

使用示例

1、Python示例代码

向量检索RAG接口调用:

```
import requests
import json

# 非流式调用
response = requests.post(
    "http://localhost:8000/rag_pipeline",
    json={
        "query": "请假如何请",
        "context_query": "请假如何请",
        "stream": False
    }
)
print(response.json())

# 流式调用方式
import requests
import json

response = requests.post(
    "http://localhost:8000/rag_pipeline",
```

知识图谱RAG接口调用:

```
import requests
import json
import sseclient

# 非流式调用
response = requests.post(
    "http://localhost:7680/graph_rag",
    json={
        "query": "腾讯最近有什么投资事件?",
        "stream": False
    }
)
print(response.json())
```

Agent接口调用:

```
import requests
import json

# 非流式调用
response = requests.post(
    "http://localhost:8000/agent",
    json={
```

```
    "query": "公司的请假制度是什么？",  
    "history": [] # 可选的对话历史  
  }  
)  
print(response.json())  
  
# 流式调用  
import requests  
import json  
  
response = requests.post(  
    "http://localhost:8000/rag_pipeline",  
    .  
    ,
```

2、JavaScript示例代码

向量检索RAG接口:

```
// 非流式调用  
async function callRagPipeline() {  
  try {  
    const response = await fetch('http://localhost:8000/rag_pipeline', {  
      method: 'POST',  
      headers: {  
        'Content-Type': 'application/json',  
      },  
      body: JSON.stringify({  
        query: "请假如何请",  
        context_query: "请假如何请",  
        stream: false  
      })  
    })  
  }  
}
```

```
    });  
    const data = await response.json();  
    console.log(data);  
  } catch (error) {  
    console.error('Error:', error);  
  }  
}
```

知识图谱RAG接口:

```
// 非流式调用  
async function callGraphRag() {  
  try {  
    const response = await fetch('http://localhost:8000/graph_rag', {  
      method: 'POST',  
      headers: {  
        'Content-Type': 'application/json',  
      },  
      body: JSON.stringify({  
        query: "腾讯最近有什么投资事件?",  
        stream: false  
      })  
    });  
    const data = await response.json();  
    console.log(data);  
  } catch (error) {  
    console.error('Error:', error);  
  }  
}
```

Agent路由接口:

```
// 非流式调用
async function callAgent() {
  try {
    const response = await fetch('http://localhost:8000/router', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        query: "公司的请假制度是什么？",
        history: []
      })
    });
    const data = await response.json();
    console.log(data);
  } catch (error) {
    console.error('Error:', error);
  }
}
```

3、Java示例代码

使用 OkHttp 库进行 HTTP 请求:

```
// 添加依赖
// implementation("com.squareup.okhttp3:okhttp:4.9.1")
```

```
// implementation("com.fasterxml.jackson.core:jackson-databind:2.12.5")

import okhttp3.*;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import java.util.ArrayList;

public class RagApiClient {
    private final OkHttpClient client = new OkHttpClient();
    private final ObjectMapper mapper = new ObjectMapper();
    private static final String BASE_URL = "http://localhost:8000";
    private static final MediaType JSON = MediaType.parse("application/json; charset=utf-8");

    // 1. 向量检索RAG接口
    public void callRagPipeline() throws IOException {
        Map<String, Object> requestBody = new HashMap<>();
    }
}
```

下一节：本章介绍(01:31)

下一节