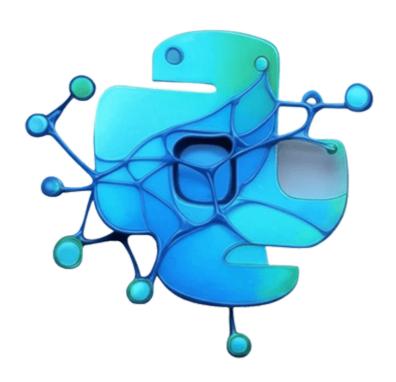
# Apex Agent 用户手册

#### Apex Agent 用户手册

- 一、基本介绍
  - 1.1 如何给 Apex Agent 下达清晰的任务指令?
  - 1.2 Apex Agent 能做什么,不能做什么? (它的"工具箱")
  - 1.3 反思机制-RAG
  - 1.4 记忆管理
- 二、代理下载安装与参数配置手册
- 2.1 下载和安装
- 2.2 初始参数配置
- 2.3 使用方法
- 三、附录--配置文件参数解析

### 一、基本介绍

欢迎使用 Apex Agent! 这款智能助手旨在帮助您自动化和完成各种技术任务。本手册将引导您了解如何与 Apex Agent 有效沟通,让它更好地为您服务。



Apex Agent 如何工作?

不同于聊天AI,这是一款AI智能体,顾名思义不仅仅可以用于聊天对话,更重要的是它可以帮您去操作和执行,完成您的任务。您可以将 Apex Agent 想象成一位非常聪明的助手。您告诉它您想做什么(您的任务),它会思考、规划,并利用各种工具(包括编写和运行代码、网络搜索、处理文件、理解图片)来一步步完成任务。它会记录工作过程,并在遇到困难或需要您确认时与您沟通。

您的角色:清晰的指令+适时的协助=成功的任务!

## 1.1 如何给 Apex Agent 下达清晰的任务指令?

清晰的指令是 Apex Agent 高效工作的第一步。请尝试遵循以下原则:

- 1. 明确您最终想要什么 (Specify Your Goal):
  - 示例: "我需要你分析一个名为 sales\_data.csv 的表格文件,找出销售额 最高的月份,并生成一个包含结果和分析图表的Word文档报告。"
  - 避免: "处理一下数据。" (太模糊, Agent 不知道具体要做什么)
- 2. 提供必要的信息和文件 (Provide Necessary Information):
  - 文件路径: 如果任务依赖数据等文件,请提供正确的文件路径或下载 网址URL。
    - 方法1:直接在任务描述中写出路径,可以使用快捷方式(输入 @即可显示出当前工作项目路径下的所有文件内容),例如: "分析文件 D:\我的项目\数据\data.csv"或"分析文件 当前项目下的data.csv"。
    - 方法2: 给出数据下载URL网址,Agent可以通过git等命令进行克 隆下载(需要电脑支持git工具或其他依赖工具)
- 3. 长文本截断 (重要):Apex Agent (以及其背后的大型语言模型) 在一次交互中能处理的信息量是有限的(这被称为"上下文Token限制")。直接让 Agent 处理非常大的数据文件或海量文件列表,可能会超出这个限制,导致 Agent 出错甚至无法正常响应。当您处理的数据文件较大时,Agent无法读取会被截断,它会采取可控的读取策略,如读取前5行等操作来进行受限读取,了解数据的基本结构,并通过脚本进行统计量计算和可视化来了解数据的基本统计信息。
- **4. 特殊要求:** 如果有特定的处理方法、计算公式(可以用  $E = mc^2$  这样的简单格式,或者更专业的 LaTeX 格式如 \sum\_{i=1}^{n} x\_i) ,请一并告知。**Agent 对常见的公式格式(如 LaTeX**)有很好的理解能力。

# 1.2 Apex Agent 能做什么,不能做什么?(它的"工具箱")

了解 Apex Agent 的能力边界,有助于您更好地利用它:

- 它能做到的:
  - 理解您的任务: 尽力理解您的自然语言描述。
  - 规划步骤: 将复杂任务分解成小步骤。
  - 编写和修改代码: 这是它的核心能力之一。
  - 执行代码: 在您指定的环境中运行Python/C/C++等代码。
  - 执行CMD指令: 可以用于查看指定路径内容、复制、删除文档、执行 C/C++等指令
  - 修复代码: 根据代码的执行结果决定是否需要修改代码。
  - 处理文件: 创建、读取、写入、列出文件和文件夹。
  - 识别图片内容: 分析图片中的物体、文字、图表趋势等。
  - **联网查询信息:** 当需要通用知识或技术资料时,它可以上网搜索(通过内部的Googel搜索和 Grok-3的DeeperSearch 工具)。
  - 与外部LLM进行交互: 当持续遇到无法解决问题时,会通过联网搜索或调用外部LLM的API进行问题咨询,与外部"专家"对话交流,避免因信息闭塞造成的阻塞。
  - 生成报告: 将任务过程、代码、结果整理成详细的报告(对于word或pdf等二进制与文本混杂的格式文件,需要您的环境中提供必要的第三方库)。
  - 与您沟通: 在遇到困难或需要您决策时,会请求您的指示。
- 它可能做不好的或不能做的 (需要您注意的):
  - 理解极其模糊或矛盾的指令: 它会尽力,但清晰的指令效果最好。
  - **处理超出其知识范围的专业领域问题:** 除非您提供足够的上下文和指导。
  - 无限制地处理超大规模数据: 尽管Agent有长期记忆管理的能力,但是 在多轮摘要压缩后,Agent不可避免出现遗忘、混乱等情况。
  - 记住非常久远或不相关的对话内容: 虽然已经配备了各种维护"记忆" 的方法(包括保留对话记忆缓冲区、实时维护工作状态文档、上下文摘要 压缩精炼等待),但是llm本质仍然是无状态、无记忆的,在多次摘要压 缩总结后,它的输出质量必然会逐渐降低。
  - 保证图像识别100%准确: 虽然它的图像识别能力强大,但对于极其复杂、模糊或低质量的图像,识别结果可能存在偏差。您的补充描述和对结果的验证依然重要。
- 功能模式:
  - 图 任务回溯与继续模式 (Task Traceback and Continuation Mode):
    - 作用: 如果之前的任务中断(比如您手动中止,或者程序关闭),在确保工作目录设置正确的前提下,勾选此项,Agent可以尝试从上次中断的地方继续执行。您只需在任务输入框中给出简单指示,如"继续执行任务",然后发送即可。注意,在单次

任务执行期间,由于Agent需要停下来和用户交互,因此在等待用户回复期间,系统会自动勾选上该选项,请不要关闭该选项,否则将会重置记忆,从零开始。

#### • 如何使用:

- i. 关键: 确保您在"工作目录"中设置的是上次任务使用的同一个文件夹,并且该文件夹内的 HistoryChat 子文件夹(Agent自动创建的)及其内容完好无损。
- ii. 在任务输入框中,输入简单的继续指令(如"继续")或基于上次进度的新的引导指令。您可以打开工作目录下的 DetailedLogReport.html 文件,它记录了详细的上下文日志,能帮助您快速了解上次任务的进展情况。

#### • 注意事项:

• Agent 会加载上次保存的工作记录。如果 HistoryChat 文件夹内容被删除或严重修改,回溯功能将无法正常工作。

#### • **十**临时添加全局规则(+ Rule Addition):

- 作用:用于在当前任务窗口中临时添加一些强制性的规则约束 (如让它每一步的输出和思考过程输出都务必为中文、代码中注 释都要用中文注释、图表绘制注意事项等等)
- 注意:该规则理论上可以在执行期间实时添加、更新、删除,但是我们建议您在启动任务前就添加,尽量避免后续任务执行期间添加和改动。一方面,LLM本质上是根据前面的序列预测后面的词元序列,因此,例如当前面都是英文的输出,后面即使加了中文输出的规则约束,也难免出现LLM继续输出英文的情况;另一方面,LLM都支持KV缓存命中,规则的添加是添加在最开始的System Prompt当中,如果在任务期间频发改动规则,则会造成缓存失效,增加模型的推理成本、降低推理效率。

#### • 知 角色卡模式 (Role Card):

- 角色卡本质上是在基础的System Prompt后面添加新的规则指令,但是该指令并非一个简单的角色定义描述,而是根据Apex Agent的工具集量身制定的特殊Work Flow,可以让Agent在一些垂直领域表现的更好。我们为您提供了常用的一些角色卡:
  - **GeneralRole**: 默认角色,通用的Work Flow,一般情况下使用该模式即可;
  - **VisExpert**: 可视化专家,专注于学术科研图表的绘制;
  - **SeniorDataAnalyst**:数据分析师,有一套成熟数据分析师的Work Flow;
  - PaperResearcher: 论文研究员,有着专门设计的论 文阅读个总结Work Flow;

- DeepDiveResearcher: 深度调研员,基于Google搜索+闭环Work Flow;
- **CodeRefactoring**: 代码重构师,专注于已有代码的重构任务。(注意,Apex Agent对Python有着更好的支持,可以系统可以自动构建各个代码间的依赖关系图)
- 自定义角色卡: 支持用户自定义角色卡,设计专门的 Work Flow (请参考下面的工具箱,以便更好的设计 Work Flow)
- 它的工具箱(了解它的能力边界):
  - (1) 更新工作状态文档 (update\_work\_status\_document) \*\*基本功能\*\*:

此工具用于更新一个名为 `apex\_agent\_status.md` 的工作状态文档(类似 to-do list)。当任务取得重要进展时,例如完成一个子任务、代码成功运行或计划发生变更,系统会调用此工具来记录最新的状态。这有助于保持对任务进度的清晰追踪。系统会强制性提示Agent维护该文档,因此无需担心Agent会忘记维护。

(2) 读取工作状态文档 (read\_work\_status\_document)

\*\*基本功能\*\*:

此工具用于读取 `apex\_agent\_status.md` 工作状态文档的内容。当需要回顾 任务的当前进展、确认下一步计划或在遇到问题后重新定位时,系统会使用此工具来 获取最新的状态信息。

(3) 咨询外部AI专家 (delegate\_to\_claude & delegate\_to\_grok) \*\*基本功能\*\*:

当遇到棘手的编程难题,尤其是在前端、UI设计和后端开发领域,且经过多次尝试仍无法解决时,可以使用此工具向 Claude或Grok 专家(这是一个名字代号,底层 LLM接谁都可以,随便接入,只要是openai格式兼容即可)进行咨询。它相当于一个最后的求助手段,用于解决特定领域的深度技术问题。

(4) 深度研究与分析 (ask\_grok\_for\_info)

\*\*基本功能\*\*:

主要调用如grok-3 deepersearch模式用来深度搜索,此工具用于对复杂主题进行跨平台的深度研究和分析。它不仅仅是简单的网页搜索,而是会主动挖掘学术数据库、技术社区、新闻档案等多个渠道的信息,并最终生成一份包含深入洞察和分析的结构化研究报告。适用于需要对某个技术趋势、解决方案进行系统性、根本性研究的场景。

(5) 谷歌搜索 (google\_search\_func)

\*\*基本功能\*\*:

此工具利用谷歌搜索引擎来查找、整合和分析实时的网络信息。它可以获取最新资讯、查询特定网站内容、研究不熟悉的领域、查找技术文档和代码示例,或进行事实核查。

#### (6) 执行 Python 代码 (execute\_python)

\*\*基本功能\*\*:

此工具用于执行 Python 脚本文件。它可以运行指定的 `.py` 文件,并支持向 脚本传递命令行参数或提供预设的输入内容。适用于需要通过运行代码来验证逻辑、处理数据或完成任务的场景。

# (7) 执行 Windows 命令 (execute\_windows\_cmd\_command) \*\*基本功能\*\*:

此工具用于在 Windows 命令提示符 (cmd.exe) 环境中执行命令。它可以运行文件操作(如 `dir`, `copy`)、编译代码(C/C++, java等代码需要配置好环境并告诉Agent如何在命令行使用)、执行程序等。需要注意的是,每次调用都是一个独立的环境,因此多个连续步骤需要使用 `&&` 连接成单条命令。此工具不支持需要实时交互的程序。(Agent可以执行一些在白名单里面的命令,对于不在白名带内的,Agent会先请示用户,需要用户批准才可使用,保证安全)

#### (8) 创建文件 (create\_file)

\*\*基本功能\*\*:

此工具用于创建一个新的文件,并向其中写入指定内容。适用于需要保存新生成的代码、报告、数据或其他文本信息的场景。

#### (9) 写入文件 (write\_to\_file)

\*\*基本功能\*\*:

此工具用于向一个已存在的文件中写入或追加内容。您可以选择覆盖文件原有内容,或是在文件末尾添加新内容。适用于更新配置文件、记录日志或修改已有代码文件的场景。

#### (10) 读取文件(安全模式) (read\_file)

#### \*\*基本功能\*\*:

此工具用于安全地读取指定文件的内容。为了防止处理超大文件时出现问题,如果文件内容过大,系统会自动截断。这是默认的文件读取工具,适用于读取代码、配置等大小可控的文件。在读取前,系统会优先检查历史记录中是否已有文件内容,避免重复读取。

#### (11) 读取完整文件内容 (read\_all\_content)

#### \*\*基本功能\*\*:

此工具用于读取文件的全部内容,不会进行截断。当您确认需要获取一个文件的完整信息,且 `读取文件(安全模式)` 工具无法满足需求时,可以使用此工具。请注意,调用此工具前系统会自动征求您的同意,以确保操作的安全性。

#### (12) Office 文档查看器 (office\_doc\_viewer)

\*\*基本功能\*\*:

此工具专门用于读取 Office 文档,支持 `.doc`, `.docx`, `.ppt`, `.pptx` 和 `.pdf` 格式。它提供两种模式:"摘要模式"可以快速获取文档的核心摘要或大纲,而"详细模式"则可以读取完整的文本内容。

#### (13) 图像内容识别 (recognize\_image\_content)

\*\*基本功能\*\*:

此工具用于理解和分析图片内容(支持 PNG, JPG 等格式)。您可以向它提问,让它从图表中提取数据趋势、识别图片中的关键特征、或读取截图中的文字。它尤其适用于验证生成的数据可视化图表(如检查图表是否正确)或从图片中获取定性分析结论。

#### (14) 请求用户指令 (consult\_user\_for\_instruction)

\*\*基本功能\*\*:

当系统遇到无法独立解决的问题(如缺少关键信息、需要您做出重要决策)或在完成任务提交最终结果之前,会使用此工具向您请求明确的指令、信息或批准。这确保了任务的执行方向符合您的预期。

#### (15) 提交最终答案 (final\_answer)

\*\*基本功能\*\*:

当最初设定的任务被完全解决,并且已经通过"请求用户指令"工具获得了您的批准后,系统会调用此工具来提交最终的、完整的任务成果。(结束出口,避免因任务结束导致Agent不知道选择什么工具)

### 1.3 反思机制-RAG

Apex Agent配置了反思机制,Agent在执行任务中遇到持续性错误,在解决后会主动总结经验(模式:问题:解决方案),当Agent在当前或另一个窗口中执行任务时,遇到同类型的错误后,会触发系统的RAG检索,根据问题进行余弦相似度匹配,将检索到的解决方案反馈给Agent,以避免重复性错误下次再次出现。

### 1.4 记忆管理

Apex Agent底层为Gemini大模型,本身支持超长上下文,但这在一个复杂任务中可能仍然 "不够看",因此,设计上下文记忆管理功能是必要的。Apex Agent采用"保留初始任务指令+引用式低损压缩+动态保留记忆缓冲区+工作状态文档"模式实现超长回合的任务执行。其 背后原理是:

#### • 保留初始任务指令:

初始任务指令是全局的目标,目标不可遗忘和丢失,确保全局至高地位

#### • 引用式低损压缩:

该机制的核心思想是**将"状态"与"叙事"分离**。我们不直接压缩包含大量原始数据(如文件内容)的上下文,因为这会导致信息丢失("有损压缩")。取而代之,我们采用以下流程:

当对话历史超出预设长度时,系统将自动激活。它并非粗暴地让大模型直接总结,而是通过一个精巧的"保护-压缩-恢复"流程。具体而言,在发给LLM执行摘要压缩前,系统会对待压缩的内容进行"预处理",系统会自动识别并保护任务中所有关键文件(如代码、文档),用一个不可压缩的占位符标识《Non-compressible placeholder index》替换其原文内容。将处理后的上下文(仅含对话逻辑和文件占位符)交由大模型进行摘要提炼,聚焦于"思考→行动→结果"的链式摘要提炼。LLM 返回精炼后的摘要。系统随后会根据全局产物映射表,将摘要中的所有《Non-compressible placeholder index》占位符,精准地替换回它们对应的文件路径引用。通过此机制,我们实现了上下文的大幅压缩,同时确保了所有核心产物的存在性(Existence)和可访问性(Accessibility)被完整保留。Agent 的记忆中不再是庞杂的文件原文,而是轻量级的路径索引,确保了在后续任务中,它可以随时按需重新加载和利用这些关键资产,避免了"长期遗忘"问题。

#### • 动态保留记忆缓冲区:

用户可以在参数配置中设计在摘要压缩时保留最近的k条,系统会自动动态调整缓冲区长度(如当总对话历史不足以保留 k 条时,系统会自动计算并保留当前可用的最大条数),这种动态调整机制确保了程序的鲁棒性,无论对话处于何种阶段都能平稳运行。

#### • 工作状态文档(apex\_agent\_status.md):

Agent 会维护一个外部的、持久化的工作状态文档,类似于一个任务的to-do list,我们设计了一个"暂存注入"机制,在每个任务回合开始时,系统会将该状态文档的内容注入到上下文的最后端,实现在每回合动态给Agent复述一遍工作状态,在LLM 生成响应之后,该注入的状态文档会立即从上下文中移除,不会被计入下一轮的历史记录。通过这样的机制,为 Agent 提供持续的宏观任务指引,解决了长期记忆与上下文长度限制之间的核心矛盾。

