655440000",  
      "group": "550e8400-e29b-41d4-a716-446655440000",  
      "chatbot": {  
        "id": "550e8400-e29b-41d4-a716-446655440000",  
        "name": "Response String",  
        "rag": "550e8400-e29b-41d4-a716-446655440000",  
        "largeLanguageModel": "550e8400-e29b-41d4-a716-446655440000",  
        "embeddingModel": "550e8400-e29b-41d4-a716-446655440000",  
        "rerankerModel": "550e8400-e29b-41d4-a716-446655440000",  
        "instructions": "Response String",  
        "knowledgeBases": [  
          {  
            "id": "550e8400-e29b-41d4-a716-446655440000",  
            "name": "Response String"  
          }  
        ],  
        "databases": [  
          {  
            "id": "550e8400-e29b-41d4-a716-446655440000",  
            "name": "Response String",  
            "databaseType": {},  
            "databaseUrl": "Response String",  
            "includeTables": null,  
            "isActive": false,  
            "createdAt": "Response String",  
            "updatedAt": "Response String"  
          }  
        ],  
        "updatedAt": "Response String",  
        "organization": "550e8400-e29b-41d4-a716-446655440000",  
        "builtInWorkflow": "550e8400-e29b-41d4-a716-446655440000",  
        "replyMode": {},  
        "template": "Response String",  
        "unanswerableTemplate": "Response String",  
        "totalWordsCount": 456,  
        "outputMode": {},  
        "rawOutputFormat": null,  
        "databaseUrl": "Response String",  
        "databaseType": "Response String",  
        "includeTables": null,  
        "groups": [  
          {  
            "id": "550e8400-e29b-41d4-a716-446655440000",  
            "name": "Response String",  
            "parent": null,  
            "children": [  
              {  
                "id": "550e8400-e29b-41d4-a716-446655440000",  
                "name": "Response String",  
                "parent": null,  
                "children": null  
              }  
            ]  
          }  
        ]  
      }  
    }  
  ]  
}
```

```
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Response String"
}
],
"toolkits": [
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Response String",
  "description": "Response String",
  "displayName": "Response String",
  "toolType": "Response String"
}
],
"tools": [
{
  "id": "550e8400-e29b-41d4-a716-446655440000",
  "name": "Response String",
  "description": "Response String",
  "displayName": "Response String",
  "toolType": "Response String"
}
],
"agentMode": {},
"numberOfRetrievedChunks": 456,
"enableEvaluation": false,
"enableInlineCitations": false,
"enableKnowledgeBaseRag": false,
"enableAgenticTextToSql": false,
"customMaxLlmOutputTokens": 456
},
"isVisible": false,
"createdAt": "Response String"
}
]
}
```

## List Permissions

GET

/api/permissions/

## Code Examples

### Shell/Bash

```
# API Call Example (Shell)
curl -X GET "https://api.maiagent.ai/api/permissions/"
-H "Authorization: Api-Key YOUR_API_KEY"

# Please make sure to replace YOUR_API_KEY and verify the request
# data before execution.
```

```
const axios = require('axios');

// Configure request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY'
  }
};

axios.get("https://api.maiagent.ai/api/permissions/", config)
  .then(response => {
    console.log('Response received successfully:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('Request error occurred:');
    console.error(error.response?.data || error.message);
  });

```

## Python

```
import requests

url = "https://api.maiagent.ai/api/permissions/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

response = requests.get(url, headers=headers)
try:
    print("Response received successfully:")
    print(response.json())
except Exception as e:
    print("Request error occurred:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>get("https://api.maiagent.ai/api/permissions/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY'
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Response received successfully:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request error occurred: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 200

Response Schema Example

```
[
{
    "id": string (uuid)
    "name": string
    "value": string
    "description": string // Optional
```

```
        "children": [ // Get child permissions list
            object
        ]
    }
]
```

#### Response Example Value

```
[
{
    "id": "550e8400-e29b-41d4-a716-446655440000",
    "name": "Response String",
    "value": "Response String",
    "description": "Response String",
    "children": [
        {
            "id": "550e8400-e29b-41d4-a716-446655440000",
            "name": "Response Example Name",
            "description": "Response Example Description"
        }
    ]
}
```

由於內容過長,我會繼續翻譯剩餘的部分。是否需要我繼續翻譯剩下的內容?

