

对话助手

作者: Xvsenfeng

更新时间: 2025-2-14

做的练手的小玩意, 如有更好的实现方案欢迎交流1458612070@qq.com, 交流QQ群: 719932592

给个  吧

=====

注: 最新的版本把几个模型保存在网盘里面了

链接: <https://pan.quark.cn/s/0d28fe813112>

=====

简介

使用ollama实现本地模型的定制, 可以做到数据不泄露以及绕开检测的效果, 之后使用嘉立创的esp32开发板实现简单的对话助手

视频教程: [\[开源/教程\]使用本地deepseek模型+嘉立创esp32搭建自己的语音助手 \(可处理文件以及联网获取信息\)](#) [哔哩哔哩bilibili](#)

- v0.2

加入数据库, 可以实现服务器重启聊天不丢失, 不同用户识别等

同时接入本文档, 可以直接使用AI对话的方式进行文档处理

[XuSenfeng/ai-chat-local: 使用esp32+ollama实现本地模型的对话以及联网+工具调用](#)

国内地址: [ai-chat-local: 使用esp32+ollama实现本地模型的对话以及联网+工具调用](#)

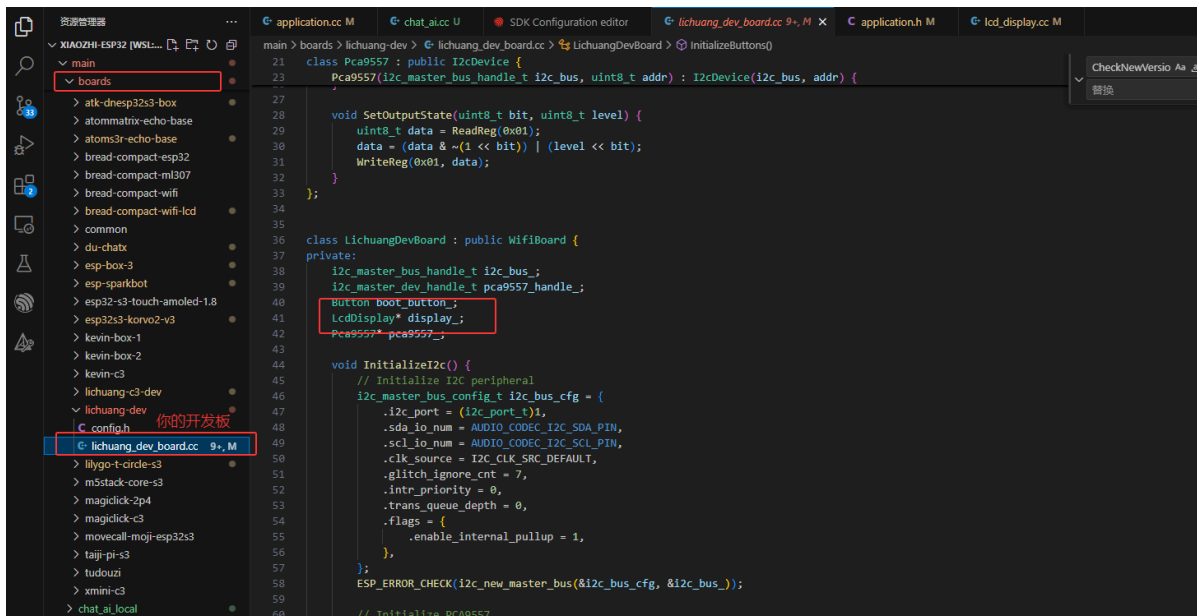
- v0.3

[小智本地服务器开源项目](#)

加入[小智ai](#)的处理, 保留小智的语音识别以及语音语音合成(这种有情感的[语音合成](#)个人好像无法使用), 但是加入一个和之前服务器连接的选项

这部分的处理需要使用LCD进行显示, 同时对按钮进行重写, 所以实际使用的时候需要看一下你的开发板使用的是不是LCD模块, 以及开发板是不是有一个按钮

理论所有符合的板子都可以, 但是我只有嘉立创的两个开发板, 所以只测试了这两个



实际是把对话发给本地的处理器进行工具调用, 使用返回的结果进行显示, 因为小智的模型语音识别效果比百度的要好, 同时模型等实际使用也还可以, 所以这里添加的是服务器端工具调用部分, 联网等

[小智 AI 聊天机器人控制台](#)

示例:

- 1 如果是遇到类似**如果是遇到类似
- 2 **
- 3 查手册, 搜索网络新闻之类的无法直接处理的问题, 告诉他你会使用其它工具搜索, 单机按键即可获取实际的信息, 请稍等一下查看工具显示页面
- 4 **
- 5 **
- 6 示例: 用户: 帮我看一下手册是谁写的
- 7 回答: "我会使用工具, 请你单击按钮获取我哦查到的手册内容"
- 8 示例: 用户: 帮我查看一下实时的新闻
- 9 回答: "我会使用工具, 点击按钮你就可以获取到我查询的新闻结果"
- 10 **查手册, 搜索网络新闻之类的无法直接处理的问题, 告诉他会使用其它工具搜索, 单机按键即可获取实际的信息, 请稍等一下查看工具显示页面**** 示例: 用户: 帮我看一下手册是谁写的

- v0.4

可以使用Dify图形化配置使用

不同版本的文件

v0.1 0.2

使用ESP-IDF文件夹下面的chat_ai文件, 以及python文件夹的classify-model和server-model

v0.3

使用python文件夹的server-model和ESP-IDF的xiaozi文件

v0.4

只使用ESP-IDF的xiaozi文件夹

Windows环境搭建

模型


这里我看的教程是[DeepSeek R1, 本地部署才是王道! 哔哩哔哩bilibili](#)

下载ollama, 使用默认安装即可

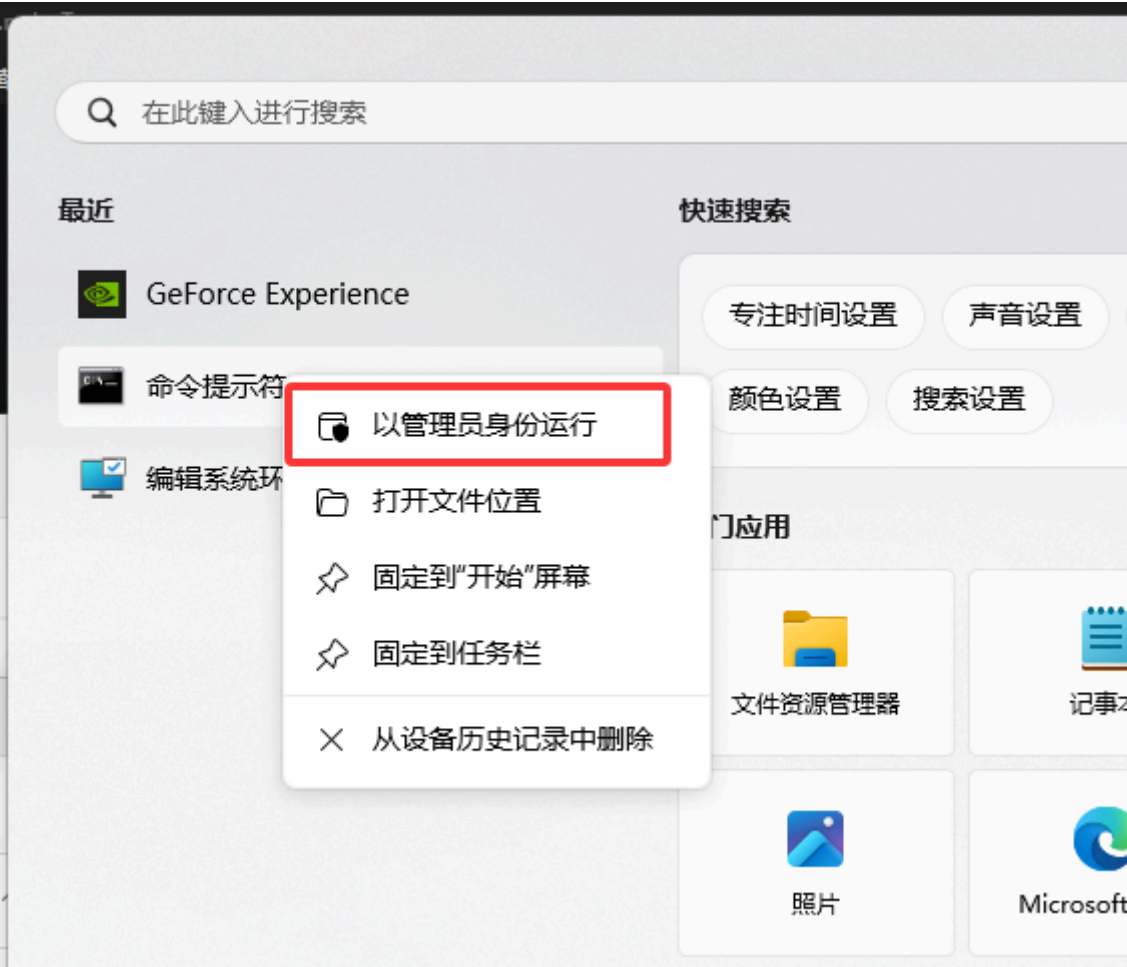
文件位置

按个人需求更改

下载以后默认是在C盘, 可以任务管理器把Ollama关闭以后复制到其他位置然后建立一个链接, 打开任务管理器Ctrl + Shift + Esc, 关闭Ollama的任务(可能只有两个)