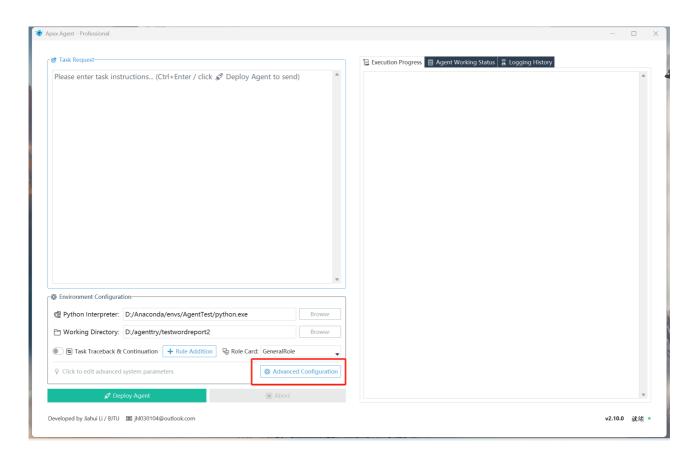
# 二、代理下载安装与参数配置手册

### 2.1 下载和安装

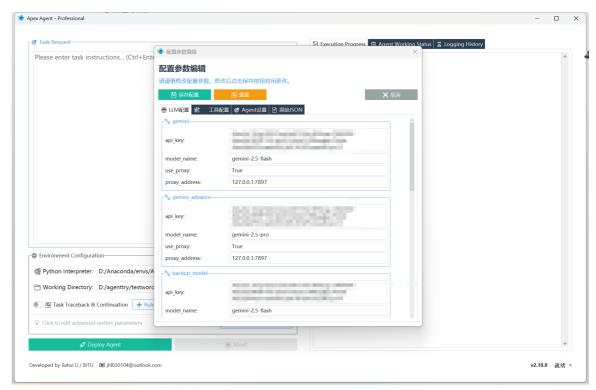
运行ApexAgent.msi 安装包,快速下载。

### 2.2 初始参数配置

首次使用需要配置智能体参数,点击主界面的Advanced Configuration按钮进行参数初始配置:



• **LLM配置:**包括Gemini LLM API、第三方中转LLM API(可选,不填可以正常使用但代理无法访问到外部LLM寻求帮助);

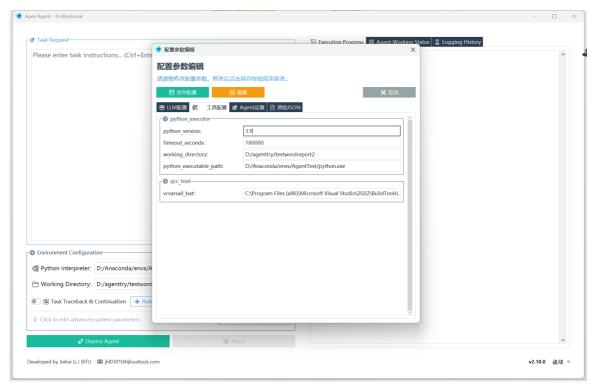


配置Gemini API时,由于系统是轮询方式,因此可以填多个API-KEY,每一行填一个API-KEY,不要添加分号或逗号隔开,示例填写如下,注意不要留有空白行,否则会被当成空API-KEY去轮询,会导致报错:

```
MYAPIKEY1
MYAPIKEY1
MYAPIKEY1
.....
```

此外,由于Gemini不支持部分地区用户使用,因此国内需要能够科学上网,并打开**Tun虚拟卡模式(极其重要)**,不要使用系统代理模式,否则部分功能将会受限导致应用卡住。无需科学上网即可使用地区的用户,可以直接在use\_proxy位置填false即可。

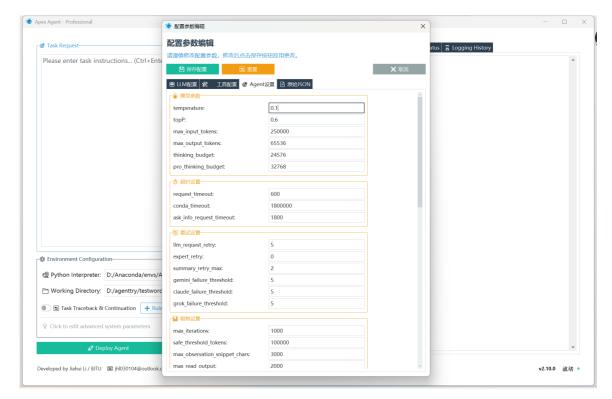
• 工具配置:需要配置给代理使用的python解释器路径和c/c++生成工具里面的vcvarsall.bat路径,python解释器也可以在主界面直接切换和配置,其余可以暂时默认;



