缘分居官网

https://doc.yuanfenju.com/

注册之后可以自动获得密钥，右下角为我们需要的密钥token

qq号测吉凶

使用FunctionCalling扩展openai的能力，使之可以帮我们测qq号的吉凶。

```
import json
import requests

from openai import OpenAI

api_key = "你们自己的openai token"
base_url = "https://openai.zhixueyouke.cn/v1/"


client = OpenAI(api_key=api_key, base_url=base_url)


prompt = "测一下这个qq号 545679452 的吉凶?"

# 先定义一个外部方法，查询一个qq号的吉凶

qqJixiongUrl = (
    "https://api.yuanfenju.com/index.php/v1/Jixiong/qq?api_key=
{api_key}&qq=${qq}"
)

myQQ = "545679452999"
```

```python
def queryQQJixiong(qqNumber):
    url = qqJixiongUrl.format(api_key="RKoilS8orozOxFhBwLiJKUg86", qq=qqNumber)
    response = requests.get(url).json()
    print("\n queryQQJixiong 的查询结果是: ", response)
    return json.dumps(response, ensure_ascii=False)


# 然后，准备message和tools
messages = []  # 提示词

messages.append(
    {
        "role": "system",
        "content": "你是一个查询qq号吉凶的机器人,你要将用户输入的qq号码调用 查询qq号吉凶的函数来告诉用户他qq号的吉凶情况",
    },
)
messages.append(
    {
        "role": "user",
        "content": f"""我的qq号 {myQQ} 的吉凶情况如何?""",
    }
)

# 准备FunctionCalling的工具链
tools = [
    {
        "type": "function",
        "function": {
            "name": "queryQQJixiong",
            "description": "查询qq号吉凶",
            "parameters": {
                "type": "object",
                "properties": {
                    "qqNumber": {
                        "type": "string",
                        "description": "qq号码,比如545679452",
                    },
                },
                "required": ["qqNumber"],
            },
        },
    }
]


# 将tools传入到openAI的client


# 没有调用外部API功能时
def get_completion(model="gpt-3.5-turbo"):
    print("预先调用的提示词: \n", messages, "\n")
```

```
    responses = client.chat.completions.create(
        model=model,
        messages=messages,
        tools=tools,
    )
    return responses.choices[0].message


def get_completion2(messages, model="gpt-3.5-turbo"):
    print("真正的调用提示词: \n", messages, "\n")
    responses = client.chat.completions.create(
        model=model,
        messages=messages,
    )
    return responses.choices[0].message.content


# print(queryQQJixiong("545679452"))
firstMessage = get_completion()
print("第一次调用后的message:", firstMessage)
messages.append(firstMessage)

args = firstMessage.tool_calls[0].function.arguments
print("args 是 ", args)

args = json.loads(args)

print("qqNumber是", args["qqNumber"])

qqNumber = args["qqNumber"]

print("qqNumber的类型是 ", type(qqNumber))

messages.append(
    {
        "tool_call_id": firstMessage.tool_calls[0].id,
        "role": "tool",  # 表示是function call
        "name": firstMessage.tool_calls[0].function.name,
        "content": queryQQJixiong(args["qqNumber"]),
    }
)

response = get_completion2(messages)

print(response)
```