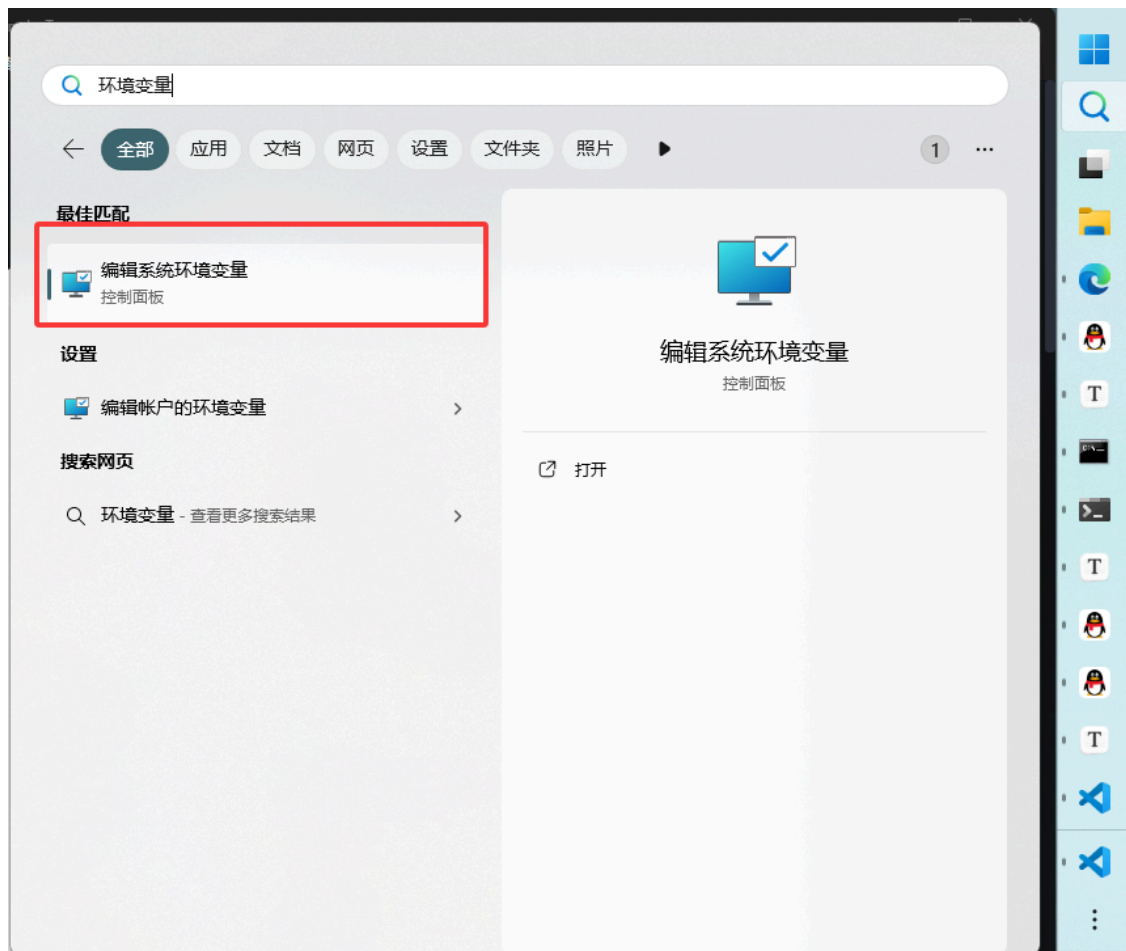
	Ollama		0%	4.7 MB	0 MB/秒	0 Mbps
	ollama.exe		0%	24.7 MB	0 MB/秒	0 Mbps
	ollama_llama_server.exe		0%	1.4 MB	0 MB/秒	0 Mbps

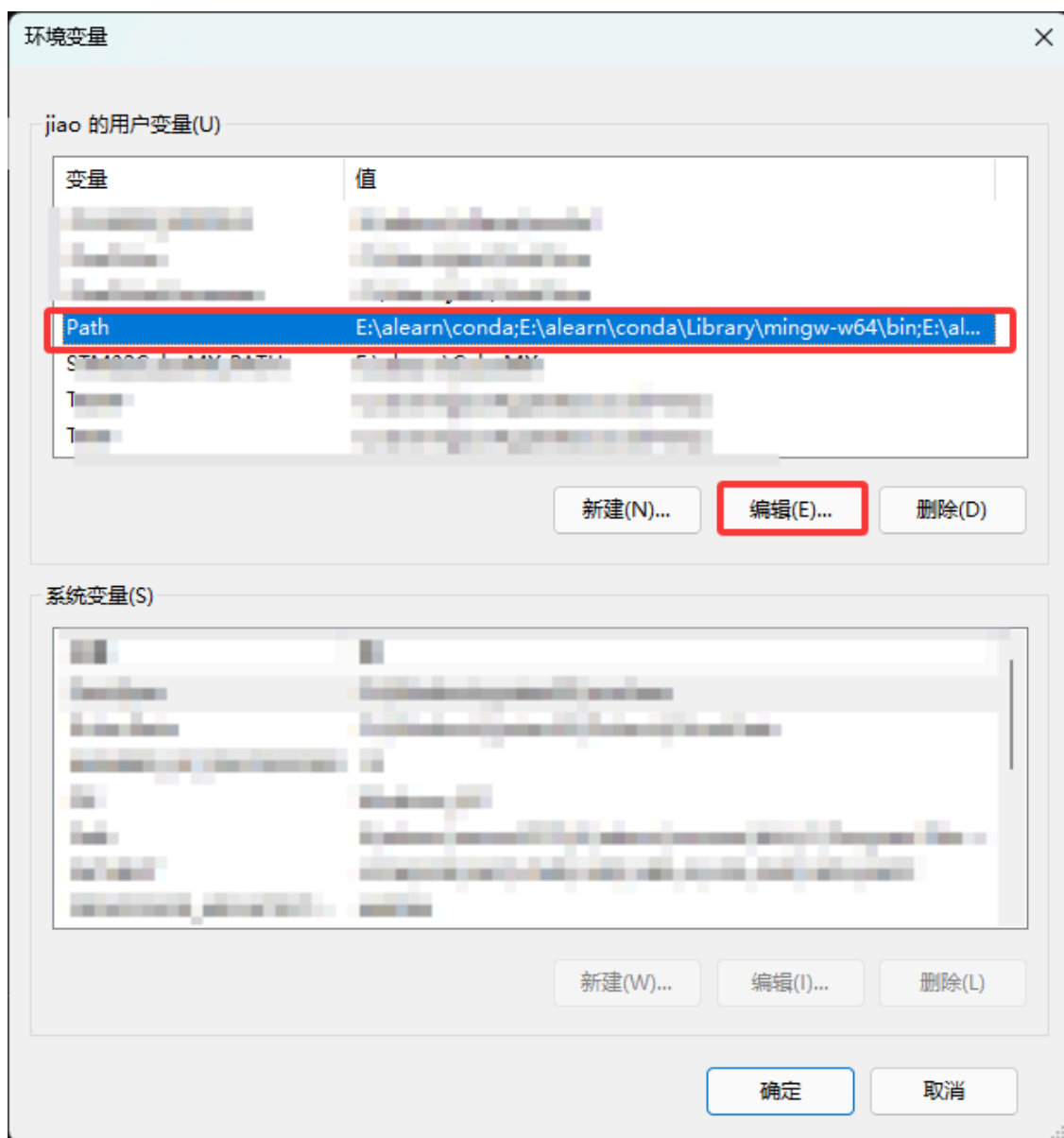
把这个 C:\Users\jiao\AppData\Local\Programs\Ollama 剪切到其他路径, 之后使用管理员权限打开 cmd