工具配置中,python\_version参数已经弃用,默认即可,无需改动(作者太懒不想动代码);

timeout\_seconds代表执行python代码和CMD命令最长的超时时间,一般按照默认即可(180000s=50h,一般程序都能执行完毕)

• **Agent设置**:这里面的参数为调节智能体参数,如温度、topP等,一般情况下都选择默认即可(作者已经调好的,一般情况下都可以正常使用的参数),由于参数太多,如需了解其余参数的详情,欢迎咨询作者(jhl030104@outlook.com)。



• 原始JSON: 高级用户(即能看明白的)可以直接编辑JSON配置。请确保JSON格式正确即可。(本质上前面3栏只是对原始JSON进行了分组分栏,修改哪一个都可以)

修改完成后,点击保存配置即可,如果过程中想要退回未修改前的,点击重置按钮即可(已经保存过的无法重置返回,因为已经写入配置文件当中了)

# 2.3 使用方法

下面截图是有关主界面的详细介绍,请参考。



## 三、附录——配置文件参数解析

注意,当前版本逻辑已经修改,多数参数已经弃用,下面图示为旧版参数介绍(作者太懒,不想更新),DevelopmentstrategistModel和DevelopmentstrategistModel\_back\_up参数已经弃用。此外所有第三方openai格式兼容的LLM都会作为备用的"外部专家",不再为主力选手。一般情况下,只需要填写Gemini的官方API-KEY即可维持Agent的正常运行,虽然会缺少一些功能(访问外部LLM、Gork-3-DeeperSearch等功能),但不影响程序正常运行,仍能实现90%以上的功能。

```
1
         "llms": {
              "gemini": {
 3
                  "api_key": [
 4
                      "your_gemini_key1",
                      "your_gemini_key2",
 6
                      "your_gemini_keyGemini官方API (有多少填多少,越多越好,背后做轮询,增加额度且不
                      "your_gemini_key等易额度超限)
 8
                      "your_gemini_key6",
"your_gemini_key7"
10
11
12
                   "model_name": "gemini-2.0-flash-thinking-exp-01-21",
13
                  "use_proxy": true,
"proxy_address": "127.0.0.1:7897"
14
15
16
              "gemini_advance": {
17
                  "api_key":
18
19
                      "your_gemini_key1",
                      "your_gemini_key2",
20
                      "your_gemini_key3",
21
                      "your_gemini_key4",
23
                      "your_gemini_key5",
                      "your_gemini_key6",
24
                      "your_gemini_key7"
25
26
                   'model_name": "gemini-2.5-flash-preview-05-20",
27
                  "use_proxy": true,
"proxy_address": "127.0.0.1:7897"
28
29
30
31
              "backup_model": {
32
                  "api_key": [
                      "your_gemini_key1",
33
                      "your_gemini_key2",
34
                                                                                                        VPN端口配置
35
                      "your_gemini_key3",
                      "your_gemini_key4",
36
                      "your_gemini_key5",
37
38
                      "your_gemini_key6",
39
                      "your_gemini_key7"
40
                   "model_name": "gemini-2.5-flash-preview-05-20"
41
                  "use_proxy": true,
"proxy_address": "127.0.0.1:7897"
42
43
44
```

```
"llms": {
    "gemini": {
        "api_key": [
            "your_gemini_key1",
            "your_gemini_key2",
            "your_gemini_key3",
            "your_gemini_key4",
            "your_gemini_key5",
            "your_gemini_key6",
            "your_gemini_key7"
                                                           备用大脑 (用于连接中断或超时等情况回退)
        "model_name": "gemini-2.0-flash-thinking-exp-01-21",
        "use_proxy": true,
"proxy_address": "127.0.0.1:7897"
     "gemini_advance": {
        "api_key": [
    "your_gemini_key1",
             "your_gemini_key2",
            "your_gemini_key3",
            "your_gemini_key4",
            "your_gemini_key5",
            "your_gemini_key6",
             "your_gemini_key7"
        "model_name": "gemini-2.5-flash-preview-05-20",
"use_proxy": true,
"proxy_address": "127.0.0.1:7897"
                                                                主脑模型
    "backup_model": {
        "api_key": [
            "your_gemini_key1",
            "your_gemini_key2",
             "your_gemini_key3",
            "your_gemini_key4",
            "your_gemini_key5",
            "your_gemini_key6",
            "your_gemini_key7"
                                                             所有编程模型的备用模型
       "model_name": "gemini-2.5-flash-preview-05-20",
"use_proxy": true,
"proxy_address": "127.0.0.1:7897"
```

```
"DevelopmentStrategistModel":
45
46
                  "api_key": "your_key(sk-)",
                 "api_url": "your_url/v1/chat/completions",
47
                                                                            openai格式兼容的API-
                  "model_name": "gpt-4.1-2025-04-14"
48
                                                                            key (sk开头) 和url (末
49
             "DevelopmentStrategistModel_back_up": {
50
                                                                            尾一般是v1/chat/
                 "api_key": "your_key(sk-)",
51
                                                                            completions即聊天补齐
                 "api_url": "your_url/v1/chat/completions",
52
                 "model_name": "claude-3-7-sonnet-20250219"
53
                                                                            模式)
54
55
