**GraphRAG API 搭建与使用**

**本地搭建**

**相关package安装**

**Uvicorn**

安装uvicorn 作为web server。

|  |
| --- |
| PowerShell pip install uvicorn |

**RabbitMQ**

安装RabbitMQ作为broker, 安装步骤取决于系统环境，具体请参考[Installing RabbitMQ | RabbitMQ](https://www.rabbitmq.com/docs/download)

|  |
| --- |
| PowerShell #下列适用于Ubuntu 22.04 #!/bin/sh  sudo apt-get install curl gnupg apt-transport-https -y  ## Team RabbitMQ's main signing key curl -1sLf "https://keys.openpgp.org/vks/v1/by-fingerprint/0A9AF2115F4687BD29803A206B73A36E6026DFCA" | sudo gpg --dearmor | sudo tee /usr/share/keyrings/com.rabbitmq.team.gpg > /dev/null ## Community mirror of Cloudsmith: modern Erlang repository curl -1sLf https://github.com/rabbitmq/signing-keys/releases/download/3.0/cloudsmith.rabbitmq-erlang.E495BB49CC4BBE5B.key | sudo gpg --dearmor | sudo tee /usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg > /dev/null ## Community mirror of Cloudsmith: RabbitMQ repository curl -1sLf https://github.com/rabbitmq/signing-keys/releases/download/3.0/cloudsmith.rabbitmq-server.9F4587F226208342.key | sudo gpg --dearmor | sudo tee /usr/share/keyrings/rabbitmq.9F4587F226208342.gpg > /dev/null  ## Add apt repositories maintained by Team RabbitMQ sudo tee /etc/apt/sources.list.d/rabbitmq.list <<EOF ## Provides modern Erlang/OTP releases ## deb [arch=amd64 signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg] https://ppa1.rabbitmq.com/rabbitmq/rabbitmq-erlang/deb/ubuntu focal main deb-src [signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg] https://ppa1.rabbitmq.com/rabbitmq/rabbitmq-erlang/deb/ubuntu focal main  # another mirror for redundancy deb [arch=amd64 signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg] https://ppa2.rabbitmq.com/rabbitmq/rabbitmq-erlang/deb/ubuntu focal main deb-src [signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg] https://ppa2.rabbitmq.com/rabbitmq/rabbitmq-erlang/deb/ubuntu focal main  ## Provides RabbitMQ ## deb [arch=amd64 signed-by=/usr/share/keyrings/rabbitmq.9F4587F226208342.gpg] https://ppa1.rabbitmq.com/rabbitmq/rabbitmq-server/deb/ubuntu focal main deb-src [signed-by=/usr/share/keyrings/rabbitmq.9F4587F226208342.gpg] https://ppa1.rabbitmq.com/rabbitmq/rabbitmq-server/deb/ubuntu focal main  # another mirror for redundancy deb [arch=amd64 signed-by=/usr/share/keyrings/rabbitmq.9F4587F226208342.gpg] https://ppa2.rabbitmq.com/rabbitmq/rabbitmq-server/deb/ubuntu focal main deb-src [signed-by=/usr/share/keyrings/rabbitmq.9F4587F226208342.gpg] https://ppa2.rabbitmq.com/rabbitmq/rabbitmq-server/deb/ubuntu focal main EOF  ## Update package indices sudo apt-get update -y  ## Install Erlang packages sudo apt-get install -y erlang-base \  erlang-asn1 erlang-crypto erlang-eldap erlang-ftp erlang-inets \  erlang-mnesia erlang-os-mon erlang-parsetools erlang-public-key \  erlang-runtime-tools erlang-snmp erlang-ssl \  erlang-syntax-tools erlang-tftp erlang-tools erlang-xmerl  ## Install rabbitmq-server and its dependencies sudo apt-get install rabbitmq-server -y --fix-missing |

**Celery**

安装Celery来进行消息列队管理

|  |
| --- |
| PowerShell pip install celery |

**目录管理**

将此文件夹放置在根目录下，如果更改路径，需要修改GraphRag\_Demo/API\_AICO/目录下的三个.py文件里的系统路径。

https://zhipu-ai.feishu.cn/drive/folder/UbdhfhX1Rlj2x4dCkpeccSU0noc

**API服务启动**

在GraphRag\_Demo/API\_AICO/目录下运行如下命令

**启动Celery与RabbitMQ服务**

|  |
| --- |
| PowerShell sudo systemctl start rabbitmq-server  ####check server status #sudo systemctl status rabbitmq-server  celery -A queue\_func.celery\_app worker -l info -P gevent --concurrency=1 |

**启动Uvicorn**

|  |
| --- |
| PowerShell uvicorn API\_main:app --host 0.0.0.0 --port 8000  ####额外可选参数#### #--reload .py文件发生更变时自动重启 #--log-level 日志等级 |

**启动Gradio 文档上传服务**

注意：目前Dify AI与AICO暂且都不支持用户上传文档至API后台，因此需要用户打开网站手动上传文档

文档上传页面部署在http://47.88.27.143/的服务器上，当前端口为8003.