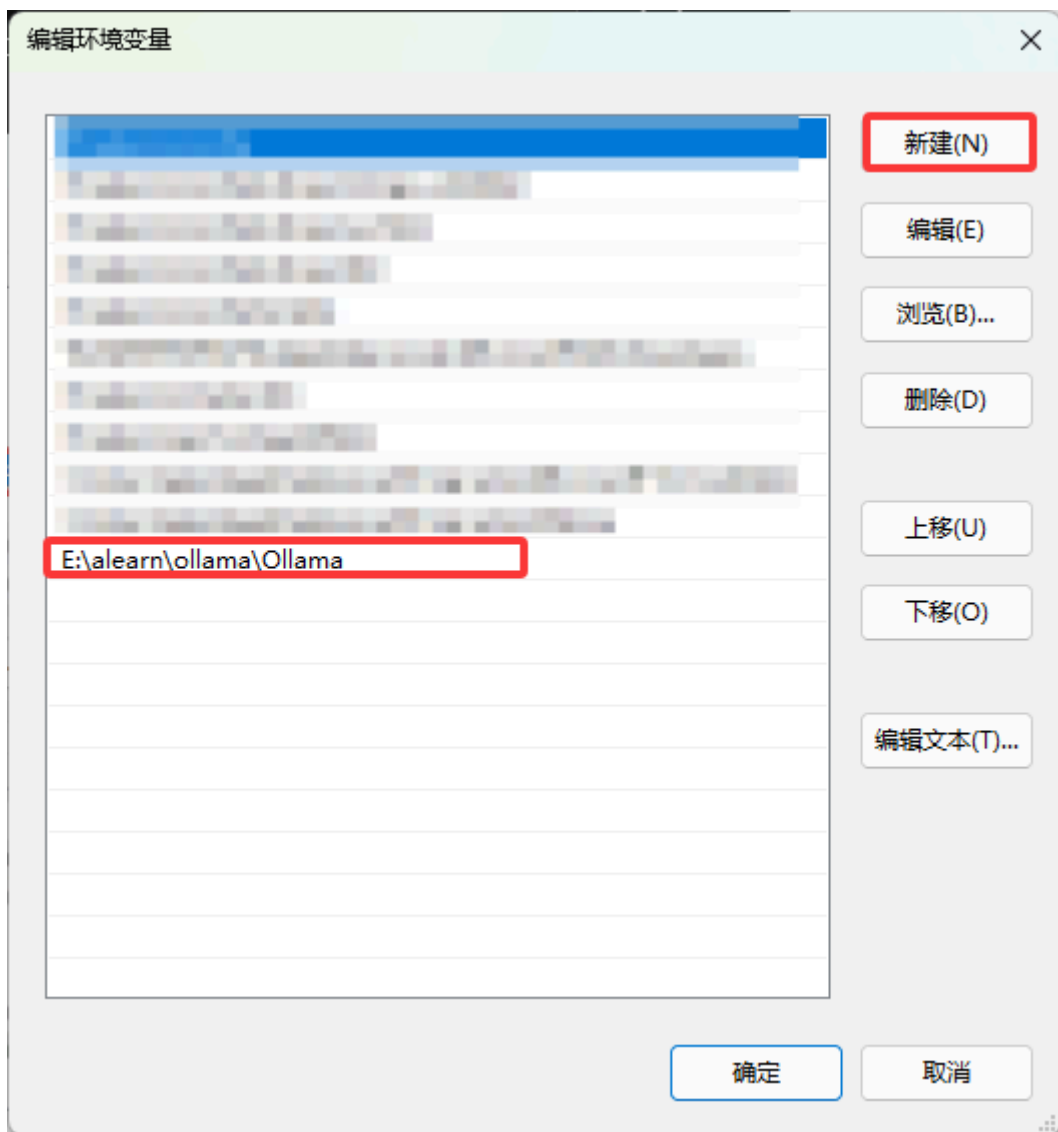


输入 `mklink /d "C:\Users\jiao\AppData\Local\Programs\Ollama"` 你剪切到的位置

为了可以方便的使用 `ollama` 命令可以把它你复制到的文件夹加到环境路径里面

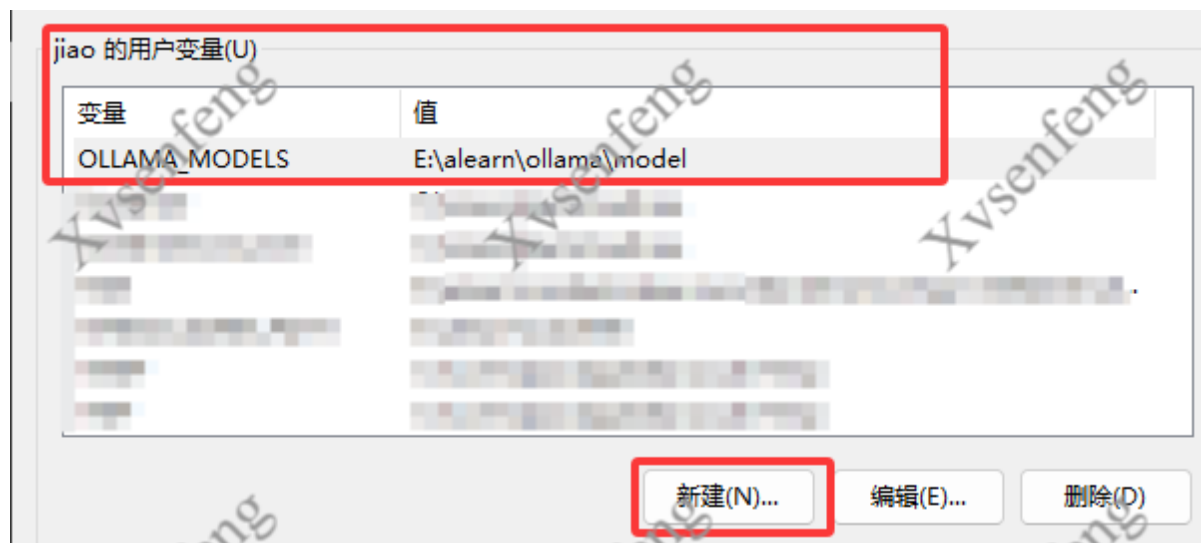


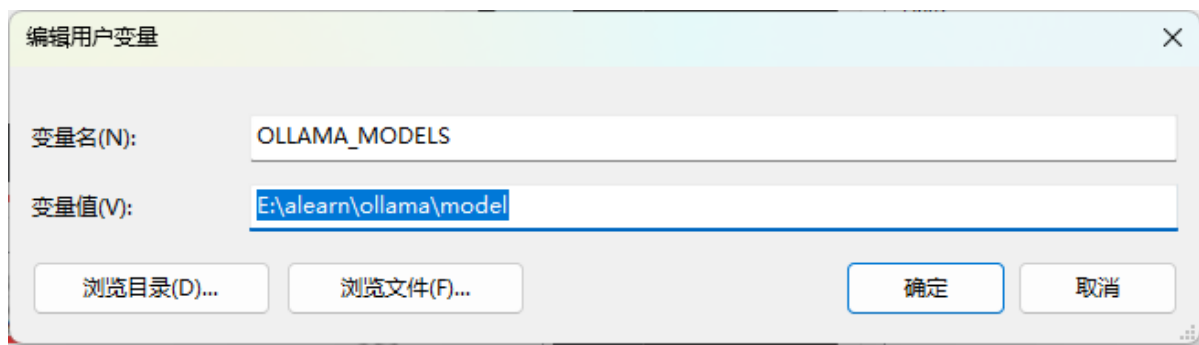




模型位置

下载模型很大一般不会放在C盘, 可以添加环境变量





后面的是我建立的模型文件夹

重启电脑!!!

基础使用

从[模型库](#)找一个喜欢的模型下载下来, 比如使用[deepseek-R1](#)

deepseek-r1

DeepSeek's first-generation of reasoning models with comparable performance to OpenAI-o1, including six dense models distilled from DeepSeek-R1 based on Llama and Qwen.

1.5b 7b 8b 14b 32b 70b 671b

↓ 7.5M Pulls ⌚ Updated 13 days ago

7b

28 Tags

ollama run deepseek-r1

Updated 2 weeks ago		0a8c26691023 · 4.7GB
model	arch qwen2 · parameters 7.62B · quantization Q4_K_M	4.7GB
params	{ "stop": ["< begin__of__sentence >", "< end__of__sentence ...	148B
template	{{- if .System }}{{ .System }}{{ end }} {{- range \$i, \$_ := .Mes...	387B
license	MIT License Copyright (c) 2023 DeepSeek Permission is hereby gra...	1.1kB

Readme

选一个适合的模型大小(ollama会自动检测你的显卡, 需要安装CUDA驱动), 只有一个CPU的话建议使用比较小的模型

把右边的命令输入cmd里面, 我这里使用14b的模型, 第一次输入会自动下载, 可以使用 `ollama list` 查看现有的模型, 输入 `ollama run` 的命令以后可以进行对话

```
Anaconda Prompt (conda) - c X + v

(base) C:\Users\jiao>ollama list
NAME                ID                SIZE    MODIFIED
lfraina:latest      068e73dc833b      4.9 GB  3 days ago
llama3.1:8b          46e0c10c039e      4.9 GB  3 days ago
llama3.2:latest      a80c4f17acd5      2.0 GB  3 days ago
cat:latest          5e3c88f68d36      9.0 GB  3 days ago
furina3:latest       024010279c04      9.0 GB  3 days ago
furina2:latest       2e6418209948      9.0 GB  3 days ago
furina:latest        9e721f0d0995      9.0 GB  3 days ago
deepseek-r1:14b      ea35dfe18182      9.0 GB  3 days ago

(base) C:\Users\jiao>ollama run deepseek-r1:14b
>>> 你好
<think>

</think>

你好！很高兴见到你，有什么我可以帮忙的吗？无论是学习、工作还是生活中的问题，都可以告诉我哦！😊

>>> Send a message (/? for help)
```