              "SummaryCompressionModel": {
                 "api_key": "your_key(sk-)",
"api_url": "your_url/v1/chat/completions",
56
57
                  "model_name": "gpt-4.1-2025-04-14"
58
59
              "SummaryCompressionModel_back_up": {
60
                 "api_key": "your_key(sk-)",
61
                 "api_url": "your_url/v1/chat/completions"
62
                 "model_name": "grok-3
64
             },
              "claude_sonnet": {
65
                  "api_key": "your_key(sk-)",
66
                 "api_url": "your_url/v1/chat/completions",
67
                  "model name": "deepseek-r1-0528"
68
69
             },
              "claude_sonnet_back_up": {
70
                 "api_key": "your_key(sk-)",
71
                 "api_url": "your_url/v1/chat/completions",
72
                 "model_name": "claude-3-7-sonnet-20250219"
73
             },
              "grok": {
75
                 "api_key": "your_key(sk-)",
76
                 "api_url": "your_url/v1/chat/completions",
77
78
                 "model_name": "grok-3"
79
             },
80
              "grok back up": {
                 "api_key": "your_key(sk-)",
81
                 "api_url": "your_url/v1/chat/completions",
82
                 "model_name": "deepseek-v3-0324"
83
84
             },
85
              "grok search": {
86
                 "api_key": "your_key(sk-)",
                 "api_url": "your_url/v1/chat/completions",
87
                  "model_name": "grok-3-deepersearch
88
89
             },
```

```
45
            "DevelopmentStrategistModel": {
                "api_key": "your_key(sk-)",
46
                                                           主深度规划模型
                "api_url": "your_url/v1/chat/completions",
47
48
                "model_name": "gpt-4.1-2025-04-14"
49
            },
50
            "DevelopmentStrategistModel back up": {
                "api_key": "your_key(sk-)",
51
                "api_url": "your_url/v1/chat/completions";
52
                                                           深度规划模型备用回退模型
                "model name": "claude-3-7-sonnet-20250219"
53
54
            },
55
            "SummaryCompressionModel": {
                "api_key": "your_key(sk-)",
56
                                                          摘要压缩总结模型
                "api_url": "your_url/v1/chat/completions",
57
                "model_name": "gpt-4.1-2025-04-14"
58
            },
59
            "SummaryCompressionModel_back_up": {
60
                "api_key": "your_key(sk-)",
61
                                                          摘要压缩总结备用回退模型
                "api_url": "your_url/v1/chat/completions",
62
                "model_name": "grok-3"
            },
64
            "claude_sonnet": {
65
                "api_key": "your_key(sk-)",
66
                                                             编程首席模型
                "api_url": "your_url/v1/chat/completions",
67
                "model name": "deepseek-r1-0528"
68
69
            },
             "claude_sonnet_back_up": {
70
                "api_key": "your_key(sk-)",
71
                                                               编程备用回退模型
72
                "api_url": "your_url/v1/chat/completions",
73
                "model_name": "claude-3-7-sonnet-20250219"
74
            },
             "grok": {
75
76
                "api_key": "your_key(sk-)",
                "api_url": "your_url/v1/chat/completions",
77
                                                            代码修复/牛成模型
78
                "model name": "grok-3"
79
            },
             "grok_back_up": {
80
                "api_key": "your_key(sk-)",
81
                                                          代码修复/生成备用回退模型
                "api_url": "your_url/v1/chat/completions",
82
83
                "model name": "deepseek-v3-0324"
84
            },
             "grok_search": {
85
                "api_key": "your_key(sk-)",
86
                                                          联网搜索、深度搜索主模型
                "api_url": "your_url/v1/chat/completions",
87
                "model_name": "grok-3-deepersearch"
88
89
            },
```

```
90
            "grok_search_back_up": {
               "api_key": "your_key(sk-)",
91
               "api_url": "your_url/v1/chat/completions",
                                                    联网搜索备用回退模型
 92
               "model_name": "grok-3"
93
 94
95
                                  该参数已弃用,这里只做占位
        "tools": {
97
            "python_executor": {
98
               "python_version":
                                         代码执行时间限制
              "timeout_seconds": 180000
99
               working_directory": "D:/agenttry/testnew",
100