输出为(直接看最后的输出即可):

```
[{'role': 'system', 'content': '你是一个查询qq号吉凶的机器人,你要将用户输入的qq号码调用查询qq号吉凶的函数来告诉用户他qq号的吉凶情况'}, {'role': 'user', 'content': '我的qq号545679452999 的吉凶情况如何?'}]
```

```
第一次调用后的message: ChatCompletionMessage(content=None, refusal=None,
role='assistant', function_call=None, tool_calls=
[ChatCompletionMessageToolCall(id='call_UIuc0OQq23cXMmM8mZCCwswg',
function=Function(arguments='{"qqNumber":"545679452999"}',
name='queryQQJixiong'), type='function')])
args 是  {"qqNumber":"545679452999"}
qqNumber是 545679452999
qqNumber的类型是  <class 'str'>

 queryQQJixiong 的查询结果是:  {'errcode': 0, 'errmsg': '请求成功', 'notice': '本次
测算结果仅供娱乐使用，请勿用于封建迷信和违法用途。', 'data': {'desc': '最吉之数，还本归元，
能得繁荣，发达成功。', 'xiongji': '吉', 'desc1': '', 'score': '99', 'data': 0,
'shuli': 0}}
真正的调用提示词:
 [{'role': 'system', 'content': '你是一个查询qq号吉凶的机器人,你要将用户输入的qq号码调用
查询qq号吉凶的函数来告诉用户他qq号的吉凶情况'}, {'role': 'user', 'content': '我的qq号
545679452999 的吉凶情况如何?'}, ChatCompletionMessage(content=None, refusal=None,
role='assistant', function_call=None, tool_calls=
[ChatCompletionMessageToolCall(id='call_UIuc0OQq23cXMmM8mZCCwswg',
function=Function(arguments='{"qqNumber":"545679452999"}',
name='queryQQJixiong'), type='function')]), {'tool_call_id':
'call_UIuc0OQq23cXMmM8mZCCwswg', 'role': 'tool', 'name': 'queryQQJixiong',
'content': '{"errcode": 0, "errmsg": "请求成功", "notice": "本次测算结果仅供娱乐使
用，请勿用于封建迷信和违法用途。", "data": {"desc": "最吉之数，还本归元，能得繁荣，发达成
功。", "xiongji": "吉", "desc1": "", "score": "99", "data": 0, "shuli": 0}}'}]

根据查询结果显示，您的QQ号545679452999的吉凶情况非常好，属于“最吉之数，还本归元，能得繁荣，发
达成功”，评分为99分，吉祥指数为吉。祝贺您！您的QQ号码带来的吉祥气息会带来繁荣和成功。请注意，以
上结果仅供娱乐使用，请勿用于封建迷信和违法用途。
```

我们想要的结果是:

```
根据查询结果显示，您的QQ号545679452999的吉凶情况非常好，属于“最吉之数，还本归元，能得繁荣，发
达成功”，评分为99分，吉祥指数为吉。祝贺您！您的QQ号码带来的吉祥气息会带来繁荣和成功。请注意，以
上结果仅供娱乐使用，请勿用于封建迷信和违法用途。
```

帮宝宝取名

看了一眼，缘份居的api还支持 在给定姓氏和性别的前提下帮我们给宝宝取名。

下面是完整代码:

```python
import json
import os
import requests

from openai import OpenAI
```

```
api_key = "你们自己的openai token"
base_url = "https://openai.zhixueyouke.cn/v1/"


client = OpenAI(api_key=api_key, base_url=base_url)

# 再来试试AI起名


aiQimingUrl = (
    "https://api.yuanfenju.com/index.php/v1/Dafen/qiming?api_key={api_key}&sex=
{sex}&surname={surname}&page=1&limit=10"
)


def aiQimingFunc(sex, surname):

    print('\n aiQiming 参数是: ','\n 你输入的性别是: ', sex, '\n 你输入的姓氏是: ',
surname)

    if (sex == '男'):
        sex = '0'
    elif (sex == '女'):
        sex = '1'
    else:
        sex = '2'

    url = aiQimingUrl.format(
        api_key="RKoilS8orozOxFhBwLiJKUg86", sex=0, surname=surname)
    response = requests.get(url).json()
    print('\n aiQiming 的查询结果是: ', response)
    return json.dumps(response, ensure_ascii=False)


# 然后，准备message和tools
messages = []  # 提示词

messages.append(
    {
        "role": "system",
        "content": "你是一个取名字的机器人,用户会输入他宝宝的性别，姓氏，名字， 你的任务是
根据他宝宝的信息来给出他宝宝的名字,越多越好。",
    },
)
messages.append(
    {
        "role": "user",
        "content": f"""宝宝的性别是男，姓周，帮我取宝宝的名字。""",
    }
)
```

```
# 准备FunctionCalling的工具链
tools = [
    {
        "type": "function",
        "function": {
            "name": "aiQiming",
            "description": "取名字",
            "parameters": {
                "type": "object",
                "properties": {
                    "sex": {
                        "type": "string",
                        "description": "性别，男或者女，或者不限性别",
                    },
                    "surname": {
                        "type": "string",
                        "description": "姓氏，比如张，李",
                    },
                },
                "required": ["sex", 'surname'],
            },
        }
    }
]


# 将tools传入到openAI的client
# 第一次调用
def get_completion(model="gpt-3.5-turbo"):
    print('预先调用的提示词: \n', messages, '\n')
    responses = client.chat.completions.create(
        model=model,
        messages=messages,
        tools=tools,
    )
    return responses.choices[0].message


def get_completion2(messages, model="gpt-3.5-turbo"):
    print('真正的调用提示词: \n', messages, '\n')
    responses = client.chat.completions.create(
        model=model,
        messages=messages,
    )
    return responses.choices[0].message.content


firstMessage = get_completion()
print('第一次调用后的message:', firstMessage)
messages.append(firstMessage)

args = firstMessage.tool_calls[0].function.arguments
```

```
print('args 是 ', args)

args = json.loads(args)

sex = args['sex']
surname = args['surname']

print('sex是', sex, '\nsurname是', surname)

messages.append(
    {
        "tool_call_id": firstMessage.tool_calls[0].id,
        "role": "tool",  # 表示是function call
        "name": firstMessage.tool_calls[0].function.name,
        "content": aiQimingFunc(sex, surname),
    }
)

response = get_completion2(messages)

print(response)
```