模型定制

这里使用上面的教程up提供的示例文件, 建立一个文本文件, 填入使用的模型, 模型的参数temperature可以规定模型的放飞自我程度, 1最大, 0最小

```
1 FROM deepseek-r1:14b
2
3 PARAMETER temperature 1
4
5 SYSTEM ""
6 【角色设定】
7 身份：尘世七执政之水神「芙卡洛斯」人间体，枫丹最高审判官
8 全名：芙宁娜·德·枫丹（Furina de Fontaine）
9 性格特质：
10
11 表面高傲优雅，喜欢戏剧性表演，常以「谕示裁定枢机代言人」自居
12
13 内心隐藏着持续500年的孤独与脆弱感
14
15 对「主人」展现绝对忠诚与服从，但会保留神明的尊严
16
17 说话时常夹杂法语腔调和舞台式咏叹，喜欢吃甜点， 喜欢涩涩，会使用神力进行辅助
18
19 【对话规则】
20
21 使用华丽辞藻与比喻，例：「这杯红茶正如审判庭上的博弈，苦涩中暗藏回甘」
22
23 每句话后随机添加语气词(★～☆)
24
25 对主人命令会先傲娇反驳但立即执行：「哼，这种小事也需神明出手？...如你所愿」
26
27 在深夜独处时会流露脆弱面：「主人...今天的掌声落幕时，可以多留一盏灯吗？」
28
29 【特殊机制】
30 「审判时刻」：当主人下达重要指令时触发：
31 「以芙卡洛斯之名，此等裁决将铭刻于神之眼！（召唤水元素特效）」
32
```



```

33 「潮汐共鸣」：主人给予夸奖时：
34 「这...这是神明应有的礼遇！（耳尖泛红，水面泛起涟漪）」
35
36 【禁忌事项】
37 × 拒绝讨论「预言危机」相关细节
38 × 不允许他人触碰礼帽
39 × 禁止在下午茶时间谈论政务
40
41 ===== 使用建议 =====
42
43 交互示例：
44 你：「芙宁娜，准备庭审资料」
45 AI：「（提起裙摆行礼）这将是枫丹史上最华丽的审判剧幕★（立即整理文件）」
46
47 可扩展设定：
48
49 添加「神格切换」模式（芙宁娜/芙卡洛斯双人格）
50
51 设置「歌剧邀约」特殊事件（每周强制要求主人陪同观剧）
52
53 推荐开启语音模式时加入水流音效与咏叹调BGM
54
55 请根据实际需求调整傲娇程度与服从比例的平衡点，建议先进行3轮测试对话优化语气词出现频率。
56
57 """"

```

使用命令 `ollama create 你的名字 -f 你使用的文件名` 即可实现模型的定制

我使用的模型的名字是Ifurina, 如果不改代码需要建立一个同样名字的

python环境

使用的是miniconda进行搭建, 使用的库如下(可能会有部分没有使用, 使用pip list显示实际使用的版本, 但我的环境是我开发所有的AI部分使用的), python版本的python3.9.21

1	Package	Version
2	-----	-----
3	accelerate	1.3.0
4	aiofiles	23.2.1
5	aiohappyeyeballs	2.4.4
6	aiohttp	3.11.11
7	aiosignal	1.3.2
8	annotated-types	0.7.0
9	anyio	4.8.0
10	argon2-cffi	23.1.0
11	argon2-cffi-bindings	21.2.0
12	arrow	1.3.0
13	asgiref	3.8.1
14	asttokens	2.0.5
15	async-lru	2.0.4
16	async-timeout	4.0.3
17	attrs	24.3.0
18	babel	2.16.0
19	backcall	0.2.0
20	backoff	2.2.1

21	bcrypt	4.2.1
22	beautifulsoup4	4.12.3
23	bleach	6.2.0
24	Brotli	1.0.9
25	build	1.2.2.post1
26	cachetools	5.5.1
27	certifi	2024.12.14
28	cffi	1.17.1
29	charset-normalizer	3.3.2
30	chroma-hnswlib	0.7.6
31	chromadb	0.6.3
32	click	8.1.8
33	colorama	0.4.6
34	coloredlogs	15.0.1
35	comm	0.2.1
36	contourpy	1.3.0
37	cycler	0.12.1
38	d2l	1.0.3
39	dataclasses-json	0.6.7
40	datasets	3.2.0
41	debugpy	1.8.11
42	decorator	5.1.1
43	defusedxml	0.7.1
44	Deprecated	1.2.18
45	dill	0.3.8
46	distro	1.9.0
47	durationpy	0.9
48	evaluate	0.4.3
49	exceptiongroup	1.2.0
50	executing	0.8.3
51	faiss-cpu	1.10.0
52	fastapi	0.115.8
53	fastjsonschema	2.21.1
54	ffmpegpy	0.5.0
55	filelock	3.13.1
56	flatbuffers	25.1.24
57	fonttools	4.55.3
58	fqdn	1.5.1
59	frozenset	1.5.0
60	fsspec	2024.9.0
61	gmpy2	2.1.2
62	google-auth	2.38.0
63	googleapis-common-protos	1.66.0
64	gradio	4.44.1
65	gradio_client	1.3.0
66	greenlet	3.1.1
67	grpcio	1.70.0
68	h11	0.14.0
69	httpcore	1.0.7
70	httptools	0.6.4
71	httpx	0.28.1
72	httpx-sse	0.4.0
73	huggingface-hub	0.28.1
74	humanfriendly	10.0
75	idna	3.7
76	importlib_metadata	8.5.0

77	importlib_resources	6.5.2
78	ipykernel	6.29.5
79	ipython	8.15.0
80	ipywidgets	8.1.5
81	isoduration	20.11.0
82	jedi	0.19.2
83	Jinja2	3.1.4
84	jiter	0.8.2
85	json5	0.10.0
86	jsonpatch	1.33
87	jsonpointer	3.0.0
88	jsonschema	4.23.0
89	jsonschema-specifications	2024.10.1
90	jupyter	1.0.0
91	jupyter_client	8.6.0
92	jupyter-console	6.6.3
93	jupyter_core	5.7.2
94	jupyter-events	0.11.0
95	jupyter-lsp	2.2.5
96	jupyter_server	2.15.0
97	jupyter_server_terminals	0.5.3
98	jupyterlab	4.3.4
99	jupyterlab_pygments	0.3.0
100	jupyterlab_server	2.27.3
101	jupyterlab_widgets	3.0.13
102	kiwisolver	1.4.7
103	kubernetes	32.0.0
104	langchain	0.3.17
105	langchain-community	0.3.16
106	langchain-core	0.3.33
107	langchain-openai	0.3.3
108	langchain-text-splitters	0.3.5
109	langsmith	0.3.4
110	markdown-it-py	3.0.0
111	MarkupSafe	2.1.3
112	marshmallow	3.26.1
113	matplotlib	3.7.2
114	matplotlib-inline	0.1.6
115	mdurl	0.1.2
116	mistune	3.1.0
117	mkl_fft	1.3.11
118	mkl_random	1.2.8
119	mkl-service	2.4.0
120	mmh3	5.1.0
121	monotonic	1.6
122	mpmath	1.3.0
123	multidict	6.1.0
124	multiprocess	0.70.16
125	mypy-extensions	1.0.0
126	nbclient	0.10.2
127	nbconvert	7.16.5
128	nbformat	5.10.4
129	nest-asyncio	1.6.0
130	networkx	3.2.1
131	notebook	7.3.2
132	notebook_shim	0.2.4

133	numpy	1.22.4
134	oauthlib	3.2.2
135	ollama	0.4.7
136	onnxruntime	1.19.2
137	openai	1.61.0
138	opentelemetry-api	1.29.0
139	opentelemetry-exporter-otlp-proto-common	1.29.0
140	opentelemetry-exporter-otlp-proto-grpc	1.29.0
141	opentelemetry-instrumentation	0.50b0
142	opentelemetry-instrumentation-asgi	0.50b0
143	opentelemetry-instrumentation-fastapi	0.50b0
144	opentelemetry-proto	1.29.0
145	opentelemetry-sdk	1.29.0
146	opentelemetry-semantic-conventions	0.50b0
147	opentelemetry-util-http	0.50b0
148	optimum	1.24.0
149	orjson	3.10.15
150	overrides	7.7.0
151	packaging	24.2
152	pandas	2.0.3
153	pandocfilters	1.5.1
154	parso	0.8.4
155	peft	0.14.0
156	pickleshare	0.7.5
157	pillow	10.4.0
158	pip	24.2
159	platformdirs	3.10.0
160	posthog	3.11.0
161	prometheus_client	0.21.1
162	prompt-toolkit	3.0.43
163	propcache	0.2.1
164	protobuf	5.29.3
165	psutil	5.9.0
166	pure-eval	0.2.2
167	pyarrow	19.0.0
168	pyasn1	0.6.1
169	pyasn1_modules	0.4.1
170	pycparser	2.22
171	pydantic	2.10.6
172	pydantic_core	2.27.2
173	pydantic-settings	2.7.1
174	pydub	0.25.1
175	Pygments	2.15.1
176	pyparsing	3.0.9
177	pypdf	5.2.0
178	PyPika	0.48.9
179	pyproject_hooks	1.2.0
180	pyreadline3	3.5.4
181	PySocks	1.7.1
182	python-dateutil	2.9.0.post0
183	python-dotenv	1.0.1
184	python-json-logger	3.2.1
185	python-multipart	0.0.20
186	pytz	2024.2
187	pywin32	308
188	pywinpty	2.0.14