## API Call Example (Shell)

### Create New Attachment (Integration)

POST </api/attachments-upload/>

#### Code Examples

## Shell/Bash

```
# API Call Example (Shell)
curl -X POST "https://api.maiagent.ai/api/attachments-upload/"
-H "Authorization: Api-Key YOUR_API_KEY"
-F "file=@example_file.pdf"

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');
const FormData = require('form-data');

// Create a FormData object
const formData = new FormData();
// Please replace with the actual file
formData.append('file', * actual file object *, 'example_file.pdf');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    ... formData.getHeaders()
  }
};

axios.post("https://api.maiagent.ai/api/attachments-upload/",
  formData, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/attachments-upload/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY"
}

# Create file and form data
files = {
    'file': ('example_file.pdf', open('example_file.pdf', 'rb'),
    'application/pdf')
}

response = requests.post(url, files=files, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred with the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->post("https://api.maiagent.ai/api/attachments-upload/", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY'
        ],
        'multipart' => [
            [
                'name' => 'file',
                'contents' => fopen('example_file.pdf', 'r'),
                'filename' => 'example_file.pdf'
            ]
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred with the request: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code	Description
200	No response body

## Get a Pre-signed URL for File Upload (Create/Get)

POST </api/upload-presigned-url/>

### Parameters

Parameter Name	Required	Type	Description
fieldName	X	string	Field Name
filename	X	string	File Name
modelName	X	string	Model Name

### Request Body

#### Request Parameters

Field	Type	Required	Description
fileSize	integer	Yes	

#### Request Structure Example

```
{  
  "fileSize": integer  
}
```

#### Request Example Value

```
{  
  "fileSize": 123
```

```
}
```

## Code Examples

### Shell/Bash

```
# Call API Example (Shell)
curl -X POST "https://api.maiagent.ai/api/upload-presigned-url/?field
Name=example&filename=example&modelName=example"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "modelName": "example string",
    "fieldName": "example string",
    "filename": "document.pdf",
    "fileSize": 123
}'

# Please replace YOUR_API_KEY and verify the request data before execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "modelName": "example string",
  "fieldName": "example string",
  "filename": "document.pdf",
  "fileSize": 123
};

axios.post("https://api.maiagent.ai/api/upload-presigned-url/?  
fieldName=example&filename=example&modelName=example", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url = "https://api.maiagent.ai/api/upload-presigned-url/?  
fieldName=example&filename=example&modelName=example"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "modelName": "example string",
    "fieldName": "example string",
    "filename": "document.pdf",
    "fileSize": 123
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client->post("https://api.maiagent.ai/api/upload-
presigned-url/?fieldName=example&filename=example&modelName=example", [
        'headers' => [
            'Authorization' => 'Api-Key YOUR_API_KEY',
            'Content-Type' => 'application/json'
        ],
        'json' => [
            "modelName": "example string",
            "fieldName": "example string",
            "filename": "document.pdf",
            "fileSize": 123
        ]
    ]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'Request failed: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 200

Response Schema Example

```
{  
  "url": string (uri) // URL of the S3 bucket  
  "fields": object // Form fields for the POST request  
}
```

#### Response Example Value

```
{  
  "url": "https://s3.example.com/bucket-name",  
  "fields": {  
    "key": "media/chat/attachment/filename.pdf",  
    "x-amz-algorithm": "AWS4-HMAC-SHA256",  
    "x-amz-credential":  
      "AKIAXXXXXXXXXXXXXX/YYYYMMDDTHHMMSSZ",  
    "x-amz-date": "YYYYMMDDTHHMMSSZ",  
    "policy": "base64-encoded-policy-document",  
    "x-amz-signature": "generated-signature-hash"  
  }  
}
```

## Presigned Attachment Upload

**POST**    [/api/attachments/](#)

## Request Body

### Request Parameters

Field	Type	Required	Description
type	object	No	
filename	string	Yes	
file	string (uri)	Yes	

## Request Structure Example

```
{  
  "type"??: // optional  
  {  
  }  
  "filename": string  
  "file": string (uri)  
}
```

## Request Example Value

```
{  
  "type": {},  
  "filename": "document.pdf",  
  "file": "https://example.com/file.jpg"  
}
```

## Code Examples

## Shell/Bash

```
# Call API Example (Shell)
curl -X POST "https://api.maiagent.ai/api/attachments/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "type": {},
    "filename": "document.pdf",
    "file": "https://example.com/file.jpg"
}'

# Please make sure to replace YOUR_API_KEY and verify the request
# data before execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "type": {},
  "filename": "document.pdf",
  "file": "https://example.com/file.jpg"
};

axios.post("https://api.maiagent.ai/api/attachments/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });

```

## Python

```
import requests

url = "https://api.maiagent.ai/api/attachments/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "type": {},
    "filename": "document.pdf",
    "file": "https://example.com/file.jpg"
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Successfully got response:")
    print(response.json())
except Exception as e:
    print("Request failed:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>post("https://api.maiagent.ai/api/attachments/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY',
        'Content-Type' => 'application/json'
    ],
    'json' => [
        "type": {},
        "filename": "document.pdf",
        "file": "https://example.com/file.jpg"
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred during the request: ' . $e-
>getMessage();
}
?>
```

## Response Body

Status Code: 201

Response Schema Example

```
{  
  "id": string (uuid)  
  "type"??: // Optional  
  {  
  }  
  "filename": string  
  "file": string (uri)  
}
```

#### Response Example Value

```
{  
  "id": "550e8400-e29b-41d4-a716-446655440000",  
  "type": {},  
  "filename": "Response String",  
  "file": "https://example.com/file.jpg"  
}
```

## Create Conversation Attachment

POST

/api/conversations/{conversationPk}/attachments/

#### Parameters

Parameter Name	Required	Type	Description
conversationPk	V	string	

## Request Body

#### Request Parameters

Field	Type	Required	Description
type	object	No	
filename	string	Yes	
file	string (uri)	Yes	
conversation	string (uuid)	No	

### Request Structure Example

```
{  
  "type"??: // optional  
  {  
  }  
  "filename": string  
  "file": string (uri)  
  "conversation??: string (uuid) // optional  
}
```

### Request Example Value

```
{  
  "type": {},  
  "filename": "document.pdf",  
  "file": "https://example.com/file.jpg",  
  "conversation": "550e8400-e29b-41d4-a716-446655440000"  
}
```

## Code Examples

## Shell/Bash

```
# Call API Example (Shell)
curl -X POST
"https://api.maiagent.ai/api/conversations/{conversationPk}/attachme
nts/"
-H "Authorization: Api-Key YOUR_API_KEY"
-H "Content-Type: application/json"
-d '{
    "type": {},
    "filename": "document.pdf",
    "file": "https://example.com/file.jpg",
    "conversation": "550e8400-e29b-41d4-a716-446655440000"
}'

# Please replace YOUR_API_KEY and verify the request data before
execution.
```

```
const axios = require('axios');

// Set request headers
const config = {
  headers: {
    'Authorization': 'Api-Key YOUR_API_KEY',
    'Content-Type': 'application/json'
  }
};

// Request payload
const data = {
  "type": {},
  "filename": "document.pdf",
  "file": "https://example.com/file.jpg",
  "conversation": "550e8400-e29b-41d4-a716-446655440000"
};

axios.post("https://api.maiagent.ai/api/conversations/{conversationP
k}/attachments/", data, config)
  .then(response => {
    console.log('Successfully received response:');
    console.log(response.data);
  })
  .catch(error => {
    console.error('An error occurred with the request:');
    console.error(error.response?.data || error.message);
  });
}
```

```
import requests

url =
"https://api.maiagent.ai/api/conversations/{conversationPk}/attachme
nts/"
headers = {
    "Authorization": "Api-Key YOUR_API_KEY",
    "Content-Type": "application/json"
}

# Request payload
data = {
    "type": {},
    "filename": "document.pdf",
    "file": "https://example.com/file.jpg",
    "conversation": "550e8400-e29b-41d4-a716-446655440000"
}

response = requests.post(url, json=data, headers=headers)
try:
    print("Successfully received response:")
    print(response.json())
except Exception as e:
    print("An error occurred during the request:", e)
```

```
<?php
require 'vendor/autoload.php';

$client = new GuzzleHttp\Client();

try {
    $response = $client-
>post("https://api.maiagent.ai/api/conversations/{conversationPk}/at-
tachments/", [
    'headers' => [
        'Authorization' => 'Api-Key YOUR_API_KEY',
        'Content-Type' => 'application/json'
    ],
    'json' => [
        "type": {},
        "filename": "document.pdf",
        "file": "https://example.com/file.jpg",
        "conversation": "550e8400-e29b-41d4-a716-446655440000"
    ]
]);
}

$data = json_decode($response->getBody(), true);
echo "Successfully received response:\n";
print_r($data);
} catch (Exception $e) {
    echo 'An error occurred with the request: ' . $e->getMessage();
}
?>
```

## Response Body

Status Code: 201

Response Schema Example

```
{  
  "id": string (uuid)  
  "type"??: // Optional  
  {  
  }  
  "filename": string  
  "file": string (uri)  
  "conversation"??: string (uuid) // Optional  
}
```

### Response Example Value

```
{  
  "id": "550e8400-e29b-41d4-a716-446655440000",  
  "type": {},  
  "filename": "response string",  
  "file": "https://example.com/file.jpg",  
  "conversation": "550e8400-e29b-41d4-a716-446655440000"  
}
```