               "python_executable_path": "D:/Anaconda/envs/DL/python.exe"
101
102
        》,项目路径和python解释器(可在UI界面内动态设置,不需要在参数配置文件里面修改)
103
```

```
"agent_settings": {
104
            "temperatur<u>e": a →</u> 主脑模型温度
105
            "topP": 0.6<u>, 丰脑模型topP参数</u>
106
107
            'max input tokens": 250000,
                                     ★主脑模型的最大输入和输出token限制
108
            "max_output_tokens": 250000,
            "thinking_budget": 24576,
109
                                       ▶主脑模型的思考预算 (2.5flash默认是8192,最大不超过24576)
            'analysis_max_output_tokens": 2500000 参数已弃用
110
                                                  → 代码脚本最大输出token限制(超出直接被截断)
             codelog_summary_max_output_tokens": 100000,
111
            "history_summary_max_output_tokens": 250000,
112
                                                             ▶ 历史摘要压缩模型的最大输出token限制
113
            "kb_days_to_keep": 90,<mark>知识库最长保留多少天的经验向量</mark>
114
             rag top k": 1.
            "rag_confidence_score": 0.8,
115
                                            RAG检索的topk和置信度阈值设置
116
            "rag_processed_confidence_score": 0.8,
            "base_sleep": 5,
                                             轮询Gemini api时, 切换休眠等待时间 (速率限制防护)
117
118
             'max_tokens":
                       250000
                                            」主脑模型每分钟最大token限制
            "openai_max_tokens": "auto
119
                                           openai格式兼容的IIm的基本参数配置(最大token限制、温度、请求超时
            "openai_temperature": 0.1,
120
                                           时间、是否需要移除VPN代理(0是否1是是));对于最大token限制和
121
            "openai_request_timeout": "auto",
                                           请求时间超时参数,如果调用的中转或逆向,不确定或不支持的参数请填
            "open_ai_pop_vpn": 0,
122
            "ask_info_request_timeout": 1800,
                                           写'auto',则会默认选择提供商设置的默认参数
123
124
            "summary_output_base_sleep": 2,
             request_timeout": 150,
                                                   联网搜索、深度搜索的等待时间限制以及总结模型轮询的休
125
126
            "max_iterations": 1000,
                                                   眠等待时间(该参数已经弃用,可以设置可以不设置)
            "safe threshold tokens": 30000,
127
             conda_timeout": 1800000,
                                                    主脑模型的请求时间超时限制、任务推进最大轮次回合限制
128
            "gemini_failure_threshold": 5,
129
                                                      (若在指定轮次回合内未解决任务则会停止)、上下文
                                      参数已弃用
130
            "claude_failure_threshold": 5,
                                                     token安全阈值设置(同下面的
131
            "grok_failure_threshold": 5,
                                                   ▶ history_summary_threshold_tokens,二者参数设置需要
            "history_summary_threshold_tokens": 30000,
132
            "threshold_token_growth_factor": 1.2,
                                           参数已弃用
133
                                                           代码脚本执行输出内容token长度阈值限制(输出超
            "max_observation_snippet_chars": 3000,
"max_read_output": 2000,
134
                                                           出该阈值会被压缩总结或截断);以及agent读取除
135
                                                           py等核心文件外其余项目文件时,读取内容token长
136
            'history_keep_last_n_messages": 10,
                                                           度限制
137
            "summary retry max": 2,
            "summarization_chunk_failure_threshold": 多参数已弃用
138
                                                             ▶ 上下文摘要压缩时保留最近几条对话作为记忆缓
            "truncation amplitude": 2.
139
                                                              冲区; 以及摘要压缩的失败重试次数
140
            "post_truncation_percentage_threshold": 0.1
                                                                      过history_summary_threshold_tokens的 (默认) 百分
            "threshold_percentage" · 0 6
                                                          之10 (0.1) 时会被截断
141
            "llm_request_retry": 5
"avrant_retry": 2
Im模型和专家模型失败响应重试次数阈值
142
                                                           系统指令+初始指令+记忆缓冲区总token阈值百分比,超出
143
                                                           后会缩减记忆缓冲区直到满足不超过阈值百分比
             analysis_model": "gemini-2.0-flash",
144
            "summarization_model": "gemini-2.0-flash", 压缩总结模型的备用回退模型,默认Gemini-2.0-flash
145
            "token_count_model": "gemini-2.0-flash-lite"参数已弃用
146
            "UI_theme": "flatly",
147
                                                                   UI界面的主题、字体放大倍数、消息弹窗间
148
            "font_size_up": 1,
                                                                   隔时间 (s)
            "quote_interval": 300,
149
```