* 安装gradio package: pip install gradio
* 切换工作目录：cd /home/ecs-user/API\_upload\_host
* 运行脚本：python upload\_gradio.py

**API 参数与使用**

因为云服务器防火墙及设置原因，本API仅接受HTTP请求。

PowerShell中使用CURL请求时请先运行 Remove-item alias:curl

**/upload/ POST**

|  |
| --- |
| PowerShell curl --location 'http://101.201.33.94:8000/upload/' --form 'file=@"Your\_Path\_to\_File"' |

本接口会将本地txt文本上传至云端，鉴于文本Indexing是一项极为耗时的过程，目前仅支持上传编码格式为utf-8且小于40kb的文本

**传入参数**

* file：string, 文本的本地绝对地址

**返回值**

* info：file ‘file\_name’ saved at './corpus/user\_id/file\_name.txt
* user\_id: 唯一user\_id

**示例**

|  |
| --- |
| PowerShell curl --location 'http://101.201.33.94:8000/upload/' --form 'file=@"C:\\GraphRag\_Demo\\rag\_ZH\\corpus\\chennanJiushi-4.txt"' |

成功返回：

{"info":"file 'chennanJiushi-4.txt' saved at'./corpus/7aacf020-c420-4fe1-88e1-399d77502db7/chennanJiushi-4.txt',\n Please KEEP Your user\_id:'7aacf020-c420-4fe1-88e1-399d77502db7'safe!",

"user\_id":"7aacf020-c420-4fe1-88e1-399d77502db7"}

**/index POST**

|  |
| --- |
| PowerShell curl -X POST "http://101.201.33.94:8000/index" -H "Content-Type: application/json" -d '{\"user\_id\": \"Your\_user\_id\"，\"use\_autotune\":\"True\"}' |

开始对上传的文档进行Index处理(4千字文档耗时约2分钟)

**传入参数**

传入方式：body

* user\_id: string, 上传文档时返回的唯一user\_id
* use\_autotune: Boolean, 是否需要使用Auto Prompt Tune

**返回值**

* task\_id：当前Index的唯一任务id
* status: 当前任务状态

**示例**

|  |
| --- |
| PowerShell curl -X POST "http://101.201.33.94:8000/index" -H "Content-Type: application/json" -d '{\"user\_id\": \"7aacf020-c420-4fe1-88e1-399d77502db7\"}' |

成功返回：

{"task\_id":"bb4b740a-5718-43c6-82c7-d6934448ee3b","status":"in Queue"}

**/status/ GET**

|  |
| --- |
| PowerShell curl -X GET "http://101.201.33.94:8000/status/?task\_id=your\_task\_id" |

开始对上传的文档进行Index处理(4千字文档耗时约2分钟)

**传入参数**

传入方式：body

* user\_id: string, index时返回的唯一task\_id

**返回值**

* task\_id：当前Index的唯一任务id
* status: 当前任务状态 (In Queue/ Auto Tune/ In Progress/ completed/ failed)
* Result: 当前任务结果 (Null/Success/Error)

**示例**

|  |
| --- |
| PowerShell curl -X GET "http://101.201.33.94:8000/status/?task\_id=009047a1-ea97-4863-b003-651323864f6d" |

成功返回：

{"task\_id":"009047a1-ea97-4863-b003-651323864f6d","status":"completed","result":"Success"}

**/globalSearch/ POST**

|  |
| --- |
| PowerShell curl -X POST "http://101.201.33.94:8000/globalSearch/" -H "Content-Type: application/json" -d '{\"id\": \"Your\_task\_id\", \"query\": \"查询内容\"}' |

对文档进行全局搜索搜索

**传入参数**

传入方式：body

* task\_id: Index时返回的唯一task\_id
* query: 查询或搜索语句

**返回值**

* message：查询结果

**示例**

|  |
| --- |
| PowerShell curl -X POST "http://101.201.33.94:8000/globalSearch/" -H "Content-Type: application/json" -d '{\"id\": \"bb4b740a-5718-43c6-82c7-d6934448ee3b\", \"query\": \"故事里发生了什 么?\"}' |

成功返回：

{"message":"# 故事概览\n\n故事主要围绕一个被称为\"I\"的学生以及他的家庭展开，描述了\"I\"在家庭、学校和社区中的成长经历和所面临的挑战。\n\n## 家庭生活与情感关系\n\n故事深入探讨了家庭成员之间的日常生活和情感关系。家庭成员之间的互动非常紧密， 他们共同分享生活的喜悦和挑战 [Data: Reports (21, 19, 11, 23, 24)]。在家庭中，\"I\"与父母和兄弟姐妹之间存在着复杂的关系，包括权威与反抗、关爱与支持等 [Data: Reports (8, 21)]。"}

**/localSearch/ POST**

|  |
| --- |
| PowerShell curl -X POST "http://101.201.33.94:8000/localSearch/" -H "Content-Type: application/json" -d '{\"id\": \"Your\_task\_id\", \"query\": \"查询内容\"}' |

对文档进行全局搜索搜索

**传入参数**

传入方式：body

* task\_id: Index时返回的唯一task\_id
* query: 查询或搜索语句

**返回值**

* message：查询结果

**示例**

|  |
| --- |
| PowerShell curl -X POST "http://101.201.33.94:8000/localSearch/" -H "Content-Type: application/json" -d '{\"id\": \"bb4b740a-5718-43c6-82c7-d6934448ee3b\", \"query\": \"故事里发生了什 么?\"}' |

成功返回：

{"message":"# 具体返回内容"}