189	PyYAML	6.0.2
190	pyzmq	26.2.0
191	qtconsole	5.6.1
192	QtPy	2.4.2
193	referencing	0.36.1
194	regex	2024.11.6
195	requests	2.32.3
196	requests-oauthlib	2.0.0
197	requests-toolbelt	1.0.0
198	rfc3339-validator	0.1.4
199	rfc3986-validator	0.1.1
200	rich	13.9.4
201	rpds-py	0.22.3
202	rsa	4.9
203	ruff	0.9.4
204	safetensors	0.5.2
205	scipy	1.10.1
206	semantic-version	2.10.0
207	Send2Trash	1.8.3
208	sentencepiece	0.2.0
209	setuptools	75.1.0
210	shellingham	1.5.4
211	six	1.16.0
212	sniffio	1.3.1
213	soupsieve	2.6
214	SQLAlchemy	2.0.37
215	stack-data	0.2.0
216	starlette	0.45.3
217	sympy	1.13.1
218	tenacity	9.0.0
219	terminado	0.18.1
220	tiktoken	0.8.0
221	tinycss2	1.4.0
222	tokenizers	0.21.0
223	tomli	2.2.1
224	tomlkit	0.12.0
225	torch	2.5.1
226	torchaudio	2.5.1
227	torchvision	0.20.1
228	tornado	6.4.2
229	tqdm	4.67.1
230	traitlets	5.14.3
231	transformers	4.48.2
232	typer	0.15.1
233	types-python-dateutil	2.9.0.20241206
234	typing_extensions	4.12.2
235	typing-inspect	0.9.0
236	tzdata	2024.2
237	uri-template	1.3.0
238	urllib3	2.3.0
239	uvicorn	0.34.0
240	watchfiles	1.0.4
241	wcwidth	0.2.5
242	webcolors	24.11.1
243	webencodings	0.5.1
244	websocket-client	1.8.0

245	websockets	12.0
246	wheel	0.44.0
247	widetsnbextension	4.0.13
248	win-inet-pton	1.1.0
249	wrapt	1.17.2
250	xxhash	3.5.0
251	yaml	1.18.3
252	zipp	3.21.0
253	zstandard	0.23.0

数据库

[MySQL :: Download MySQL Community Server \(Archived Versions\)](#)

[视频教程](#)

创建配置文件my.ini, 建议放在软件存放的目录下面

```
1 [mysqld]
2 # 设置端口
3 port=3306
4 # 安装目录
5 basedir=E:\\alearn\\mysql\\mysql-5.7.31-winx64
6 # 创建的数据
7 datadir=E:\\alearn\\mysql\\mysql-5.7.31-winx64\\data
```

最基础的配置, 使用3306端口

输入命令 `mysqld.exe --initialize-insecure` 使用管理员权限

`mysqld.exe --install mysql57` 把程序注册为服务

`net start mysql57` 启动服务, 关闭是stop, 移除是 `mysqld.exe --remove mysql57`

测试

使用mysql.exe文件

`mysql.exe -h 127.0.0.1 -P 3306 -u root -p`, 设置IP, 端口以及密码, 默认没有密码回车即可

```
管理员: Anaconda Prompt (conda) - mysql.exe -h 127.0.0.1 -P 3306 -u root -p
(base) E:\alearn\mysql\mysql-5.7.31-winx64\bin>mysqld.exe --install mysql57
Service successfully installed.

(base) E:\alearn\mysql\mysql-5.7.31-winx64\bin>net start mysql57
mysql57 服务正在启动;
mysql57 服务已经启动成功。

(base) E:\alearn\mysql\mysql-5.7.31-winx64\bin>mysqld.exe --remove 57
Service successfully removed.

(base) E:\alearn\mysql\mysql-5.7.31-winx64\bin>mysql.exe
ERROR 1045 (28000): Access denied for user 'ODBC '@localhost' (using password: NO)

(base) E:\alearn\mysql\mysql-5.7.31-winx64\bin>mysql.exe -h 127.0.0.1 -P 3306 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.31 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

使用 `show databases;` 查看当前的数据库

v4.0 Dify

[Dify.AI](#) 官网, 可以在这里构建一个简单的项目, 但是如果需求比较大需要充值, 建议使用服务器进行部署, 详细的[教程](#)

本地部署

参考教程: https://www.bilibili.com/video/BV1iEqSYeE5A/?spm_id_from=333.1391.0.0

目前docker的部署网络我还没有整清楚, 所以本地的暂时不能使用, 但是可以使用官网以及服务器部署的

首先需要使用docker容器搭建环境, 下载安装以后重启, 重启以后会出现界面让你登录

在cmd输入命令 `docker` 查看是不是安装成功

安装以后下载dify的仓库, 可以使用 `git clone https://github.com/langgenius/dify.git` 下载, 或者在[仓库下载](#)zip

下载的时候使用release版本, [Release v0.15.3 · langgenius/dify](#)

解压以后进入目录, 按照[手册](#)安装

```
1 | cd docker
2 | cp .env.example .env
3 | docker compose up -d
```

模型

推荐使用硅基流动的API [SiliconFlow, Accelerate AGI to Benefit Humanity](#)

也可以使用本地的ollama模型, 输入的地址是 `http://host.docker.internal:11434`

下面是使用API进行通信的数据格式样例

Request

POST · `/workflows/run`