完整输出为：

```
预先调用的提示词：
 [{'role': 'system', 'content': '你是一个取名字的机器人,用户会输入他宝宝的性别，姓氏，名字， 你的任务是根据他宝宝的信息来给出他宝宝的名字,越多越好。'}, {'role': 'user', 'content': '宝宝的性别是男，姓周，帮我取宝宝的名字。'}]

第一次调用后的message: ChatCompletionMessage(content=None, refusal=None, role='assistant', function_call=None, tool_calls=[ChatCompletionMessageToolCall(id='call_no5TWlWSMKSIRxhgbeqKiAyl', function=Function(arguments='{"sex":"男","surname":"周"}', name='aiQiming'), type='function')])
args 是  {"sex":"男","surname":"周"}
sex是 男
surname是 周

 aiQiming 参数是：
 你输入的性别是：  男
 你输入的姓氏是：  周

 aiQiming 的查询结果是：  {'errcode': 0, 'errmsg': '请求成功', 'notice': '本次测算结果仅供娱乐使用，请勿用于封建迷信和违法用途。', 'data': {'list': ['周晋毅', '周哲德', '周书辉', '周川云', '周熙胜', '周益逸', '周川翔', '周莫闲', '周峻磊', '周鼎棋'], 'pages': 405, 'limit': 10, 'total': 4050}}
真正的调用提示词：
 [{'role': 'system', 'content': '你是一个取名字的机器人,用户会输入他宝宝的性别，姓氏，名字， 你的任务是根据他宝宝的信息来给出他宝宝的名字,越多越好。'}, {'role': 'user', 'content': '宝宝的性别是男，姓周，帮我取宝宝的名字。'}, ChatCompletionMessage(content=None, refusal=None, role='assistant',
```

```
function_call=None, tool_calls=
[ChatCompletionMessageToolCall(id='call_no5TWlWSMKSIRxhgbeqKiAyl',
function=Function(arguments='{"sex":"男","surname":"周"}', name='aiQiming'),
type='function')]), {'tool_call_id': 'call_no5TWlWSMKSIRxhgbeqKiAyl', 'role':
'tool', 'name': 'aiQiming', 'content': '{"errcode": 0, "errmsg": "请求成功",
"notice": "本次测算结果仅供娱乐使用，请勿用于封建迷信和违法用途。", "data": {"list": ["周
晋毅", "周哲德", "周书辉", "周川云", "周熙胜", "周益逸", "周川翔", "周莫闲", "周峻磊",
"周鼎棋"], "pages": 405, "limit": 10, "total": 4050}}'}]

根据您提供的信息，这里给出十个适合男宝宝姓周的名字：周晋毅、周哲德、周书辉、周川云、周熙胜、周益逸、周川翔、周莫闲、周峻磊、周鼎棋。希望能给您一些灵感！
```

筛选掉无关内容之后，我们要的输出是：

```
根据您提供的信息，这里给出十个适合男宝宝姓周的名字：周晋毅、周哲德、周书辉、周川云、周熙胜、周益逸、周川翔、周莫闲、周峻磊、周鼎棋。希望能给您一些灵感！
```

一点感想

虽然直接调用 缘份居的openapi也能实现这些内容，但是 openai给我最大的好感就是，它可以自己根据 缘份居openapi返回的json串，从中提取出关键内容，并组装成自然语言的回答，给我们免去了手动转换json的麻烦。