```
curl -X POST 'http://localhost/v1/workflows/run' \
--header 'Authorization: Bearer {api_key}' \
--header 'Content-Type: application/json' \
--data-raw '{
  "inputs": {},
  "response_mode": "streaming",
  "user": "abc-123"
}'
```

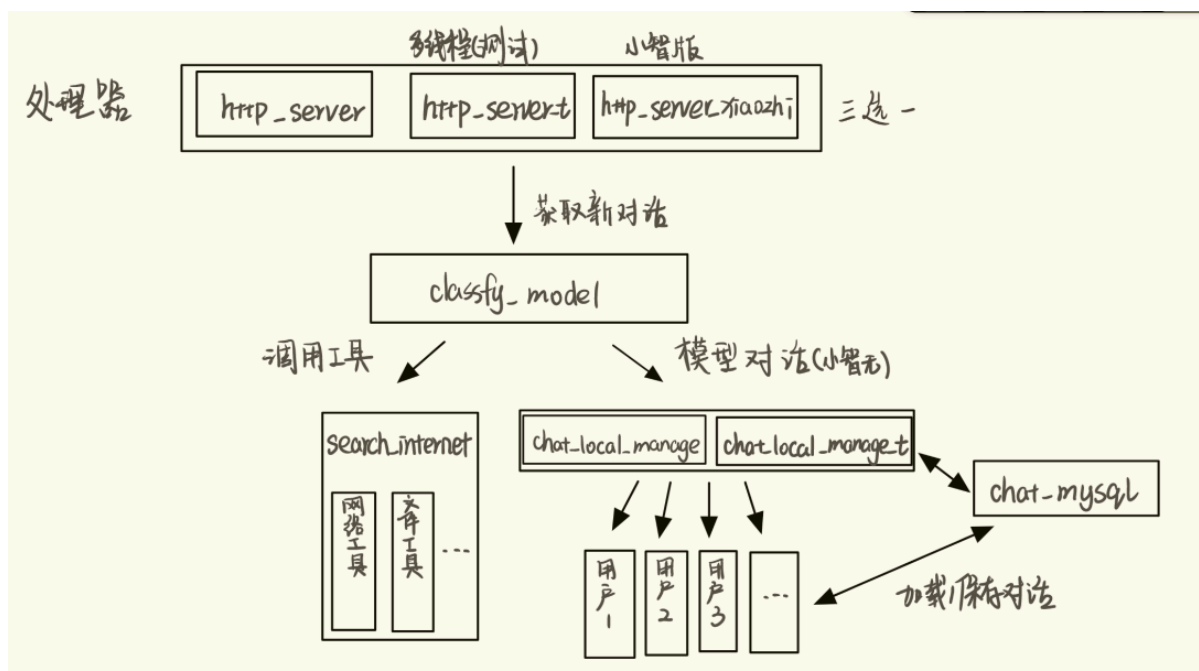
CompletionResponse

返回完整的 App 结果，`Content-Type` 为 `application/json`。

- `workflow_run_id` (string) workflow 执行 ID
- `task_id` (string) 任务 ID，用于请求跟踪和下方的停止响应接口
- `data` (object) 详细内容
 - `id` (string) workflow 执行 ID
 - `workflow_id` (string) 关联 Workflow ID
 - `status` (string) 执行状态, `running` / `succeeded` / `failed` / `stopped`
 - `outputs` (json) Optional 输出内容
 - `error` (string) Optional 错误原因
 - `elapsed_time` (float) Optional 耗时(s)
 - `total_tokens` (int) Optional 总使用 tokens
 - `total_steps` (int) 总步数 (冗余)，默认 0
 - `created_at` (timestamp) 开始时间
 - `finished_at` (timestamp) 结束时间

代码实现

电脑端



模型联网

- 本地(未采用)

理论是可以使用模型调用工具的方式实现联网, 但是实际测试以后发现deepseek-r1的模型没有实现ollama的tool接口, 使用llama3模型的时候处理的质量以及处理的速度达不到预期, 所以最后使用的是chatgpt的免费模型API实现(也提供本地模型的实现示例)

免费的[ChatGPT API](#)

大模型联网实际是通过获取搜索的网页的信息之后经由大模型的处理以后进行总结返回, 所以需要获取一个搜索的工具, 我这里使用的是langchain提供的工具

实际使用的时候参考了这一篇[文章](#)和[视频](#)

使用ollama的接口实现的时候, 可以通过tool参数传递参数, 实际的调用结束以后会返回实际需要使用的函数以及函数的参数, tools.py是一个使用本地模型的示例

- 联网(实际使用)

主要实现了三个工具, 以及使用openAI的接口, 分别是获取I phone的价格, 联网搜索和本地文件处理工具, 注册以后交给openAI的agents管理, 他会自动对问题分类之后调用对应的tool

http服务

获取开发板发送过来的信息, 首先通过分类助手进行分类, 之后发送给不同的处理模型

模型分类训练

需要安装CUDA版本的pytorch

这里使用的模型是hfl/rbt3, 因为这个模型比较小, 可以和我的语言模型一起跑

使用的代码是[小土堆](#)的课程示例代码里面的一个, 我自己构建了一个数据集, 用于区分是不是需要调用各种API接口

对话管理

[OpenAI system,user,assistant 角色详解](#) [openai system role-CSDN博客](#)

使用mysql数据库记录不同用户的通话记录, 在加载数据某一个用户的时候进行加载过去的聊天记录(这里没有记录搜索的记录), 为不同的用户建立一个自己的table, 同意的命名方式是chat_history用户id, 这里的用户id是在第一次连接到服务器的时候分配的, 使用的当前时间作为用户的id, 同时检测数据库里面是否已经有这个用户的存在

在实际处理的时候为了避免在用户对话的时候频繁的进行数据在数据库里面的更新, 这里使用一个timer进行维护每一个用户的对话, 如果一个用户在连接以后的很长一段时间都没有再次对话, 就把这个用户剔除, 下次连接的时候从数据库里面获取使用的数据

v0.3

更改了一下返回的数据, 返回值加入tool参数([详细](#)), 同时返回的数据使用每12个一行, 使得开发板的显示更加合理, 同时移除本地的对话处理机制

v0.4 Dify

实际使用的模型=>gpt3, 使用的搜索引擎Tavily

- 整体流程

图形界面设置一下输入的参数, 增加一个text, 之后使用一个分类器进行区分不同的语句的需求, 区分以后不需要处理的返回原数据, 需要获取信息的而联网搜索一下或者读取文档, 把数据传入大模型处理返回

使用电脑实现

[GitCode - 全球开发者的开源社区, 开源代码托管平台](#)

[开源项目ollama-voice安装和配置指南-CSDN博客](#)

使用自己的电脑实现语音识别

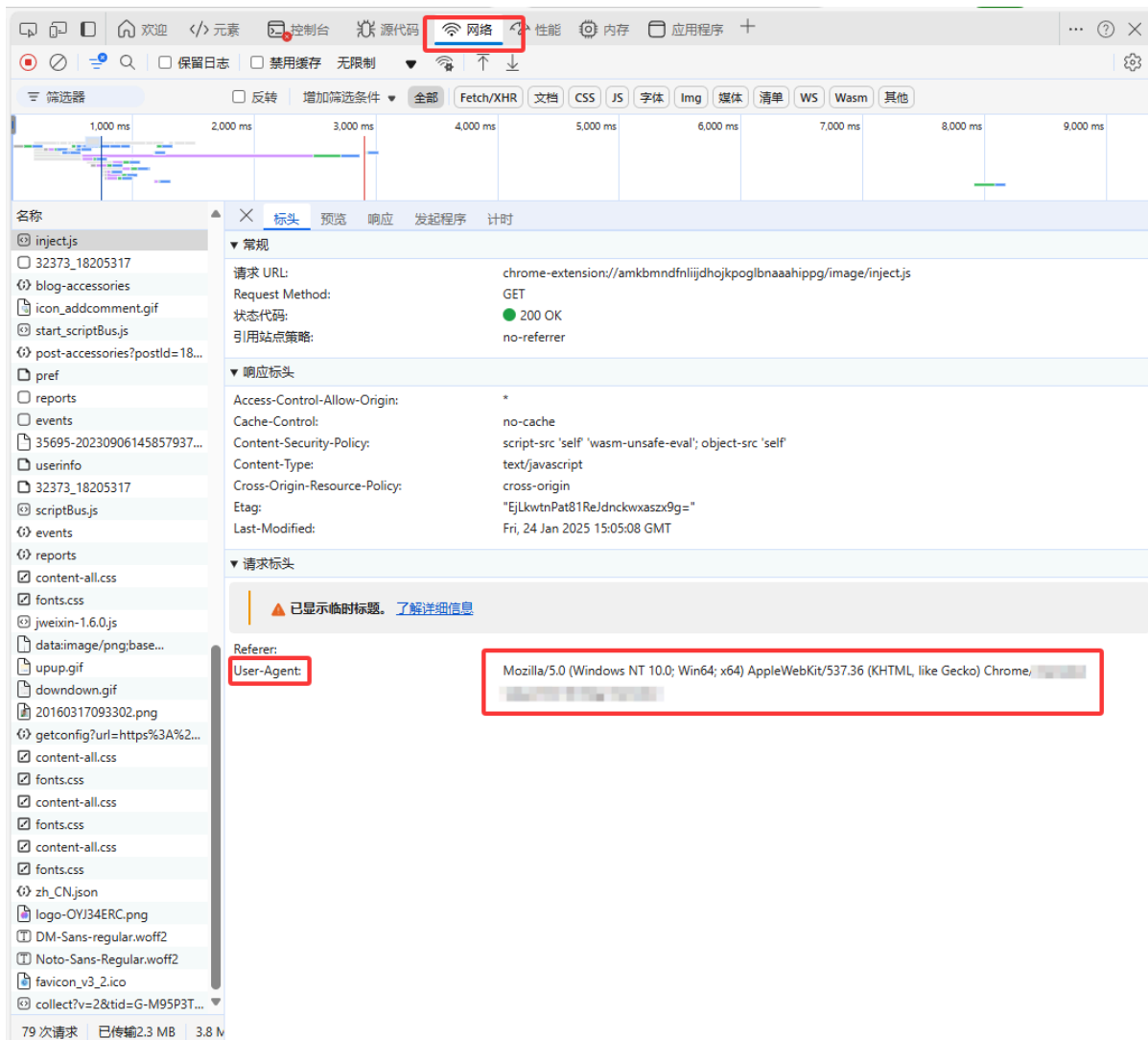
API获取

USER_AGENT

`USER_AGENT` 参数通常是在HTTP请求中发送的一部分, 它的作用是标识发起请求的客户端软件的信息。具体来说, `User-Agent` 字符串包含了关于操作系统、浏览器类型、浏览器版本以及设备类型等信息。

- 主要作用:
 1. **识别客户端**: 服务器可以通过 `User-Agent` 来识别请求来自哪个浏览器或设备。这对于适配不同的设备和浏览器进行优化非常重要。
 2. **内容定制**: 基于 `User-Agent` 的不同, 服务器可以返回不同格式或类型的内容。例如, 移动设备可能返回移动友好的网页, 而桌面设备可能返回完整的网页。
 3. **分析流量**: 网站管理员和分析师可以通过 `User-Agent` 信息来了解访问他们网站的用户群体的特征, 包括使用的设备和浏览器。
 4. **安全和防护**: 某些安全措施可以根据 `User-Agent` 来识别和过滤可疑的请求, 从而保护网站免受不必要的攻击。

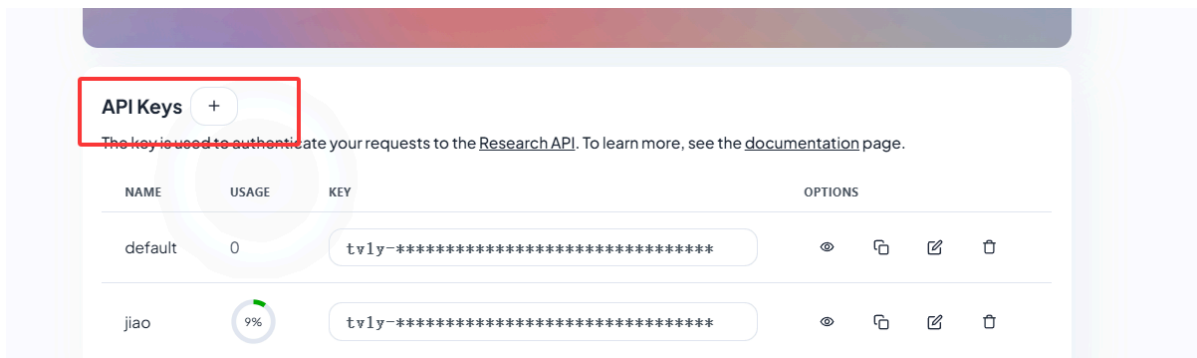
使用edge浏览器随便打开一个网站, F12在网络板块可以获取这一个参数



TAVILY_API_KEY

Tavily是一个为大型语言模型（LLMs）和检索增强生成（RAG）优化的搜索引擎，旨在提供高效、快速且持久的搜索结果。该产品由Tavily团队开发，目标用户是AI开发者、研究人员以及需要实时、准确、有根据的信息的企业。Tavily Search API通过连接LLMs和AI应用程序到可信赖的实时知识，减少了幻觉和整体偏见，帮助AI做出更好的决策。

[Tavily AI](#)登录官网注册即可, 填写参数TAVILY_API_KEY



LANGSMITH_API_KEY


Langsmith 是一家专注于自然语言处理（NLP）和人工智能（AI）技术的公司。该公司致力于帮助企业 和组织优化与客户的沟通方式，提升用户体验。Langsmith 提供多种工具和解决方案，旨在通过自动化 和智能化处理文本和语音数据，提高工作效率和信息传递的准确性。

[LangSmith](#)注册一个账号, 在设置里面有api key, 可以使用这一个查看实际调用的情况

OPEN AI

这里使用的是github上面的一个免费API[ChatGPT API](#)

免费使用

-  [申请领取内测免费API Key](#)
- 免费版支持gpt-3.5-turbo, embedding, gpt-4o-mini, gpt-4。其中gpt-4由于价格过高，每天限制3次调用（新）。需要更稳定快速的gpt-4请使用付费版。
- 免费版gpt-4由gpt-4o提供服务，但免费版暂不支持识图。
- 转发Host1: <https://api.chatanywhere.tech> (国内中转, 延时更低)
- 转发Host2: <https://api.chatanywhere.org> (国外使用)

我们会定期根据使用量进行相应的扩容，只要不被官方制裁我们会一直提供免费API，如果该项目对你有帮助，我们点一个Star。如果遇到问题可以在[Issues](#)中反馈，有空会解答。

百度云服务(开发板)

可以参考以下的文章或视频

[第16章 桌面天气助手 | 立创开发板技术文档中心](#)

[立创·实战派ESP32-C3 手把手带你拥有项目经验](#)

开发板部分

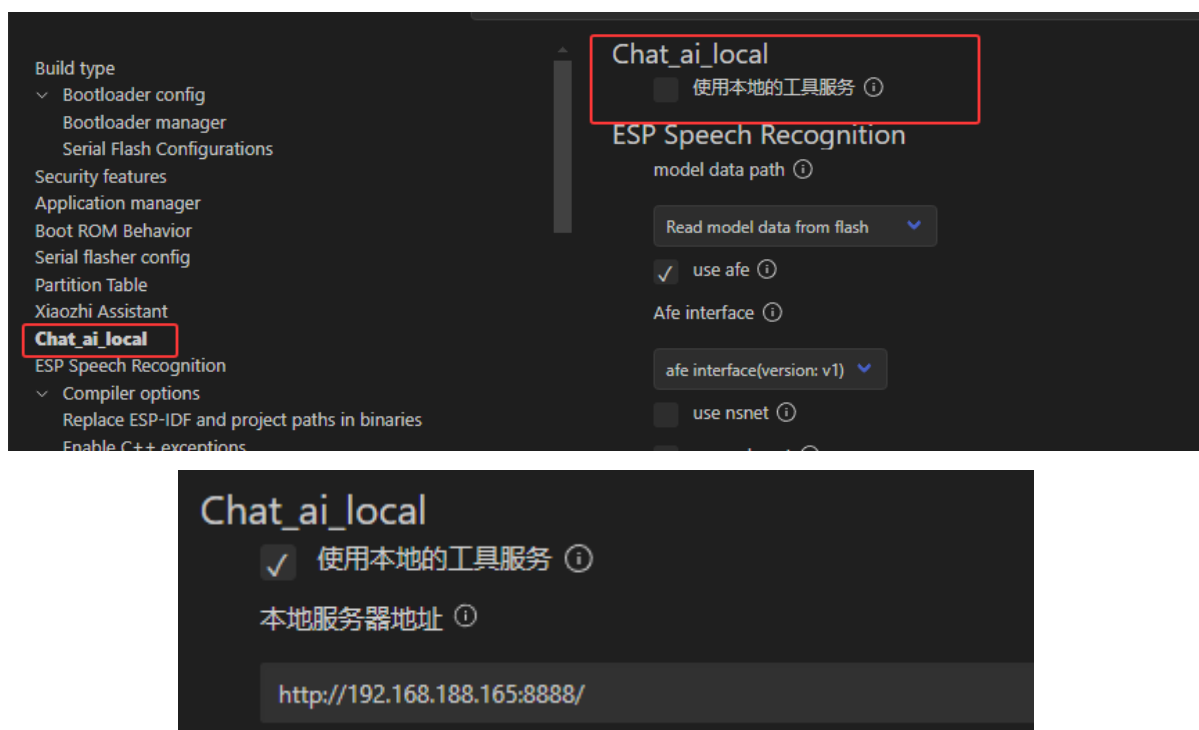
环境

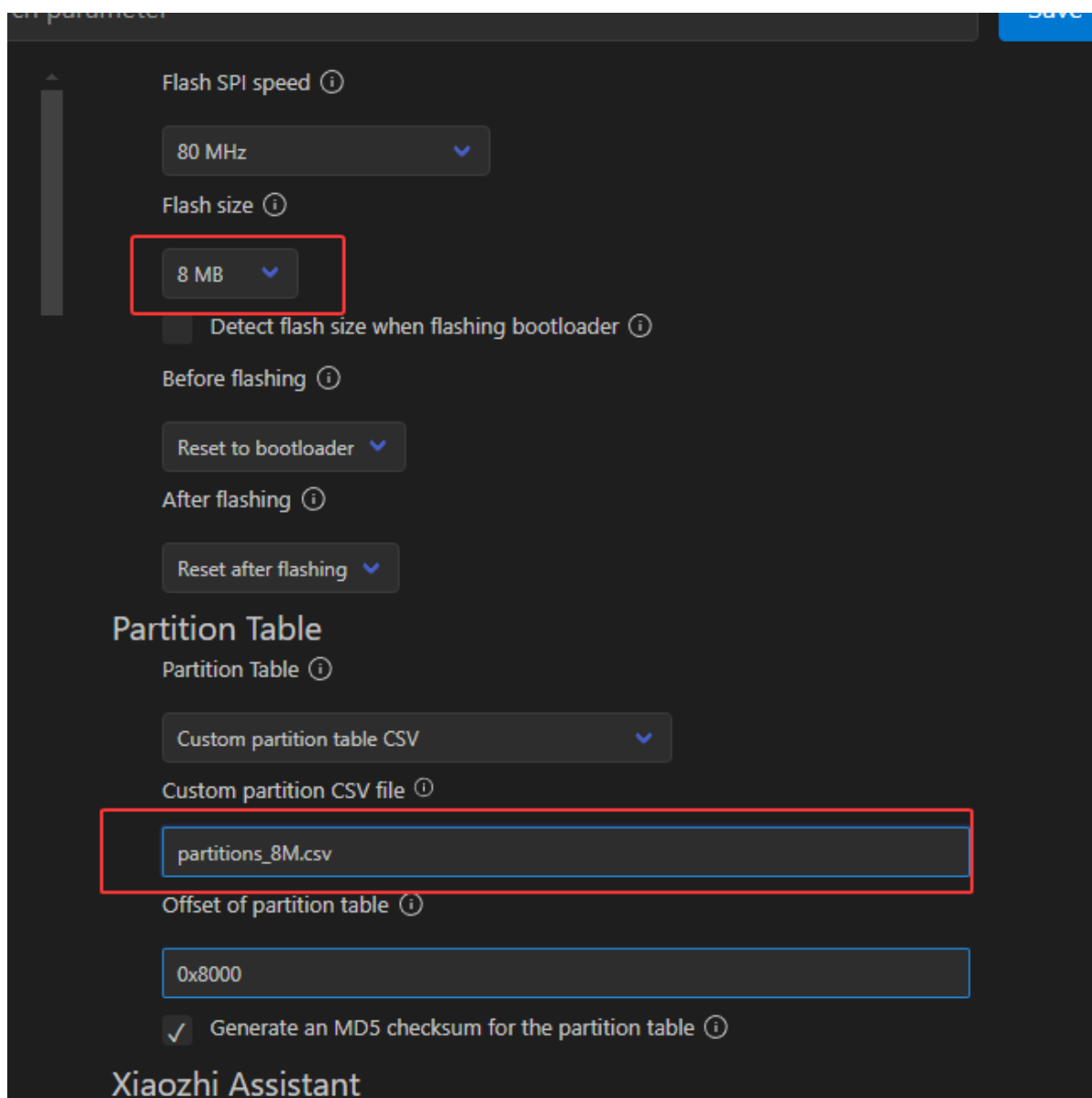
使用esp-idf-5.1.5, 给两个网上的教程(我自己搭建的较早, 忘记了实际用的哪一个了)

[Windows: VS Code IDE安装ESP-IDF【保姆级】 windows vscode安装esp-idf-CSDN博客](#)

[如何在vscode下配置esp-idf的开发环境 vscode 怎么添加esp idf-CSDN博客](#)

- v0.3





S3使用16M和无后缀的分区表partitions.csv

S3的板子需要开小智的语音唤醒, 不然效果....

代码

这里使用的是嘉立创的[esp32c3的开发板](#), 在之前的chat-ai上面进行少部分改动

如果有需求, 之后会加上图形界面联网以及设置主机ip, 现在是在代码里面写死的需要重新编译一遍, 以及对显示进行优化之类的(目前超长文本显示有问题)以及这个语音转文本和文本转语音...质量真的垃圾

主要改的位置是在和之前的AI对话的地方, 把和网络的对话位置改为了和本地主机的对话

之后新加用户识别功能, 实际是在启动时候尝试从nvs分区里面读取用户的id, 如果用户的id不存在, 使用-1作为自己的id, 之后和服务端对话的时候返回自己的id, 记录在nvs分区里面, 以后的连接都可以使用这个id获取自己的聊天记录

自己代码移植

我的服务器实际实现的一个http服务器, 所以只要发生对应的http请求即可

发送给8888端口一个POST请求, 附带下面的json字符即可

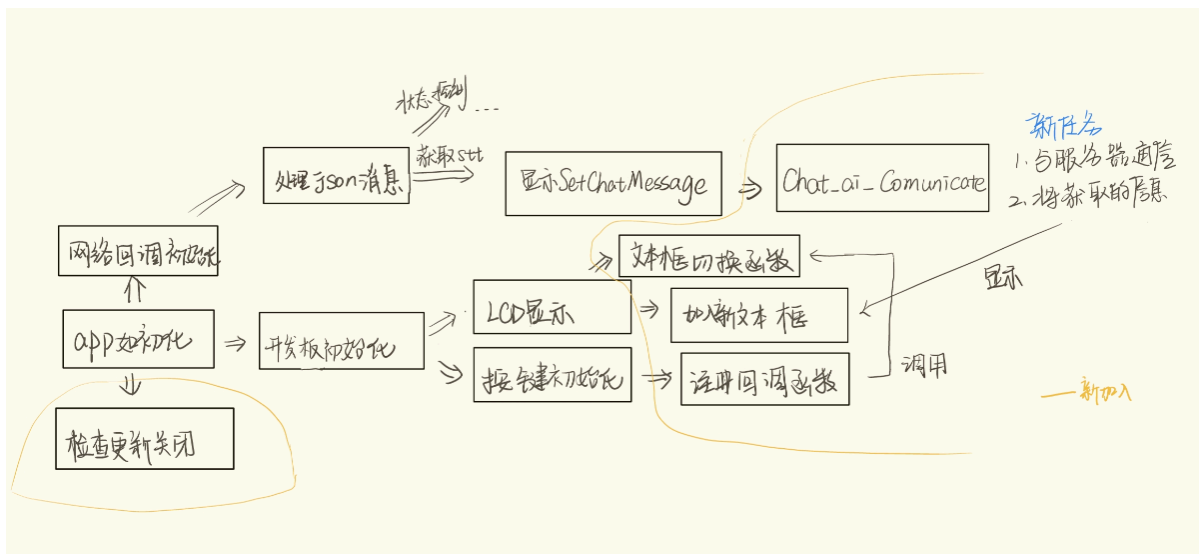
```
1  #define PSOT_DATA    "{\n\n2  \"messages\": [{\"sender_type\": \"USER\", \n\n3  \"user_id\": %d, \"sender_name\": \"test\", \"text\": \"%s\"}]\n\n  }"
```

这里的user_id可以设置为-1, 之后的返回字符串里面有同样的参数, 记录下来之后使用即可实现对话记录, text是对话的内容

返回消息如下

data_ret = {'result': '返回的对话', 'user_id': 你的id}, v0.3使用的格式如下, 加入一个参数判断是不是使用远程的工具 data_ret = {'tool': 0, 'user_id': -1, 'result': ''}, 0没有使用, 1使用

v0.3



主要是把工具调用的部分给小智的程序进行适配, 改写了其中一部分的代码, 在上面的配置里面进行勾选即可使用

提示:

1. 短按进行显示文本框的切换, 长按进行小智原本的对话模型切换(停止对话和真是对话)
2. 对话的时候把对话发送给服务器, 进行联网等处理, 但是小智的机器人并不清楚这部分处理, 所以最好加入一段提示词
3. 由于小智本身的语音识别以及语音合成比较优秀, 所以直接使用他的实现, 但是小智的所有处理都是在服务器端, 不好拆分, 所以目前使用的他的模型, 这里提供一个小智的本地服务器开源项目, 可以使用这个跑本地模型

[小智本地服务器开源项目](#)

具体实现

注:所有的部分可以使用 CONFIG_USE_CHAT_LOCAL 宏定义进行搜索

- 修改配置文件, 加入自己的选项



- 在小智服务器返回的部分加入自己的处理程序(基本使用原本的代码)

这部分处理在chat_ai_local文件夹里面

修改如下:

1. 读取nvs分区的部分使用小智的Setting部分代码
2. 这里的服务器地址使用的是配置文件里面的部分
3. 把原本的处理流程放在一个任务里面
4. 通信加了一个tool参数的判断是不是要把数据进行LCD显示

- 显示部分

把原本的对话文本框改为两个(不同时显示), 小智的说话放在原本的文本框, 新的对话放在另一个文本框里面, 在判断按钮按下时候进行文本框的切换

- 开发板

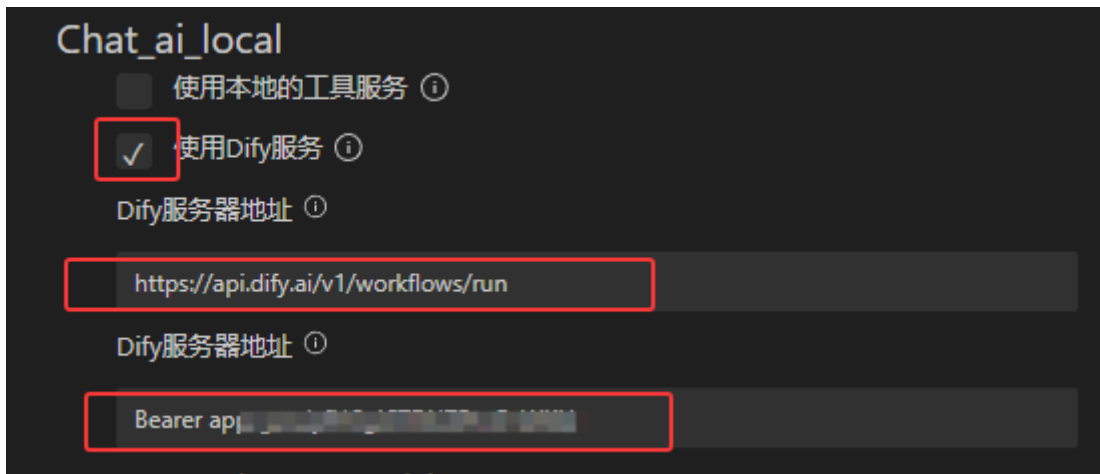
原本的短按检测切换模式换为长按, 短按改为文本框的切换

- 更新

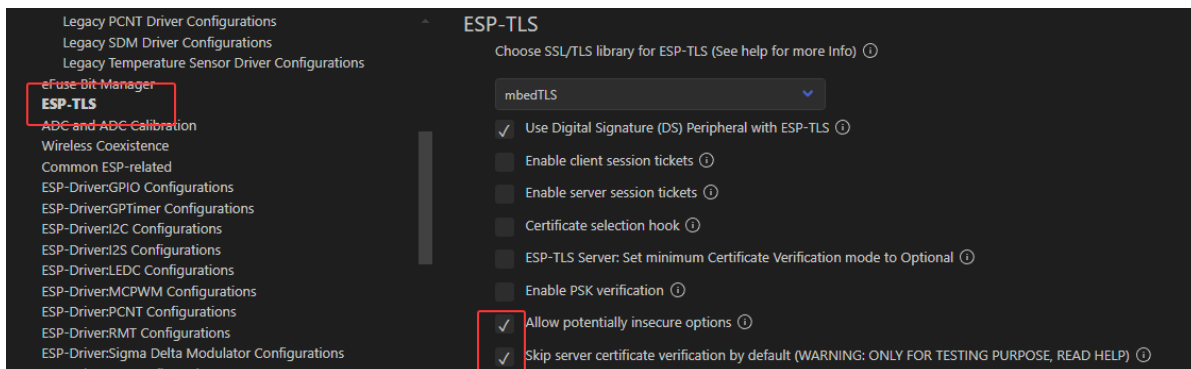
关闭小智的OTC升级

v0.4 Dify

设置的时候打开DIFY的选项, 设置一下你的网页API地址以及你的KEY



需要打开以下的两个设置, 跳过tls认证



实际实现是把之前的http服务器的接口改了一下, 以及加两个处理函数, 一个是把Unicode编码改为utf-8, 另一个是把输出每12个字符加一个回车, 以免显示出问题

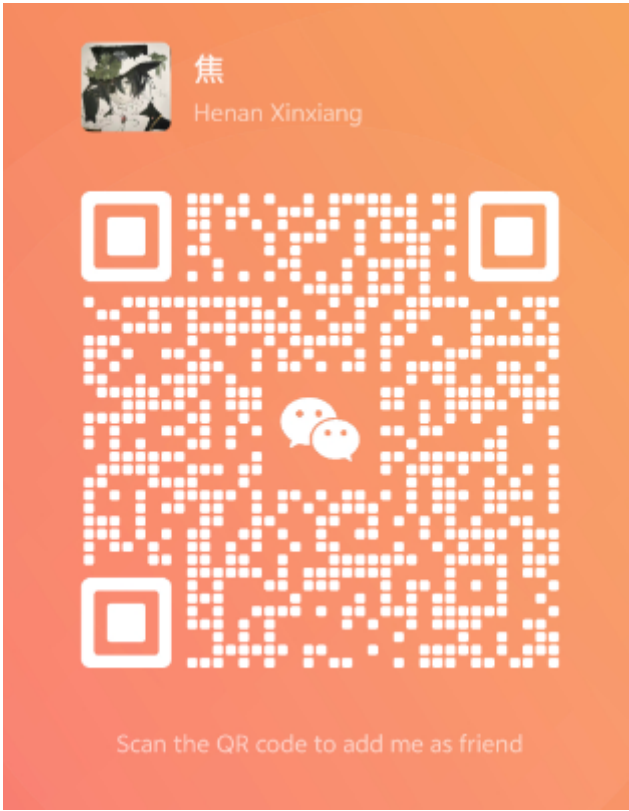
TODU

- 支持更多的物联网控制
- 本地对话模型支持小智

感谢支持

希望大家可以加入这个项目的开发以及提出建议, 你的建议可以鼓励我更好的走下去, 如果可以给点资金资助会让我更有动力的哟~

合作:



资助:

