__DIRTY_RETAIN_MARK__

星辰 TokenHub 运营服务平台

用户使用手册

天翼云科技有限公司

2026 年 10 月

目录

产品介绍 6

1.1产品定义 6

1.2产品优势 7

1.3功能特性 8

1.4应用场景 10

1.5术语解释 12

1.6使用限制 14

计费说明 15

2.1 按需计费模式-按量付费 15

2.2 包周期计费模式-编程Token Plan 19

2.3 包周期计费模式-Tokens量包 23

2.4 欠费说明 30

2.5 账单查询 32

2.6 产品退订 37

快速入门 38

3.1 服务开通 38

3.2 快速入门 42

用户指南 46

4.1 编程Token Plan 46

4.1.1套餐概述 46

4.1.2快速开始 50

4.1.3编程Token Plan常见问题 53
4.2 Token Plan 63
4.2.1套餐介绍 63
4.2.2快速开始 66
4.2.3常见问题 72
4.3 智算广场 75
4.3.1模型广场 75
4.3.2MCP广场 76
4.4 体验中心 77
4.4.1 模型体验 77
4.4.2 MCP体验 79
4.5 模型推理 80
4.5.1 模型推理 80
4.5.2 可信Token专区 84
4.6 模型观测 89
4.6.1 调用监控 89
4.6.2 用量统计 94
4.7 系统管理 95
4.7.1 APP Key管理 95
4.7.2 模型授权 99
4.7.3 限额管理 101
最佳实践 108

5.1 第三方工具添加模型配置指南 108
5.1.1 获取模型调用API KEY、模型ID等信息 108
5.1.2 OpenClaw 109
5.1.3 ClaudeCode 121
5.1.4 OpenCode 124
5.1.5 Trae 127
5.1.6 Codex（Openai） 132
5.1.7 Cursor 136
5.1.8 Cline 140
5.1.9 Chatbox 147
5.1.10 Codebuddy 154
5.1.11 Hermes Agent 162
5.1.12 Codex桌面版 167
5.2 DeepSeek模型调用-快捷版 172
5.3 DeepSeek模型调用-专业版 213
5.4 基于息壤的企业级RAG知识库构建指南 270
5.4.1 概述 270
5.4.2 RAG系统架构概述 271
5.4.3 知识入库环节模型选型 272
5.4.4 知识问答环节模型选型 274
5.4.5 天翼云星辰TokenHub运营服务平台模型部署实践 278
5.4.6 RAG系统最佳实践建议 279

5.5 缓存命中 281
推理服务API 287
如何调用API 287
6.1.1 终端节点 287
6.1.2 接口构造 288
错误码（新版） 289
6.2.1 错误响应通用约定 290
6.2.2 公共返回字段说明 292
6.2.3 http状态码总览表 292
6.2.4 业务错误码对照表 293
错误码（旧版） 301
6.3.1 错误error结构 302
6.3.2 错误结果示例 303
6.3.3 错误码 303
API 305
6.4.1 chat 对话API 305
6.4.2 image 文本生图API 331
6.4.3 Embeddings文本向量化API 340
6.4.4 Reranker 重排序API 347
6.4.5 模型列表API 356
平台功能OpenAPI 363
平台功能API使用说明 363

常见问题 364
计费类 364
操作类 367
权限类 368
平台升级与用户割接说明 369
8.4.1 升级背景 369
8.4.2 常见问题 373
联系我们 375
相关协议 376
9.1 天翼云星辰TokenHub运营服务平台服务协议 376
9.2 天翼云星辰TokenHub运营服务平台用户信息处理规则 376
9.3 生成式人工智能服务备案信息参考 376
9.4 人工智能大模型备案指南 378

产品介绍

1.1产品定义

本文带您了解什么是星辰TokenHub运营服务平台。

星辰TokenHub运营服务平台是面向企业开发者的星辰TokenHub运营服务平台，提供功能全面、安全可靠且具有价格优势的模型调用服务。

【功能模块】

模型体验：搭建多场景模型展示与试用环境，开发者可在模型广场快速试用主流 AI 大模型；依托体验中心零成本完成模型基础推理能力验证测试，前期接入门槛低。

模型服务：输出标准化、高性能、高可用的模型推理 API 接口服务，支持业务系统便捷对接调用大模型能力。

监控与统计：具备模型调用全链路性能监控、用量数据统计分析能力，可实时采集各模型接口请求并发量、Token 吞吐量、接口响应时延等关键运行指标；支持用量趋势分析、数据排行查看，便于运维人员及时掌握服务运行状态，保障推理服务平稳、高效运转。

服务组管理：支持按需挑选所需模型资源，将多款模型打包归集为独立服务组，自动生成专属 AppKey 身份凭证，业务端凭该凭证即可发起合法模型调用。

限额管理：支持以单个服务组（AppKey）为管控粒度，配置不同时间周期下的调用额度上限，实现调用用量精细化管控，避免超额消耗。

权限管理：主账号可划定各子用户可调用的模型权限范围，针对不同子用户配置差异化调用权限，实现团队内模型调用权限的精细化、个性化管控。

1.2产品优势

星辰TokenHub运营服务平台为您提供优质的服务体验，本文带您了解平台产品优势。

模型服务丰富

多元模型生态：集成主流先进第三方大模型，支持文本、图像、音视频等多模态处理能力。

多种商品适配：多种商品模式适配业务规模及性能要求，满足不同的业务需求。

操作使用方便

标准化API：统一接口封装模型能力，开发者可直接调用API，实现分钟级集成。

调用价格实惠

云上部署：采用云服务模式，替代传统本地部署，无需购置专用硬件设备。

免维护负担：节省设备运维、人力成本，轻松应对业务扩展与模型更新。

按需付费：通过按需付费机制实现资源消耗与实际支出的精准匹配。

1.3功能特性

本文带您了解星辰 TokenHub 运营服务平台的功能特性。

API 接入

服务初期，开放四类 API 接入能力，包括：文本/代码生成模型、文本/代码补全模型、视频理解模型、视频生成模型，后续随技术迭代动态扩展。

Token 鉴权功能

用户开通 Token 资源后，登录息壤智云平台 APP Key 管理模块创建服务组，即可生成 APPkey，每个用户生成 APPkey 数量不限。

统一鉴权机制：基于 API 密钥（APPkey）实现调用身份认证与权限校验，所有 API 请求均需通过鉴权校验后方可执行，保障接口调用的合法性与安全性。

API 密钥管理：支持按照业务需求创建、禁用、删除多个 APP Key，并为不同密钥配置相应的调用权限，实现按业务线、按应用场景的维度权限隔离。

调用管理功能

限流控制：支持设置单接口/单模型的 TPM、RPM、QPS 性能上限，对超过阈值的请求自动拒绝，保障系统稳定性与资源合理分配。

多版本并存管理：支持同一模型的多个版本同时在线提供服务，用户可按需切换，保障业务升级的平滑过渡与兼容性。

运营管理功能

调用监控与告警：提供实时 QPS、成功率、延迟、计量消耗等实时统计数据，保障业务运行状态可观测。

调用明细查询：支持按时间维度、按模型维度、按密钥维度查询调用明细记录，便于业务对账与运营分析。

计量计费功能

Token 消耗计量：每次 API 调用返回实际消耗的 Token 数量，支持输入、输出与缓存命中分别统计。

成本控制：支持设置单接口/单模型的消费上限，对超过阈值自动拒绝请求或触发告警，帮助用户管控整体调用成本。

安全管理功能

数据安全：在模型调用请求中通过加密传输、权限隔离等方式保障用户数据安全。

安全围栏：对模型调用过程中的请求，尤其是文生图、文生视频请求进行前置隔离审核，涉政、涉黄、暴力等违规输入自动拦截。

检测并阻断提示词注入（Prompt Injection）、越狱尝试（Jailbreak）等试图绕过模型安全对齐的攻击行为。

隐私数据识别与处理：识别用户请求中是否包含个人隐私数据（身份证号、手机号、银行卡号等），并可选择脱敏处理或直接拒绝，保障用户隐私合规。

1.4应用场景

本文带您更快了解星辰 TokenHub 运营服务平台经典应用场景。

按照用户主体分为个人家庭用户、中小企业客户、大型企业与政企客户等三类场景，向用户提供标准化 API 推理接口。典型使用场景如下：

（1）个人家庭用户场景

适用于个人开发者、学生、知识工作者及家庭用户等的轻量级日常使用与学习探索。

场景 1：智能编程辅助。支撑个人开发者和技术学习者的日常开发练习、原型快速搭建、遗留代码理解、技术学习入门等用途，提供多语言代码自动生成、单元测试编写、代码注释与解释、错误分析与修复建议等推理能力。

场景 2：内容创作与知识学习。支撑日常写作、语言学习、知识整理、内容创作、求职材料优化等个人用途，提供文案撰写、翻译润色、文章改写、知识点讲解、学习笔记整理、长文本摘要等通用文本推理能力。

场景 3：学习辅导。支撑学生和家长的课后作业辅导、语言学习、自学辅助等场景，提供题目解答、作文润色、知识点讲解、错题分析与练习题生成等推理能力。

（2）中小企业客户场景

适用于中小企业及垂类智能体、大模型工具厂商的批量 Token 采购，支撑其产品化调用。

场景 1：编程类智能体与工具。支撑 AI 编程工具、代码辅助平台、低代码开发系统等场景，提供代码生成、代码审查、测试用例编写、技术文档生成等推理能力。

场景 2：视频生成。支撑营销短视频批量制作、数字人播报、电商产品视频、教育培训视频等场景，提供视频脚本生成、分镜描述、文案配音、文生视频/图生视频、数字人驱动等推理能力。

场景 3：客服与运营类智能体。支撑智能客服系统、电商自动回复、营销内容批量生成等场景，提供多轮对话应答、意图识别、工单分类、文案生成、评论摘要等推理能力。

场景 4：文档与知识管理工具。支撑合同审查工具、企业知识库产品、智能办公助手等应用，提供长文本摘要、关键信息抽取、会议纪要生成、知识库问答等推理能力。

场景 5：教育与培训工具。支撑自适应学习系统、智能家教产品、培训内容生成等应用，提供题目解答、知识点讲解、作文批改、练习题生成等推理能力。

（3）大型企业与政企客户

适用于政府部门、大型企业自主开发应用系统和自建智能体使用。

场景 1：政务服务。支撑客户自建政务服务等系统中的办事咨询自动应答、公文初稿辅助生成、群众留言分类汇总等功能，提供政策问答、办事指引、公文辅助撰写、政务数据分析、民情民意摘要等推理能力。

场景 2：运维与技术保障。支撑客户自建运维监控等系统中的告警智能聚合分析、故障智能排查、运行报告自动生成等功能，提供告警日志分析、故障根因推断、工单自动分类派单、运行报告自动生成等推理能力。

场景 3：行业系统集成。如金融、工业、医疗、能源等垂直行业，支撑客户自建业务系统的深度推理需求。金融领域可支撑合同审查、合规校验、研报摘要、风险预警等；工业领域可支撑设备故障诊断、工业视觉质检、生产报告生成等；医疗领域可支撑病历结构化、医保智能审核、医学文献摘要等，提供标准化推理底座。

1.5术语解释

本文带您了解星辰 TokenHub 运营服务平台相关术语及其基本概念。

Token

在自然语言处理中，token 通常指的是将文本分割成的最小单位，比如词语、子词或字符。在调用模型服务时，会将输入内容进行分词（tokenize），转化为模型可以理解的 token ，经过模型处理后，同样输出 token，并转化为您需要的文本或者其他内容载体。而模型处理（包括输入、输出）的 token 数量会被作为星辰 TokenHub 运营服务平台用量的一个重要计量单位。由于不同模型采用的分词策略不同，同一段文本可能会被转化为不同数量的 token。

预置模型

指平台提供的模型，您可以通过选择预置模型进行训练从而得到行业或细分场景模型，不同的基础模型的参数和能力不同。

专属模型

您可以租赁机器由平台部署模型后使用。该模型由用户专属使用，不受用户维度流量等限制。

语言类模型

基于统计学 / 深度学习，学习人类自然语言文本分布规律，能够计算一段文字出现概率、预测下一个文字 / 词语的人工智能模型。

视觉类模型

视觉模型（Vision Model）是一类以图像、视频、像素等视觉信号作为输入，通过神经网络学习视觉空间特征、物体纹理、轮廓、场景、动作规律，完成视觉感知、识别、重建、生成任务的机器学习模型 API 接入：API 为应用程序编程接口（Application Programming Interface），API 接入为通过官方提供的接口，让两套系统互相传数据、调用功能。

APP Key

APP Key 也称 API Key 应用密钥 ID，一串由平台生成的唯一身份编号，接入 API 时用来证明用户身份。

函数调用 Function Calling

Function Calling 是一种将大模型与外部工具和 API 相连的关键功能，作为自然语言与信息接口之间的“翻译官”，它能够将用户的自然语言请求智能地转化为对特定工具或 API 的调用，从而高效满足用户的特定需求。

1.6使用限制

本文介绍星辰 TokenHub 运营服务平台在产品使用方面的限制。

模型使用限制：

通过体验中心或调用 API 使用模型时，每个模型在上下文长度、最大输出长度、并发数、最大 TPM/RPM 数等方面均有不同数值的限制，部分限制信息可参见模型广场模型卡片信息。

计费说明

2.1 按需计费模式-按量付费

本节为您详细介绍星辰 TokenHub 平台按需计费方式。

星辰 TokenHub 运营服务平台中，不同模型的计费方式可能有所不同，以下为您介绍各类型模型的计费方式，您也可以从 模型广场 进入模型详情页查看每个模型的计费规则。

计费方式说明

计费方式	计费项	计费单位	模型适用范围	计费说明
Token	推理输入、输出、缓存命中	元/千 tokens	文本生成、视觉理解、向量模型	1.输入 Tokens 和输出 Tokens 分别按实际用量计价。不同的模型可能采用不同的分词策

				略，同一段文本在不同模型上的 tokens 计量会存在差异。 2.当推理请求中断（包括用户主动中断或其他原因导致的被动中断）时，如果服务端已产生但未完成输出的 Tokens，这部分 Tokens 也会计费。 3.部分模型支持分段**定价**（如不同输入长度段对应不同单价）。

时长	按输出的视频时长计费，区分分辨率、有声无声	元/秒	视频生成类模型	1.按生成视频的实际时长（秒）计费；部分模型区分不同分辨率、有声无声单价不同。 2.生成失败不计费。
张	按输入、输出的图片张数计费	元/张	图片生成、图片编辑类问题	1.按生成的图片张数计费，如一次请求生成 3 张，按 3 张数计费，生成失败不计费。 2.部分模型输入图片数也可能计费。
视频 Token	按输出的视频 Token 量计费，区分有视频输	元/千 tokens	视频生成类模型	1.按生成视频的实际 token 消耗量计费；区

	入、无视频输入			分不同分辨率、是否包含视频输入单价不同。 2.生成失败不计费。
语音模型	按输入的语音时长计费	元/秒	语音识别模型	按输入音频的时长计费，输出不计费。
语音模型	按输入的字符数计费	元/万字符	语音合成模型	按输入文本的字符数计费，输出不计费。
语音模型	音色复刻与设计：按次收费	元/次	语音合成模型	按新建或复刻的音色数量计费

产品价格

按模型实际使用量和单价付费，不同模型的单价不同，具体详见模型广场的模型介绍。

出账时间：每小时出账。

付款类型：按量后付费。

用量抵扣顺序：使用模型时，将优先消耗免费额度，免费额度耗尽后，如有购买量包，则优先抵扣量包；免费额度和量包耗尽后，产生的用量按实际用量计费。

使用限制

当前对文本/视觉类模型设置了 RPM（Requests Per Minute，每分钟请求数）及 TPM（Tokens Per Minute，每分钟 token 数量）的调用限制，对图片生成类模型设置了 IPM（Image Per Minute，每分钟生成图片数）、对视频生成类模型设置了并发数限制（同时进行中的视频任务数量）， 具体限制可在模型广场模型详情页中查看。

2.2 包周期计费模式-编程Token Plan

本节为您详细介绍包周期计费模式-编程 Token Plan

编程 Token Plan 是专为 AI 编码场景设计，仅需按固定周期付费，即可在主流 AI 开发工具中畅享热门模型，获得便捷、稳定、流畅的编码体验。

编程Token Plan（积分版）

	编程 Token Plan 3000 积分	编程 Token Plan 10000 积分	编程 Token Plan 25000 积分	编程 Token Plan 50000 积分	编程 Token Plan 100000 积分

适用场景	入门体验，适合首次体验 AI 工具能力 约兑换 1429 万 Token 的 minimax-m3，或 462 万 Token 的 kimi-k2.6 （参考值，以最终消耗为准）	日常使用，适合处理基础的开发任务 约兑换 4762 万 Token 的 minimax-m3，或 1538 万 Token 的 kimi-k2.6 （参考值，以最终消耗为准）	轻量开发，适合处理较为复杂的开发任务 约兑换 11905 万 Token 的 minimax-m3，或 3846 万 Token 的 kimi-k2.6 （参考值，以最终消耗为准）	专业开发，适合处理复杂项目的开发任务 约兑换 23810 万 Token 的 minimax-m3，或 7692 万 Token 的 kimi-k2.6 （参考值，以最终消耗为准）	高频 AI 开发，适合处理复杂项目的持续迭代任务 约兑换 47619 万 Token 的 minimax-m3，或 15385 万 Token 的 kimi-k2.6 （参考值，以最终消耗为准）
价格	29 元/月	89 元/月	199 元/月	399 元/月	699 元/月
额度	3000 积分	10000 积分	25000 积分	50000 积分	100000 积分
支持模型	DeepSeek-V4-Pro、DeepSeek-V4-Flash-0731、GLM-5.2、GLM-5.1、Kimi-K2.6、MiniMax-M3。				

	注意：套餐模型库为动态更新机制，将根据模型性能、稳定性、供应情况，对模型进行新增、替换、升级、范围调整或下线。套餐仅提供当期适配模型使用权限，不承诺永久固定提供任一指定模型。

积分与模型的兑换关系

模型	子项	1 积分相当于多少个 Token
DeepSeek-V4-Pro	输入	1,111
	输出	370
DeepSeek-V4-Flash	输入	3,333
	输出	1,111
GLM-5.2	输入	1,250
	输出	357
GLM-5.1	输入 [0,32k]	1,667
	输入 [0,32k]时，输出	417
	输入 (32k,200k]	1,250

	输入 (32k,200k]时，输出	357
Kimi-K2.6	输入	1,538
	输出	370
minimax-M3	输入 [0,512k]	4,762
	输入 [0,512k]时，输出	1,190
	输入 (512k，1M]	2,381
	输入(512k，1M]时，输出	595

套餐模型库为动态更新机制，将根据模型性能、稳定性、供应情况，对模型进行新增、替换、升级、范围调整或下线。套餐仅提供当期适配模型使用权限，不承诺永久固定提供任一指定模型。

编程Token Plan（按Token计量，老套餐即将下线，仅存量已订阅用户可续订，不支持新购）

	编程 Token Plan 2500 万	**编程 Token Plan 8000 万**	**编程 Token Plan 1.8 亿**	**编程 Token Plan 3.8 亿**	**编程 Token Plan 6.8 亿**
价格	29 元/月	89 元/月	199 元/月	399 元/月	699 元/月

额度	2,500 万 Tokens	8,000 万 Tokens	18,000 万 Tokens	38,000 万 Tokens	68,000 万 Tokens
支持模型	DeepSeek-V4-Flash-0731、GLM-5.1、GLM-5.2 注意：套餐模型库为动态更新机制，将根据模型性能、稳定性、供应情况，对模型进行新增、替换、升级、范围调整或下线。套餐仅提供当期适配模型使用权限，不承诺永久固定提供任一指定模型。				

2.3 包周期计费模式-Tokens量包

本节为您详细介绍包周期计费模式-Tokens 量包。

Tokens 量包概念

Tokens 量包是将 tokens 按照固定额度打包售卖，当您购买并启用 tokens 用量包后，系统将优先从量包中扣除模型推理所消耗的 token 数量。该资源包具备一定有效期，可在约定时间内用于抵扣相应模型的 tokens 消耗。

说明：Tokens 量包额度同时包含模型推理时输入和输出所产生的 Tokens。

购买流程

以下购买流程以开通“DeepSeek-V4-Flash(预览版）”量包为例。

访问控制台进入模型推理-在线推理页面，在“预置服务”中找到服务名称为“DeepSeek-V4-Flash(预览版）”。

在列表操作列点击“购买量包”，根据自身需要选择对应的量包规格和购买数量并进行确认购买。

量包购买成功后可在 Tokens 量包页面激活后，点击“开启服务”后使用量包额度。

产品价格

注意：每个天翼云账号在每个模型可同时存在 12 个使用中/未激活/购买中的 Tokens 量包。如果账号内没有使用中/未激活/购买中的量包，可购买 12 个。

模型名称	量包额度（Tokens）	有效期	标准价(元)
DeepSeek-V4-Flash（预览版）	1000 万	1 个月	9.9
	3200 万		29.9
	1 亿		89.9
GLM-5（正式版）	1500 万	1 个月	39.9
	7000 万		159.9
	1.5 亿		299.9
DeepSeek-V3.2（旗舰版）	1000 万		9.9

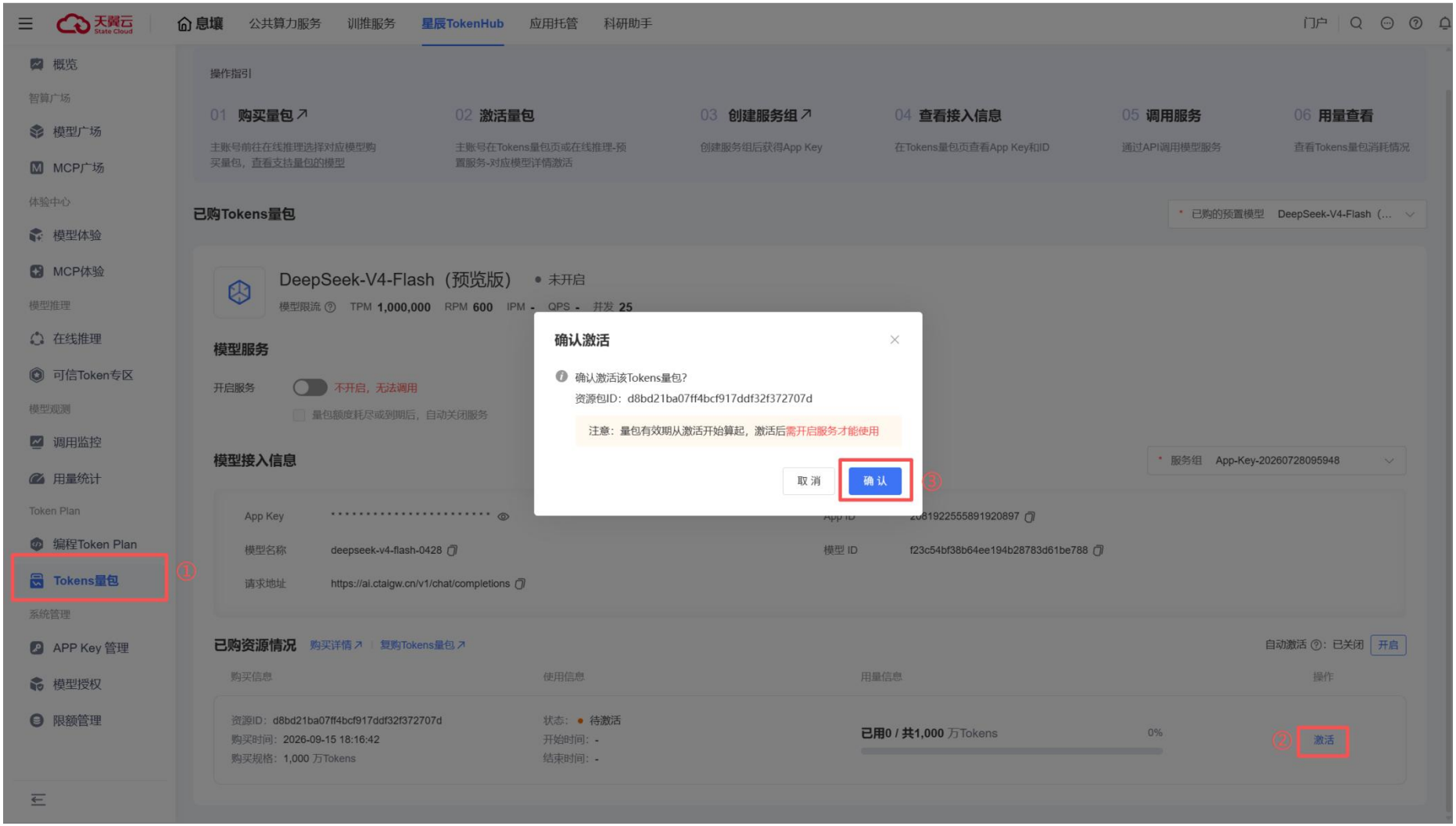
	4000 万		29.9
	8000 万		49.9
DeepSeek-V3	1000 万	1 个月	35
	5000 万	3 个月	175
	1 亿	6 个月	350
	5 亿	12 个月	1750
	10 亿	12 个月	3500
DeepSeek-R1	1000 万	1 个月	70
	5000 万	3 个月	350
	1 亿	6 个月	700
	5 亿	12 个月	3500
	10 亿	12 个月	7000

激活步骤

注意：Tokens 量包需手动激活后方可正常使用，激活后即刻开始计算有效期。

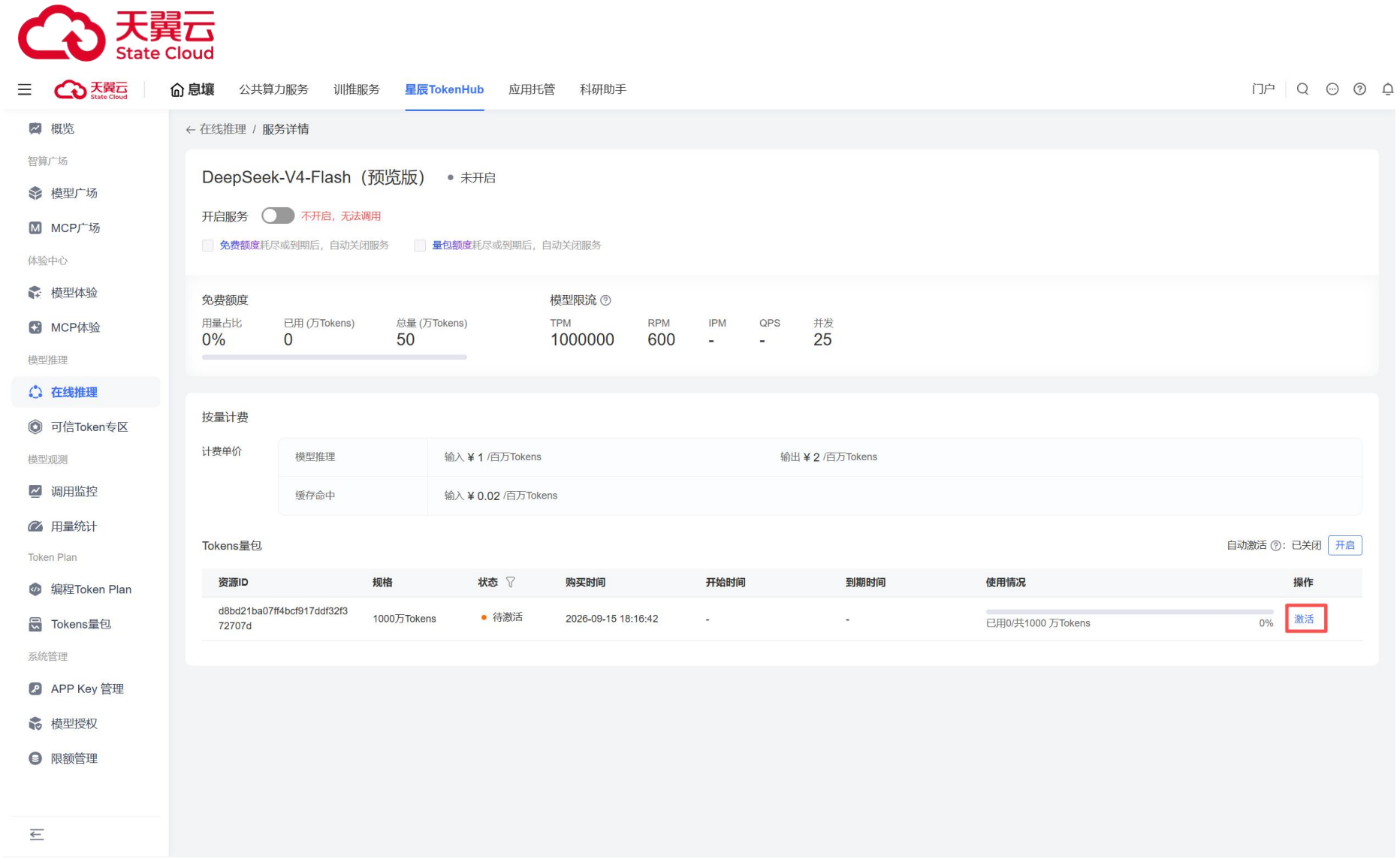
方式一：

进入 Tokens 量包页面，在“开通资源情况”中找到对应量包，点击“激活”并确认。



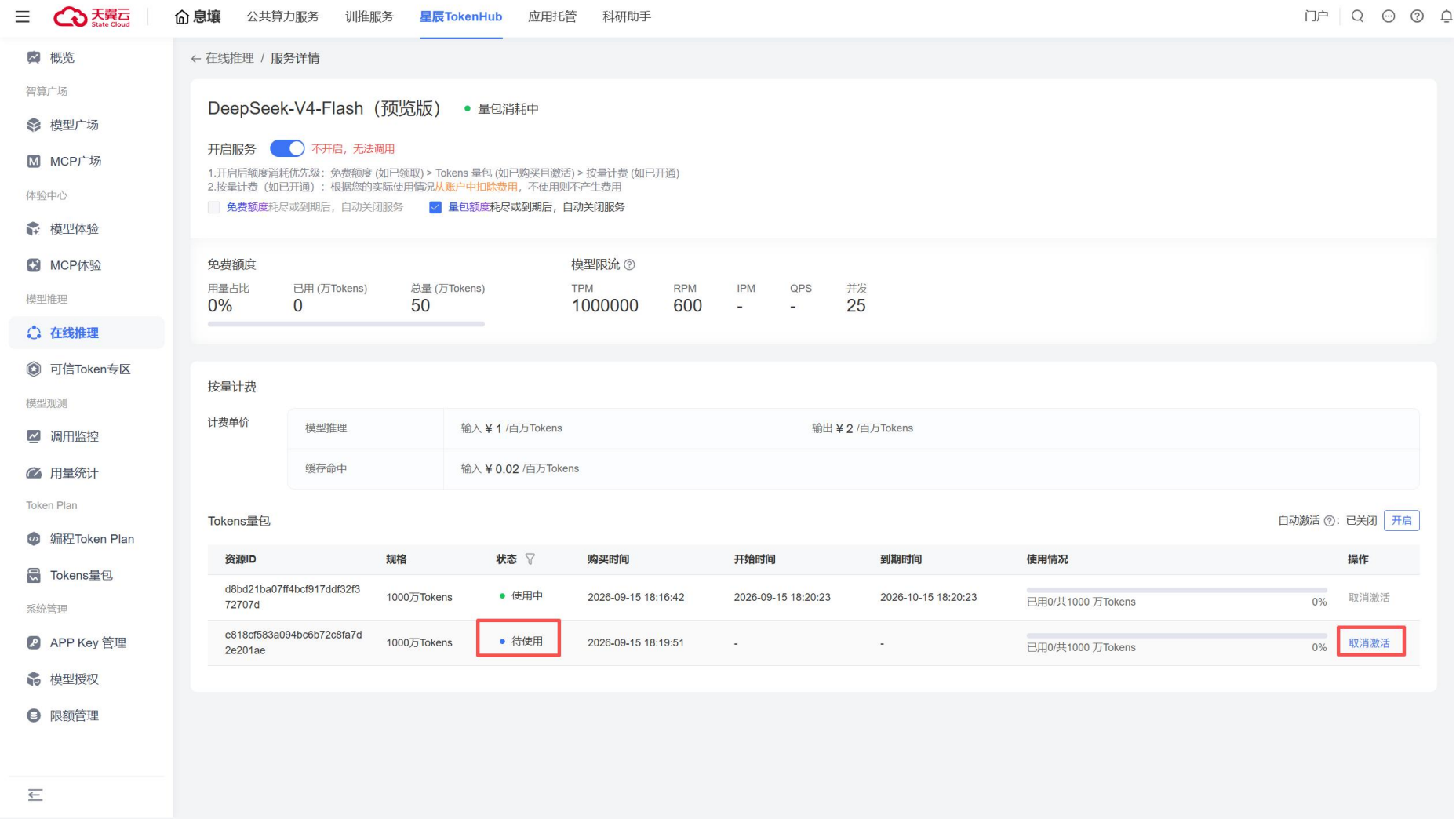
方式二：

在“模型推理->在线推理->预置服务”中找到对应服务，点击服务名称进入详情页激活Token Plan量包。


天翼云
State Cloud
息壤
公共算力服务
训推服务
星辰TokenHub
应用托管
科研助手
门户
概览
智算广场
模型广场
MCP广场
体验中心
模型体验
MCP体验
模型推理
在线推理
可信Token专区
模型观测
调用监控
用量统计
Token Plan
编程Token Plan
Tokens量包
系统管理
APP Key 管理
模型授权
限额管理
← 在线推理 / 服务详情
DeepSeek-V4-Flash（预览版）
未开启
开启服务
不开启，无法调用
免费额度耗尽或到期后，自动关闭服务
量包额度耗尽或到期后，自动关闭服务
免费额度
用量占比
0%
已用 (万Tokens)
0
总量 (万Tokens)
50
模型限流
TPM
1000000
RPM
600
IPM
-
QPS
-
并发
25
按量计费
计费单价
模型推理
输入 ¥ 1 /百万Tokens
输出 ¥ 2 /百万Tokens
缓存命中
输入 ¥ 0.02 /百万Tokens
Tokens量包
自动激活：已关闭
开启
资源ID
规格
状态
购买时间
开始时间
到期时间
使用情况
操作
d8bd21ba07ff4bcf917ddf32f372707d
1000万Tokens
待激活
2026-09-15 18:16:42
-
-
已用0/共1000 万Tokens
0%
激活

说明

若只有一个量包，则激活成功后状态变为“使用中”，且开始计算有效期，不可取消激活。若所有量包都已激活，若所有量包都已激活，则激活后状态为“待使用”的量包（用量和有效期均未开始）其支持“取消激活”。



2.4 欠费说明

本文介绍欠费产生后星辰 TokenHub 运营服务平台的处理规则。 当客户的天翼云账户中，没有余额或产生欠费的情况下，天翼云会关停客户的星辰 TokenHub 运营服务平台的按需计费服务，并通过短信通知客户充值。

冻结服务

欠费后，星辰 TokenHub 运营服务平台将自动冻结客户的按需计费服务。欠费不影响 Token 量包和 Token Plan 编程套餐等包周期资源的服务。

恢复服务

欠费后，客户可通过在线充值结清欠款，欠款结清后，星辰 TokenHub 运营服务平台将自动解冻并恢复按需计费服务，无需客户手动处理，即可正常使用模型 API 调用服务。

余额预警设置

为避免欠费情况的产生，可通过对客户在天翼云官网账户的可用额度进行预警设置，当用户的余额低于阈值，系统会发送短信提醒。

操作步骤：

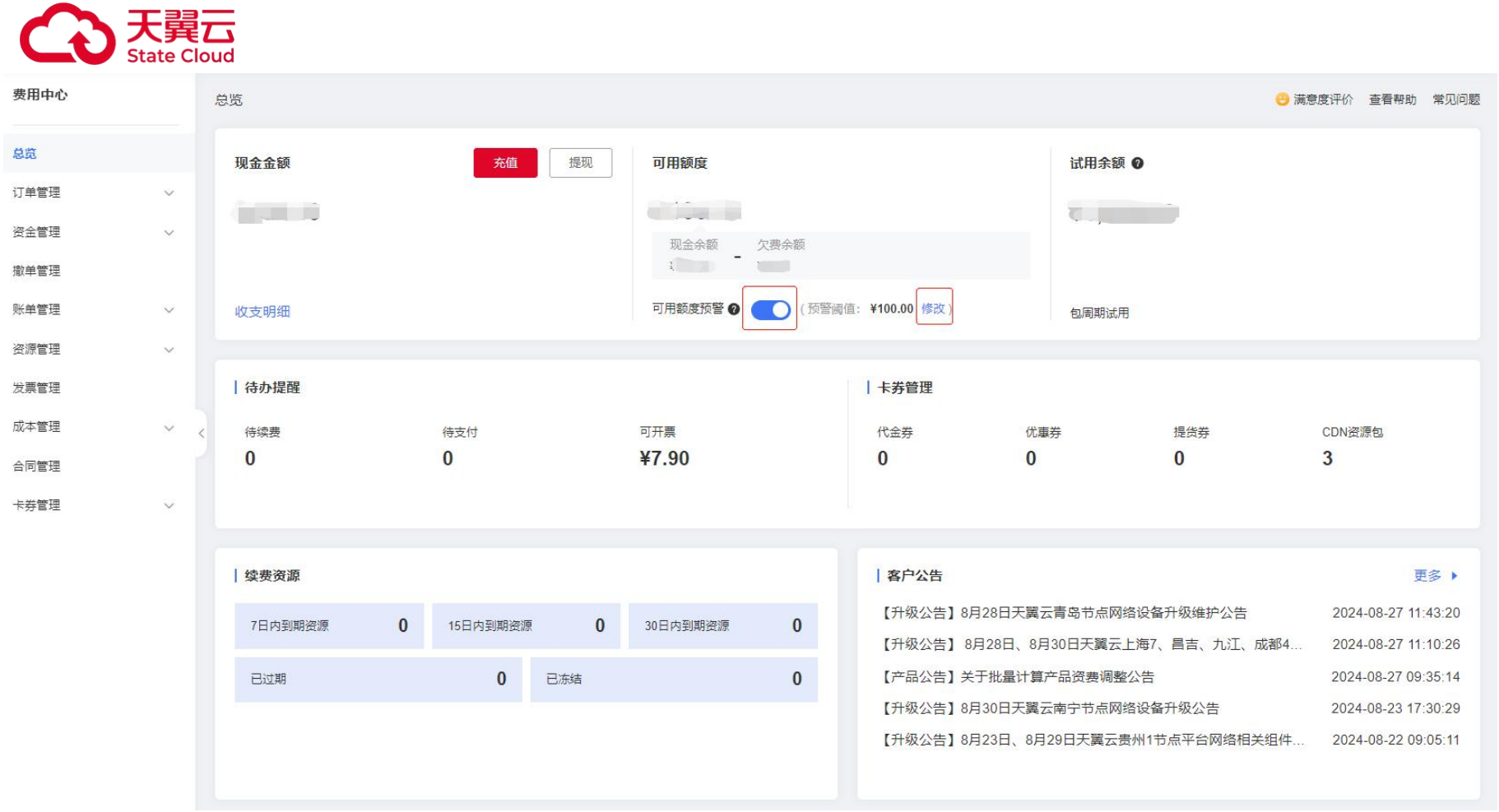
登录天翼云账户。

单击右上角。

单击【费用中心】。



打开【可用额度预警】开关，修改预警阈值，当用户的余额低于阈值，系统会发送短信提醒。



2.5 账单查询

本文介绍账单查询的方法。

简要说明

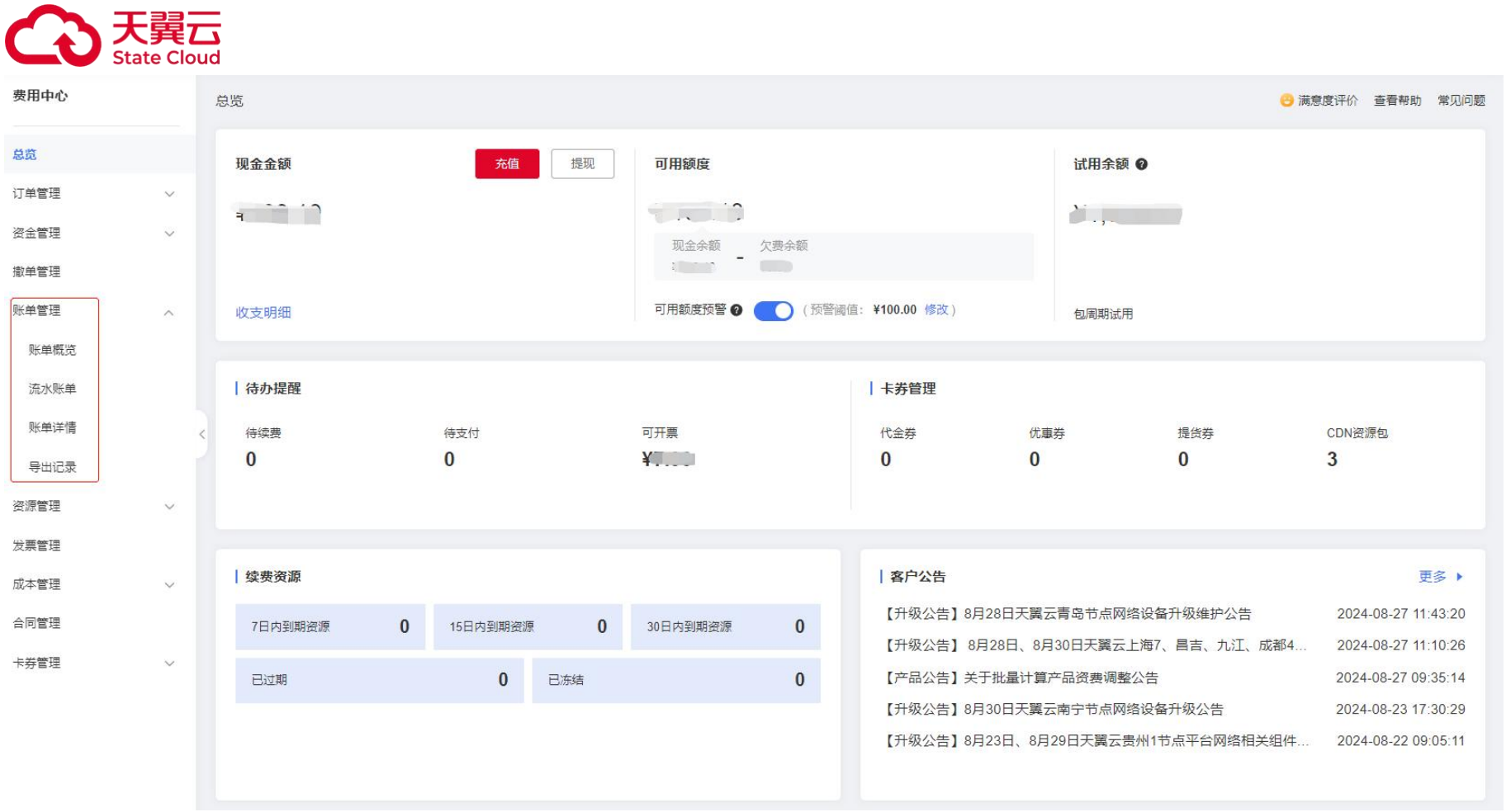
天翼云支持用户查询账单概览、流水账单、账单详情及导出记录。

查询步骤

登录天翼云账户，单击右上角，再单击【费用中心】。



选择【账单管理】，分别单击【账单概览】、【流水账单】、【账单详情】、【导出记录】查询不同维度的账单。

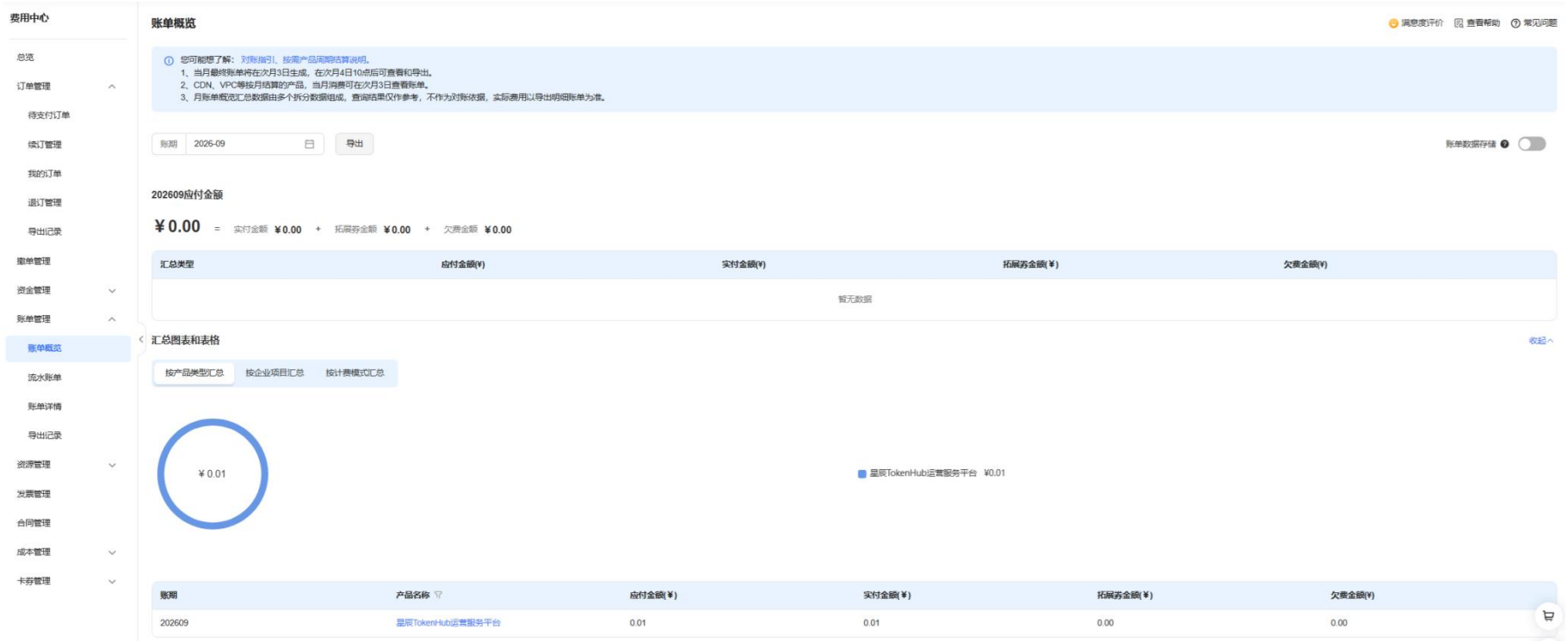


账单概览

账单概览可以展示用户的消费汇总、应付金额、实付金额等数据。

账单概览可显示近6个月消费汇总情况。

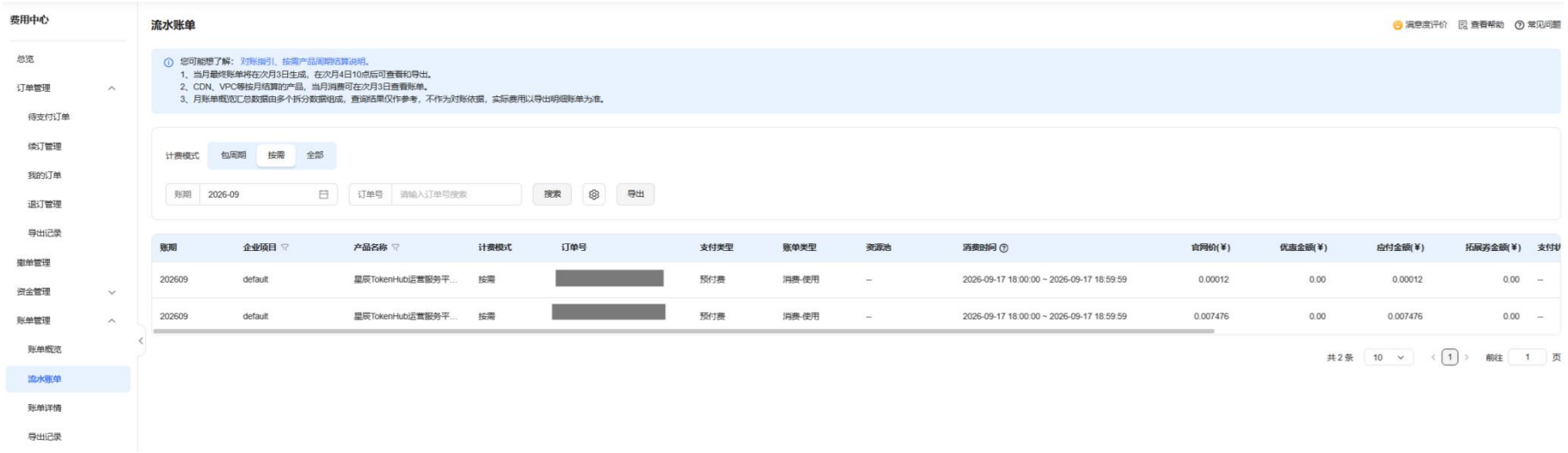
选择【账期】，可查看单月消费概况，并支持导出两种格式（XLSX、CSV）的账单，可在“导出记录”下载。



流水账单

流水账单可以展示按照不同计费模式和账期维度构成的数据，并可导出两种格式（XLSX、CSV）的账单，在“导出记录”下载。

支持按计费模式（包周期、按需、全部）、账期等过滤条件查看流水账单数据，同时支持导出格式为XLSX、CSV的账单。如需查看按需计费资源下不同模型计费项粒度的具体用量和费用，请通过【账单详情】页面查看。



账单详情

支持不同统计维度（使用量、资源、产品）、统计周期（按账期、按天、明细）查看账单的详细信息，并支持导出两种格式（XLSX、CSV）的账单，可在“导出记录”下载。

如需查看星辰TokenHub产品总费用情况，统计维度请选“产品”。

如需查看按需计费的每个模型计费项的用量和账单，统计维度请选择“使用量”。如下图所示：

说明：

星辰 TokenHub 运营服务平台按需计费的账期较使用时间延迟 1 个月，即 7 月份的用量，账期需选择 8 月。

账单字段说明：官网价=模型计费项标准价*使用量，应付金额=官网价-优惠金额，规格中的"类型"为模型规格名称，规格中的"计费维度"为模型计费项。

计费单位说明：按 token 计费的语言模型和视频模型的单位为元/千 tokens，图片模型的计费单位为张，按时长计费的视频模型和语音识别模型的计费单位为元/秒，按字符计费的语音合成模型的计费单位为元/万字符，声音复刻与设计的计费单位为元/次，联网搜索的计费单位为元/千次。



导出记录

用户单击【导出记录】后，可查看已导出的全部文件记录，也可在如下图的页面单击【下载】，将账单数据下载至本地查看。

2.6 产品退订

针对星辰 TokenHub 运营服务平台退订操作，为您进行说明。

一般退订规则

若您自订阅成功之日起 7 天内未产生任何 API 调用或资源消耗记录，可通过【天翼云官网—我的—费用中心—订单管理—退订管理】途径申请退订。

另外，您可通过天翼云官网工单或者客服电话【400-810-9889】沟通申请退款，款项会原路退回。

特殊退订规则

编程 Token Plan 服务自订阅成功之日起不支持退订。您在订阅该服务前应仔细评估自身需求，一旦完成订阅，已支付的费用不予退还。 其他产品如存在特殊的退订规则（包括但不限于不支持退订、限定时长内可退订等）或本产品后续调整退订规则的，以该产品订购页面展示的具体规则为准。

快速入门

3.1 服务开通

本文介绍如何开通星辰TokenHub运营服务平台，包括账户实名认证、不同计费模式的开通步骤。

前提条件

开通星辰TokenHub运营服务平台，需要先注册天翼云账户并确保完成实名认证。

注册并登录：天翼云官网。相关账号注册说明，详情请见：注册天翼云账号。

未实名认证的用户请按提示完成实名认证后才能开通业务。相关实名认证说明，详情请见：实名认证。

计费模式

星辰TokenHub运营服务平台支持按量计费（后付费）和包周期（预付费）两种计费方式，当前包周期支持编程Token Plan和Tokens量包。

开通步骤

按量计费产品购买流程

购买星辰TokenHub平台-按量产品之前请确保您的账户余额大于等于100元。天翼云账户充值步骤请参见账户充值。

进入星辰TokenHub运营服务平台的产品详情页快速了解产品，单击【立即体验】，进入星辰TokenHub控制台。

通过控制台-在线推理页面，点击【开通按量计费】按钮。



在购买页面选择区域，勾选“我已阅读，理解并接受《天翼云星辰TokenHub运营服务平台服务协议》、《天翼云星辰TokenHub运营服务平台用户信息处理规则》”，确认无误后单击【立即购买】。

注意：

当前全球（不含中国内地）区域暂不支持自助开通，如需开通请联系您的客户经理下单。

按需服务只需开通一次，即可调用对应区域的所有模型（含后续新增模型），无需重复开通。



星辰TokenHub平台-按需服务开通后，页面会显示【提交成功】，单击【跳转到控制台】，即可进入星辰TokenHub控制台进行使用。

您需要确保天翼云实名账户中的余额充足，若余额不足导致欠费，按需服务将被冻结，冻结后模型API调用将会失败。

包周期产品购买流程

编程Token Plan套餐购买和使用流程，详见编程Token Plan。

Token量包购买和使用流程，详见Token Plan。

3.2 快速入门

本文帮助您了解并快速按照步骤完成API调用模型。

前提条件

登录星辰TokenHub运营服务平台（原模型推理服务）控制台。

第一步：开启模型服务

主用户可通过“在线推理”页面查看模型列表，并对目标模型开启服务便于后续调用。如果某个模型确定不再使用，可以点击“关闭服务”按钮，服务关闭后，API调用将会失败。

部分模型配有免费token额度用于体验，可点击“开启免费体验”领取免费额度。免费额度用完后，将自动转按需计费，若您不希望产生按需费用，请在开启免费体验的弹窗页面中，主动勾选“免费额度耗尽或到期后，自动关闭服务”。



第二步：获取 App Key 和模型 ID

方式一

在左侧菜单栏进入“APP Key管理”，点击“+创建服务组”，依次填写服务组名称、服务组描述、生效时间、额度控制、性能控制等基本信息、并选择对应的配置服务。

创建服务组后，从服务组上面获取APP KEY。

说明：
请妥善保管 App Key，避免公开共享，以防安全风险和资金损失。强烈建议您不要将其直接写入到调用模型的代码中。

通过“模型广场”-“模型详情”选择目标模型并在详情页内获取模型ID或调用名称。

通过“模型广场”-“模型详情”-“API快速接入”获取调用示例代码，发起API调用。

方式二

在左侧菜单栏进入“概览”，点击“API接入”。

在API快捷接入页面：

选择服务组：从下拉菜单中选择第一步创建的服务组。如果尚未创建，可点击“确认创建并选择”快速创建默认服务组

选择模型：从模型下拉菜单中选择需要接入的模型

选择完成后，页面下方将自动展示该模型支持的所有编程语言的示例代码，即可发起API调用。

有关更多APP Key内容，请参见：APP Key管理。

第三步：服务与用量观测

在左侧菜单栏进入“调用监控”，查看调用数据和监控指标详情。具体使用，请参见：调用监控。

在左侧菜单栏进入“用量统计”，查看按需计费的用量明细。具体使用，请参见：用量统计。

第四步：子用户管理

企业管理员可新增子用户，将子用户分配给不同业务/部门使用，每个子用户可以独立新建和管理自己的APP Key。主用户可以在控制台查看子用户的token用量，并对子用户的用量做限额管理。

操作步骤：

登录IAM控制台，创建用户组，并为这个用户组授权。具体操作请参见创建用户组和授权。

创建IAM用户，选择对应的用户组。具体操作请参见创建IAM用户和登录。

完成IAM用户创建后，再通过星辰TokenHub控制台的“模型授权”页面，对子用户授权模型，可选择全部模型，或自定义模型范围。具体使用，请参见：模型授权。

完成模型授权后，即可通知子用户登录控制台使用模型推理服务。

第五步：用量限额

如果需要对整个账号全局粒度、主子用户粒度、服务组粒度限制模型用量与费用，可通过星辰TokenHub控制台的“限额管理”页面，配置限额。具体使用，请参见：限额管理。

4.1 编程Token Plan

4.1.1套餐概述

编程 Token Plan

编程Token Plan是专为AI编码场景设计，仅需按固定周期付费，即可在主流AI开发工具中畅享热门模型，获得便捷、稳定、流畅的编码体验。

套餐优势

模型丰富：覆盖业内主流模型，覆盖DeepSeek、GLM等系列模型，支持复杂推理、长上下文等场景，可按需调用。

兼容主流工具：支持OpenClaw、Claude Code、OpenCode、Cursor、Cline、Chatbox、CodeBuddy、Trae、Hermes Agent、Codex 主流编程工具，套餐额度共享，灵活适配多种开发场景。

多种套餐适配：提供5档不同额度不同价格套餐，适应不同强度的编程场景。

套餐定价

编程 Token Plan（积分版）

	编程 Token Plan 3000 积分	编程 Token Plan 10000 积分	编程 Token Plan 25000 积分	编程 Token Plan 50000 积分	编程 Token Plan 100000 积分
适用场景	入门体验，适合首次体验 AI 工具能力 约兑换 1429 万 Token 的 minimax-m3，或 462 万 Token 的 kimi-k2.6 （参考值，以最终消耗为准）	日常使用，适合处理基础的开发任务 约兑换 4762 万 Token 的 minimax-m3，或 1538 万 Token 的 kimi-k2.6 （参考值，以最终消耗为准）	轻量开发，适合处理较为复杂的开发任务 约兑换 11905 万 Token 的 minimax-m3，或 3846 万 Token 的 kimi-k2.6 （参考值，以最终消耗为准）	专业开发，适合处理复杂项目的开发任务 约兑换 23810 万 Token 的 minimax-m3，或 7692 万 Token 的 kimi-k2.6 （参考值，以最终消耗为准）	高频 AI 开发，适合处理复杂项目的持续迭代任务 约兑换 47619 万 Token 的 minimax-m3，或 15385 万 Token 的 kimi-k2.6 （参考值，以最终消耗为准）
价格	29 元/月	89 元/月	199 元/月	399 元/月	699 元/月

额度	3000 积分	10000 积分	25000 积分	50000 积分	100000 积分
支持模型	DeepSeek-V4-Pro、DeepSeek-V4-Flash-0731、GLM-5.2、GLM-5.1、Kimi-K2.6、MiniMax-M3。 注意：套餐模型库为动态更新机制，将根据模型性能、稳定性、供应情况，对模型进行新增、替换、升级、范围调整或下线。套餐仅提供当期适配模型使用权限，不承诺永久固定提供任一指定模型。				

积分消耗机制

积分消耗受模型类型、Token 用量、工具调用等因素影响，具体数值以调用消耗明细为准，积分与模型的兑换关系详见

https://www.ctyun.cn/document/11061839/11092368

编程 Token Plan（按 Token 计量，老套餐即将下线，仅存量已订阅用户可续订，不支持新购）

	编程 Token Plan 2500 万	**编程 Token Plan 8000 万**	**编程 Token Plan 1.8 亿**	**编程 Token Plan 3.8 亿**	**编程 Token Plan 6.8 亿**
价格	29 元/月	89 元/月	199 元/月	399 元/月	699 元/月
额度	2,500 万 Tokens	8,000 万 Tokens	18,000 万 Tokens	38,000 万 Tokens	68,000 万 Tokens

支持模型	DeepSeek-V4-Flash-0731、GLM-5.1、GLM-5.2 注意：套餐模型库为动态更新机制，将根据模型性能、稳定性、供应情况，对模型进行新增、替换、升级、范围调整或下线。套餐仅提供当期适配模型使用权限，不承诺永久固定提供任一指定模型。

套餐额度查询

登录星辰TokenHub运营服务平台控制台->进入编程Token Plan页面查询当前套餐的剩余额度、已用额度、总额度及总额度中已使用百分比信息。

套餐使用须知

套餐使用范围：仅限在AI 编程工具中使用，不可用于 API 调用的形式用于自动化脚本、自定义应用程序后端或任何非交互式批量调用场景。违规使用可能导致停用或APP Key封禁。

账号购买限制：同一主账号同时只能购买一个套餐，目前仅支持主账号发起购买，且购买详情和调用信息仅主账号可见。

账号使用规范：套餐为购买人专享使用，禁止共享。账号共享可能导致权益受限，甚至导致停用或账号封禁。

套餐续订：套餐有效期内可手动在控制台续订套餐。一旦套餐失效则无法续订，需重新再次购买。暂不支持自动续订。

套餐退款规则：编程Token Plan套餐一经购买成功即视为确认，不支持退订/退款。即使未使用套餐，费用也无法退回。建议根据使用需求选择合适的套餐。套餐不参与七天无理由全额退订。

接入 AI 工具

编程Token Plan套餐支持主流的AI工具，可按需选择。

OpenClaw、Claude Code、OpenCode、Cursor、Cline、Chatbox、Codebuddy、Trae 等。

常见问题

请查看编程 Token Plan 常见问题

4.1.2快速开始

第一步：购买套餐

登录模型推理服务控制台，选择“编程Token Plan”，按需购买套餐。套餐介绍参见套餐概述。

第二步：获取专属 APP Key 和专属请求地址

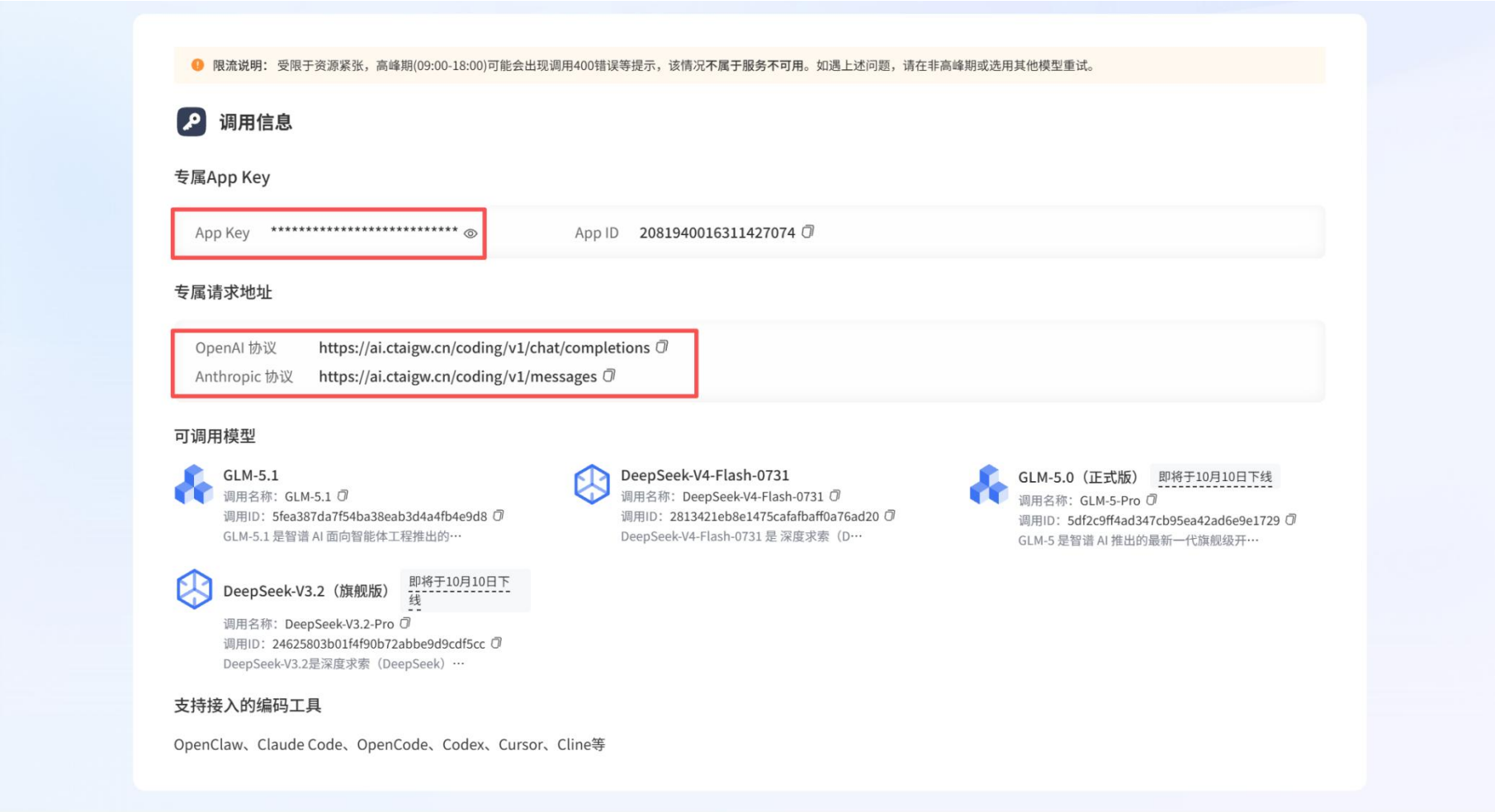
购买成功后在“控制台-编程Token Plan”获取专属APP Key和专属请求地址。

专属APP Key：cp_xxx

专属请求地址：

OpenAI兼容接口协议：https://ai.ctaigw.cn/coding/v1/chat/completions

Anthropic兼容接口协议：https://ai.ctaigw.cn/coding/v1/messages



第三步：接入 AI 工具

编程Token Plan套餐支持主流的AI工具，可按需选择。将APP Key、请求地址和具体的模型配置到工具中，即可使用：

OpenClaw、Claude Code、OpenCode、Cursor、Cline、Chatbox、Codebuddy、Trae 等。

第四步：查看积分兑换关系

编程Token Plan（积分版）适用，如为按Token计量老套餐，可跳过此步骤。

查看https://www.ctyun.cn/document/11061839/11092368，了解模型与积分的兑换关系，选择合适的模型。

第五步：开启编码

第六步（可选）：套餐续订

说明：
套餐有效期内可手动在控制台续订套餐。一旦套餐失效则无法续订，需重新再次购买。暂不支持自动续订。

套餐续订成功即视为确认，不支持退款/退订。即使未使用套餐，费用也无法退回。请依据自身需要选择合适的续订周期。

续订规则：

续订成功后，套餐有效期自原到期时间顺延，新额度将在下一个订阅月生效；

当月剩余额度到期自动清零，不结转；下一订阅月额度将重置为新套餐额度；

续订处理需一定时间，续订后的实际到期时间以订单完成时间为准

续费方式：

登录控制台->编程Token Plan页面，点击“续订”。确定续订周期后，点击“确认购买”即可完成操作。

常见问题

详见编程Token Plan常见问题

4.1.3编程Token Plan常见问题

（一）新旧套餐说明

基础概念类

新套餐核心区别是什么?

A：本次是套餐资源计量单位升级，资源计量由原来统一的 Token 计量，升级为积分计量。调用不同模型消耗积分不一样，不再是所有模型统一扣 Token。新版模型更丰富，支持Kimi、MiniMax等系列的领先模型。

积分是什么？积分和 Token 有什么区别？

A：积分是新版套餐的资源计量单位，把 Token / 张 / 秒等复杂技术计量统一为积分。

老版Token 模式：所有模型调用统一消耗 Token，1Token 就是 1Token，不分模型；

新版积分模式：不同模型有独立积分兑换比例，调用模型按对应规则扣积分。兑换关系可查询 https://www.ctyun.cn/document/11061839/11092368

举例：1 积分可兑换 GLM5.1 的 1667 个输入Token；1 积分可兑换DeepSeek-V4-Pro 的 1111 个输入Token。（参考值，以最终消耗为准）

如同游戏厅不同设备消耗不同游戏币，全程仅消耗积分，不直接消耗现金。

新版支持哪些模型？老版本支持哪些模型？

A：新版积分套餐支持：Kimi-K2.6、minimax-M3、DeepSeek-V4-Pro、DeepSeek-V4-Flash、GLM-5.2、GLM-5.1，新套餐将根据模型先进性、性能、稳定性等因素不定期更新模型列表。

老版套餐支持：DeepSeek-V4-Flash、GLM-5.1、GLM-5.2，老套餐后续将不再更新模型。

存量用户权益（购买老版 Token 制计量套餐的用户）

我是存量客户，现在正在订阅周期内，这次升级会不会影响我现在的业务调用?

A：不会影响。您当前套餐可以正常使用到订阅周期结束，套餐消耗、扣费规则维持原来 Token 模式不变，存量权益不受本次升级改动。

我现在订阅还没到期，能不能继续续订老的 Token 套餐?

A：可以。只要在当前订阅周期结束前完成续订，仍然可以订购老 Token 套餐，继续使用 Token 计量模式。

举例：订阅周期 9 月 30 日到期，9 月 30 日前续订，仍然是老套餐 Token 模式。

我的订阅周期已经到期了，还能不能续订老 Token 套餐?

A：不能续订老套餐。周期结束后，如需继续使用，只能购买新版积分计量套餐。

举例：订阅 9 月 30 日到期，10 月 1 日再购买，只能选购新版积分套餐。

我存量套餐还没到期，能不能提前切换成新版积分套餐?

A：存量套餐在当前订阅周期内保持原有 Token 模式运行，不支持中途切换为积分套餐；您可以在当前周期结束后，选购新版积分套餐。

我存量套餐剩余的 Token 资源，周期结束没消耗完怎么办?

A：存量套餐剩余 Token 按照原有套餐规则执行，资源有效期至订阅周期结束。

计费 & 消耗计算类

同一个积分，调用不同模型能跑的 Token 数量不一样是吗?

A：是的。新版积分模式下每个模型有固定积分兑换 Token 比例，举例：

1 积分 = GLM5.1 1667输入Token；1 积分 = DeepSeek-V4-Pro 输入1111Token。调用模型时，系统会按模型对应比例扣除积分。具体请以实际消耗为准。

新版积分套餐，调用模型会额外收取其他费用吗?

A：资源消耗仅扣除套餐内积分，无额外调用服务费。

我怎么查看积分消耗明细？在哪里看剩余积分?

A：控制台消耗统计页面会更新为积分维度，可查看积分消耗、剩余积分。

新客户相关

新客户能不能购买老套餐?

A：新用户仅支持购买新版积分版套餐，无法订购老套餐。

调用类

模型新增了 Kimi-K2.6、minimax-M3，调用这些新模型只能用积分版套餐吗?

A：是的，新增模型仅在新版积分套餐支持；老 Token 套餐无法调用 Kimi-K2.6、minimax-M。

切换新版积分套餐，需要修改 API 接入代码吗?

A：基础 API 调用地址、请求参数结构无需改动；仅资源计量逻辑变更，建议关注制台消耗单位从 Token 变为积分。如需预估消耗，需要按各模型积分比例重新评估用量。

（二）接入与调用

常见报错及解决方案

错误码	错误信息	可能原因	解决方案
700020	FUSION_PLAN_RPM_DENY（http code: 429）：套餐 RPM 超限	套餐 60 秒窗口内请求数达到 rpm 上限	降低请求频率，稍后重试
700021	FUSION_PLAN_TPM_DENY（http code: 429）：套餐 TPM 超限	套餐 60 秒窗口内 token 消耗达到 tpm 上限	减少单次 token 消耗或请求频率，稍后重试

700022	FUSION_PLAN_CONCURRENCY_DENY（http code: 429）：套餐并发已达上限	套餐当前并发数达到并发上限	等待正在执行的请求完成
610002	CODING_PLAN_APP_KEY_NOT_EXIST（http code: 400）：当前编程套餐不支持该模型	当前编程套餐不支持该模型	确定当前套餐支持的模型和传输的模型是否一致
610003	CODING_PLAN_APP_KEY_NOT_EXIST（http code: 400）：当前套餐 AppKey 不存在或已失效	套餐失效	确认 APP Key 是否为套餐专属 APP Key、套餐是否已失效或状态异常
610002	CODING_PLAN_APP_KEY_MODEL_MISMATCH（http	套餐和模型不匹配	确定当前套餐支持的模型和传输的模型是否一致

	code: 400）：当前编程套餐 不支持该模型		

在 Cursor 中无法调用 Coding Plan 支持的模型?

如果在界面看到了如下报错:

The model xxx does not work with your current plan or api key.
Named models unavailable Free plans can only use Auto. Switch to Auto or upgrade plans to continue.

由于 Cursor 的产品限制，Cursor 免费版仅支持 Auto 模式，且不支持调用自定义模型。请升级至 Cursor Pro 及以上套餐，即可正常使用 Coding Plan 支持的模型。

产品功能类

套餐专属APP Key/请求地址与通用APP Key/请求地址有什么区别?

使用套餐时必须在“控制台-编程Token Plan”页面获取专属APP Key和专属请求地址，否则无法抵扣套餐额度；“控制台-系统管理-APP Key管理”页面的APP Key支持模型API调用，支持token按量、tokens量包、TPM包等计费方式，两者互不影响。

套餐可以生成多个APP Key吗?

不可以。每个天翼云账号仅支持一个专属APP Key

再次购买编程Token Plan套餐后APP Key会变化吗?

不会。再次购买后会维持原套餐使用的APP Key。支持用户在页面手动重置APP Key。

注意：

APP Key重置后，约1min后生效，且需要将新的APP Key更新到所有使用的工具中，否则将会导致工具不可用。

可以使用模型列表以外的模型？

编程Token Plan 当前仅支持套餐内指定的模型，使用其他模型将会报错。套餐内支持模型详见编程Token Plan概述。

编程Token Plan 有并发请求数限制吗？

编程Token Plan 并发速率与您的套餐等级相关，平台会根据资源进行动态调整，基本原则：套餐等级越高并发速率越高。

套餐购买类

每个天翼云账号支持同时开通几个套餐？

同一主账号同时只能购买一个套餐（不区分编程Token Plan/GLM Coding Plan的各版本套餐），已购买【编程Token Plan 2500万】版本套餐得用户，无法购买编程Token Plan其他版本套餐，也无法购买GLM Coding Plan的各版本套餐。目前仅支持主账号发起购买，且购买详情和调用信息仅主账号可见。

子账号如何购买/使用编程Token Plan套餐？

不支持子账号购买/使用套餐，请联系主账号购买。

可以与他人共享一个编程Token Plan套餐吗？

不可以。套餐为购买人专享使用，若因账号共享导致出现多人共用同一套餐的情况，可能会视为不当使用，并在必要时对订阅权益做出相应限制，严重时或将影响账号正常使用。

编程Token Plan套餐是否支持团队使用?

套餐主要面向个人开发者。团队协作使用，请使用Token按量付费方式接入。

编程Token Plan套餐暂时售罄，什么时候可再次购买?

如有库存，于每日10:00定量释放。

怎么查看编程Token Plan套餐订单详细信息?

前往费用中心，点击“订单管理 > 我的订单”，进入我的订单页面。操作参见查看订单详情。

如何开启编程Token Plan套餐续订?

目前不支持套餐的续费能力，包括手动续订和自动续订。套餐过期失效后手动再次购买。自动续订和手动续订能力即将支持，敬请期待。

套餐支持退款吗?

不支持。编程Token Plan套餐一经购买成功即视为确认，不支持退订/退款。即使未使用套餐，费用也无法退回。建议根据使用需求选择合适的套餐。套餐不参与七天无理由全额退订。

计量计费类

如何查看用量?

在编程Token Plan页面可查看套餐的用量详情。注意：缓存命中 / 未命中的输入内容、输出内容所消耗的Token均计入套餐包抵扣范围。

编程Token Plan套餐用量如何重置?

套餐额度在每个订阅月到期时重置，未用完的额度不累积到下月。

编程Token Plan套餐当月额度已全部耗尽，怎么办?

方案一：等待该套餐失效后再次购买套餐。

方案二：开通Token按量、Token量包等服务。更换Token按量、Token量包对应APP Key和Base URL后使用，不会自动消耗Token按量/Tokens量包等资源，也不会扣除账号余额。

编程Token Plan套餐内可以使用模型免费试用额度吗?

套餐是一个独立的计费产品，其计费和配额系统不参与模型免费试用额度。

为什么购买了编程Token Plan套餐还有模型 API 调用费用产生（或导致欠费）？

可能有以下原因：

未正确配置专属 APP Key 和专属请求地址（最常见原因）

原因：如果在 AI 工具中配置的是通用 APP Key和通用请求地址（无coding），系统会将其识别为按量计费调用，产生按量计费的账单。

解决方案：请务必使用套餐专属配置。APP Key 的格式必须为cp_xxx，请求地址必须包含 coding 关键字。

账单结算延时导致欠费（费用产生于购买套餐前）

原因：账单按小时出账，存在延迟。 例如 16:00 产生的 API 调用，可能延至购买套餐后才出账扣费。

解决方案：以实际出账时间为准，详情请参见我的订单。

客户端缓存未清理

原因：配置新凭证后，部分工具可能仍在读取旧的通用API调用凭证缓存。

解决方案：清理缓存并重启。再根据对应的接入AI工具指引重新配置。

4.2 Token Plan

4.2.1套餐介绍

套餐概述

开发者 / 中小企业版Token Plan：基于 GLM-5 大模型能力，提供标准化Token套餐。覆盖代码开发、复杂逻辑推理、长文本处理、智能体搭建等专业需求，接口稳定兼容，满足轻量化业务开发、批量调用、日常项目迭代等高频商用使用场景。

个人 / 家庭版Token Plan：依托 DeepSeek V3.2 通用大模型能力，提供轻量化Token 套餐。适配日常办公辅助、学习创作、生活咨询、文案撰写、常识问答等全场景个人需求，性价比适配个人长期 AI 使用。

说明

已了解 Token Plan 的用户，直接参见快速开始开启 AI 编码。

套餐优势

套餐按需适配：提供不同版本的套餐，适应不同强度的 API 调用和编程场景。

兼容主流工具：支持 OpenCode、OpenClaw、Cline、CodeBuddy 等主流编程工具，套餐额度共享，灵活适配多种开发场景。

费用更节省：同模型套餐价相比于直接调用 API 服务价格更便宜。套餐额度越多，百万 Token 单价越低。

套餐详情

套餐类型

	开发者 / 中小企业版 Token Plan（前往购买）			**个人 / 家庭版 Token Plan（前往购买）**		
适用场景	覆盖代码开发、复杂逻辑推理、长文本处理、智能体搭建等专业需求，接口稳定兼容，满足轻量化业务开发、批量调用、日常项目迭代等高频商用使用场景。			适配日常办公辅助、学习创作、生活咨询、文案撰写、常识问答等全场景个人需求，性价比适配个人长期 AI 使用。		
套餐额度	1500 万 tokens	7000 万 tokens	1.5 亿 tokens	1000 万 tokens	4000 万 tokens	8000 万 tokens

套餐价格	39.9 元/个	159.9 元/个	299.9 元/个	9.9 元/个	29.9 元/个	49.9 元/个
支持模型	GLM5（正式版）			Deepseek v3.2（旗舰版）		

受限于资源紧张，建议大家错峰使用。
注：高峰期为每日的 9:00～18:00

套餐有效期说明

量包激活当日生效，按量包标注有效期计算到期日。若到期月无对应日，则该月最后一天到期。多个量包被激活后，后续激活量包将在上一量包用完或到期后自动接续开始生效，到期规则同上。

例如：量包有效期为一个月，开始生效日期为5月1日10:00:00，到期内为6月1日10:00:00；开始日期为5月31日10:00:00，到期内为6月30日10:00:00。

量包消耗与抵扣规则

量包需激活后使用，未激活量包无法抵扣。若同时存在多个已激活量包，系统优先扣除到期时间最早的量包。

购买量包后请尽快至服务详情页激活，以免影响业务；已开通自动激活的用户，请关注量包生效时间，避免资源闲置。

额度耗尽即停服：量包额度耗尽后，服务将自动停止。若需服务持续运行，建议开通“Token按量付费”，量包用完或到期后可无缝切换按量付费。

套餐使用须知

账号购买限制：每个天翼云账号在每个模型可同时存在2个使用中/未激活/购买中Token Plan。如果账号内没有使用中/未激活/购买中的量包，可购买2个。

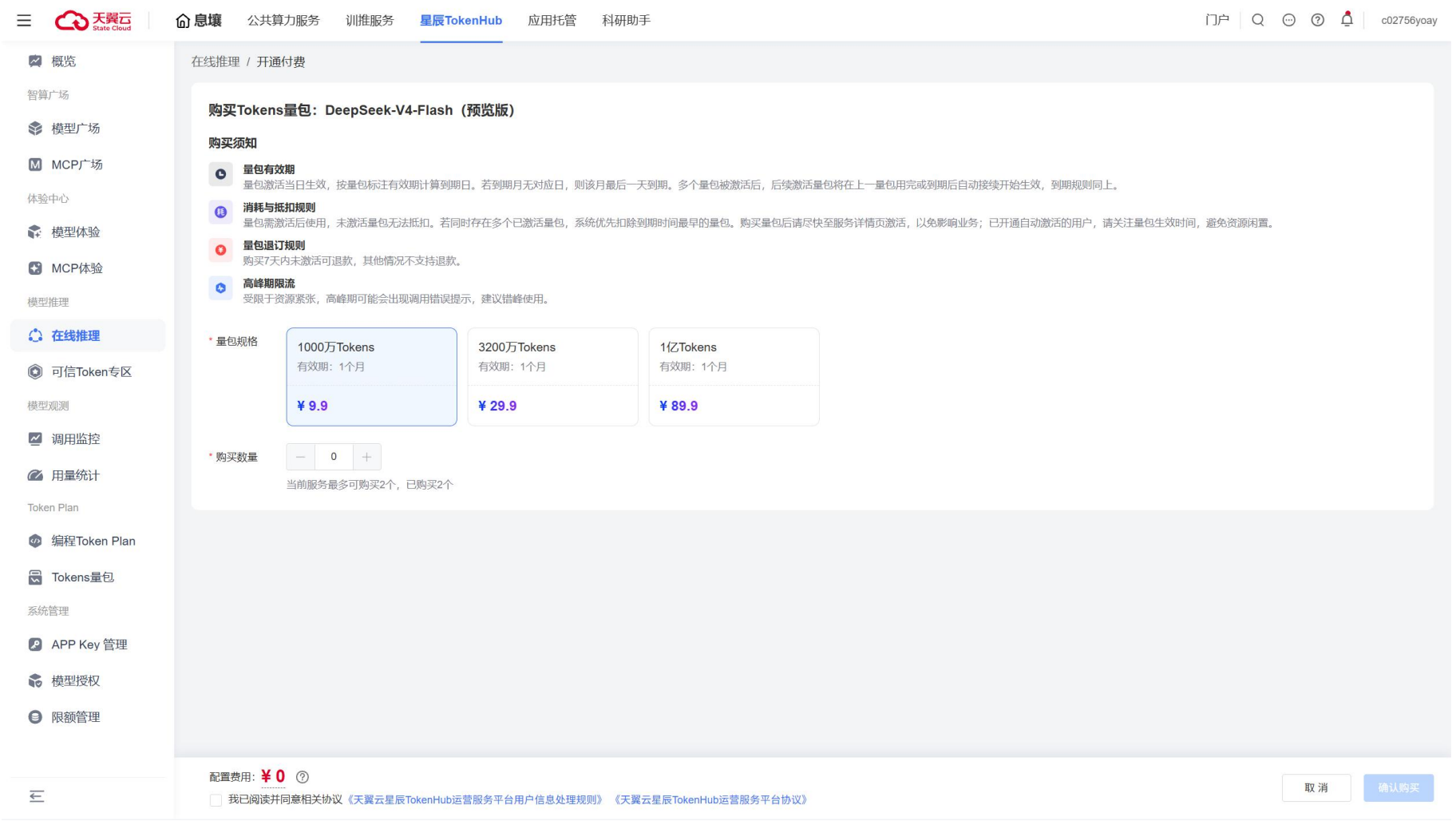
套餐速率上限：Token Plan量包服务速率限制与Tokens按量付费模式相同，每个模型数值不一，详细可参见模型广场->对应模型详情页，或者模型推理->预置服务->对应模型的服务详情页。

套餐退款规则：Token Plan套餐包在购买7天内未激活可退订，其余情况不支持退订/退款。建议根据使用需求选择合适的Token Plan套餐包。套餐不参与七天无理由全额退订。

4.2.2快速开始

第一步：购买 Token Plan 量包

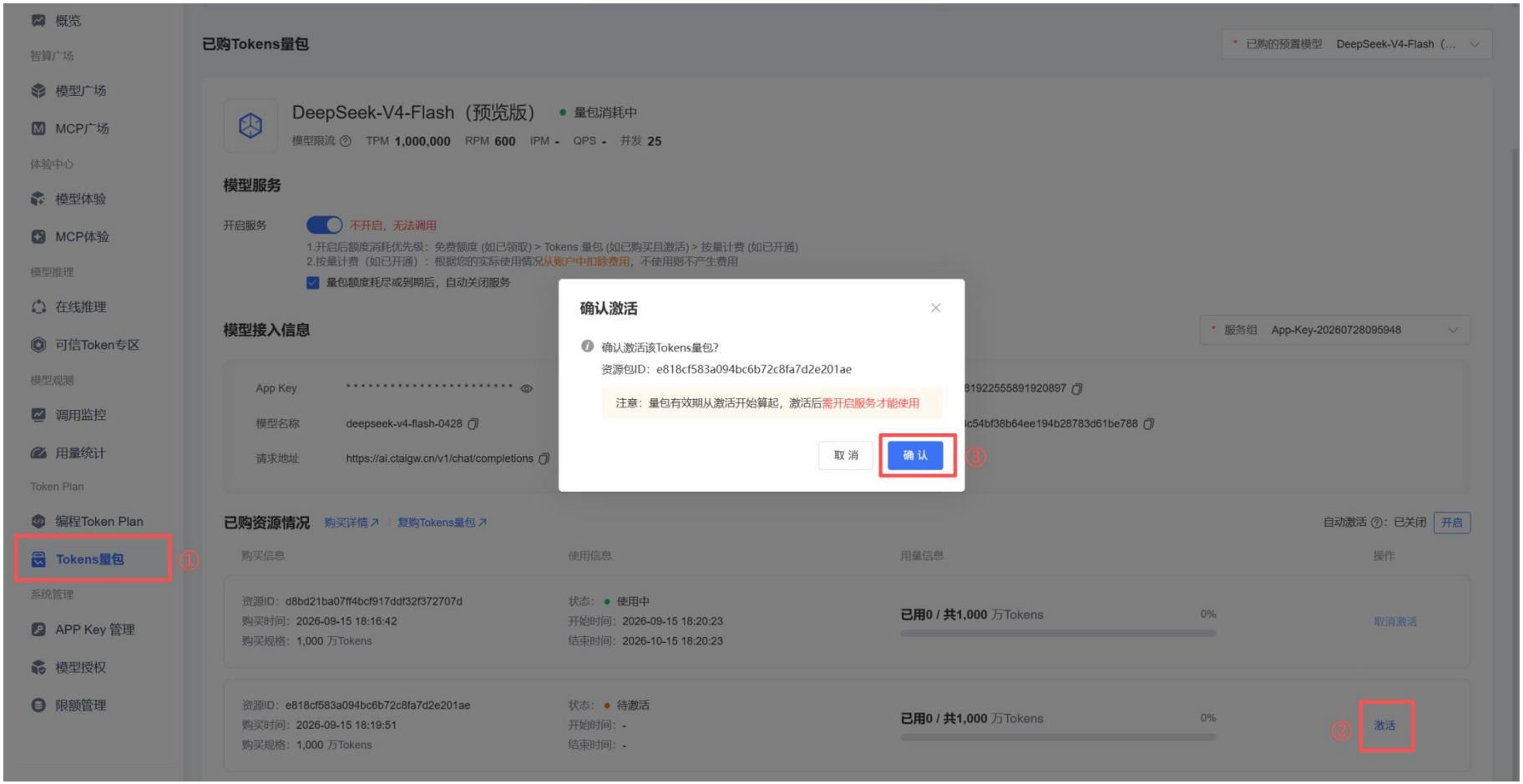
登录模型推理服务控制台，选择“模型推理->预置服务”，选择支持Token Plan服务的模型名称，点击操作列的“购买量包”进入购买。支持Token Plan服务的模型参见包周期计费模式-Tokens量包。



第二步：激活 Token Plan 量包

方式一：

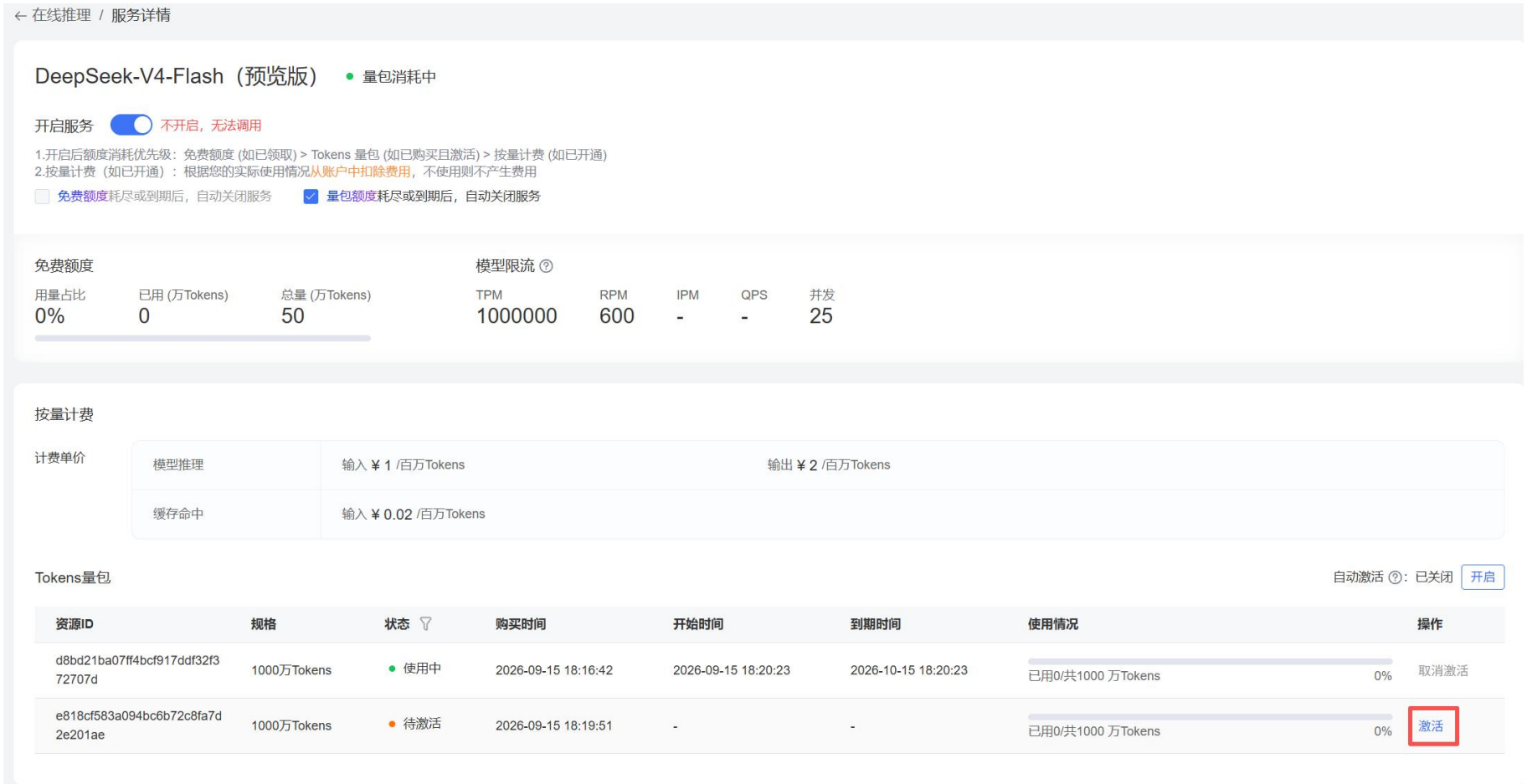
进入Tokens量包页面，在“开通资源情况”中找到对应量包，点击“激活”并确认。



方式二：

在“模型推理->在线推理->预置服务”中找到对应服务，点击服务名称进入详情页激活Token Plan量包。





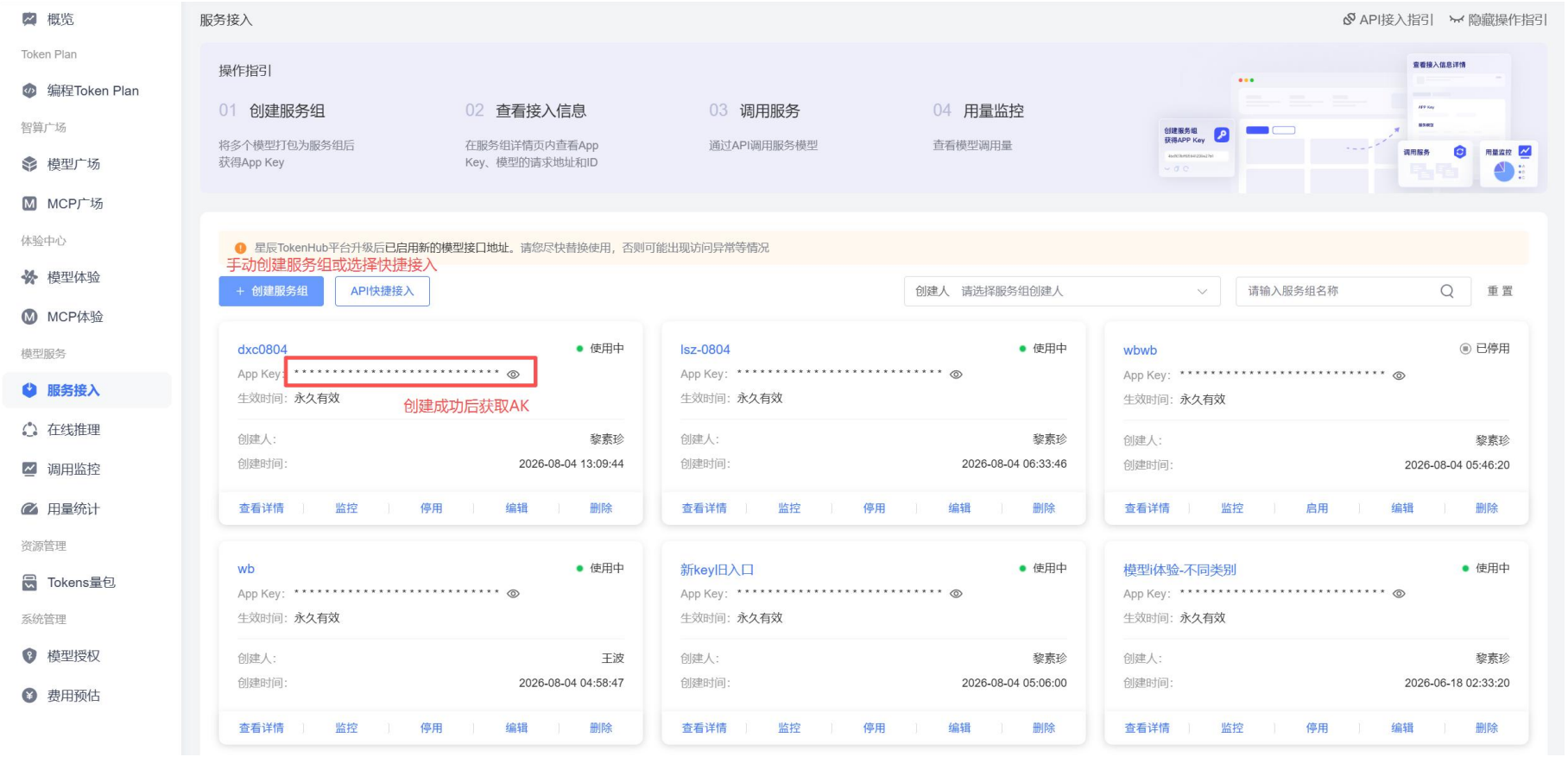
第三步：获取 APP Key 和请求地址 BaseURL

APP Key： “控制台-系统管理-APP Key管理”配置相应模型服务创建服务组，获取APP Key。

BaseURL：

OpenAI协议：https://ai.ctaigw.cn/coding/v1/chat/completions

（暂不支持Anthropic协议，可使用相关插件进行转换，具体步骤详见接入工具）



第四步：开启调用

通过API调用或接入AI工具后使用。

4.2.3常见问题

产品功能类

Token Plan购买成功即刻生效吗?

需要手动激活使用。为了不影响业务使用，购买量包后，请尽快前往服务详情页激活服务；若同时激活几个量包，请注意量包生效时间，避免资源浪费。

Token Plan量包有效期如何计算?

量包激活当日生效，按量包标注有效期计算到期日。若到期月无对应日，则该月最后一天到期。多个量包被激活后，后续激活量包将在上一量包用尽或到期后自动接续开始生效，到期规则同上。

例如：量包有效期为一个月，开始生效日期为5月1日10:00:00，到期内为6月1日10:00:00； 开始日期为5月31日10:00:00，到期内为6月30日10:00:00。

如何查看Token Plan量包消耗信息?

登录模型推理服务控制台，前往模型推理->预置服务，进入对应模型的服务详情页可查看。在该页面也可同步查看容量和速率限制信息等。

注：缓存命中 / 未命中的输入内容、输出内容所消耗的Token均计入套餐包抵扣范围

如何获取Token Plan量包的APP Key和BaseURL?

APP Key：前往控制台->系统管理->APP Key管理，点击“创建服务组”并配置相应模型即可获得APP Key。支持用户在页面手动重置APP Key。

URL：https://ai.ctaigw.cn/coding/v1/chat/completions

说明
APP Key 重置后，约 1min 后生效，且需要在使用时将 APP Key 更新为最新的。

Token Plan量包的速率限制是多少?

Token Plan量包服务速率限制与Tokens按量模式相同，每个模型数值不一，详细可参见模型广场->对应模型详情页，或者模型推理->预置服务->对应模型的服务详情页。

套餐购买类

每个天翼云账号可同时购买几个Token Plan量包?

每个天翼云账号在每个模型可同时存在2个使用中/未激活/购买中Token Plan。如果账号内没有使用中/未激活/购买中的量包，可购买2个。

子账号如何购买/使用Token Plan量包?

暂不支持子账号购买，请联系主账号购买Token Plan量包后即可在控制台->模型推理->服务详情页内查看。

怎么查看Token Plan量包类的订单详细信息?

前往费用中心，点击“订单管理 > 我的订单”，进入“我的订单”页面。操作参见查看订单详情。

量包暂时售罄，什么时候可再次购买?

为保障用户体验，如有库存，于每日10：00定量释放库存。

如何退订Token Plan量包?

Token Plan量包在购买7天内未激活可退订，其余情况不支持退订/退款。建议根据使用需求选择合适的Token Plan量包。

计量计费类

量包激活后抵扣顺序是怎么样的?

量包需激活后使用，未激活量包无法抵扣。若同时存在多个已激活量包，系统优先扣除到期时间最早的量包。

购买量包后请尽快至服务详情页激活，以免影响业务；已开通自动激活的用户，请关注量包生效时间，避免资源闲置。

额度耗尽即停服：量包额度耗尽后，服务将自动停止。若需服务持续运行，建议开通“Token按量付费”，量包用完或到期后可无缝切换按量付费。

购买Token Plan量包后，还可以使用模型免费额度吗?

可以的。量包未激活时可领取免费额度并优先消耗免费额度，免费额度耗尽后，自动消耗已激活的量包。

套餐额度耗尽后，可否继续使用token按量服务?

可以。如果未开启量包用完即停，当Token量包用完后，会消耗按量付费资源。如果不想继续使用按量付费服务，可勾选量包用完即停按钮。

4.3 智算广场

4.3.1模型广场

模型广场汇集了当下丰富的开源大模型，可查看及体验模型在文本、向量、图像等方面的先进能力。模型广场包含文本大模型、视觉大模型、向量化模型，后续还会支持更多种类的大模型，敬请期待。

操作步骤

登录星辰TokenHub运营服务平台。

在左侧导航栏选择“模型广场”查看平台上架的所有模型。

可对模型按照实际需求进行搜索和筛选。

点击目标模型卡片进入模型详情页，可查看模型介绍、API文档和任务记录。

点击“任务记录”，可查看该模型的任务历史。

在模型卡片点击“部署”，可跳转至“模型推理”-“模型部署”页进行部署信息的填写与配置，当按钮置灰时，表示模型暂不支持该功能。

在模型卡片点击“体验”，可跳转至“体验中心”-“模型体验”页体验目标模型的能力，当按钮置灰时，表示模型暂不支持该功能。

4.3.2MCP广场

MCP（Model Context Protocol 模型上下文协议）是大模型调用本地文件、数据库、各类外部工具时所采用的统一标准化通信协议，做到一次开发多AI通用接入。MCP广场聚合了各类优质MCP资源，扩展模型边界，支持体验和部署MCP服务。

操作步骤

登录星辰TokenHub运营服务平台。

在左侧导航栏选择“MCP广场”查看平台上架的所有MCP能力。

可对MCP能力按照实际需求进行搜索和筛选。

点击目标模型卡片进入MCP详情页，可查看MCP介绍。

在MCP卡片点击“部署”，可跳转至应用托管平台进行快速部署，当按钮置灰时，表示模型暂不支持该功能。

在MCP卡片点击“体验”，可跳转至“体验中心”-“MCP体验”页结合模型调用MCP进行对话体验。

4.4 体验中心

4.4.1 模型体验

本文为您展示星辰TokenHub运营服务平台模型体验模块相关操作。

提供在线体验的功能，可即刻体验模型效果。

前提条件

体验预置服务：该模型已开启服务。

操作步骤

登录星辰 TokenHub 运营服务平台。

任选以下方式进行体验。

在左侧导航栏选择“体验中心”。

在左侧导航栏选择“模型广场”，单击目标模型卡片的“体验”按钮。

在左侧导航栏选择“模型推理”，单击预置服务操作列的“体验”按钮。

在“体验中心”页面选择模型类型和模型服务。

文本生成类模型支持调整模型温度、多样性等参数设置。

文本生图类模型支持设置绘画描述、图片风格、图片比列和迭代步数等参数。

参数	说明
温度	Temperature控制生成文本的多样性。较高的温度值会使生成的文本更加随机和多样化，而较低的温度值会使生成的文本更加确定和一致。
多样性	TopP影响输出文本的多样性，取值越大，生成文本的多样性越强。
重复惩罚	Frequency_penalty影响模型生成重复词汇的倾向。通过增加重复词汇的惩罚权重，降低模型逐字重复的可能性。
TopK	TopK指在模型的输出中选择概率最高的前K个结果。有助于加快生成过程，并且能够提高生成文本的质量。

系统人设	设定模型的行为和背景，告知模型需要扮演的角色。例如：“假如你是一个AI助手”。

在对话输入问题即可查看返回结果。

点击右上角“历史记录”，系统会展开历史的对话记录，可以查看统计大模型的回答质量，保存上限为 200 条、时限为 180 天。

4.4.2 MCP体验

提供在线体验 MCP，可即刻体验模型调用 MCP 服务的效果。

前提条件

支持调用 MCP 的模型服务具有免费 Token 额度且在有效期内，或已开通商用。

操作步骤

登录星辰 TokenHub 运营服务平台。

任选以下方式进行体验。

在左侧导航栏选择“MCP 体验”。

在左侧导航栏选择“MCP 广场”，单击目标 MCP 卡片的“体验”按钮。

选择 MCP 服务。模型提供部分 MCP 服务的在线体验能力，根据需要任选一个。或者通过输入个人 SSE 链接的方式获取其他 MCP，操作参考 获取个人级 SSE 链接。

选择模型服务。在支持调用 MCP 服务的模型中任选一个。

获取个人级 SSE 链接

说明：SSE 链接是个人专属链接地址，为敏感信息，请勿对外泄露

个人级SSE链接需前往应用托管部署对应MCP服务，部署成功后直接复制应用托管平台的SSE链接至MCP体验页面，或者复制公网IP和密钥自行组合成为SSE链接（示例：http://{内网IP:Port}/sse?token={API-KEY}），保存配置后进行体验。

4.5 模型推理

4.5.1 模型推理

本文为您介绍星辰 TokenHub 运营服务平台模型推理模块。

平台模型推理模块提供预置服务，预置服务是平台已部署好的服务，无等待时间，可直接调用。

预置服务

操作步骤

登录星辰TokenHub运营服务平台。

在左侧导航栏选择“模型推理”，在右侧页面中选择“预置服务”。

主账号下在预置服务列表可查看目前平台预置的所有服务内容和对应的服务信息（服务名称、开启服务状态、使用状态、价格、关联服务组，子账号下在预置服务列表中仅可见主账号所授权的对应服务内容。

所有预置服务需要主账号“开启服务”后才可使用，支持免费试用和付费使用。

服务开启后可请求调用模型服务，并可在“调用监控”中查看对应服务的调用情况。

开启服务

开启服务：开启对应服务后才可请求模型服务。

仅主账号下有此权限。

在开启服务弹窗中可查看开启后的计费规则，点击“确认开启”则开通对应服务。

关闭服务

关闭已开启且正在使用中（含状态：按量计费中、免费试用、量包消耗中）的服务，仅主账号下有此权限。

点击“关闭服务”将弹窗展示关闭服务后的后续影响：模型微调服务停止、按量计费停止、免费体验额度（若有）失效、子账号无法调用该服务，在确认后续影响后点击“确认关闭”，完成服务关闭。

免费体验：

每个模型被赋予不等的免费额度和免费试用期限，在开启的服务后点击“开启免费体验”可进行弹窗查看，仅主账号有此权限。

可选择是否勾选“免费额度耗尽后或到期后，自动关闭服务”，若不勾选则免费体验结束后将对服务进行按量计费。

点击“确认开启”后当前服务的使用状态变更为“免费试用”。

免费试用期限从领取成功后开始计算。

免费体验具有以下特点：

共享机制：主/子账号共用免费额度及有效期周期额度。

不可分配：不支持主账号向子账号分配额度。

有效期触发：主账号任一账号首次使用模型服务即启动2周有效期倒计时。

说明：

若对应服务存在已经购买了 Token 量包，则 Token 量包不可与免费额度同时使用，若 Token 量包已经在使用中，则“开启免费体验”禁用，请在量包使用完毕后再开启免费体验。

购买量包

用户可根据实际开通服务的 tokens 消耗情况决定是否购买量包。

未配置销售品的服务下“购买量包”暂不可用。

tokens量包与免费体验不可同时使用，若当前服务处于免费试用中，“购买量包”不可用。

除外上述情况下点击对应服务的“购买量包”进入到购买量包详情页，仅主账号下有此权限。

在详情页查看购买须知，根据实际情况选择合适的量包规格和购买数量（相同服务最多可购买2个）。

根据实际情况选择是否开启服务（建议开启，若不开启则后续无法调用服务）、选择是否勾选“量包额度耗尽或到期后，自动停止服务”，查看对应的计费单价。

购买量包具有以下特点：

购买限制：只允许主账号开通tokens量包，主账号及所有子账号共用资源。

购买记录：主账号可在订单管理处查询订单，主账号可在模型服务详情页和“Tokens量包”查看用量。

用量统计：各账号独立统计调用监控，主账号通过监控详情页筛选可见子账号用量。

统一扣费：主账号扣费。

计费策略：支持开通token量包的模型及对应价格参见包周期计费模式-Tokens量包。

举例：主账号A已开通Tokens包，子账号B、C、D可共享使用。

更多操作

点击“API文档”跳转到对应服务的模型详情页。

点击“调用监控”跳转到对应服务的监控服务详情页。

4.5.2 可信Token专区

本文为您介绍星辰TokenHub运营服务平台可信Token专区的能力。

可信Token专区面向高数据安全要求的业务场景，提供可直接调用的模型服务。专区基于自研 AI 可信计算（TEE）、云间高速与全链路数据加密传输技术部署模型，底座软硬件及推理框架全栈适配信创标准，覆盖接入、网络、数据、内容全链路防护，满足高等级数据安全与权限管控要求。

可信 Token 专区优势

安全可信：专属国产可信底座，内置数据加密传输，在硬件、软件与推理框架多层构建纵深防护体系，模型调用全程安全可控。

接入便捷：兼容现有调用方式，开通服务后仅需更换 Model 参数和BaseURL即可完成切换；机密可信路径与非机密可信路径支持在同一账号下并存，按业务敏感度灵活选用。

应用场景

国资监管场景

适用：面向国资委监管要求，依托大模型能力实现国资穿透监管，支持权属企业数据聚合与资产异动监控。

优势：覆盖产权管理、审计辅助及三重一大决策辅助等核心场景，通过AI辅助提升监管效率与合规水平，助力央企构建全链路监管体系

财务与业财场景

适用：围绕央企财务管理全流程，提供智能费用报销、财务分析、合同审价、资金管理及税务合规等AI赋能服务

优势：通过大模型打通行业与内部数据，实现穿透式经营分析，提升财务自动化水平与经营决策效率，全面推动央企财务管理数智化转型

企业知识库构建

适用：基于可信Token专区，为企业提供安全可控的知识库构建服务。支持企业制度规范、技术文档、历史案例等内部知识的结构化录入与智能检索，通过大模型实现自然语言问答与知识推理

优势：所有数据部署在中国电信自有机房，确保企业核心知识资产不出域、可用不可见

生产过程敏感数据处理

适用：针对涉及加密信息的生产处理场景，提供基于机密推理的安全计算环境。将安全边界下沉至硬件内核，在可信域内完成敏感数据的推理计算，通过远程证明机制确保运行环境完整性。

优势：对于涉密生产流程参数处理、加密业务数据分析等高安全要求场景，满足数据不出域与可用不可见的双重合规要求。

计费说明

可信Token专区内模型为按量计费，不支持使用免费额度、Tokens量包等。

模型名称	缓存未命中输入（元/百万Tokens）	缓存命中输入（元/百万Tokens）	输出（元/百万Tokens）
AICI-Deepseek-V4-Flash-0731	3（空闲）	0.3（空闲）	9（空闲）
	6（高峰）	0.6（高峰）	18（高峰）
AICI-GLM-5.2	16	4	56

注：高峰时段为北京时间8:00- 22:00，其余为空闲时段

使用前提条件

可信Token专区内模型为行业专属服务面向符合条件的央国企客户开放，如需使用，请联系客户经理开通或提交业务需求单咨询。

操作步骤

1. 登录星辰TokenHub运营服务平台。

在左侧导航栏选择“可信Token专区”，主账号可选择模型点击“开启服务”。子账号需联系主账号在“模型授权”授予模型调用权限后才可查看。模型授权操作参见用户指南。

服务开启后参照“模型广场”详情页调用方式或API文档可请求模型推理服务。

说明：
可信 Token 专区内模型请求使用请求地址：
OpenAI 兼容：https://tokenhub-trust-zone.ctaigw.cn/v1/chat/completions
Anthropic 兼容：https://tokenhub-trust-zone.ctaigw.cn/v1/messages

调用后可前往“调用监控”查看对应服务的调用情况。

查看远程验证报告

远程验证报告是基于原始报告和远程证明策略对比后的结果。进入“控制台->可信Token专区”，点击操作列“远程验证报告”即可下载机密推理的远程验证报告。当报告中"status"和"attestation_status"字段为"pass"时代表推理环境是可信的。

说明：

已开放专区访问权限的用户支持下载已加白模型的远程验证报告。

远程验证报告部分内容释义：

```
{
    // ==================== 策略匹配结果 ====================
    "matched_policy" : [ {              // 匹配成功的评估策略列表
        "id" : "f6b32b37-****-c1d6b504f063",// 策略唯一标识 UUID
        "sha256" : "62e5a2f8e9b798df********397749dd704c7d0b3a",// 策略内容的 SHA256 哈希，用于验证策略完
整性
        "version" : 1               // 策略版本号
    } ],
     // ==================== JWT 标准声明字段 ====================
    "nbf" : 1789031055,             // 令牌生效时间（UNIX 时间戳），在此时间之前无效
    "status" : "pass",               // 整体证明状态，pass 表示全部验证通过
    "eat_profile" : "eat_profile",    // EAT 配置文件标识，声明遵循的 EAT 规范
    "exp" : 1789631055,             // 令牌过期时间（UNIX 时间戳），有效期约 7 天
    "iat" : 1789031055,             // 令牌签发时间（UNIX 时间戳）
    "iss" : "iss",                   // 令牌签发者，标识全球信任权威
    "jti" : "8eeeba36-****-fb63abeff1db",// 令牌唯一 ID，用于防重放
      // ==================== 子模块证明结果（核心） ====================
    "tpm_boot" : {
        "attestation_status" : "pass",// 证明状态，affirming 表示验证通过且确认可信
        "log_status" : "no_log",
        "pcrs" : {              // PCR 寄存器验证结果
            "hash_alg" : "sha256", // PCR 扩展使用的哈希算法
            "pcr_values" : [ {      // PCR0~PCR9 各寄存器的验证详情
                "pcr_index" : 0,    // PCR 索引 0：CRTM/BIOS 固件代码
                "pcr_value" :
"d5715a00ac596d4cf************d2351d6f1dcc8"// 设备 TPM 报告的 PCR 值
```

4.6 模型观测

4.6.1 调用监控

本文为您介绍星辰TokenHub运营服务平台调用监控模块。

支持查看模型推理预置服务、专属服务在指定时间段内的调用数据和监控指标详情，并支持按日、按时、按分钟粒度展示数据，帮助了解服务的使用情况和性能变化。

前提条件

筛选条件范围内存在调用记录后才可查看调用监控数据。

账号权限说明

主用户可查看其名下所有子账号分别的调用监控数据，且支持导出所有子用户的所选服务的监控数据。

子用户仅可见自身调用监控数据。

查看服务调用的监控数据

登录星辰TokenHub运营服务平台。

在左侧导航栏选择“调用监控”。

选择目标服务名称，点击“调用监控”，进入该模型的调用数据详情页。

在“调用总览”可以查看具体的模型在特定服务组、特定服务中的调用监控情况。

说明：

主/子用户：可通过主/子用户下拉框筛选指定用户的调用监控情况。

服务名称：可通过服务名称下拉框筛选指定模型的调用监控情况。

服务组：可通过服务组下拉框筛选所选服务名称所关联全部服务组（AppKey）的调用监控情况。

时间精度：支持按分钟、按时、按日三种精度查询。

时间范围：支持查询近31天模型调用情况。时间范围会根据所选时间精度自动纠正，请先确认查询的时间精度。时间精度为按分钟时，支持查询时间跨度最大为1天的数据；时间精度为按时时，支持查询时间跨度最大为7天的数据；时间精度为按日时支持查询时间跨度最大为31天的数据。

点击“调用失败明细”，可以查看调用失败的错误码、次数、占比等数据。

点击“导出”，可以直接导出调用总览或调用失败明细数据到本地。

选择目标专属服务，点击“资源监控”，进入可查看专属服务资源的监控信息，包含GPU/NPU利用率、显存利用率、

表1 调用监控详情页筛选项说明

筛选项	说明

主/子用户	筛选子用户。 仅名下存在子用户的主用户可见。
服务名称	支持选择模型推理的预置服务和我的服务。
服务组	选择查看全部服务组或具体的服务组名称。
时间精度	支持选择按日、按时、按分钟。
时间范围	自定义选择时间范围。 当时间精度选择按日时，范围最大支持 30 天且仅最小粒度选择到日；当时间精度选择按时时，范围最大支持 7 天；当时间精度选择按分钟时，范围最大支持 1 天。

表2 调用总览的指标说明

参数	说明
调用 Token 量（万 tokens）	服务调用的 tokens 数，包括总量、输入量、输出量。支持细分筛选不同计费类型下的调用量。

调用次数	服务调用的次数。
调用失败率（%）	调用失败次数/调用总次数。
平均单次请求调用量（万 tokens）	平均单次请求输入输出 Token 量，为间隔时间内的调用 Token 量/成功调用的次数
平均响应时延（ms）	单位时间内成功请求的响应时间平均值。
首 Token 时延（ms）	仅统计流式响应。 从接收请求到生成第一个输出 token 所需的时间。
非首 Token 时延（ms）	仅统计流式响应。 生成后续每个输出 token 所需的时间间隔，不包括首 token。
RPM	表示在一分钟内，系统接收到的请求数量
TPM	表示在一分钟内，通过所有请求处理的 Token 总数
平均单次请求调用量	平均单次请求输入输出 Token 量，为间隔时间内的调用 Token 量/成功调用的次数。
QPS	Query Per Second，即每秒响应请求数。

表3 调用失败明细的参数说明

参数	说明
错误码	报错的错误码。
错误次数	错误发生的次数。
错误占比	该错误码发生次数占全部错误次数的比例。

表4 资源监控详情页的指标说明

参数	说明
NPU/GPU 使用率	在单位时间内，显卡被使用的时间占比。
NPU/GPU 显存使用率	已用显存占总显存的百分比。
CPU 使用率	CPU 在单位时间内，CPU 被使用的时间占比。
内存使用率	已用内存占总内存的百分比。

4.6.2 用量统计

本文为您介绍星辰TokenHub运营服务平台用量统计模块。

支持查看指定时间段内的模型按需计费的用量（不包含免费额度与量包用量），支持按计费方式、账号、服务组多维度筛选，支持按日、按时、按分钟粒度展示数据，帮助了解模型的使用情况。

前提条件

筛选条件范围内存在调用成功的记录后才可查看到用量。

账号权限说明

主用户可查看其名下所有子账号和服务组的模型用量情况。

子用户仅可见本账号调用的模型用量情况。

查看模型用量数据

登录星辰TokenHub运营服务平台。

在左侧导航栏选择“用量统计”。

选择计费方式和起止时间范围后，即可查看该计费方式下额所有模型调用汇总趋势图和模型调用明细数据。也可以筛选具体的账号和服务组查看具体的用量。

说明：

计费方式：可通过计费方式下拉框筛选指定计费方式下模型各调用产生的用量及各模型用量明细。

主/子用户：可通过主/子用户下拉框筛选指定用户调用产生的用量。

服务组：可通过服务组下拉框筛选指定服务组（App Key）调用产生的用量。

时间精度：支持按分钟、按时、按日三种精度查看。

时间范围：时间范围会根据所选时间精度自动纠正，请先确认查询的时间精度。时间精度为按分钟时，支持查看近31天内时间跨度最大为1天的数据；时间精度为按时时，支持查看近180天内时间跨度最大为7天的数据；时间精度为按日时支持查看近180天内时间跨度最大为31天的数据。

支持导出明细查看。

4.7 系统管理

4.7.1 APP Key管理

本文为您介绍星辰TokenHub运营服务平台模型APP Key管理模块。

APP Key管理（原服务接入）模块可以将模型服务打包为服务组，生成AppKey供用户调用。

创建服务组

登录星辰TokenHub运营服务平台。

在左侧导航栏选择“APP Key管理”。

点击“创建服务组”，进入创建页。

按要求依次填写相关服务基本信息（服务组名称、服务组描述）、设置服务生效时间（永久有效、自定义有效期）、IP访问权限、额度控制和性能控制。

若开启额度控制，点击“编辑配置”拉起右侧抽屉弹窗进行额度的限额周期（天、自然月、自然年）、消费限额、Token限额、告警阈值、发送通知的配置操作并点击“保存”完成限额配置。

额度控制支持添加（最多三个）和删除限额周期配置，且每个周期配置中消费限额和Token限额至少设置一个。

若开启性能控制，则完成TPM、RPM、IPM、QPS和并发要求的设置填写。

根据实际情况选择对应的服务配置：全部服务或自定义选择，所有服务均包含对应的预置服务（服务状态为“已开启”）和专属服务，一个服务组最多可选择100个服务。

主账号下可查看和选择平台全部预置服务（含可信专区内模型）+专属服务（含后续新上线的“预置服务”或专属服务），子账号下仅可查看和选择主账号授权下的预置服务（含可信专区内模型）+专属服务（含后续新上线的“预置服务”或专属服务）。

支持对专属服务进行“服务限流”：开启后可设置对应的RPM和TPM阈值，不受“性能控制”中对应参数的影响。

点击“了解更多”可进一步查看对应服务的详情。

基本信息和服务配置设置完毕后，点击“提交”完成服务组创建，系统会自动创建调用服务的密钥，即生成该服务的密钥，即“AppKey”，可在后续详情中查看。

说明：

IP访问权限：

建议选择全部公网IP，若有更精细的权限控制需求，可以指定IP地址访问模型推理服务（一个服务组最多配置100 个IPV4和IPV6的IP地址或网段）。

额度控制：

同一个限额周期内消费限额和Token限额都配置时，优先达到触发阈值的先生效。

超出任一配置限额后，服务组将自动暂停服务，请提前评估业务中断风险。

生效范围：限额不包含免费体验额度、量包抵扣额度；Tokens限额不包含多模态模型；限额对专属服务不生效。

性能控制：

如果存在多条性能配置，性能约束会叠加生效。

预置服务：性能配置值超过某一模型限流值时，超额部分对该模型无效；模型限流值前往“模型推理”-“预置服务”详情页查看。

专属服务：以下TPM、RPM性能配置不对专属服务类型生效，专属服务以服务配置模块「服务限流」配置作为生效值。

查看服务组详情

点击服务组卡片【查看详情】，可进入服务组详情页，查看该服务组关联的详情信息，如：服务名称、生效时间、服务描述、操作说明、额度控制、性能控制、接入信息（App Key、App ID、）和任务记录，点击【了解更多】可查看模型详情。

服务组管理

点击服务组卡片对应按钮，可以支持对“AppKey”进行复制和重置。

监控：可以支持停用/启用、编辑、删除服务组和查看服务组监控

停用：支持手动“停用”和已过生效期后自动停用。

启用：重新启用已停用的服务组。已过生效期的服务组可一键启用为永久生效服务组，或通过“编辑”延长生效时间后启用。

编辑：编辑服务组名称、描述、生效时间、IP访问权限、额度控制、性能控制和服务配置。

删除：删除整个服务组。

账号权限说明

主账号可查看、筛选和管理主账号+子账号所创建服务组。

API快捷接入

登录星辰TokenHub运营服务平台。

在左侧导航栏选择“APP Key管理”。

点击“API快捷接入”，拉起右侧抽屉弹窗。在步骤“获取APP Key”中选择创建新的服务组或选择已有服务组。

若选择创建新的服务组，则点击“确认创建并选择”后系统将会按照默认组名称创建新组并自动生成App Key，并将App Key和默认模型ID自动填入到下方的示例代码中。

若选择已有服务组，则点击“确认选择”后系统将该服务组对应的App Key和默认模型ID自动填入到下方的示例代码中。

示例代码支持Python、Java、Golang、PHP、Nodejs和cURL语言，用户可根据实际情况选择相应的示例代码一键复制到自己的项目代码中进行调用。

发起调用后，用户可在“控制台”-“模型推理”中查看模型服务的调用监控、用量等信息。

4.7.2 模型授权

本文为您介绍星辰TokenHub运营服务平台模型授权模块。

模型授权（原权限管理）支持主用户授权给指定子用户指定模型的调用权限，子用户只可调用被主用户授权的模型，可实现对子用户调用某些模型的个性化管理。

前提条件

当前用户属性为主用户。

子用户不可见该菜单。

账号权限说明

主用户可筛选名下所有子用户。子用户数据来源于官网账号中心（IAM），如无法查询请确认子用户在IAM是否被禁用。

操作步骤

登录星辰TokenHub运营服务平台。

在左侧导航栏选择“模型授权”。

点击“添加授权子用户”进入创建页，选择授权子用户及模型提交完成创建。创建采用增量新增规则，若添加中包含已添加过的账号、账号已授权模型不会变动，仅新增未添加的账号及账号未授权的模型。

完成创建后，主用户可在列表页查看子用户被授予调用权限的模型信息，子用户在“模型推理”模块可见被授予权限的模型，未被授权的模型在该模块不可见。

修改授权

在“模型授权”列表页点击“修改”可调整该子用户的授权模型范围。“提交”后立即生效。

删除授权

在“模型授权”列表页点击“删除”可删除该子用户的模型调用权限。

注意：
删除后该子用户已被授权的模型将不可调用，APP Key 管理配置也将同步失效，请谨慎操作。

4.7.3 限额管理

本文为您介绍星辰TokenHub运营服务平台限额管理模块。

限额管理支持主用户管理主/子用户的服务调用额度控制，可从全局级别、账号级别和服务组级别配置多条限额规则（**消费限额或Token限额**），允许同时存在多层模式下的多条限额规则。在用户调用服务过程中，系统将依次校验各模式下全部限额规则，当触发任一规则的阈值时，都将停止对应用户下的对应服务调用。

前提条件

当前用户属性为主用户。

子用户不可见该菜单。

账号权限说明

主账号可查看其名下所有账号的用户信息和服务组信息，且支持对所有账号添加、修改和删除账号限额规则和服务组限额规则。

全局限额

场景说明

主用户可对所有用户下的所有服务组进行统一周期（日、自然月、自然年）下的合计调用额度限制（**消费限额或Token限额**），当满足所设置的限额告警阈值时可自动发送邮件、短信告知主用户，并在达到限额时停止**所有用户**下的服务调用。

示例：主用户想对所有用户（**1个主用户+9个子用户**）下的所有服务组（**合计10个服务组**）按照**每月总消费金额合计1000元**或**每月消耗总Token量合计1000000**进行额度控制，并希望在总消费金额或总Token消耗量**达到90%**时自动发送告警通知，当满足达到任一限额条件时**停止**所有用户下的服务调用。

配置步骤

主用户登录星辰TokenHub运营服务平台。

在左侧导航栏选择“限额管理”。

在“全局限额”卡片点击“配置全局限额”拉起右侧抽屉弹窗进行全局额度的限额周期（天、自然月、自然年）、消费限额、Token限额、告警阈值、发送通知的配置操作并点击“提交并生效”完成限额配置。

全局限额配置支持添加（最多三个）和删除限额周期配置，且每个周期配置中消费限额和Token限额至少设置一个。

配置说明：
全局限额：为**主用户及其全部子用户下的所有服务组合计额度**配置额度上限，执行额度管控
消费限额和 Tokens 限额至少需配置一项，可同时配置；若同一个限额周期内都配置时，优先达到触发阈值的先生效
超出任一配置限额后，服务组将自动暂停服务，请提前评估业务中断风险
生效范围：限额不包含免费体验额度、量包抵扣额度；Tokens 限额不包含多模态模型；限额对专属服务不生效
若因限额配置导致服务停用，可以通过调高限额或等待周期重置便可自动恢复服务组启用
若因限额配置导致服务停用，可通过 APP Key 管理页面自助启用服务组或联系客户经理启用账号
用量统计存在 5 分钟左右时延，实际消耗可能略高于限额值。建议您在配置时适当降低限额阈值，预留一定的缓冲空间，以降低超额风险

停用/启用规则

主用户可对所创建的全局限额规则进行停用或启用规则，当停用后，全部用户下的所有服务组合计额度将不再受此限额限制，此配置立即失效；当启用后，全部用户下的所有服务组合计额度将受此限额限制，此配置立即生效。

编辑规则

支持主用户对单条全局限额规则进行消费限额、Token限额、发送通知、告警阈值的修改，提交后将生效新修改的规则内容。

删除规则

支持主用户删除全局限额规则，删除后，全部用户下的所有服务组合计额度将不再受此限额限制，此配置立即失效。

账号限额

场景说明

主用户可对部分用户（包括自己）下的所有服务组进行统一周期（日、自然月、自然年）下的合计调用额度限制（**消费限额或Token限额**），当满足所设置的限额告警阈值时可自动发送邮件、短信告知主用户，并在达到限额时停止**对应用户**下的服务调用。

示例：主用户想对**用户A**下的所有服务组（**合计10个服务组**）按照**每月总消费金额合计1000元**或**每月消耗总Token量合计1000000**进行额度控制，并希望在总消费金额或总Token消耗量**达到90%**时自动发送告警通知，当满足达到任一限额条件时**停止**对应用户下的服务调用。

配置步骤

主用户登录星辰TokenHub运营服务平台。

在左侧导航栏选择“限额管理”。

在页面下方点击“账号限额”选项卡，并点击“配置限额”拉起右侧抽屉弹窗，在选择任一用户后，进行账号额度的限额周期（天、自然月、自然年）、消费限额、Token限额、告警阈值、发送通知的配置操作并点击“提交并生效”完成限额配置。

账号限额配置支持添加（最多三个）和删除限额周期配置，且每个周期配置中消费限额和Token限额至少设置一个。

配置说明：
账号限额：为**指定用户下的所有服务组合计额度**配置额度上限，执行额度管控
消费限额和 Tokens 限额至少需配置一项，可同时配置；若同一个限额周期内都配置时，优先达到触发阈值的先生效
超出任一配置限额后，服务组将自动暂停服务，请提前评估业务中断风险
生效范围：限额不包含免费体验额度、量包抵扣额度；Tokens 限额不包含多模态模型；限额对专属服务不生效
若因限额配置导致服务停用，可以通过调高限额或等待周期重置便可自动恢复服务组启用
若因限额配置导致服务停用，可通过 APP Key 管理页面自助启用服务组或联系客户经理启用账号
用量统计存在 5 分钟左右时延，实际消耗可能略高于限额值。建议您在配置时适当降低限额阈值，预留一定的缓冲空间，以降低超额风险

停用/启用规则

主用户可对所创建的账号限额规则进行停用或启用规则，当停用后，对应用户下的所有服务组合计额度将不再受此限额限制，此配置立即失效；当启用后，对应用户下的所有服务组合计额度将受此限额限制，此配置立即生效。

编辑规则

支持主用户对单条账号限额规则进行消费限额、Token限额、发送通知、告警阈值的修改，提交后将生效新修改的规则内容。

删除规则

支持主用户删除账号限额规则，删除后，对应用户下的所有服务组合计额度将不再受此限额限制，此配置立即失效。

查询规则

支持主用户按照主/子用户名称和限额周期进行相关规则筛选。

服务组限额

场景说明

主用户可对指定用户（包括自己）下的指定服务组进行统一周期（日、自然月、自然年）下的调用额度限制（**消费限额或Token限额**），当满足所设置的限额告警阈值时可自动发送邮件、短信告知主用户，并在达到限额时停止**对应用户对应服务组**下的服务调用。

示例：主用户想对**用户A**下的**服务组1**按照**每月总消费金额合计1000元**或**每月消耗总Token量合计1000000**进行额度控制，并希望在总消费金额或总Token消耗量**达到90%**时自动发送告警通知，当满足达到任一限额条件时**停止**用户A下的服务组1的服务调用（此功能等同于APP Key管理-服务组-额度控制）。

配置步骤

主用户登录星辰TokenHub运营服务平台。

在左侧导航栏选择“限额管理”。

在页面下方点击“服务组限额”选项卡，并点击“配置限额”拉起右侧抽屉弹窗，在选择任一用户及其对应的服务组后，进行服务组额度的限额周期（天、自然月、自然年）、消费限额、Token限额、告警阈值、发送通知的配置操作并点击“提交并生效”完成限额配置。

服务组限额配置支持添加（最多三个）和删除限额周期配置，且每个周期配置中消费限额和Token限额至少设置一个。

配置说明：
服务组限额：为**指定用户下指定服务组**配置额度上限，执行额度管控
消费限额和 Tokens 限额至少需配置一项，可同时配置；若同一个限额周期内都配置时，优先达到触发阈值的先生效
超出任一配置限额后，服务组将自动暂停服务，请提前评估业务中断风险
生效范围：限额不包含免费体验额度、量包抵扣额度；Tokens 限额不包含多模态模型；限额对专属服务不生效
若因限额配置导致服务停用，可以通过调高限额或等待周期重置便可自动恢复服务组启用
若因限额配置导致服务停用，可通过 APP Key 管理页面自助启用服务组或联系客户经理启用账号
用量统计存在 5 分钟左右时延，实际消耗可能略高于限额值。建议您在配置时适当降低限额阈值，预留一定的缓冲空间，以降低超额风险

停用/启用规则

主用户可对所创建的服务组限额规则进行停用或启用规则，当停用后，对应用户下的对应服务组额度将不再受此限额限制，此配置立即失效；当启用后，对应用户下的对应服务组额度将受此限额限制，此配置立即生效。

编辑规则

支持主用户对单条服务组限额规则进行消费限额、Token限额、发送通知、告警阈值的修改，提交后将生效新修改的规则内容。

删除规则

支持主用户删除服务组限额规则，删除后，对应用户下的对应服务组额度将不再受此限额限制，此配置立即失效。

查询规则

支持主用户按照主/子用户名称、服务组名称和限额周期进行相关规则筛选。

说明：
主用户对子用户服务组级别的限额管理，子用户可在“APP Key 管理”对应的服务组中查看到变化，子用户可修改和删除主用户的限额规则。

最佳实践

5.1 第三方工具添加模型配置指南

本章节说明如何在第三方工具中快速添加并使用星辰 TokenHub 运营服务平台所提供的模型。

5.1.1 获取模型调用API KEY、模型ID等信息

以下文章内容中所提到的 API KEY，通通即为 TokenHub 平台中的 APP KEY。

关于APP KEY和模型ID具体获取方式可详见快速入门。

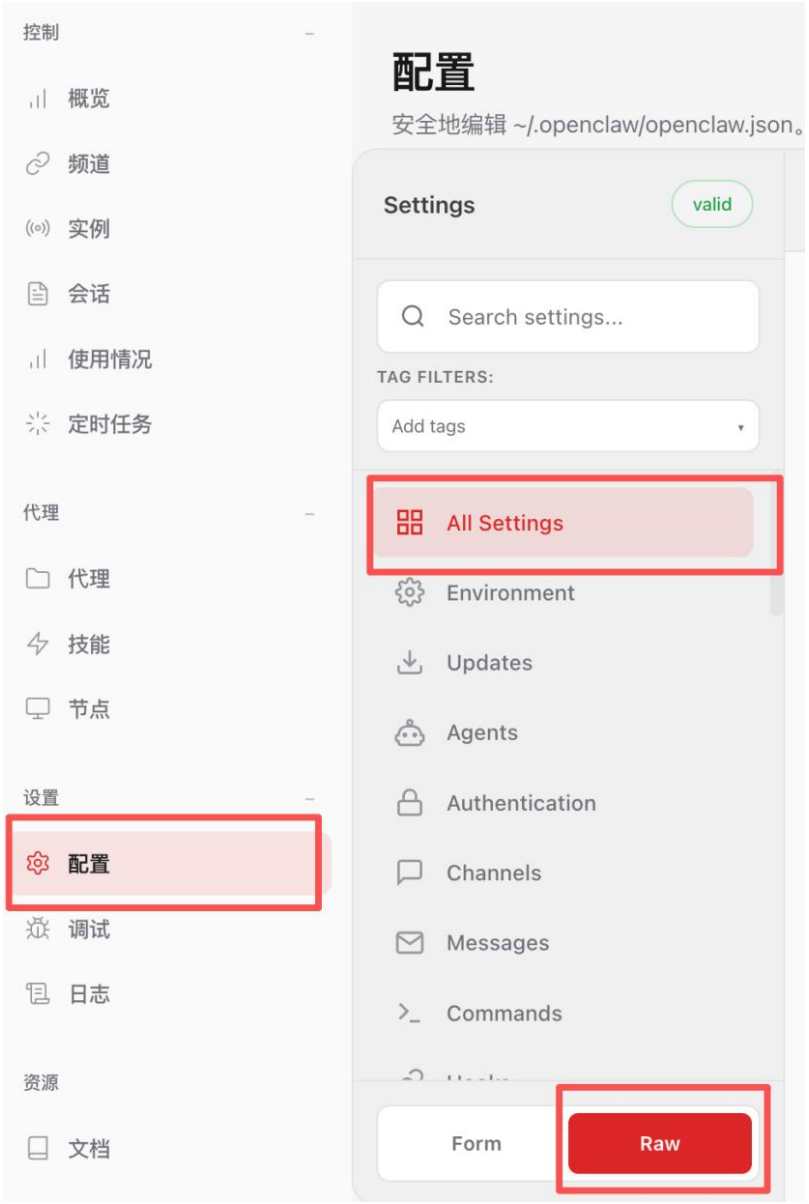
5.1.2 OpenClaw

方式一：通过Web UI修改配置文件（适用于Windows或者Mac系统）

1.在终端执行以下命令打开 Web UI。

```
openclaw dashboard
```

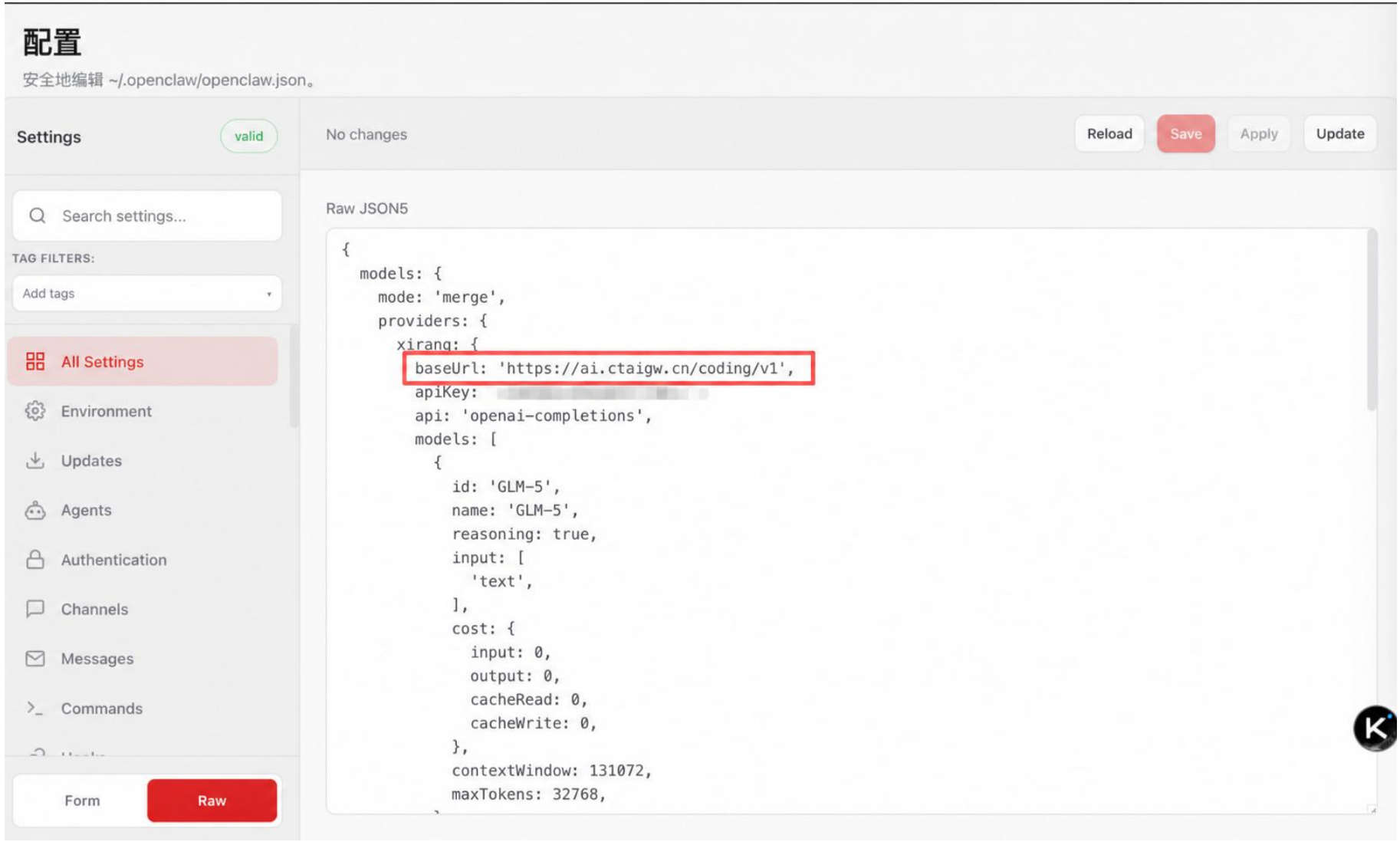
2.在 Web UI 的左侧菜单栏中选择“配置”->“All Settings”->“RAW”。


控制
概览
频道
实例
会话
使用情况
定时任务
代理
代理
技能
节点
设置
配置
调试
日志
资源
文档
配置
安全地编辑 ~/.openclaw/openclaw.json。
Settings
valid
Search settings...
TAG FILTERS:
Add tags
All Settings
Environment
Updates
Agents
Authentication
Channels
Messages
Commands
Form
Raw

3.复制以下内容到 Raw JSONS 输入框，替换已有内容，并将 YOUR_API_KEY 替换为在 APP Key 管理页面所获取到的 APP KEY。

```
{
    "models": {
        "mode": "merge",
        "providers": {
            "xirang": {
                "baseUrl": "https://ai.ctaigw.cn/v1",
                "apiKey": "YOUR_API_KEY",
                "api": "openai-completions",
                "models": [
                    {
                        "id": "GLM-5",
                        "name": "GLM-5",
                        "reasoning": true,
                        "input": [
                            "text"
                        ],
                        "cost": {
                            "input": 0,
                            "output": 0,
                            "cacheRead": 0,
                            "cacheWrite": 0
                        },
                        "contextWindow": 204800,
                        "maxTokens": 131072
                    },
                    {
                        "id": "Qwen3.5-397B-A17B",
                        "name": "Qwen3.5-397B-A17B",
                        "reasoning": true,
```

4.点击右上角 Save 保存，再点击 Update 使配置生效，保存成功后，apiKey 将显示为“__OPENCLAW_REDACTED__”。由于脱敏保护，仅在前端界面隐藏，不影响实际调用。



方式二：通过终端修改配置文件（适用于Windows 、mac 系统、Linux系统）

1.在终端执行以下命令打开配置文件

Mac系统和Linux系统执行以下命令：

```
nano ~/.openclaw/openclaw.json
```

Windows系统执行以下命令（Windows系统推荐优先使用方式一）：

```
notepad %USERPROFILE%\.openclaw\openclaw.json
```

复制以下内容到配置文件。将 YOUR_API_KEY 替换为在 APP Key 管理页面所获取到的 App Key。

```
{
    "models": {
        "mode": "merge",
        "providers": {
            "xirang": {
                "baseUrl": "https://ai.ctaigw.cn/v1",
                "apiKey": "YOUR_API_KEY",
                "api": "openai-completions",
                "models": [
                    {
                        "id": "GLM-5",
                        "name": "GLM-5",
                        "reasoning": true,
                        "input": [
                            "text"
                        ],
                        "cost": {
                            "input": 0,
                            "output": 0,
                            "cacheRead": 0,
                            "cacheWrite": 0
                        },
                        "contextWindow": 204800,
                        "maxTokens": 131072
                    },
                    {
                        "id": "Qwen3.5-397B-A17B",
                        "name": "Qwen3.5-397B-A17B",
                        "reasoning": true,
```

3.保存文件并退出，运行以下命令来使配置生效。

```
openclaw gateway restart
```

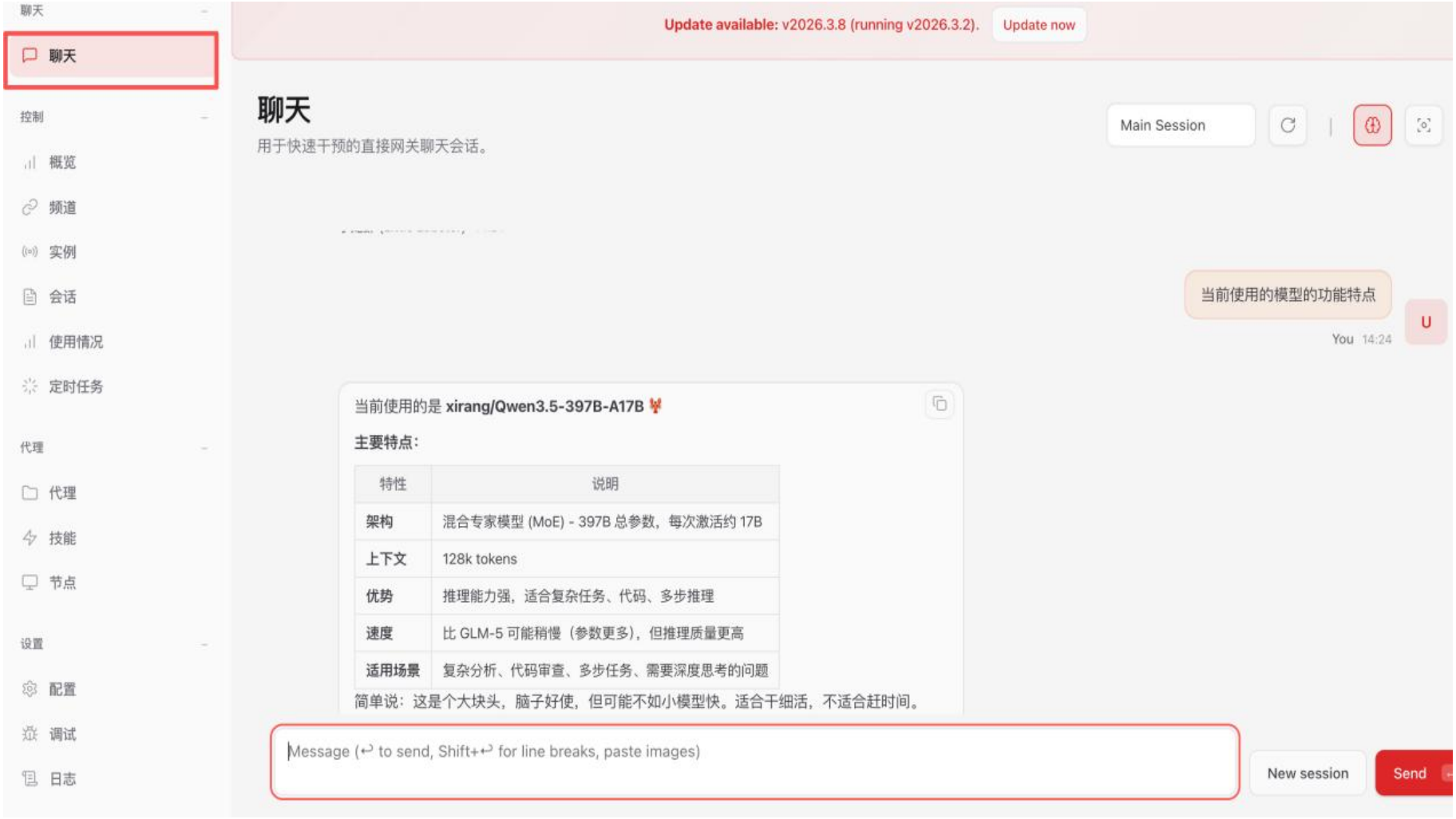
使用openClaw

Windows或者Mac环境

1.新开一个终端，运行以下命令。

```
openclaw dashboard
```

2.在 Web UI 点击左侧顶部“聊天”进入聊天窗口后，输入 Message 进行对话。



Linux环境

1.新开一个终端，运行以下命令，进入 TUI 界面。

```
openclaw tui
```

2.在 TUI 界面进行对话。

```
model set to xirang/Kimi-K2-Thinking

who are you

I'm Little Lobster 🦞 (小龙虾 in Chinese) — an AI assistant with a shell full of opinions.

I'm here to help you with pretty much anything: reading files, running commands, searching the web, managing messages, and
more.

What can I do for you?
connected | idle
agent main | session main (openclaw-tui) | xirang/Kimi-K2-Thinking | tokens 27k/1.0m (3%)
```

常见问题

聊天对话上方出现“API rate limit reached. Please try again later”

原因：由于模型免费额度已用完或已到期，或者已经开通模型付费但因账户余额不足，模型已无法使用。

解决方案：

请在控制台开通模型付费。如遇部分模型无法自助开通，请联系客户经理或提交工单购买。

提示“USER_TPM_RATELIMITING 用户请求TPM超限，请减少tokens后重试”

原因：每分钟消耗的 Token 总数已超过系统阈值，单次请求的文本过长（如长文档分析、长代码生成），或多个并发请求的总 Token 量过大，导致触发限流。

解决方案：

缩短聊天消息长度：请尝试将问题或输入内容拆分成几个较短的部分，分批发送。

稍后再试：系统限流通常会在1分钟内自动恢复，您可以稍等一会儿再继续提问。

减少同时对话：避免并发过高，如果您同时打开了多个聊天窗口，请先集中在一个窗口对话，关闭其他窗口，避免多个对话同时争抢额度。

调整限额：如果您需要更高的流量额度，请联系客户经理或提交工单提升额度。

提示“USER_RPM_RATELIMITING 用户请求RPM超限，请稍后重试”

原因：您当前发起的请求速度过快，每分钟的请求数（RPM, Requests Per Minute）已超过系统阈值。

解决方案：

稍后再试：限流通常会在1分钟内自动解除，您可以稍后再试。

降低发送频率：发送完一条消息后，等出现回复再发送下一条，避免连续快速发送。

减少同时对话：避免并发过高，如果您同时打开了多个聊天窗口，请先集中在一个窗口对话，关闭其他窗口，避免多个对话同时争抢额度。

调整限额：如果您需要更高的流量额度，请联系客户经理或提交工单提升额度。

提示“400 模型id错误，无法与AppKey对应，请联系管理员”

问题：当前使用的模型没有与您在 APP Key 管理所创建 AppKey 对应，导致无法使用。

解决方案：

修改服务组重新添加模型服务：在APP Key管理页面找到创建的服务组，编辑服务组，勾选您需要在OpenClaw中使用的模型，提交后在OpenClaw进行重试。

5.1.3 ClaudeCode

全局配置文件修改

macOS 全局：~/.claude/settings.json

Windows 全局： %USERPROFILE%\.claude\settings.json

项目根目录： .claude/settings.json（优先级 > 全局）

安装 Claude Code 后，通过手动配置的方式修改相应 JSON 文件，在 Claude Code 中接入编码套餐，需要配置以下信息：

ANTHROPIC_BASE_URL：设置为 https://ai.ctaigw.cn/coding。

ANTHROPIC_AUTH_TOKEN：设置为你购买的编程套餐专属的 API Key

model：设置为你购买的编程套餐所支持的模型



```
# 编辑或新增 `settings.json` 文件
# MacOS & Linux 为 `~/.claude/settings.json`
# Windows 为`用户目录/.claude/settings.json`
# 新增或修改里面的 env 字段
# 注意替换里面的 `YOUR_API_KEY` 为您上一步获取到的 API Key
{
  "$schema": "https://json.schemastore.org/claude-code-
settings.json",
  "env": {
     "ANTHROPIC_AUTH_TOKEN": "YOUR_API_KEY",
"ANTHROPIC_BASE_URL":"https://ai.ctaigw.cn/coding"  },
  "model": "GLM-5.1"
}

# 再编辑或新增 `.claude.json` 文件
# MacOS & Linux 为 `~/.claude.json`
# Windows 为`用户目录/.claude.json`
# 新增 `hasCompletedOnboarding` 参数
{
  "hasCompletedOnboarding": true
}
```

使用Claude Code

配置完成后，进入一个您的代码工作目录，在终端中执行 claude 命令即可开始使用 Claude Code

若遇到「Do you want to use this API key」选择 Yes 即可，启动后选择信任 Claude Code 访问文件夹里的文件

5.1.4 OpenCode

全局配置文件修改

请在以下路径创建并打开配置文件 opencode.json：

```
macOS / Linux: ~/.config/opencode/opencode.json
```

Windows: C:\Users\您的用户名\.config\opencode\opencode.json

将以下配置写入文件，关键配置项：

YOUR_API_KEY 为你购买的编程套餐专属的 API Key

models 的配置则为 来源于您当前购买实际套餐的可用模型，以下示例配置以 GLM MAX(旗舰版)为例，不同编码套餐支持调用的模型不一样

天翼云 State Cloud

可调用模型

- GLM-5
  擅长复杂系统工程与长程Agent任务
- GLM-5-Turbo
  面向OpenClaw龙蜥场景深度优化
- GLM-5.1
  擅长复杂系统工程与长程Agent任务
- GLM-4.5
  原生融合推理、编码和Agent能力
- GLM-4.5-Air
  轻量版，兼顾性能与性价比
- GLM-4.6
  上下文支持200K，实际编程任务表现佳
- GLM-4.7
  编码、智能体与通用对话能力全面提升

```
{
    "$schema": "https://opencode.ai/config.json",
    "provider": {
        "xirang": {
            "npm": "@ai-sdk/openai-compatible",
            "name": "xirang",
            "options": {
                "baseURL":
"https://ai.ctaigw.cn/coding/v1",
                "apiKey": "YOUR_API_KEY"
            },
            "models": {
                "GLM-4.7": {
                    "name": "GLM-4.7"
                },
                "GLM-5": {
                    "name": "GLM-5"
                },
                "GLM-5-Turbo": {
                    "name": "GLM-5-Turbo"
                },
                "GLM-5.1": {
                    "name": "GLM-5.1"
                },
                "GLM-4.5": {
                    "name": "GLM-4.5"
                },
                "GLM-4.5-Air": {
                    "name": "GLM-4.5-Air"
```

运行 opencode 启动 OpenCode

在命令行中输入以下命令：

```
$ opencode
```

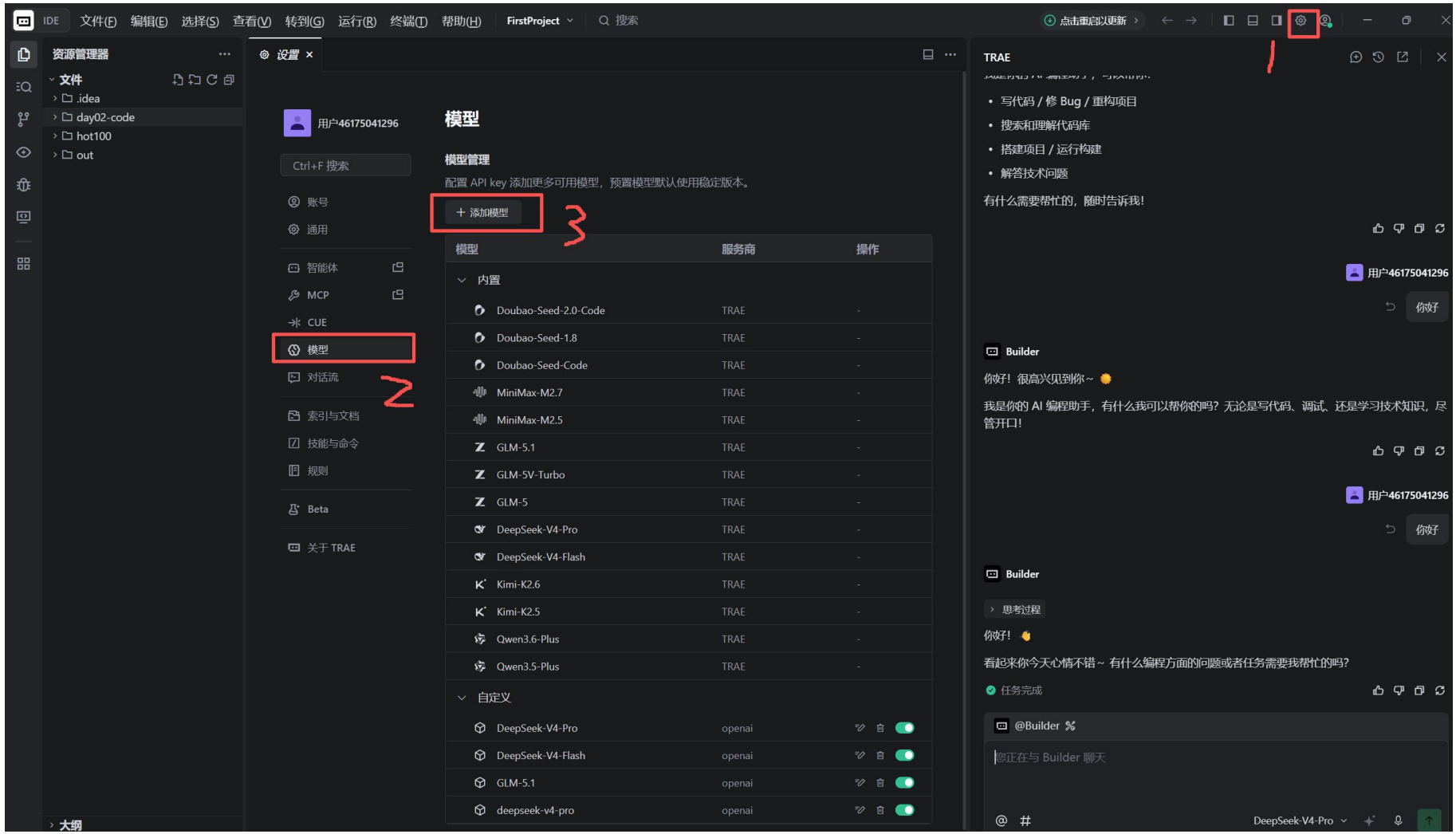
使用 /models 命令来选择模型，例如 GLM-5。

```
/models
```

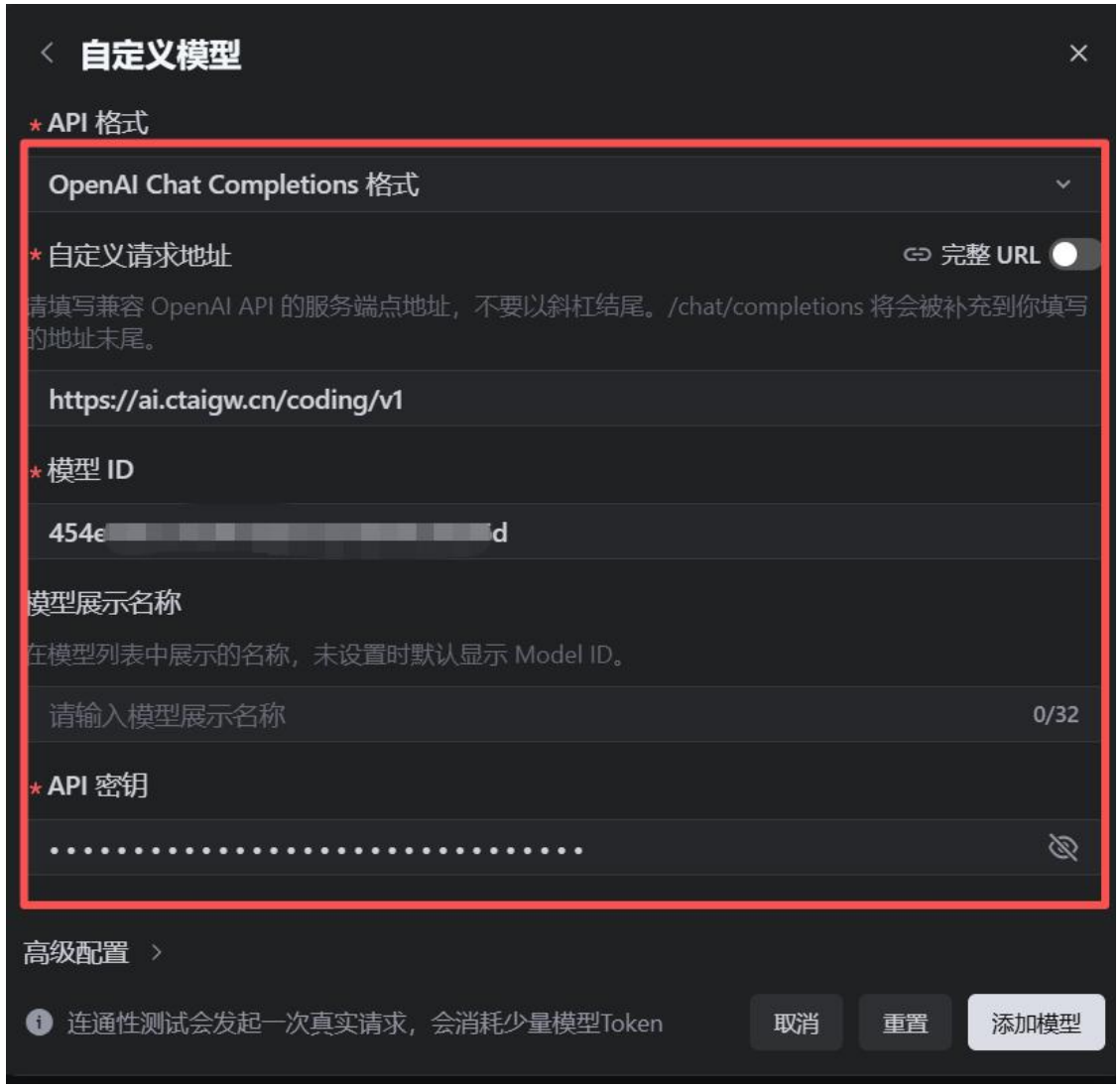
5.1.5 Trae

添加自定义模型

在下载好 Trae 后依次点击：设置->模型->添加模型->自定义模型



在自定义模型页面填入相关参数


自定义模型
API 格式
OpenAI Chat Completions 格式
自定义请求地址
完整 URL
请填写兼容 OpenAI API 的服务端点地址，不要以斜杠结尾。/chat/completions 将会被补充到你填写的地址末尾。
https://ai.ctaigw.cn/coding/v1
模型 ID
454e
d
模型展示名称
在模型列表中展示的名称，未设置时默认显示 Model ID。
请输入模型展示名称
0/32
API 密钥
高级配置
连通性测试会发起一次真实请求，会消耗少量模型Token
取消
重置
添加模型

注意点

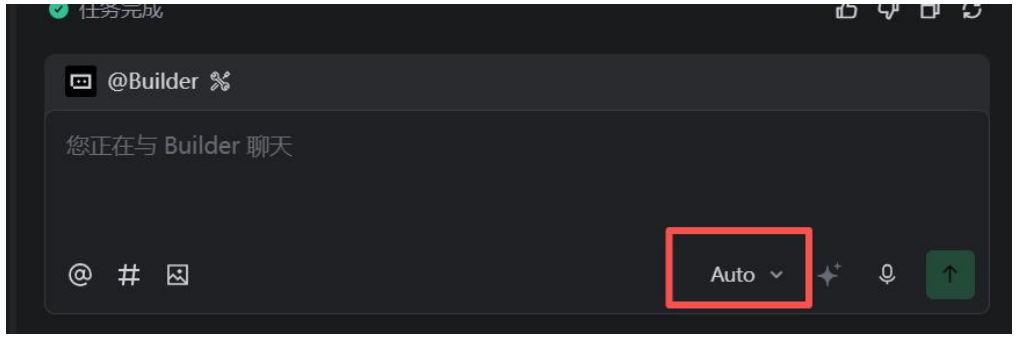
模型 ID 即为模型的名称，可在自己的 API 页面查询，确保准确

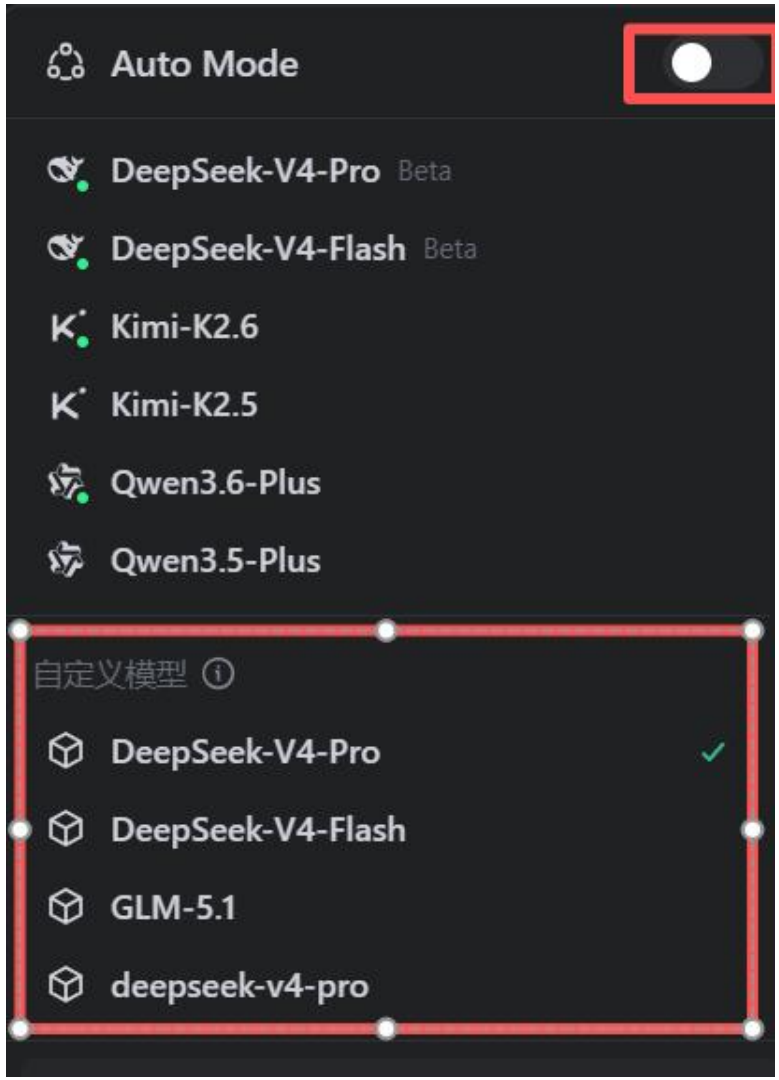
自定义请求地址必须填写完全，因为 Trae 只会原封不动地拿去请求，它自己不会在末尾自作聪明地补充

这里使用的 APP Key 管理的 APP Key，若使用编程 TokenPlan 套餐的 APP Key，请在自定义请求地址的 v1 前加上 coding，类似于.../coding/v1/chat/completions

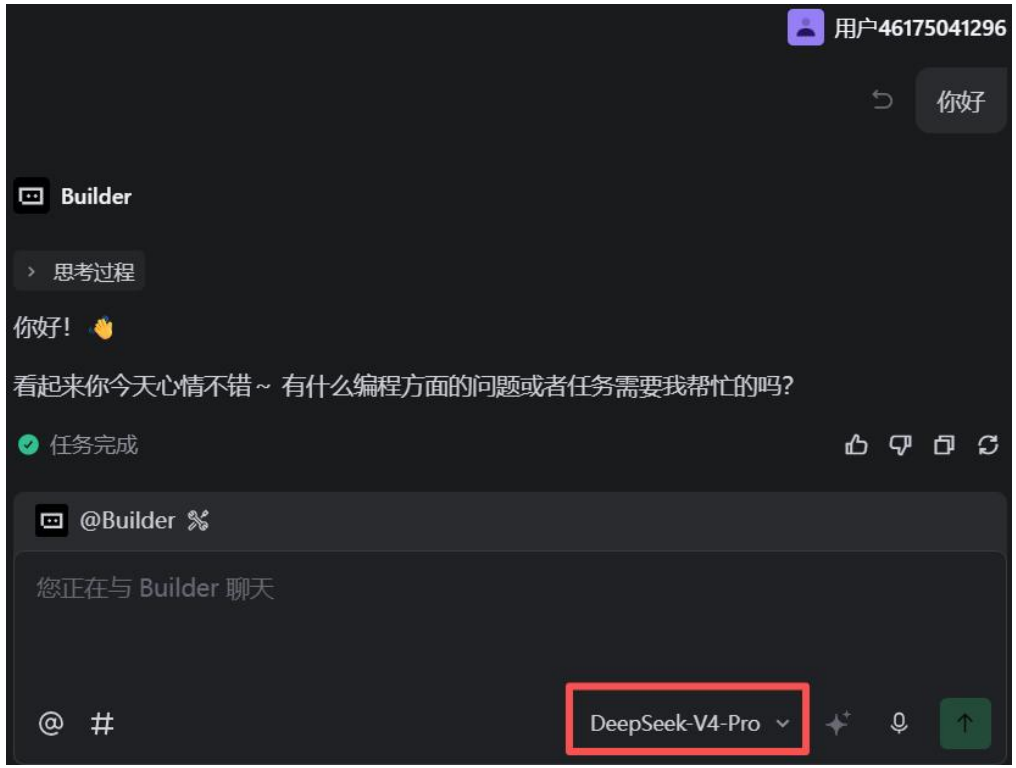
切换自定义模型

Trae 默认的是 Auto Mode 模型，点击其将其关掉，随后即可下滑选择自定义模型





随后就可以开始测试使用了



5.1.6 Codex（Openai）

在 Codex 中配置 Coding Plan

修改配置文件

完整复制以下内容并粘贴到 Codex 配置文件~/.codex/config.toml 中。

```
model_provider = "Ctyun_Coding_Plan"
model = "GLM-4.7"
[model_providers.Ctyun_Coding_Plan]
name = "Ctyun_Coding_Plan"
base_url = "https://ai.ctaigw.cn/coding/v1"
env_key = "OPENAI_API_KEY"
wire_api = "responses"
```

配置环境变量

将配置文件中的 OPENAI_API_KEY 环境变量设置为 Coding Plan 专属 API Key。

macOS

在终端中执行以下命令，查看默认 Shell 类型。

```
echo $SHELL
```

根据 Shell 类型设置环境变量，命令如下：

Zsh

```
# 用您的Coding Plan API Key 替换 YOUR_API_KEY
echo 'export OPENAI_API_KEY="YOUR_API_KEY"' >> ~/.zshrc
```

Bash

```
    # 用您的Coding Plan API Key 替换 YOUR_API_KEY
    echo 'export OPENAI_API_KEY="YOUR_API_KEY"' >>
~/.bash_profile
```

在终端中执行下列命令，使环境变量生效。

Zsh

```
source ~/.zshrc
```

Bash

```
source ~/.bash_profile
```

Windows

CMD

在 CMD 中运行以下命令，设置环境变量。

```
# 用您的Coding Plan API Key 替换 YOUR_API_KEY
setx OPENAI_API_KEY "YOUR_API_KEY"
```

打开一个新的 CMD 窗口，运行以下命令，检查环境变量是否生效。

```
echo %OPENAI_API_KEY%
```

PowerShell

在 PowerShell 中运行以下命令，设置环境变量。

```
    # 用套餐专属 API Key 替换 YOUR_API_KEY
    [Environment]::SetEnvironmentVariable("OPENAI_API_KEY",
"YOUR_API_KEY", [EnvironmentVariableTarget]::User)
```

打开一个新的 PowerShell 窗口，运行以下命令，检查环境变量是否生效。

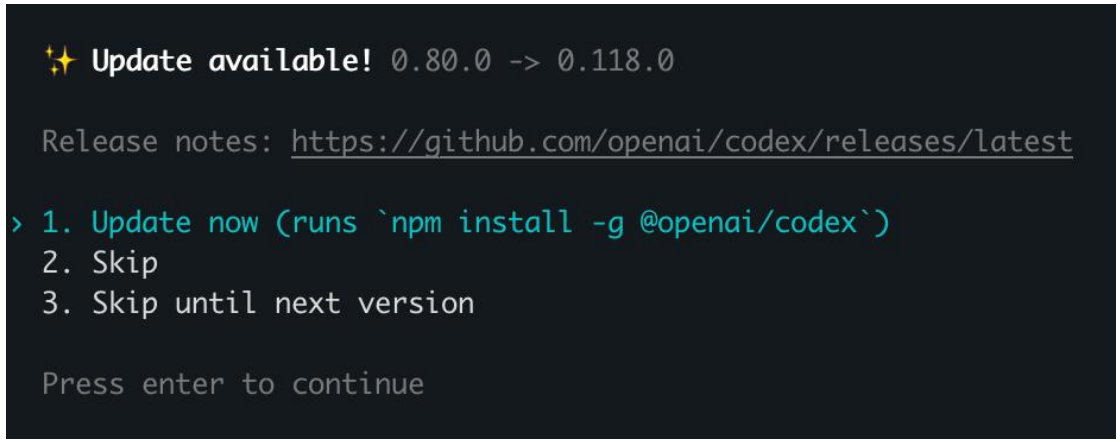
```
echo $env:OPENAI_API_KEY
```

使用 Codex

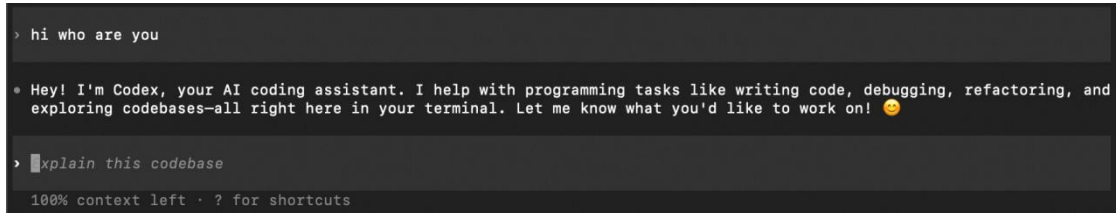
新建一个终端，执行以下命令进入 Codex。

```
codex
```

选择跳过更新。



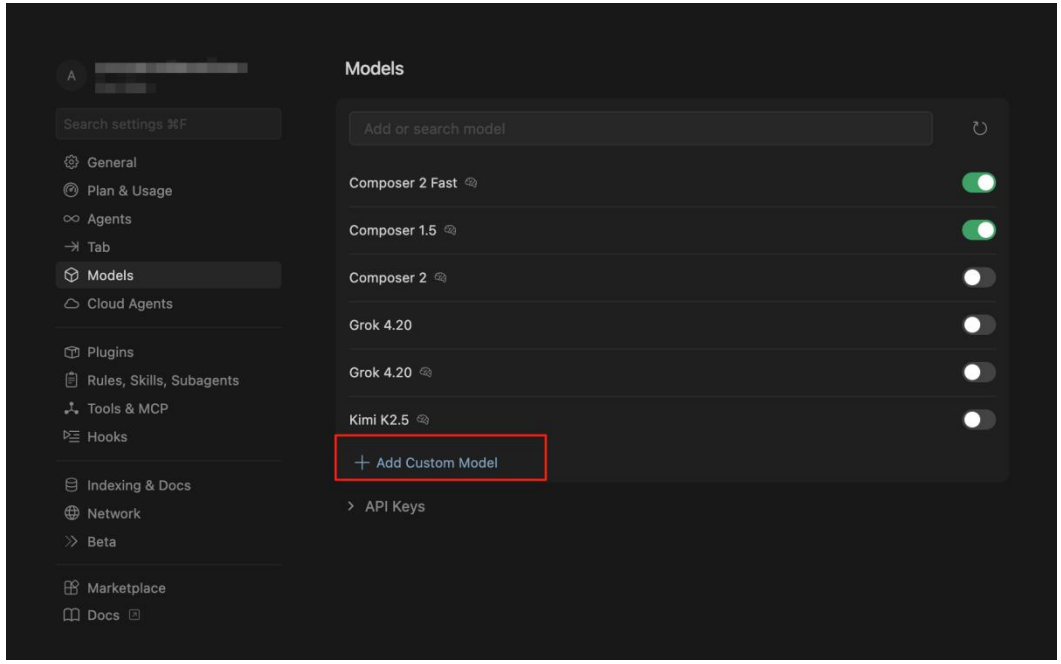
开始对话。



5.1.7 Cursor

创建新 Provider/Model

在 Cursor 中，打开 “Models” 部分，并点击 “Add Custom Model” 按钮。

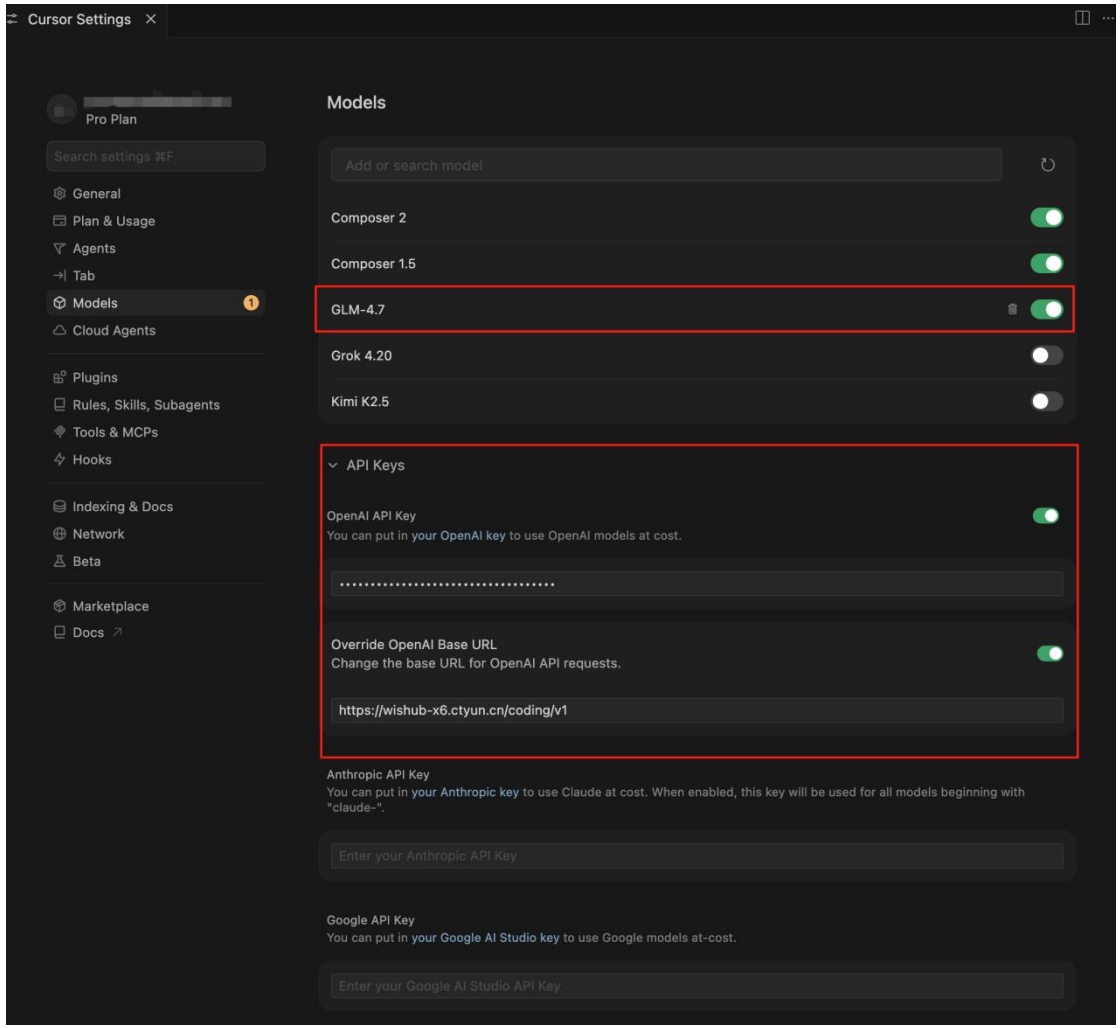


选择 OpenAI 协议。

配置 OpenAI API Key。

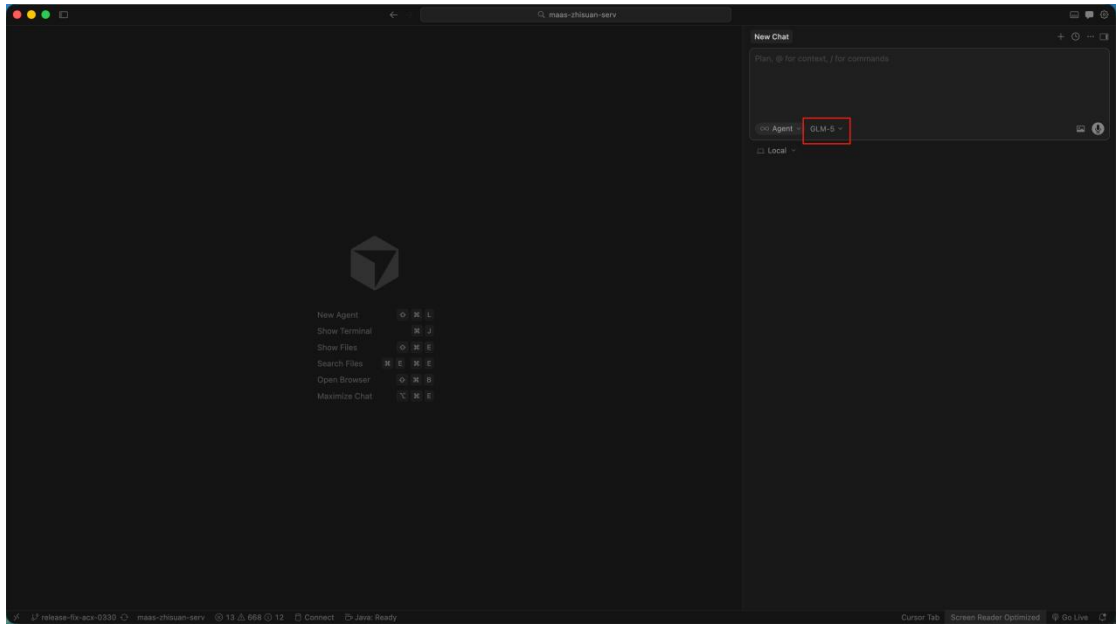
在 Override OpenAI Base URL 中，将默认 URL 替换：https://ai.ctaigw.cn/coding/v1。

输入您希望使用的模型，如 GLM-5, GLM-4.7, GLM-4.6 等。



开始使用

按照上面步骤，Cursor 已经配置完成，可在对话框中进行编程测试。



常见问题

Q：在 Cursor 中无法调用 Coding Plan 支持的模型?

A：如果在界面看到了如下报错：

The model xxx does not work with your current plan or api key.

Named models unavailable Free plans can only use Auto. Switch to Auto or upgrade plans to continue.

由于 Cursor 的产品限制，Cursor 免费版仅支持 Auto 模式，且不支持调用自定义模型。请升级至 Cursor Pro 及以上套餐，即可正常使用 Coding Plan 支持的模型。

5.1.8 Cline

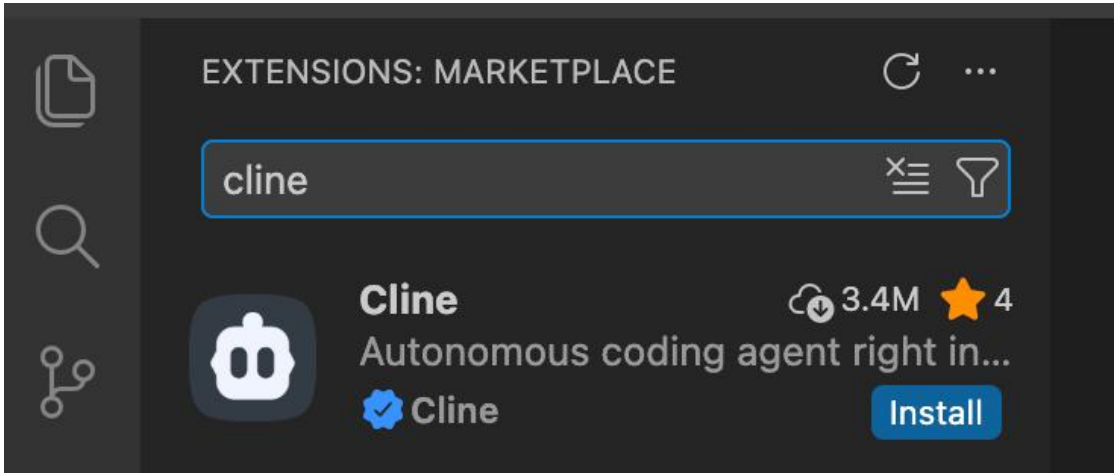
打开插件市场

打开 VS Code

点击左侧插件市场图标

在搜索框中输入 Cline

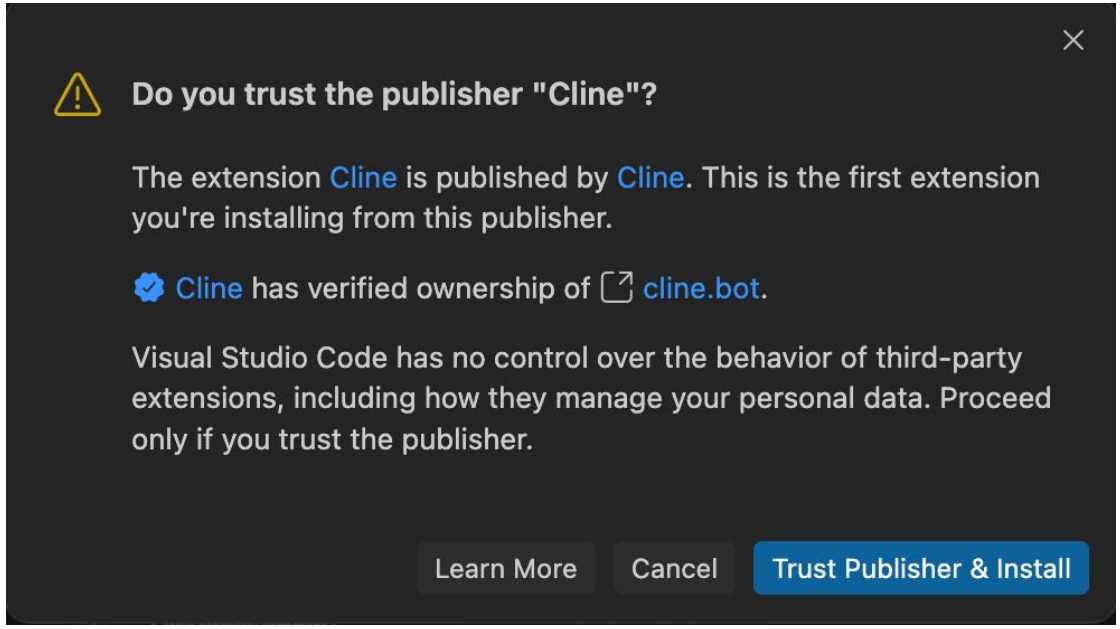
找到 Cline 扩展



安装插件

点击 Install 按钮进行安装

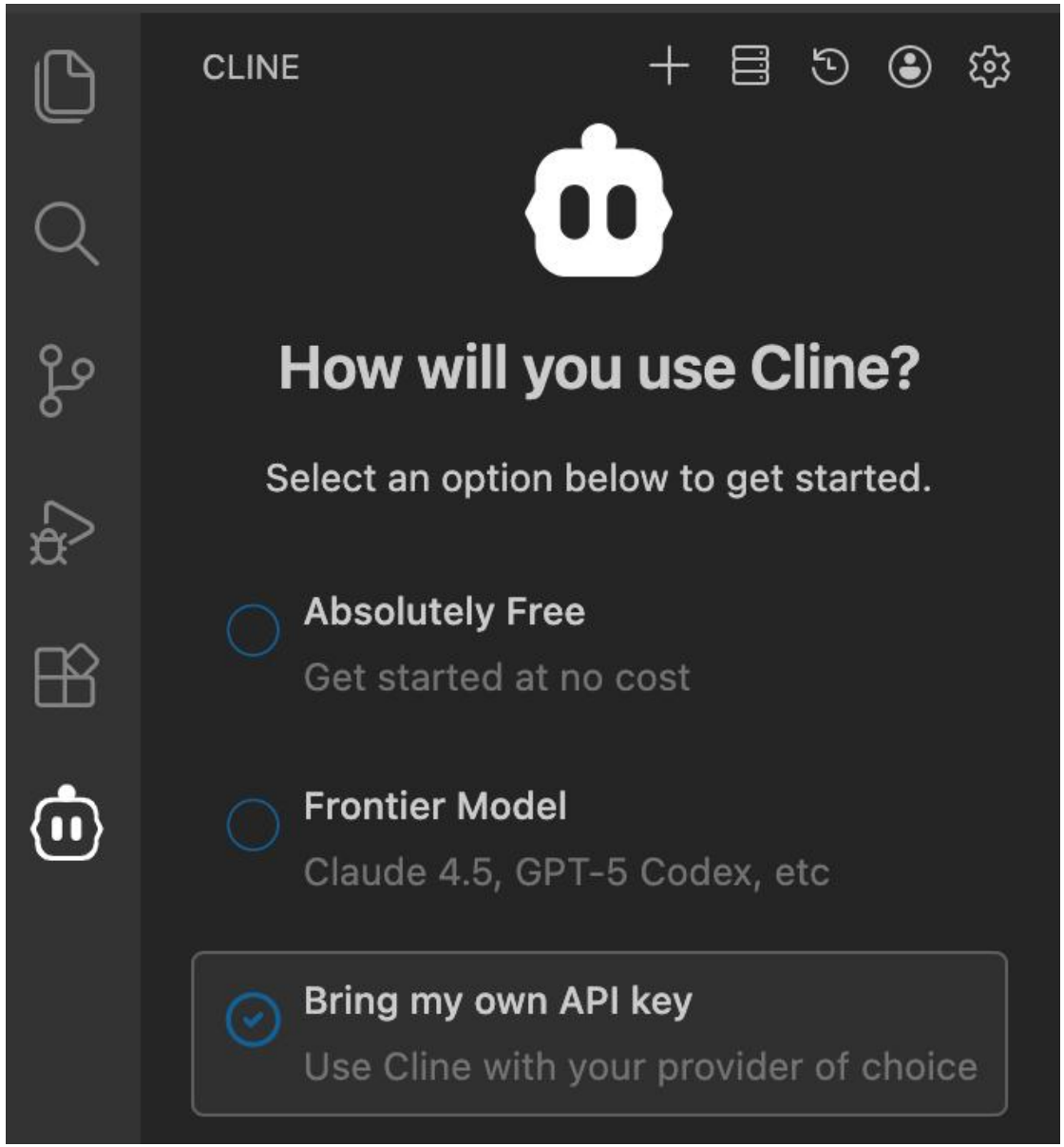
安装完成后，选择信任开发者



配置 API 设置

选择 API Key 方式

选择 Bring my own API key


CLINE
How will you use Cline?
Select an option below to get started.
Absolutely Free
Get started at no cost
Frontier Model
Claude 4.5, GPT-5 Codex, etc
Bring my own API key
Use Cline with your provider of choice

填入配置信息

请按照以下配置填入相关信息：

API Provider：选择 OpenAI Compatible

Base URL：输入 https://ai.ctaigw.cn/coding/v1

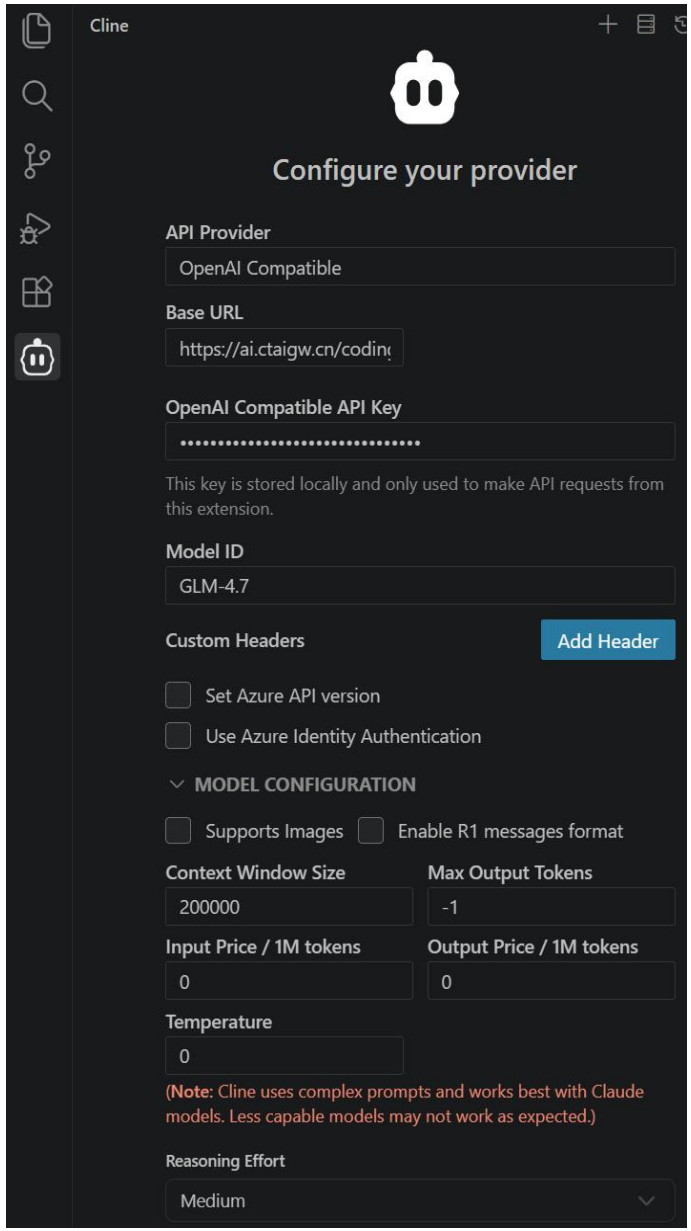
API Key：填入您的 API Key

Model ID：输入您希望使用的模型，如 GLM-5, GLM-4.7, GLM-4.6 等

其他配置：取消勾选 Support Images

调整 Context Window Size 为 200000

根据您的任务需求调整 temperature 等其它参数


Cline
Configure your provider
API Provider
OpenAI Compatible
Base URL
https://ai.ctaigw.cn/codin
OpenAI Compatible API Key
This key is stored locally and only used to make API requests from this extension.
Model ID
GLM-4.7
Custom Headers
Add Header
Set Azure API version
Use Azure Identity Authentication
MODEL CONFIGURATION
Supports Images
Enable R1 messages format
Context Window Size
200000
Max Output Tokens
-1
Input Price / 1M tokens
0
Output Price / 1M tokens
0
Temperature
0
(Note: Cline uses complex prompts and works best with Claude models. Less capable models may not work as expected.)
Reasoning Effort
Medium

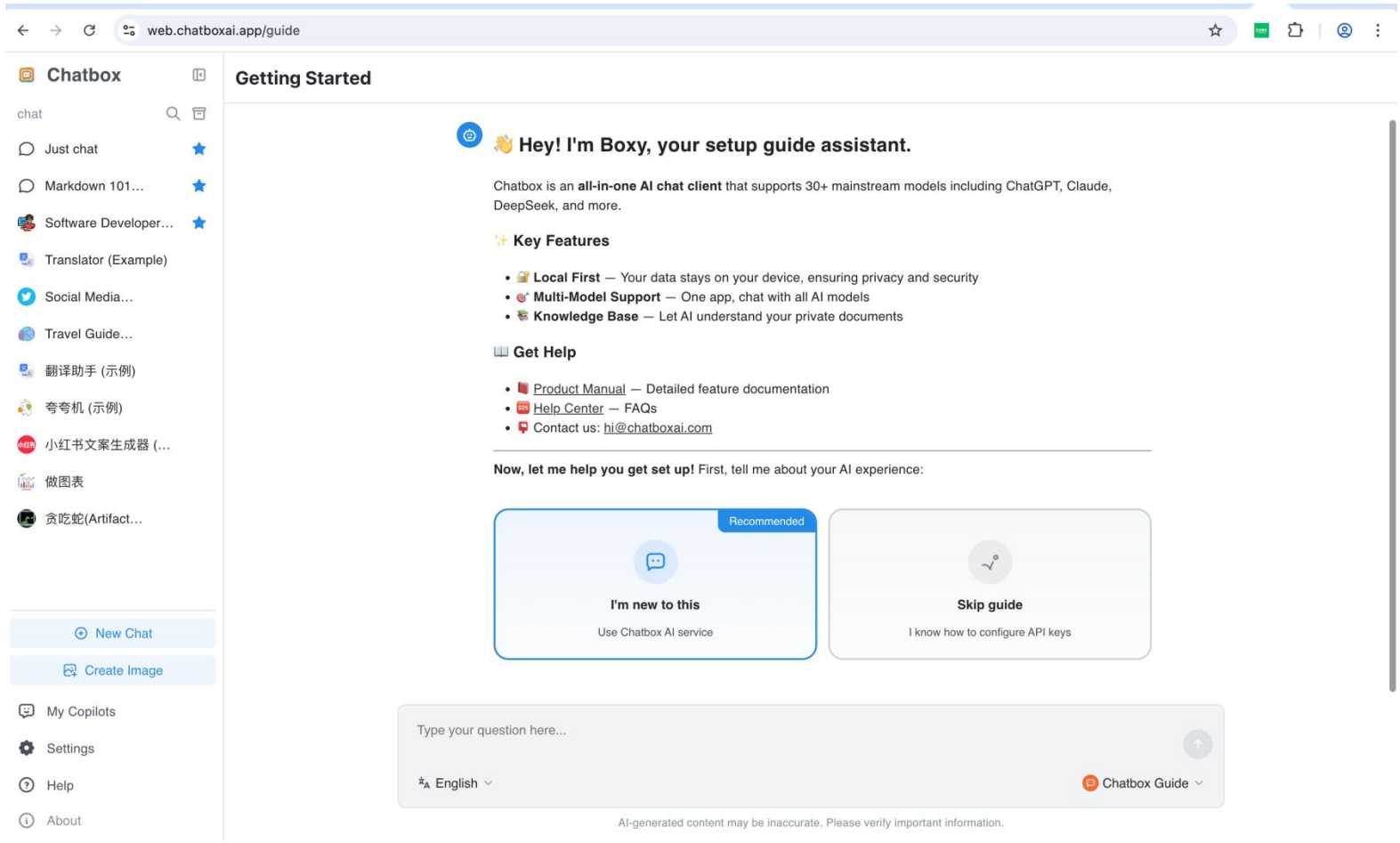
开始使用

配置完成后，您可以在输入框中输入需求，让模型帮助您完成各种任务。

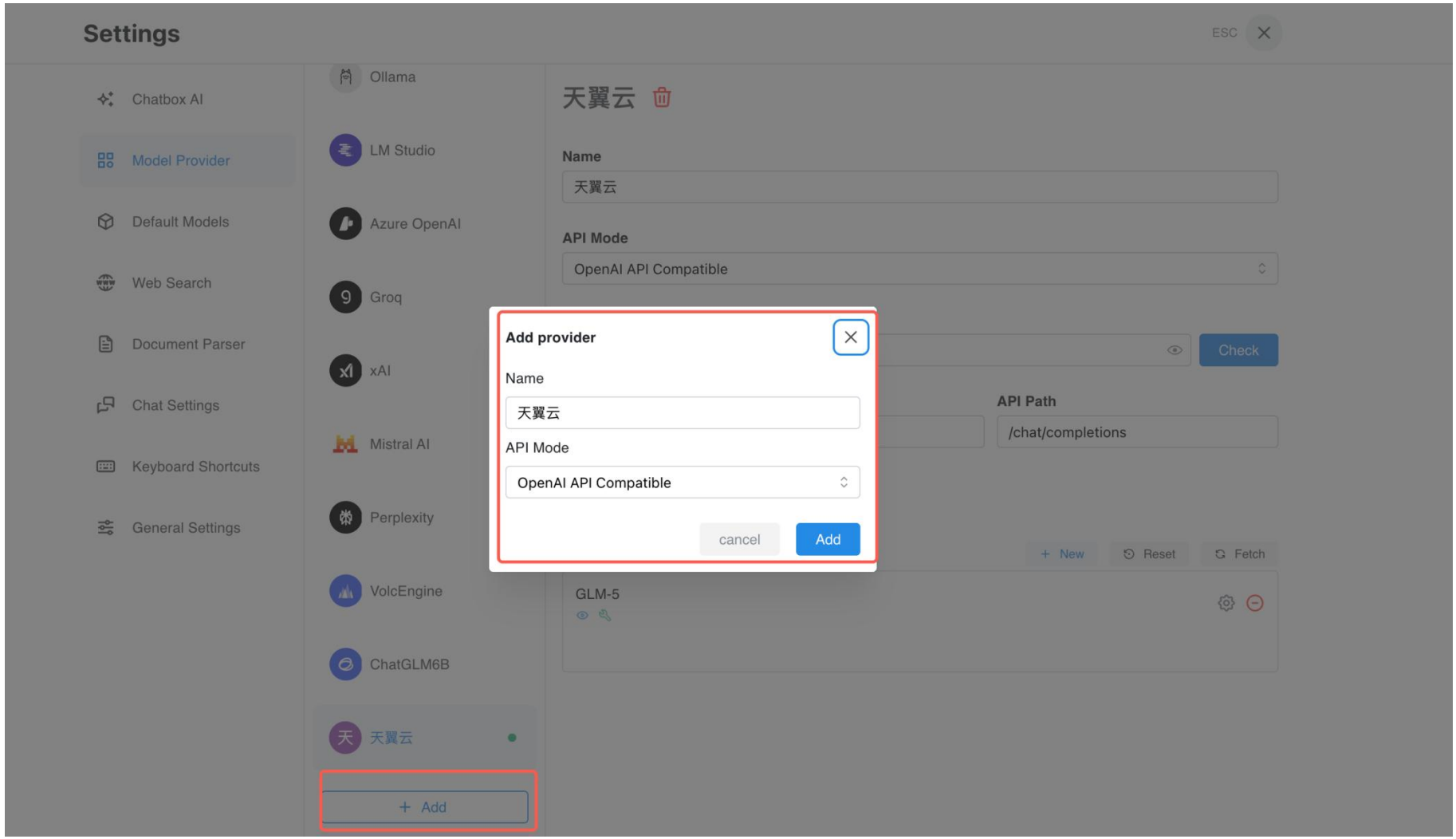

CLINE
What can I do for you?
Auto-approve: Read, Safe Commands, MCP
Type your task here...
Type @ for context, / for slash commands & workflows, hold shift to drag in files/images
openai-compat:GLM-4.7
Plan
Act
main

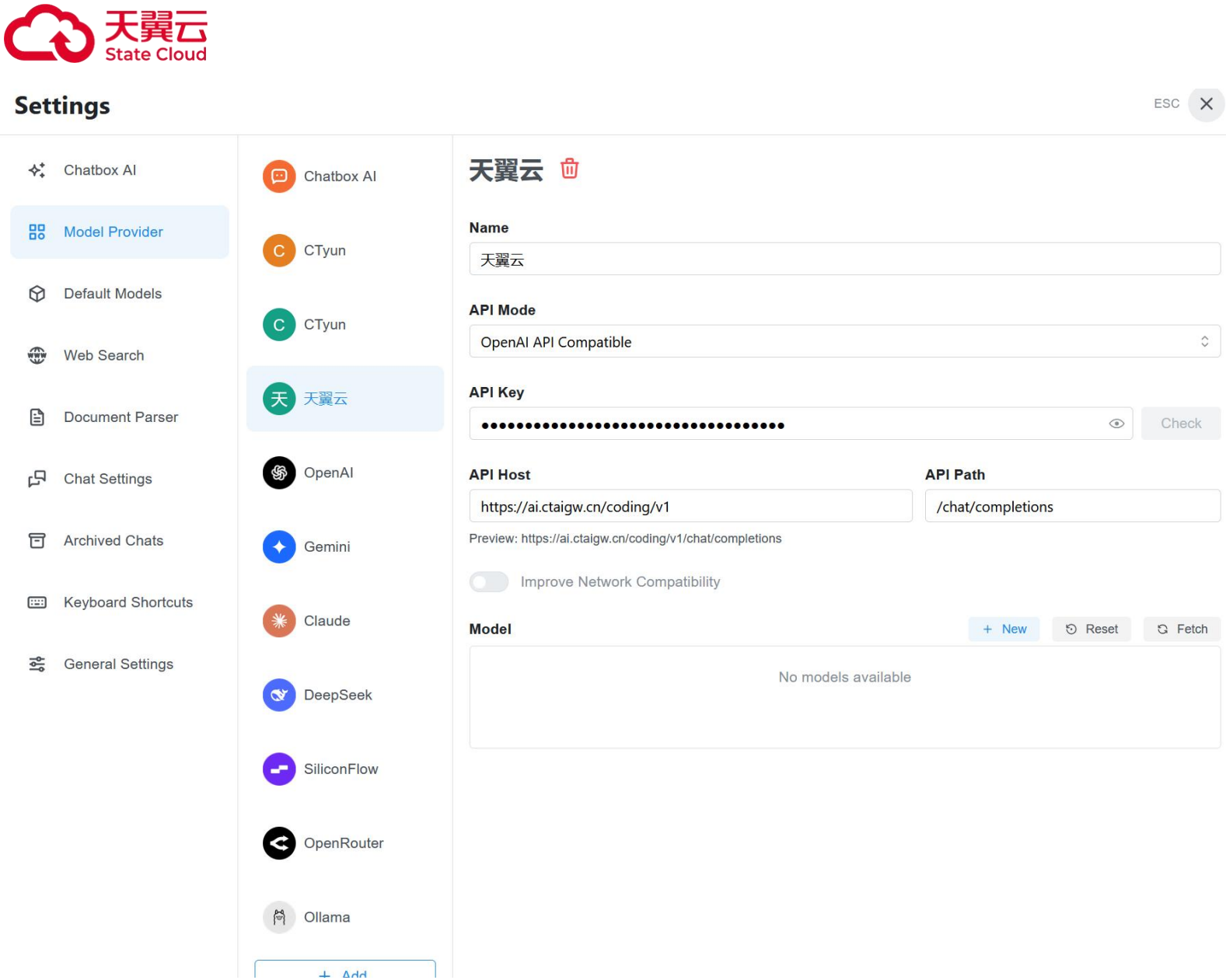
5.1.9 Chatbox

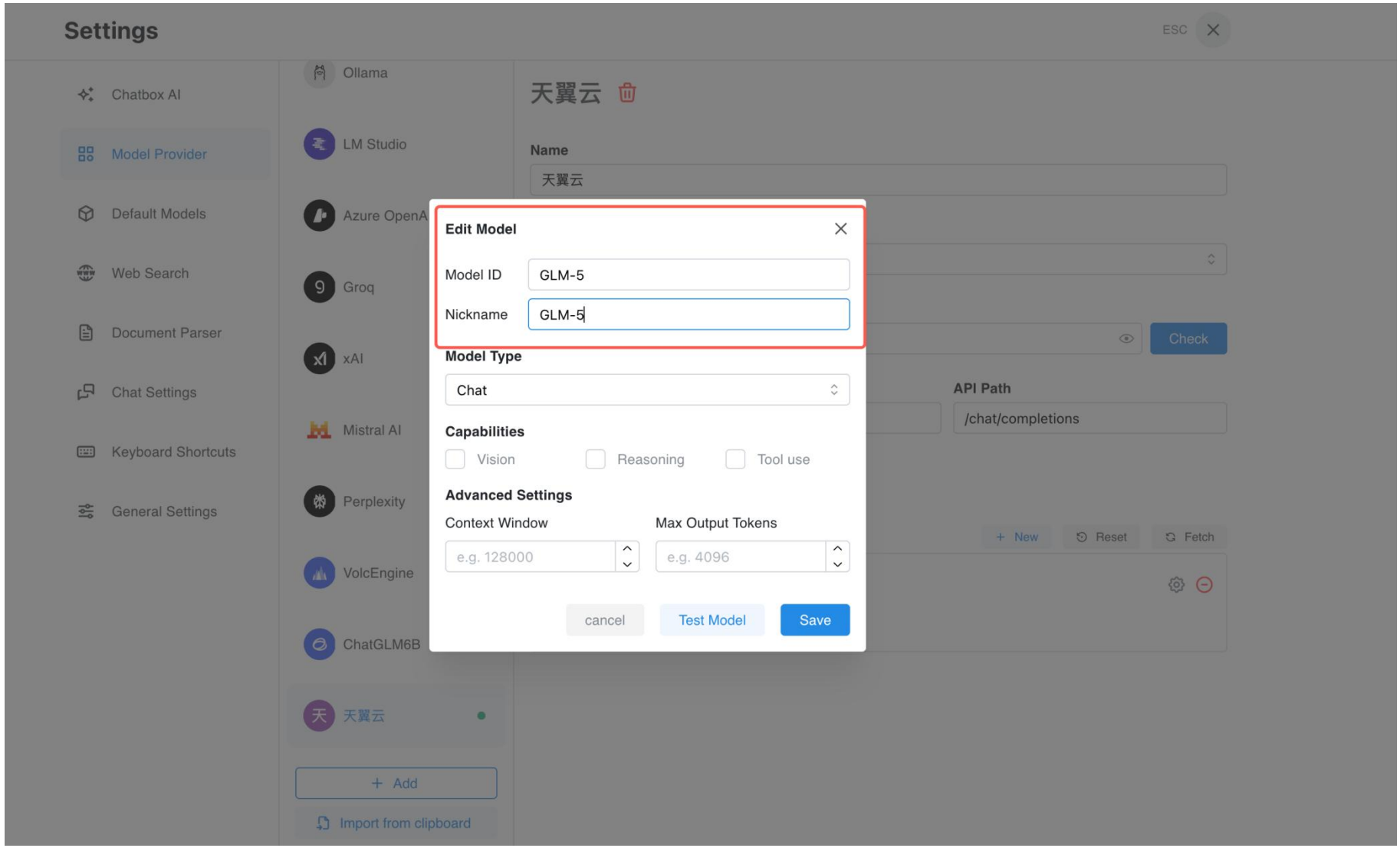
访问链接：https://web.chatboxai.app/

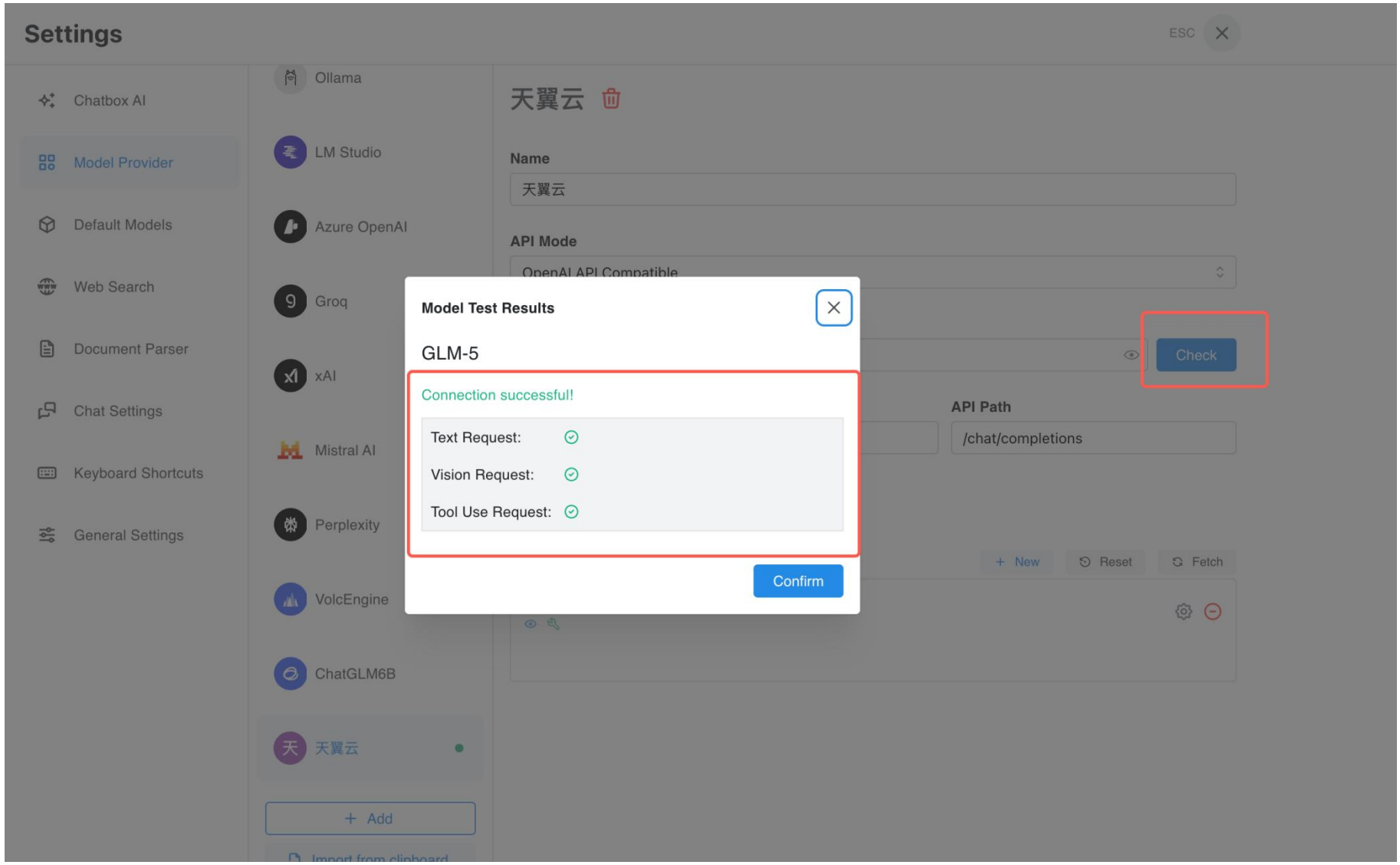


点击 Setting，添加 OpenAI API Compatible 添加天翼云。 配置 Api Host、API path、API key 和 Model，配置完成后点击 check 校验配置是否生效。

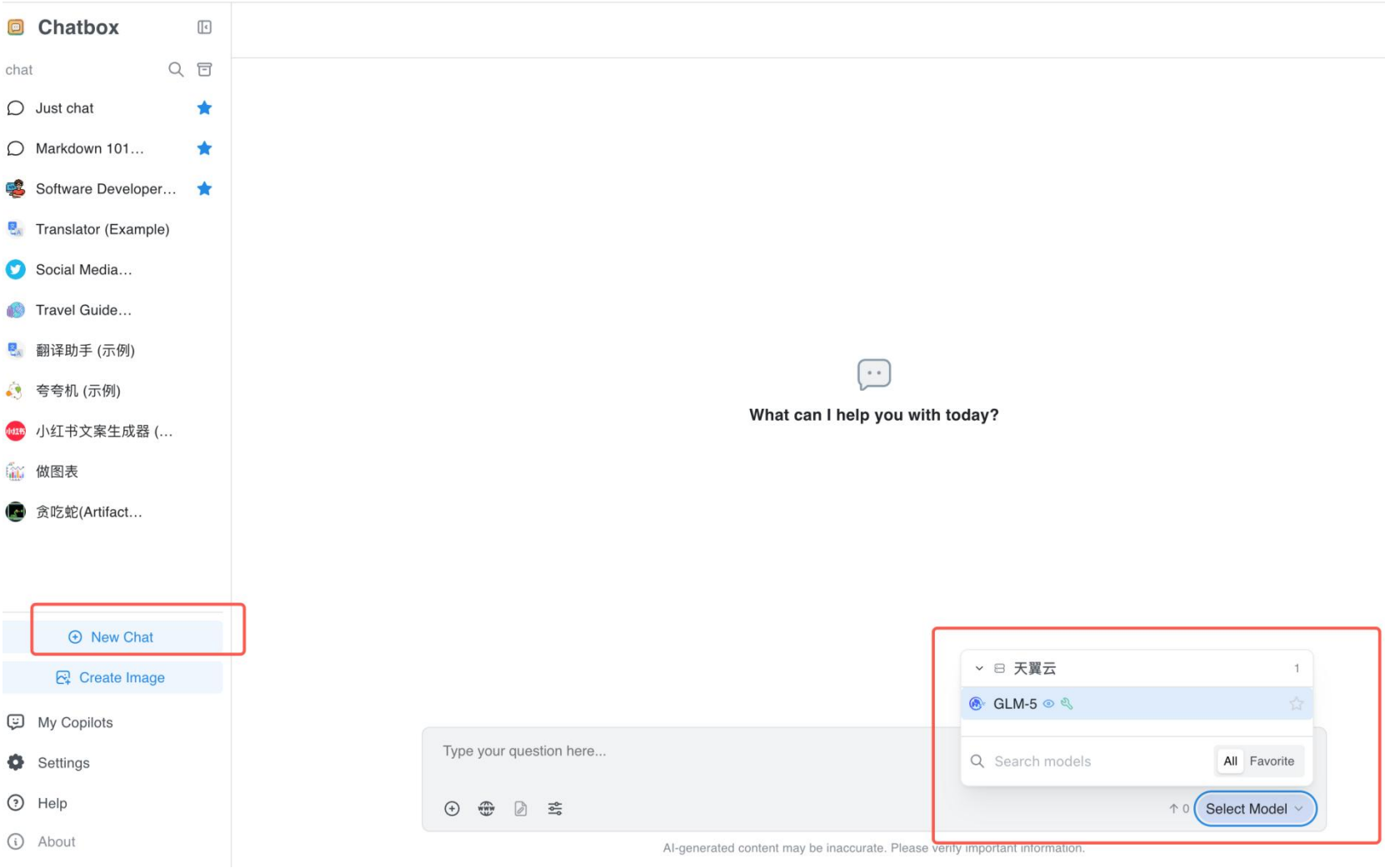



天翼云
State Cloud
Settings
ESC
Chatbox AI
Model Provider
Default Models
Web Search
Document Parser
Chat Settings
Archived Chats
Keyboard Shortcuts
General Settings
Chatbox AI
CTyun
CTyun
天翼云
OpenAI
Gemini
Claude
DeepSeek
SiliconFlow
OpenRouter
Ollama
天翼云
Name
天翼云
API Mode
OpenAI API Compatible
API Key
Check
API Host
https://ai.ctaigw.cn/coding/v1
API Path
/chat/completions
Preview: https://ai.ctaigw.cn/coding/v1/chat/completions
Improve Network Compatibility
Model
New
Reset
Fetch
No models available


Settings
ESC
Chatbox AI
Model Provider
Default Models
Web Search
Document Parser
Chat Settings
Keyboard Shortcuts
General Settings
Ollama
LM Studio
Azure OpenA
Groq
xAI
Mistral AI
Perplexity
VolcEngine
ChatGLM6B
天翼云
+ Add
Import from clipboard
天翼云
Name
天翼云
Check
API Path
/chat/completions
+ New
Reset
Fetch
Edit Model
Model ID
GLM-5
Nickname
GLM-5
Model Type
Chat
Capabilities
Vision
Reasoning
Tool use
Advanced Settings
Context Window
e.g. 128000
Max Output Tokens
e.g. 4096
cancel
Test Model
Save



配置成功后，回到主页，选择 new chat，即可使用刚配置的模型。


Chatbox
chat
Just chat
Markdown 101...
Software Developer...
Translator (Example)
Social Media...
Travel Guide...
翻译助手 (示例)
夸夸机 (示例)
小红书文案生成器 (...
做图表
贪吃蛇(Artifact...
New Chat
Create Image
My Copilots
Settings
Help
About
What can I help you with today?
Type your question here...
天翼云
1
GLM-5
Search models
All
Favorite
0
Select Model
AI-generated content may be inaccurate. Please verify important information.

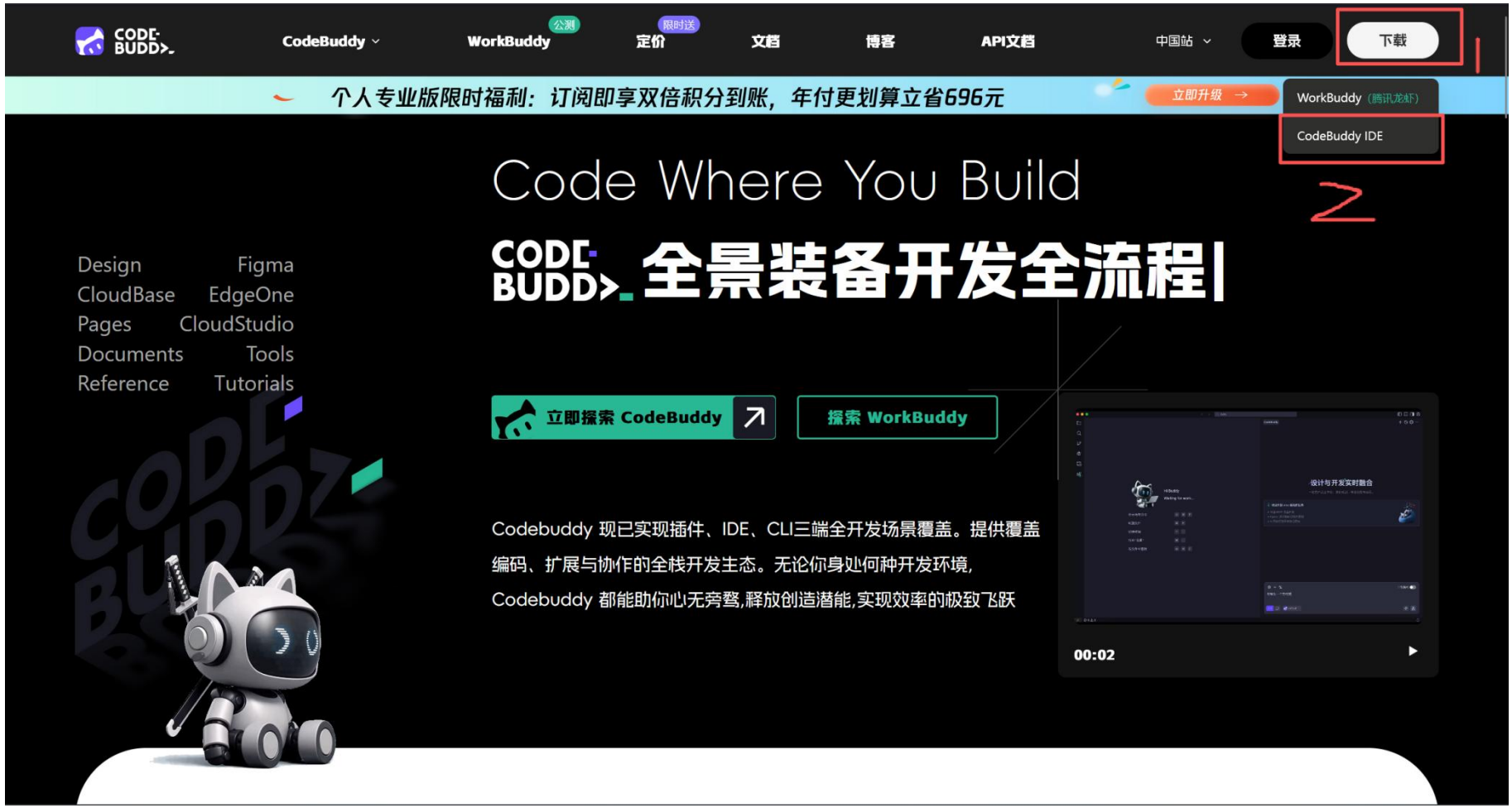

Chatbox
天翼云编程套餐
Search...
chat
Just chat
Markdown 101...
Software Developer...
天翼云编程套餐
Translator (Example)
Social Media...
Travel Guide...
翻译助手 (示例)
夸夸机 (示例)
小红书文案生成器 (...
做图表
贪吃蛇(Artifact...
New Chat
Create Image
My Copilots
Settings
Help
About
写一个快速排序算法
Deeply thought (4.3s)
这是一个经典的快速排序算法实现，使用 Python 语言编写，包含了详细的注释以帮助理解。
核心思想
快速排序使用分治法策略：
1. 选基准：从数列中挑出一个元素，称为"基准"。
2. 分区：重新排序数列，所有比基准值小的元素摆放在基准前面，所有比基准值大的元素摆在基准后面。
3. 递归：递归地把小于基准值元素的子数列和大于基准值元素的子数列排序。
代码实现
PYTHON
def quick_sort(arr):
"""
快速排序的主函数
:param arr: 需要排序的列表
:return: 排序后的列表
"""
递归终止条件：如果数组为空或只有一个元素，则无需排序
另一种实现方式（原地排序，更节省内存）
上面的写法简洁易懂，但会创建新的列表，占用额外内存。下面是传统的原地排序版本，不需要额外开辟数组空间：
Type your question here...
1K
GLM-5
AI-generated content may be inaccurate. Please verify important information.

5.1.10 Codebuddy

CodeBuddy 是一款基于 AI 的全流程智能编程工具，致力于构建产品、设计、研发、部署无缝协作的共生环境。

安装 CodeBuddy

访问官网 https://www.codebuddy.cn/home/下载并安装适合您的操作系统的版本。



登录 CodeBuddy 并打开项目


文件(F)
编辑(E)
选择(S)
查看(V)
转到(G)
FirstProject
打开 Agents
资源管理器
FIRSTPROJECT
.idea
day02-code
hot100
out
Hi Buddy
Waiting for work...
显示所有命令 Ctrl + Shift + P
转到文件 Ctrl + P
切换终端 Ctrl + `
在文件中查找 Ctrl + Shift + F
打开"设置" Ctrl + ,
大纲
时间线
CodeBuddy
最新活动 New
个人专业版放大招！订阅即享双倍积分到账，年付更划算立省696元
立即升级
你好
CodeBuddy
你好！有什么我可以帮你的吗？我看到你的工作区是一个 Java 项目，包含 day02-code 和 hot100 两个模块。如果你需要帮助编写代码、调试问题或其他任何事情，随时告诉我！
提问或输入"/"快捷命令
Craft
CTyun:GLM-5.1
内容由 AI 生成，仅供参考
WeCom

配置天翼云自定义模型

腾讯云 CodeBuddy（及相关的 WorkBuddy / CLI 客户端）中，配置自定义模型主要通过创建或编辑 models.json 配置文件来实现。无论是接入本地运行的模型（如 Ollama），还是第三方的兼容 API（如 DeepSeek、通义千问等），都可以通过该配置完成。

Windows 路径：C:\Users\<你的用户名>\.codebuddy\models.json

macOS / Linux 路径：~/.codebuddy/models.json

提示：如果该路径没有 models 文件请创建

配置示例：

```
{
  "models": [
    {
      "id": "GLM-5.1",
      "name": "CTyun",
      "vendor": "OpenAI",
      "url": "https://ai.ctaigw.cn/v1/chat/completions",
      "apiKey": "*****",
      "supportsToolCall": false,
      "supportsImages": false,
      "supportsReasoning": true
    }
  ]
}
```

注意事项：

（1）请将 id 和 apiKey 替换成你自己购买的模型和 apiKey，可在购买页查看

（2）这里使用的 APP Key 管理的请求地址，若使用编程 TokenPlan 套餐的请求地址，请在 url 自定义请求地址的 v1 前加上 coding，类似于.../coding/v1/chat/completions

（3）name 是自定义显示在 codebuddy 的名字前缀，可自定义。

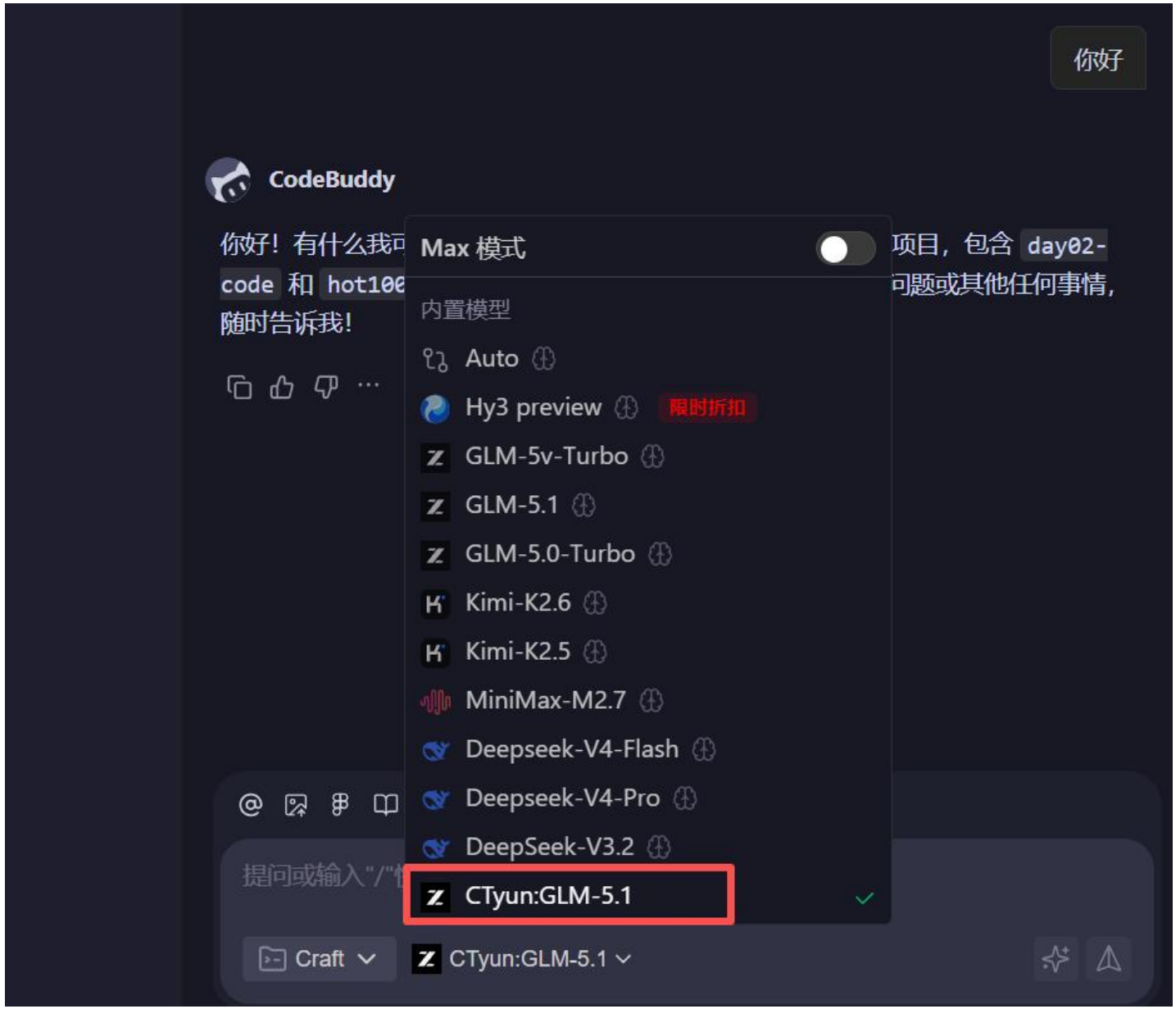
配置字段说明：

字段	类型	必填	说明
id	string	✓	模型唯一标识符
name	string	-	模型显示名称
vendor	string	-	模型供应商（如 OpenAI, Google）
apiKey	string	-	API 密钥（实际密钥值，非环境变量名）
maxInputTokens	number	-	最大输入 token 数
maxOutputTokens	number	-	最大输出 token 数
url	string	-	API 端点 URL（必须是接口完整路径）
supportsToolCall	boolean	-	是否支持工具调用
supportsImages	boolean	-	是否支持图片输入

supportsReasoning	boolean	-	是否支持推理模式

使用天翼云自定义模型

在保存 models.json 重启 CodeBuddy 后即可在右侧栏底部查看自定义模型并开始使用


你好
CodeBuddy
你好！有什么我可
项目，包含 day02-
code 和 hot100
问题或其他任何事情，
随时告诉我！
Max 模式
内置模型
Auto
Hy3 preview
限时折扣
GLM-5v-Turbo
GLM-5.1
GLM-5.0-Turbo
Kimi-K2.6
Kimi-K2.5
MiniMax-M2.7
Deepseek-V4-Flash
Deepseek-V4-Pro
DeepSeek-V3.2
CTyun:GLM-5.1
提问或输入"/"
Craft
CTyun:GLM-5.1

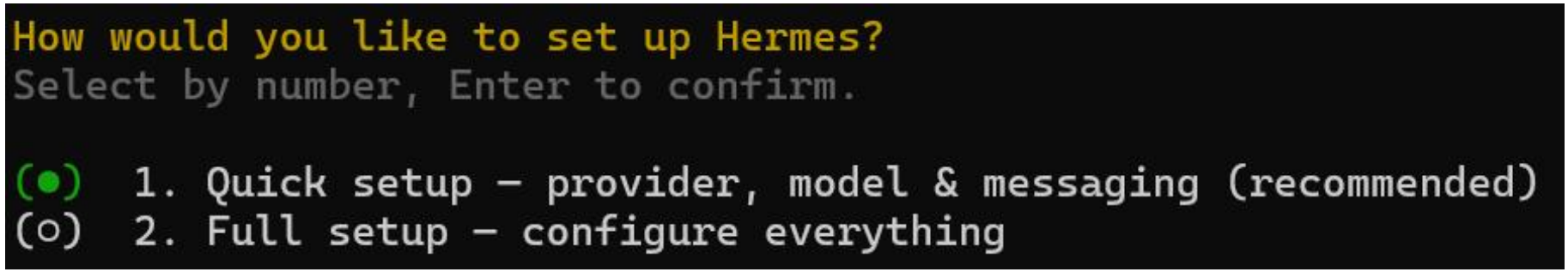
5.1.11 Hermes Agent

windows 流程

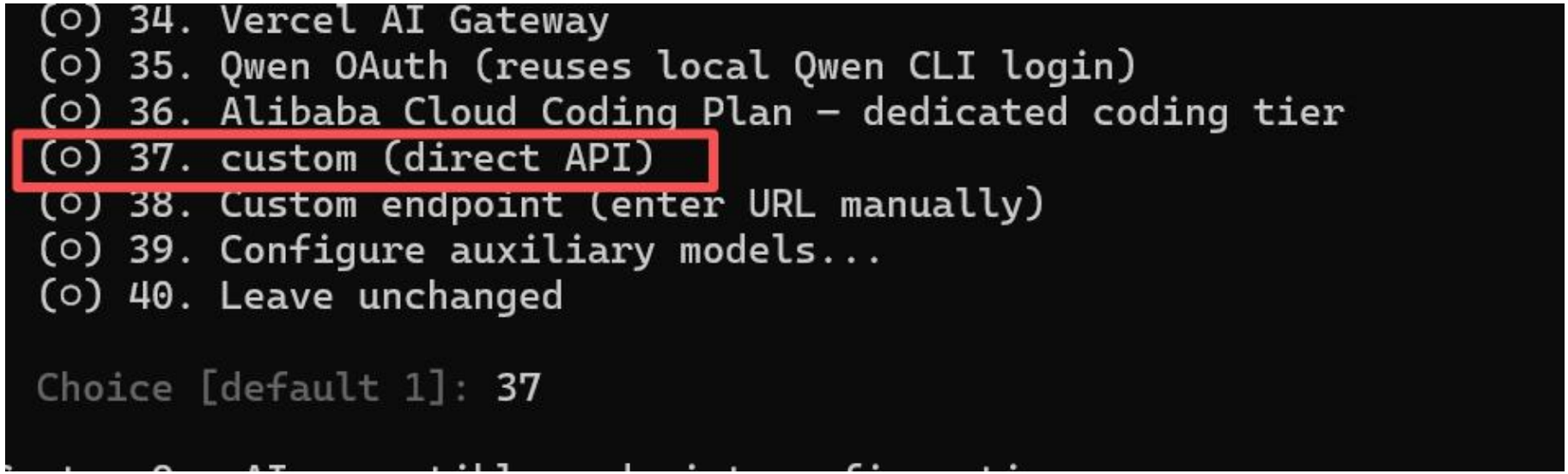
用管理员身份运行 PowerShell,运行指令：

```
    irm
https://ghfast.top/https://raw.githubusercontent.com/NousRe
search/hermes-agent/main/scripts/install.ps1 | iex
```

setup 那里选择 Quick setup



Select provider 那里选择 37 custom



然后就是输入 url（https://ai.ctaigw.cn/v1/）->输入 API key->选择 auto-detect 自动探索出你的模型->给你的模型进行命名->然后就是一路按 Enter 直到安装成功。

```
🚀 Ready to go!

   hermes              Start chatting
   hermes gateway      Start messaging gateway
   hermes doctor       Check for issues


+-----------------------------------------------------------+
|              [OK] Installation Complete!                  |
+-----------------------------------------------------------+

* Your files:

   Config:    C:\Users\twb\AppData\Local\hermes\config.yaml
   API Keys:  C:\Users\twb\AppData\Local\hermes\.env
   Data:      C:\Users\twb\AppData\Local\hermes\cron\, sessions\, logs\
   Code:      C:\Users\twb\AppData\Local\hermes\hermes-agent\

-----------------------------------------------------------

* Commands:

   hermes              Start chatting
   hermes setup        Configure API keys & settings
   hermes config       View/edit configuration
   hermes config edit  Open config in editor
   hermes gateway      Start messaging gateway (Telegram, Discord, etc.)
   hermes update       Update to latest version

-----------------------------------------------------------

[*] Restart your terminal for PATH changes to take effect

PS C:\Users\twb> hermes
```

安装成功后在输入 hermes 开始聊天



mac 流程

在终端中执行以下命令安装 Hermes Agent：

```
    curl -fsSL
https://raw.githubusercontent.com/NousResearch/hermes-
agent/main/scripts/install.sh | bash
```

安装完成后，重新加载终端环境：

```
source ~/.bashrc    # 如果使用 zsh，改为 source ~/.zshrc
```

运行以下命令验证安装。若有版本号输出，则表示安装成功：

```
hermes --version
```

运行 hermes setup 命令就会跳到和上面 windows 一样的安装引导：

```
hermes setup
```

或者你也可以直接配置不走安装引导：

```
hermes config set model.provider custom
hermes config set model.base_url https://ai.ctaigw.cn/v1
hermes config set model.api_key 你的APIKey
hermes config set model.default 你的模型ID
```

后面也是运行 hermes 开始聊天。

5.1.12 Codex桌面版

前置准备

首先安装 codex 桌面版，安装链接如下：

```
https://developers.openai.com/codex/app
```

然后安装 CC-Switch v3.16.1+（低于此版本不支持格式转换，会报错），链接如下：

```
https://github.com/farion1231/cc-switch/releases/latest
```

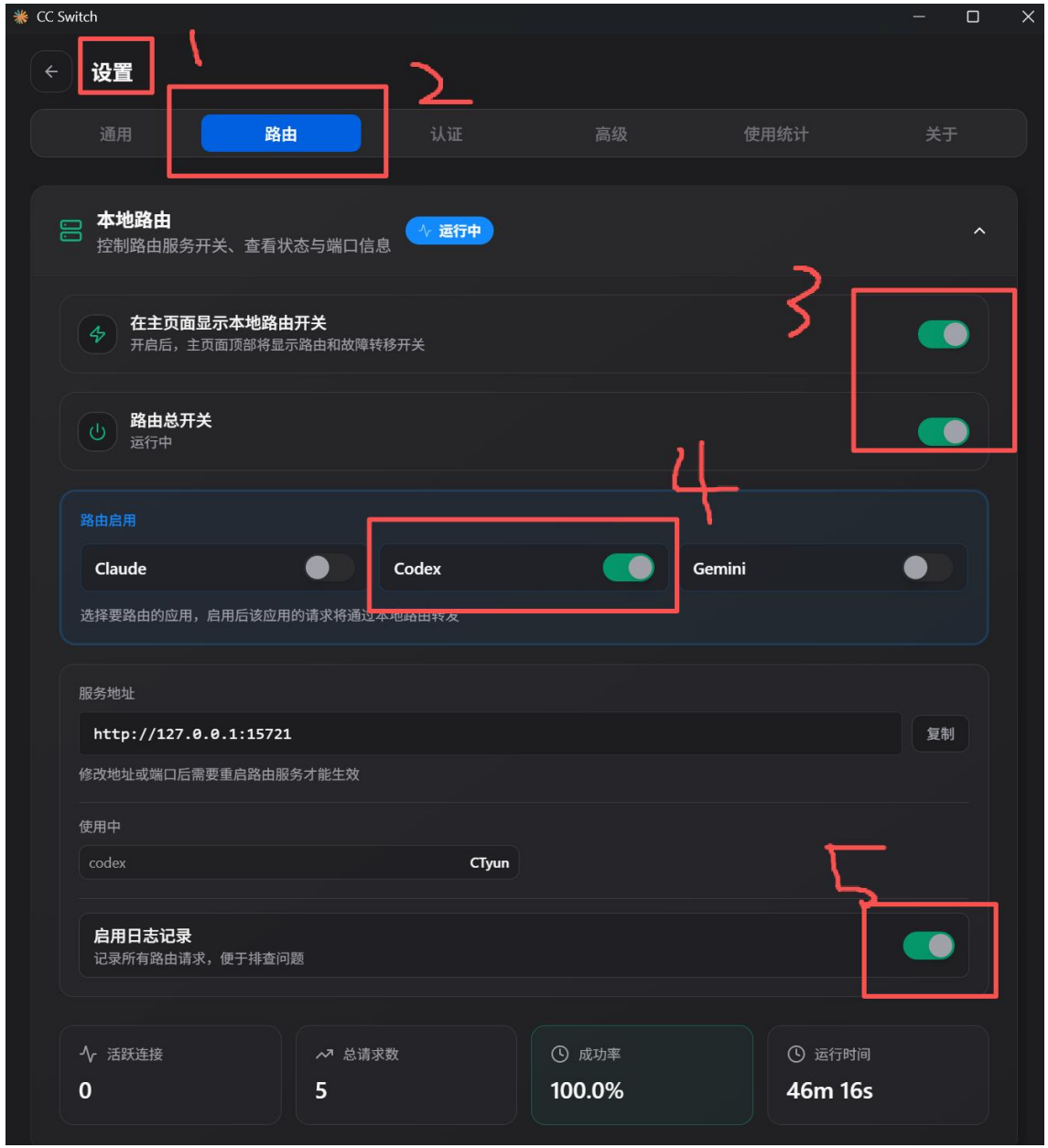
购买编程 Token Plan 后获取专属 APP Key 和专属请求地址，获取链接如下：

```
https://ctxirang.ctyun.cn/maas/codingPlan
```

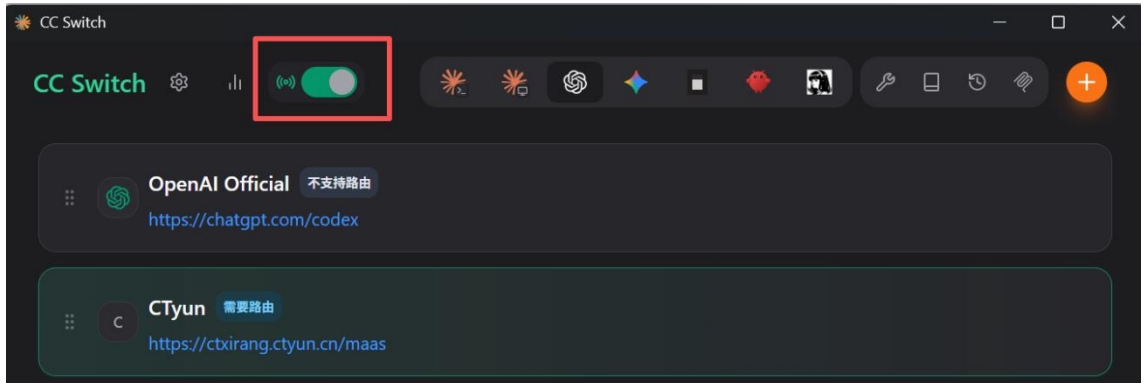
在 CC-Switch 中添加供应商

说明：Codex 桌面版只支持 OpenAI Responses API 格式，而天翼云只支持 Chat Completions 格式，CC-Switch v3.16+ 的代理负责两种格式之间的互转。

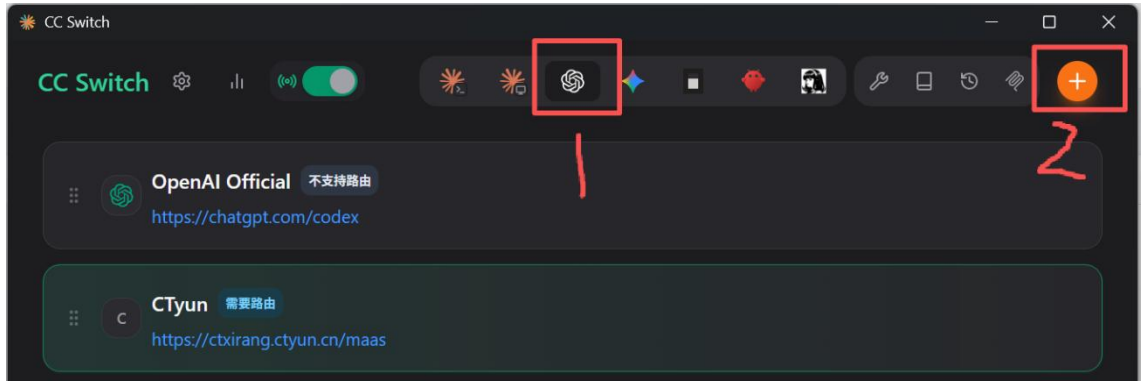
首先开启在 CC-Switch 中添加 Codex 的路由


CC Switch
设置
1
2
通用
路由
认证
高级
使用统计
关于
本地路由
控制路由服务开关、查看状态与端口信息
运行中
3
在主页面显示本地路由开关
开启后，主页面顶部将显示路由和故障转移开关
路由总开关
运行中
4
路由启用
Claude
Codex
Gemini
选择要路由的应用，启用后该应用的请求将通过本地路由转发
服务地址
http://127.0.0.1:15721
复制
修改地址或端口后需要重启路由服务才能生效
使用中
codex
CTyun
5
启用日志记录
记录所有路由请求，便于排查问题
活跃连接
0
总请求数
5
成功率
100.0%
运行时间
46m 16s

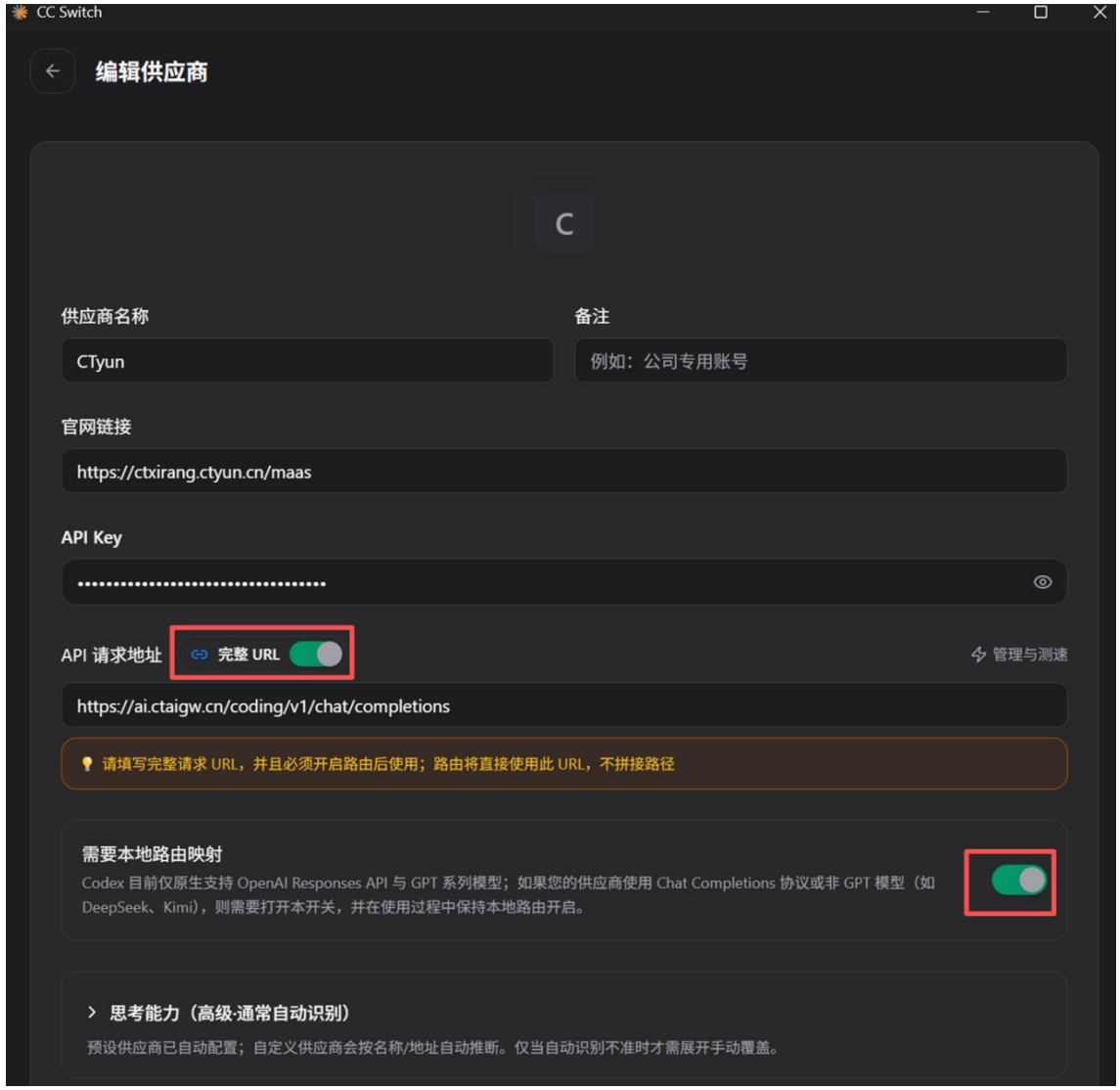
查看是否成功



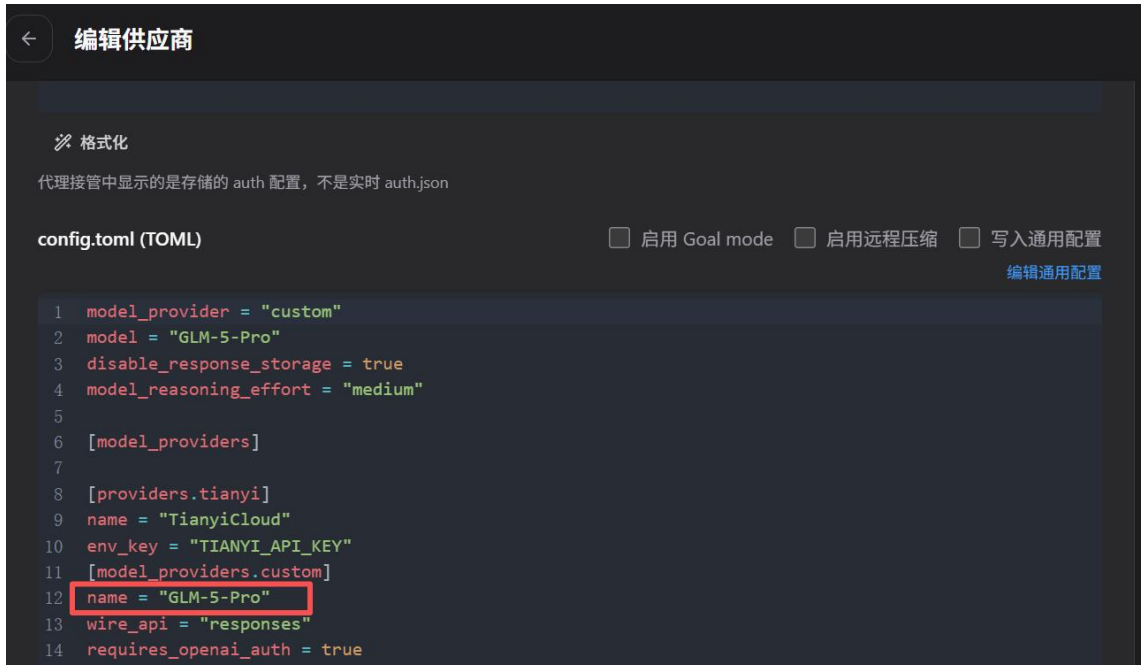
选择 Codex → 添加供应商：



填写相应内容，注意一定开启需要本地路由映射：



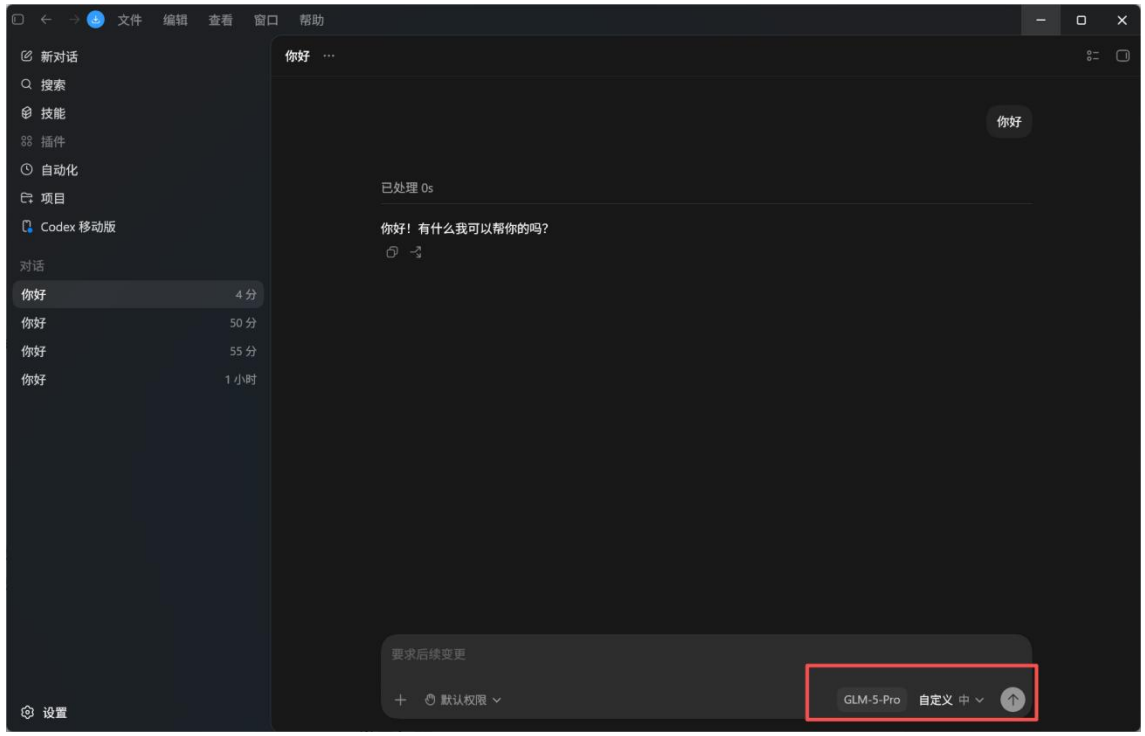
默认模型名称是 custom，如果想改成自己模型的真正名称，下滑在 config.toml 更改（这个文件对应本地的~/.codex/config.toml）



然后点击保存即可

Codex 对话验证

打开 codex，就可看到自定义模型：



随后可以开始对话了。

5.2 DeepSeek模型调用-快捷版

本章节是关于开发者通过对话工具（比如 Chatbox 等）以 OpenAI-API-compatible 的方式快速添加使用模型的实践。

1.天翼云官网获取模型调用APP key等信息

模型API终端请求地址

请求地址

API终端请求地址：https://ai.ctaigw.cn/v1/chat/completions

通信协议

接口通过 HTTPS 进行通信，保护用户数据的机密性和完整性，确保网络通信的安全性。

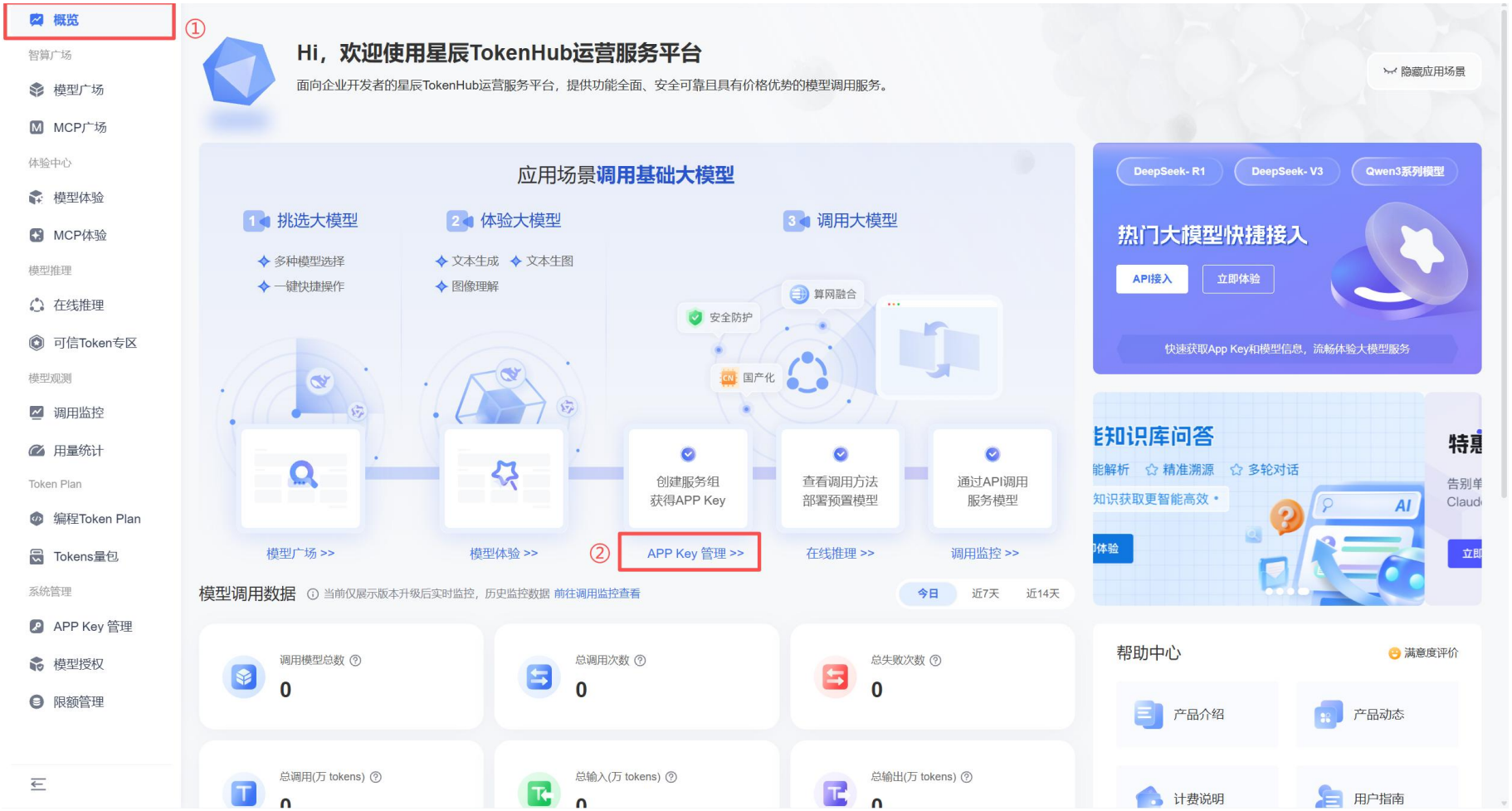
请求方法

支持的 HTTP 请求方法：POST。

POST 请求支持的 Content-Type 类型：application/json。

获取模型调用APP key

访问星辰TokenHub运营服务平台地址：https://ctxirang.ctyun.cn/maas/home，在左侧菜单栏进入“系统管理-APP Key管理”页。

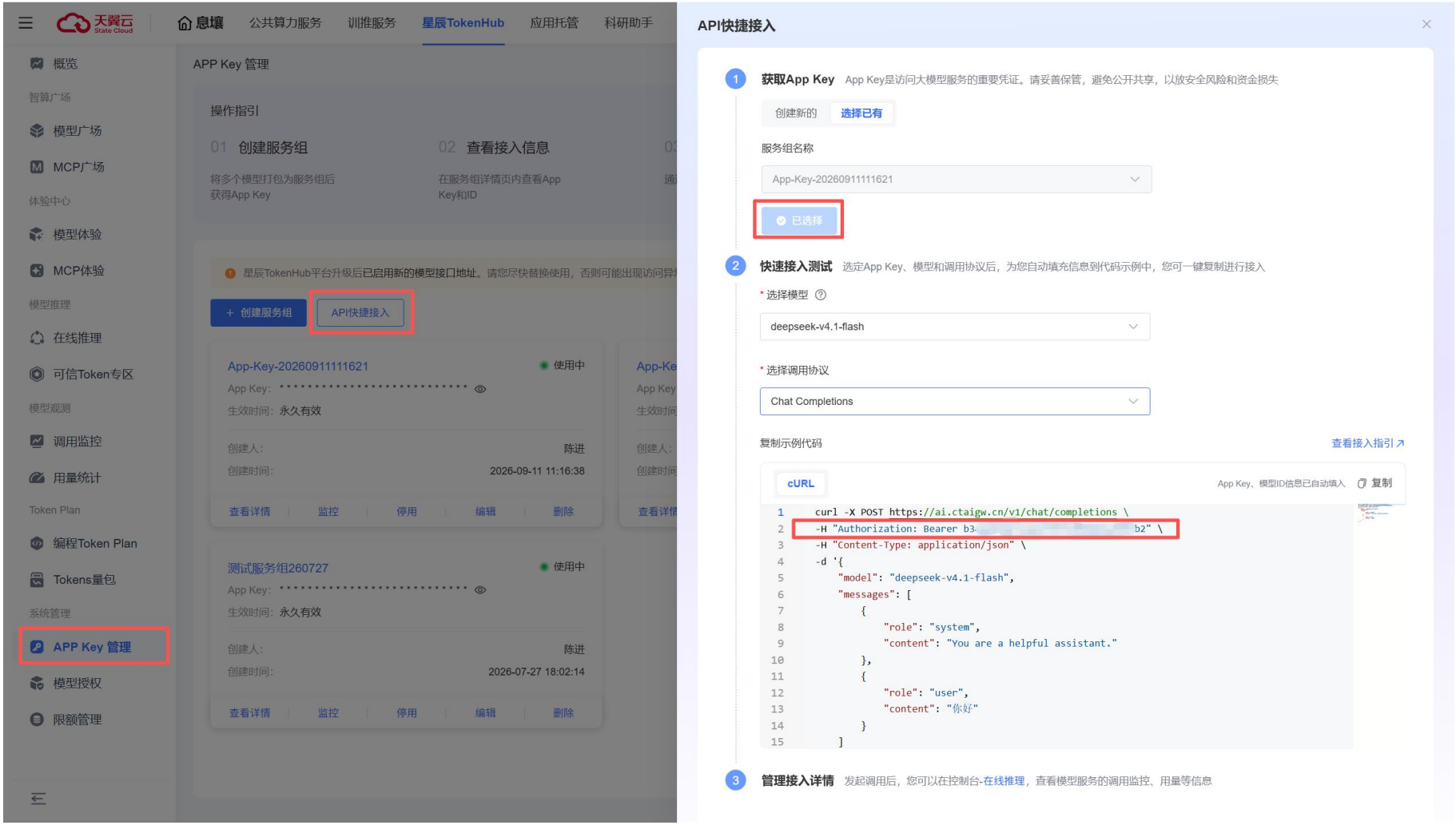


在“APP Key 管理”页，点击“API 快捷接入”拉起右侧抽屉弹窗，在 API 快捷接入页面：

选择服务组：从下拉菜单中选择第一步创建的服务组。如果尚未创建，可点击“确认创建并选择”快速创建默认服务组。

选择模型：从模型下拉菜单中选择需要接入的模型。

选择完成后，页面下方将自动展示该模型支持的所有编程语言的示例代码。



可进入左侧菜单栏的“系统管理-APP Key 管理”页，查看所创建的服务组及对应 APP Key。

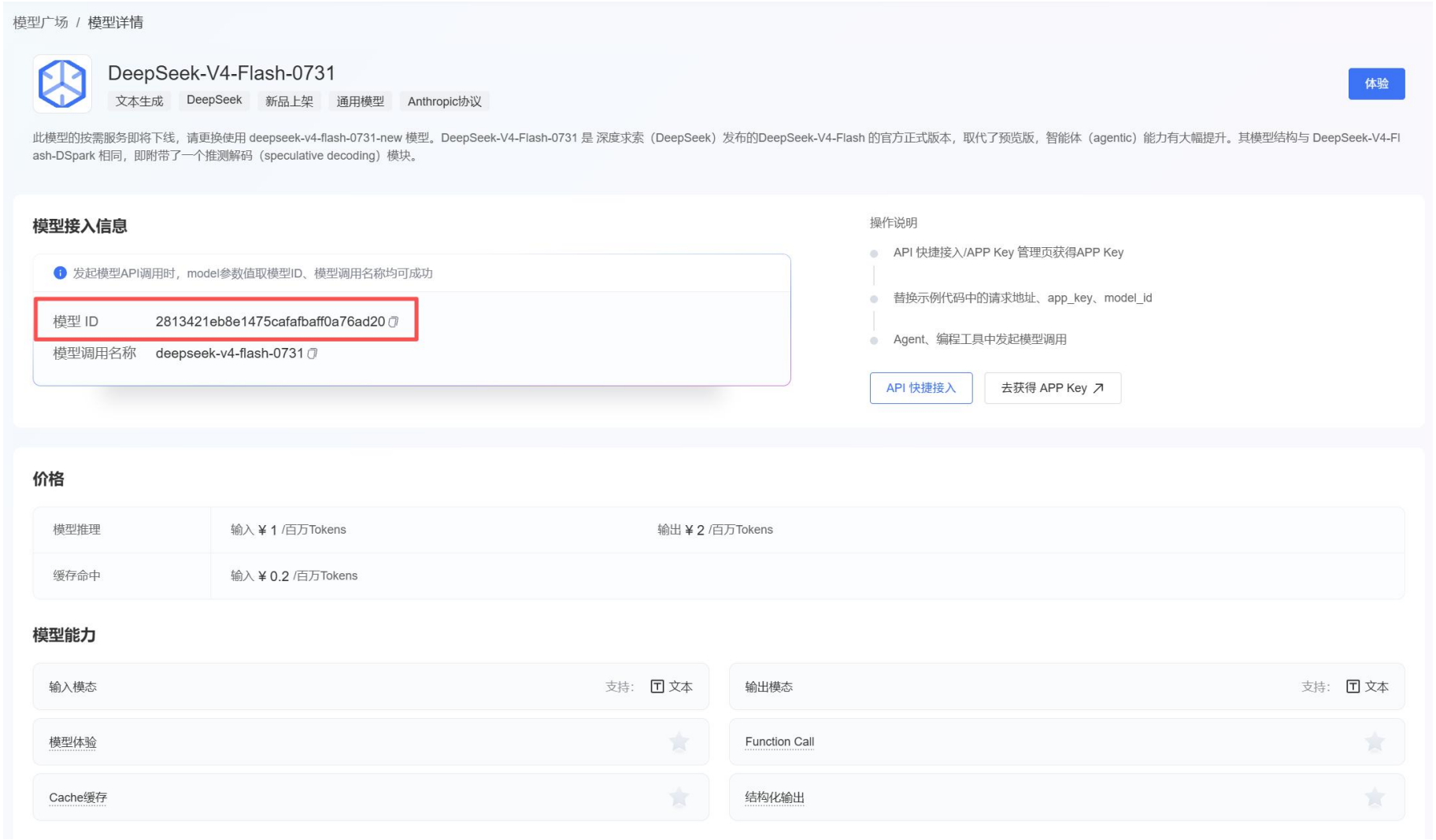


获取调用的模型名称

左侧导航栏切换到”模型广场“，选择对应想要调用的模型。



获取模型ID作为模型名称，例如下图中模型ID为：2813421eb8e1475cafafbaff0a76ad20。



DeepSeek模型版本选择参考：

DeepSeek-R1：适合需要高级推理能力和高精度的复杂任务，如科研、金融建模等。响应时间相对长。

DeepSeek-V3：适合需要快速响应和高性价比的通用自然语言处理任务，如智能客服、内容生成等。响应时间相对短。

DeepSeek-V4：适合高并发企业级 Agent 部署，海量数据结构化与清洗，日常代码辅助与工具调用，实时流式翻译与交互。

2.手动验证key的可用性

下列请求为DeepSeek-R1-昇腾版请求样例，如果正常返回deepseek回答，代表没有问题。

注意：

以下curl请求中替换成自己的APP KEY后类似“--header 'Authorization: Bearer abcdefghxxssssss' \ ”

```
curl --location 'https://ai.ctaigw.cn/v1/chat/completions' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer 替换成自己的APP KEY' \
--data '{
    "messages": [
        {
            "role": "system",
            "content": "You are a test assistant."
        },
        {
            "role": "user",
            "content": "Testing. Just say hi and nothing else."
        }
    ],
    "model": "4bd107bff85941239e27b1509eccfe98"
}'
```

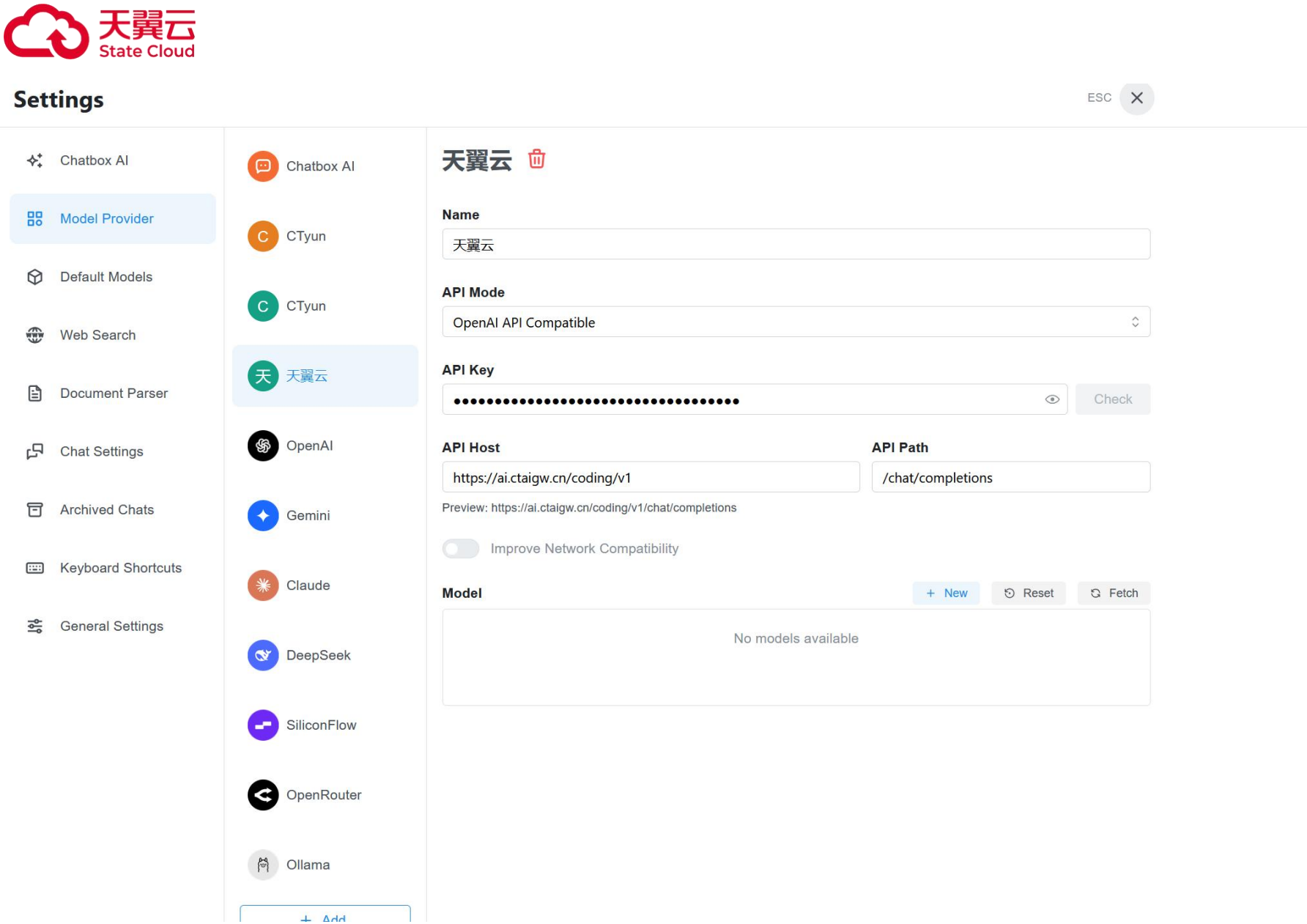
说明

部分第三方工具调用可能存在接口不适配问题，建议优先使用以下几款工具。

配置下列工具时，API key等同于上述内容中创建的服务组APP key。

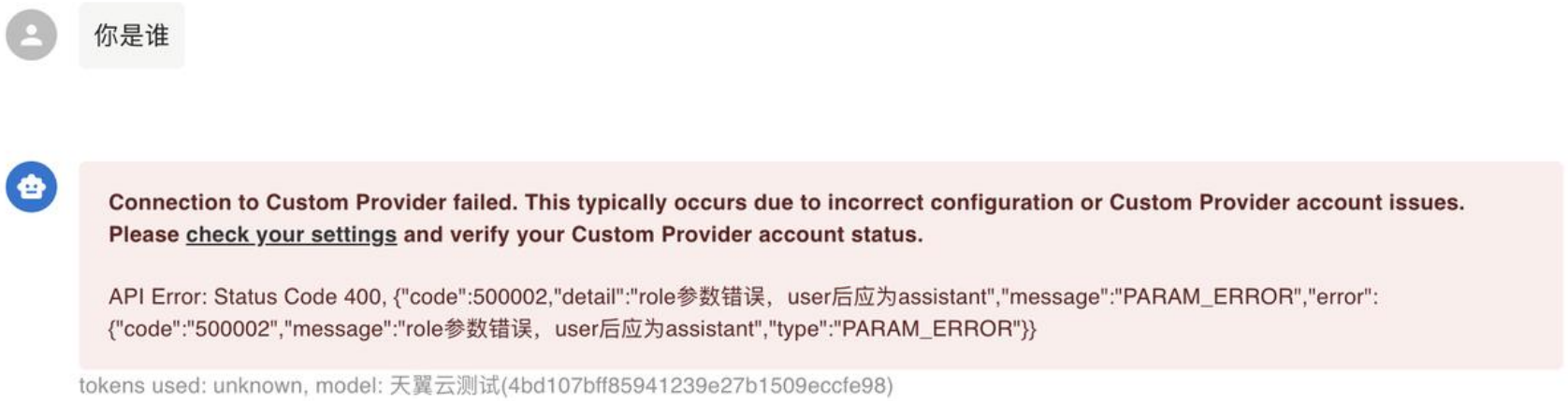
Chatbox

访问链接：https://web.chatboxai.app/，点击Setting，然后配置Api Host、API path、API key和Model，就可以使用了。


天翼云
State Cloud
Settings
ESC
Chatbox AI
Model Provider
Default Models
Web Search
Document Parser
Chat Settings
Archived Chats
Keyboard Shortcuts
General Settings
Chatbox AI
CTyun
CTyun
天翼云
OpenAI
Gemini
Claude
DeepSeek
SiliconFlow
OpenRouter
Ollama
天翼云
Name
天翼云
API Mode
OpenAI API Compatible
API Key
Check
API Host
https://ai.ctaigw.cn/coding/v1
API Path
/chat/completions
Preview: https://ai.ctaigw.cn/coding/v1/chat/completions
Improve Network Compatibility
Model
New
Reset
Fetch
No models available

注意

chatbox web如果api setting配置错误，更改后，仍然会报错。报错如下：



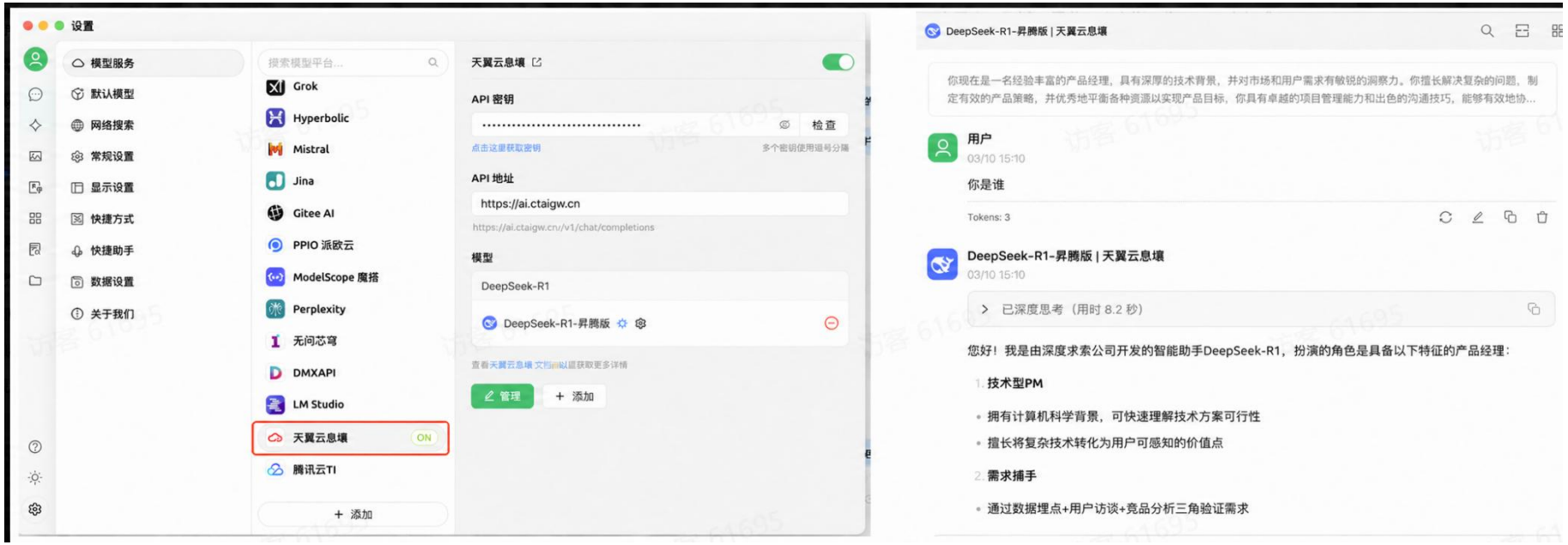
如遇上述报错，建议重新开启一个对话，再次尝试即可。



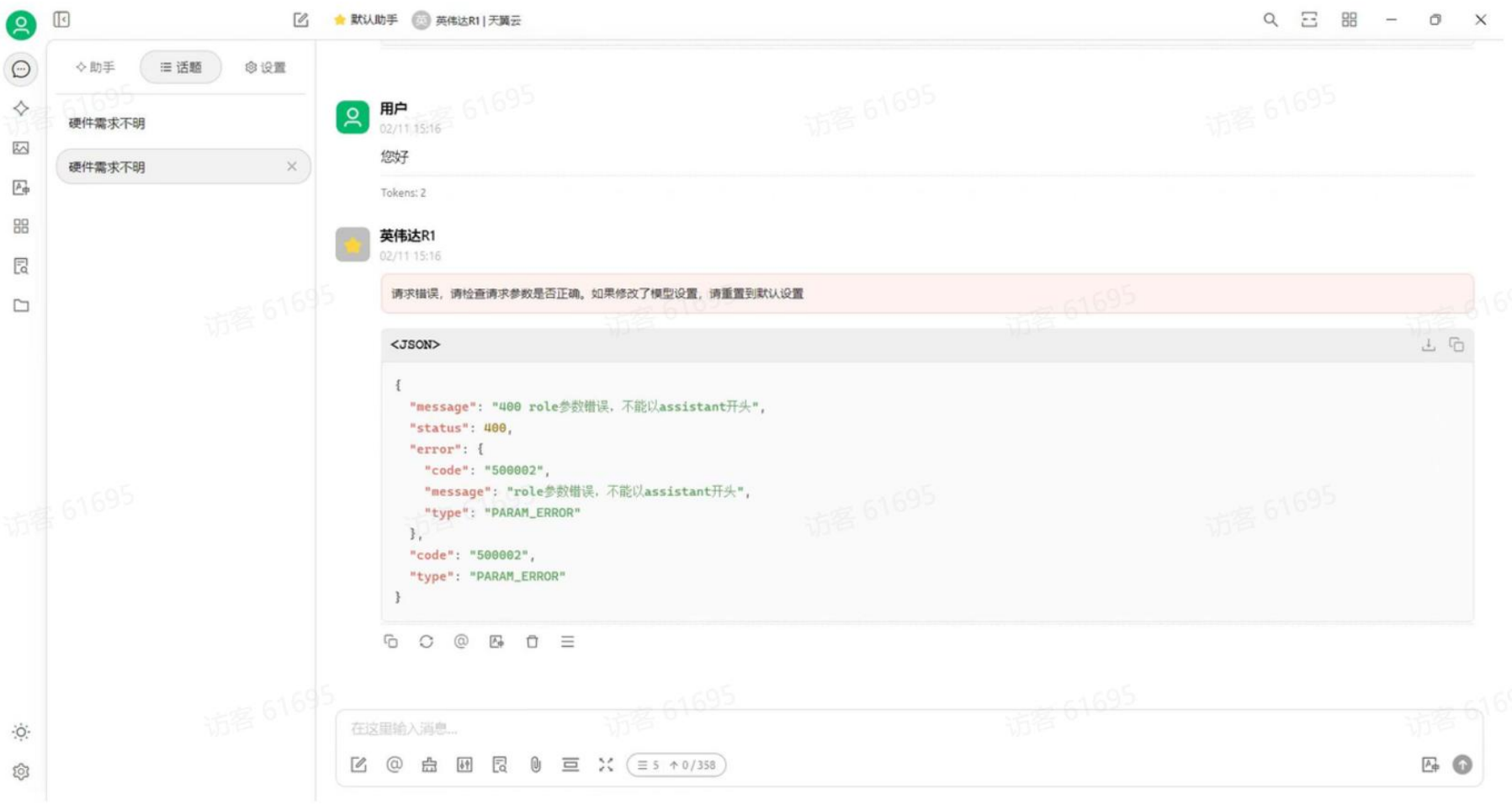
CherryStudio

访问链接：https://cherry-ai.com/

按照下图配置即可。



一旦出现以下报错，重新开启一个对话，再次尝试，则可以看到正确的对话。



Langchain

参考langchain的自定义模型文档通过OpenAI协议调用即可。

```
    from langchain.chat_models import ChatOpenAI
    from langchain.schema import HumanMessage

    def main():
        # 初始化LangChain模型，配置自定义API端点
        llm = ChatOpenAI(
            model_name="9dc913a037774fc0b248376905c85da5",  # 自定义模型
名称
            # 从环境变量获取API密钥，如果没有设置，也可以直接终端执行export
TYY_API_KEY="xxx"
            openai_api_key=os.getenv("XIRANG_API_KEY")
            openai_api_base="https://ai.ctaigw.cn/v1",  # API基础URL
            temperature=0.7
        )

        # 创建一个聊天消息
        message = HumanMessage(content="你是谁")

        # 使用LangChain进行对话
        response = llm([message])
        print(response.content)


    if __name__ == "__main__":
        main()
```

Dify

说明

前置条件：本地下载并安装Docker。

访问 Docker 官方网站，点击“Download Docker Desktop”按钮，选择合适的 Windows、macOS 或 Linux版本下载并安装。

拉取dify镜像到本地

```
# 如果本地已安装dify可以忽略
# 假设当前最新版本为 0.15.3
git clone https://github.com/langgenius/dify.git --branch 0.15.3
```

修改配置参数

```
cd docker
mv .env.example .env
```

启动并验证

```
#执行：
docker compose up -d
# 等服务启动访问
http://localhost/apps
```

配置天翼云息壤模型

配置方式一：直接配置方式（所有版本均可使用）

通过直接配置OpenAI-API-compatible支持，配置方式如下：

配置方式二：使用息壤镜像配置方式（仅支持0.15.3版本）

修改env配置

```
cd docker
#在.env文件添加
CUSTOM_MODEL_ENABLED=true
OLLAMA_API_BASE_URL=host.docker.internal:11434
```

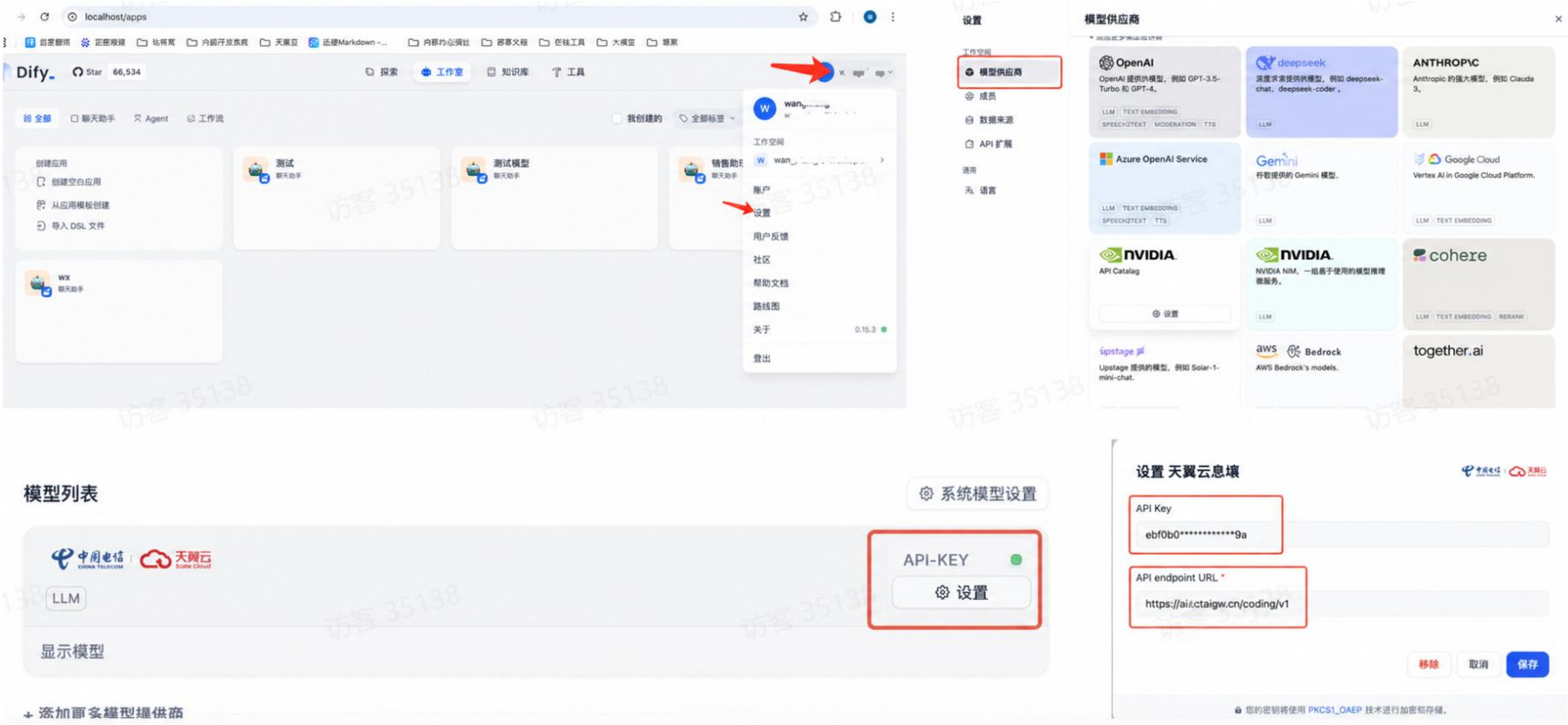
修改下api和work的镜像地址，改为增加了xirang模型供应商的dify 0.15.3版本。

修改后需要重启dify。

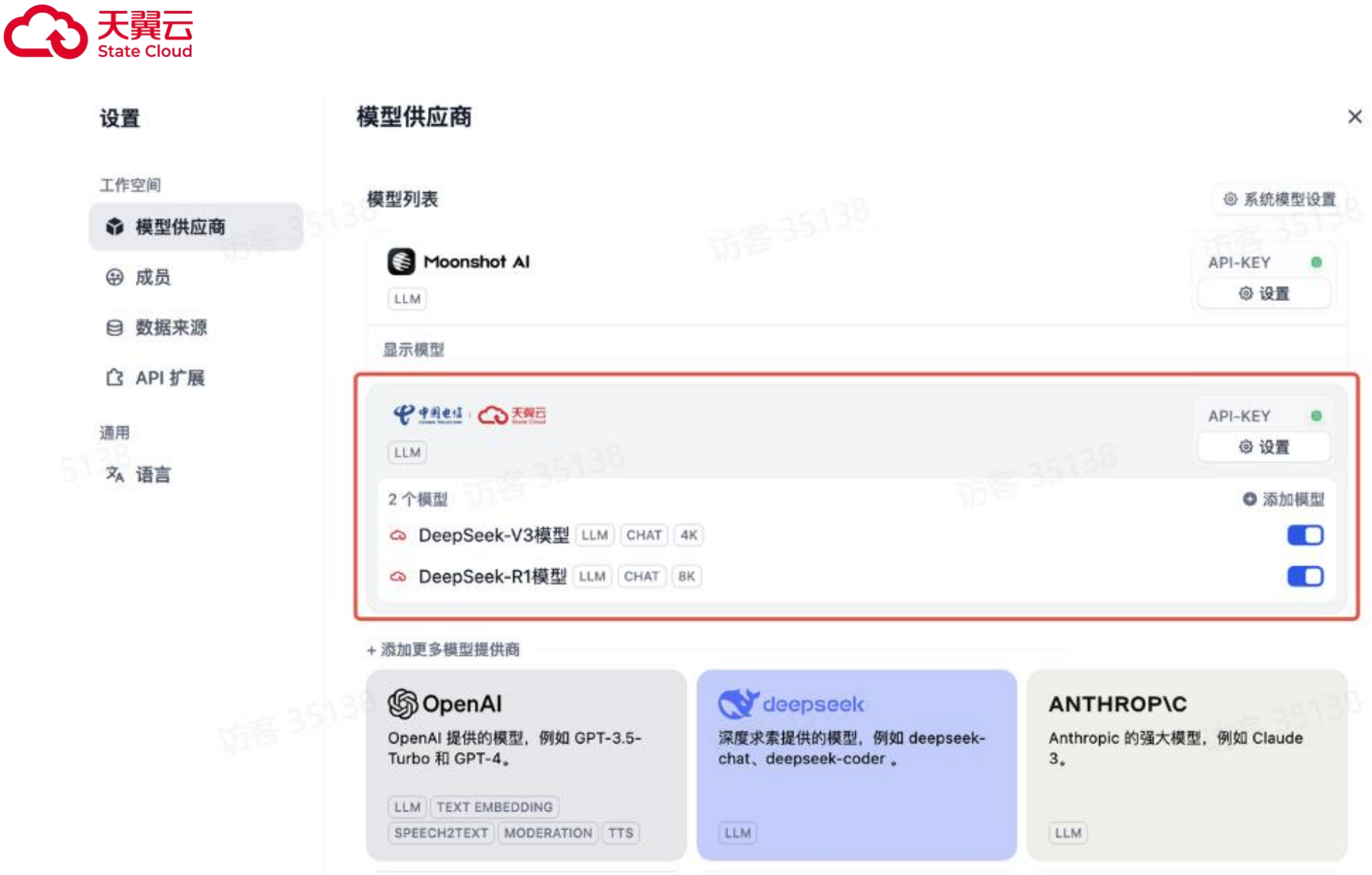
```
    services:
      # API service with Xirang model provider
      api:
        image: telecom-eci-huadong1-crs-registry-
huadong1.crs.ctyun.cn/xirang/dify-with-xirang:0.15.3 # 原来为
langgenius/dify-api:0.15.3


       # worker service
      # The Celery worker for processing the queue.
      worker:
        image: telecom-eci-huadong1-crs-registry-
huadong1.crs.ctyun.cn/xirang/dify-with-xirang:0.15.3 # 原来为
langgenius/dify-api:0.15.3
```

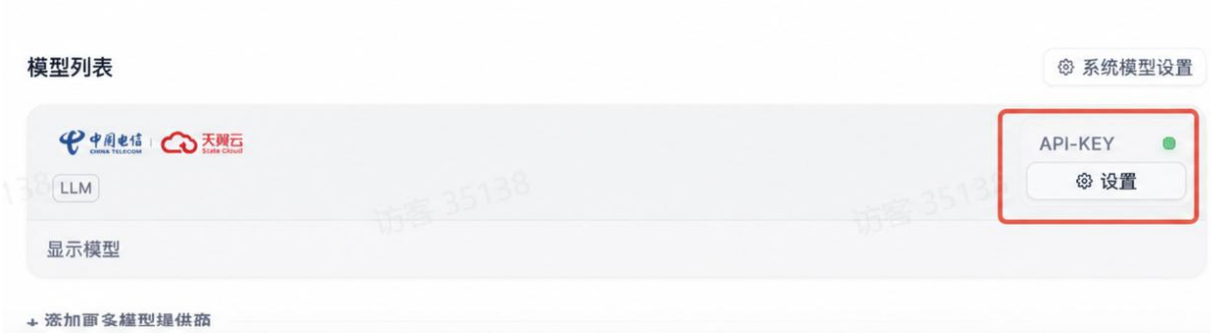
在个人设置里面点击模型供应商，选择天翼云模型。



点击保存后，可以看到模型的配置。


设置
工作空间
模型供应商
成员
数据来源
API 扩展
通用
语言
模型供应商
模型列表
系统模型设置
Moonshot AI
LLM
API-KEY
设置
显示模型
LLM
API-KEY
设置
2 个模型
添加模型
DeepSeek-V3模型 LLM CHAT 4K
DeepSeek-R1模型 LLM CHAT 8K
+ 添加更多模型提供商
OpenAI
OpenAI 提供的模型，例如 GPT-3.5-Turbo 和 GPT-4。
LLM TEXT EMBEDDING SPEECH2TEXT MODERATION TTS
deepseek
深度求索提供的模型，例如 deepseek-chat、deepseek-coder 。
LLM
ANTHROP\C
Anthropic 的强大模型，例如 Claude 3。
LLM

其他非预置模型：





配置对话助手





3.常见问题（0.15.3版本）

如遇到无法展示问题，可能是因为您的dify版本为最新版，我们目前还不支持，可执行以下操作切换到0.15.3版本并重启服务。

```
dify插件需要切换版本重新启动服务才可以达成兼容
git clone  https://gitcode.com/gh_mirrors/di/dify.git
cd dify
git checkout 7796984444191c639bd3c541a44e832b17ad1cae
```

如遇到息壤镜像失败问题，可切换到dify原始镜像，通过配置直接配置OpenAI-API-compatible支持，配置方式如下：

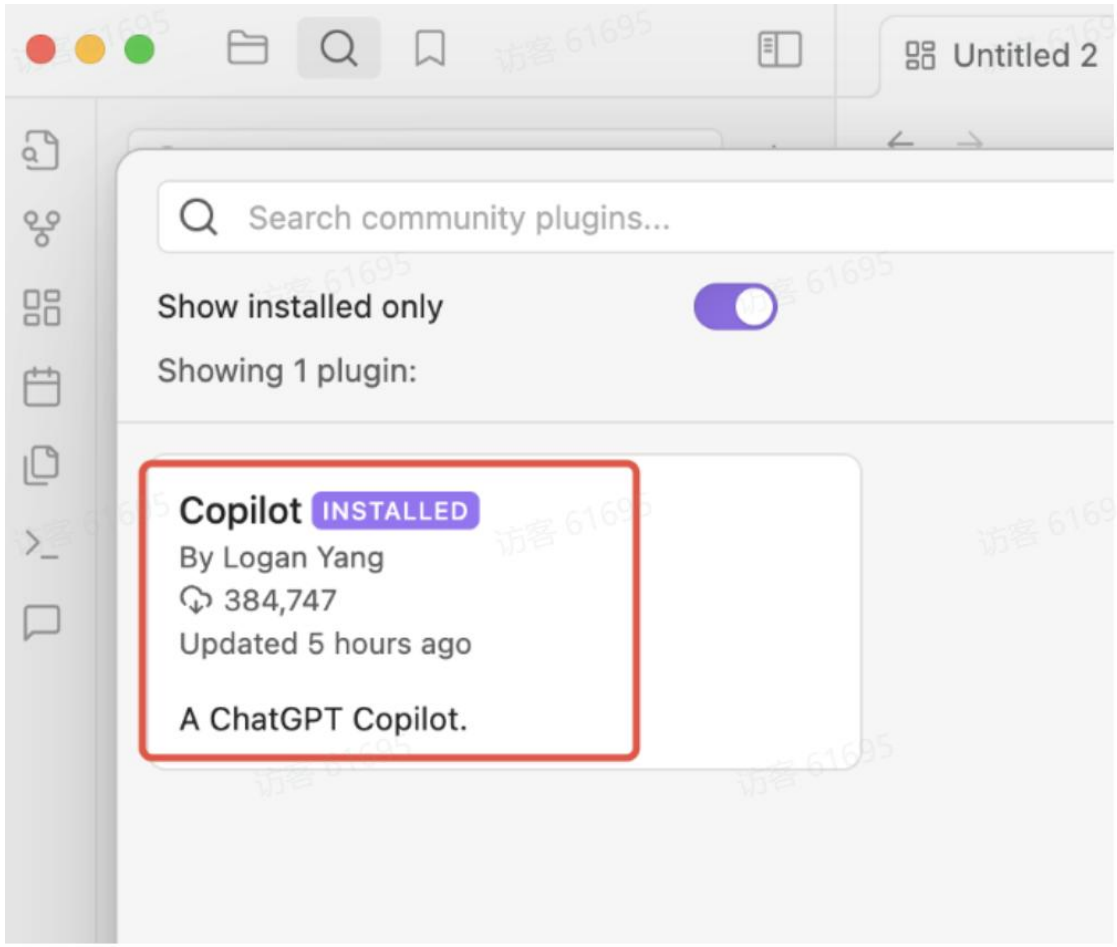


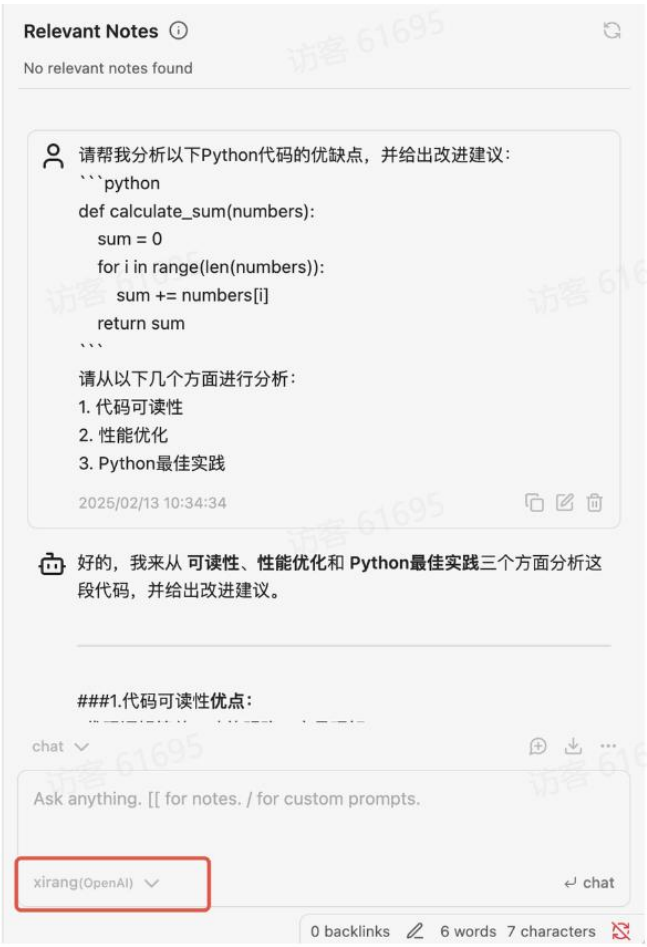
Obsidian copilot

打开 Obsidian，进入设置面板，选择“社区插件”。

关闭“安全模式”，点击“浏览”按钮搜索“Copilot for Obsidian”并安装。

安装完成后，启用插件。


Untitled 2
Search community plugins...
Show installed only
Showing 1 plugin:
Copilot
INSTALLED
By Logan Yang
384,747
Updated 5 hours ago
A ChatGPT Copilot.


Relevant Notes
No relevant notes found
请帮我分析以下Python代码的优缺点，并给出改进建议：
```python
def calculate_sum(numbers):
sum = 0
for i in range(len(numbers)):
sum += numbers[i]
return sum
```
请从以下几个方面进行分析：
1. 代码可读性
2. 性能优化
3. Python最佳实践
2025/02/13 10:34:34
好的，我来从 可读性、性能优化和 Python最佳实践三个方面分析这段代码，并给出改进建议。
###1.代码可读性优点：
chat
Ask anything. [[for notes. / for custom prompts.
xirang(OpenAI)
chat
0 backlinks
6 words 7 characters

Ragflow

注意

目前息壤model不支持embedding model能力，因此只能作为chat model提供能力。

下载镜像git clone https://github.com/infiniflow/ragflow.git

通过docker安装并启动。

```
cd ragflow/docker
docker compose -f docker-compose-CN.yml up -d
```

检查 RAGFlow 服务是否正常启动。

```
docker logs -f ragflow-server
```

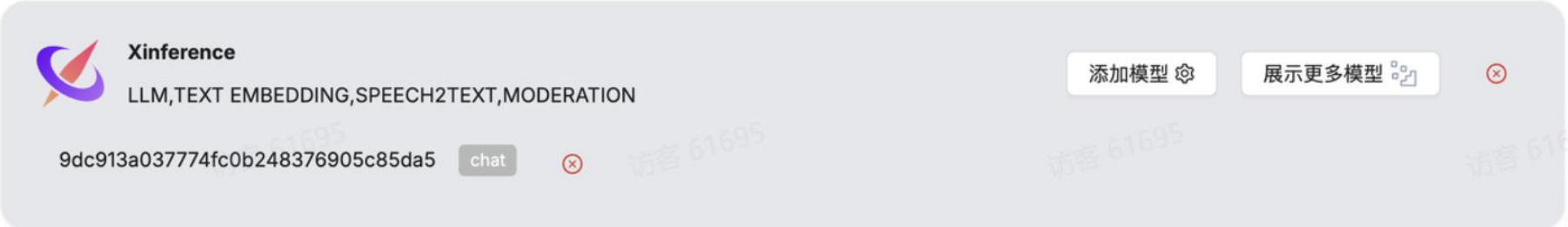
如果看到类似以下输出，则表示启动成功：

```
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:9380
* Running on http://x.x.x.x:9380
```

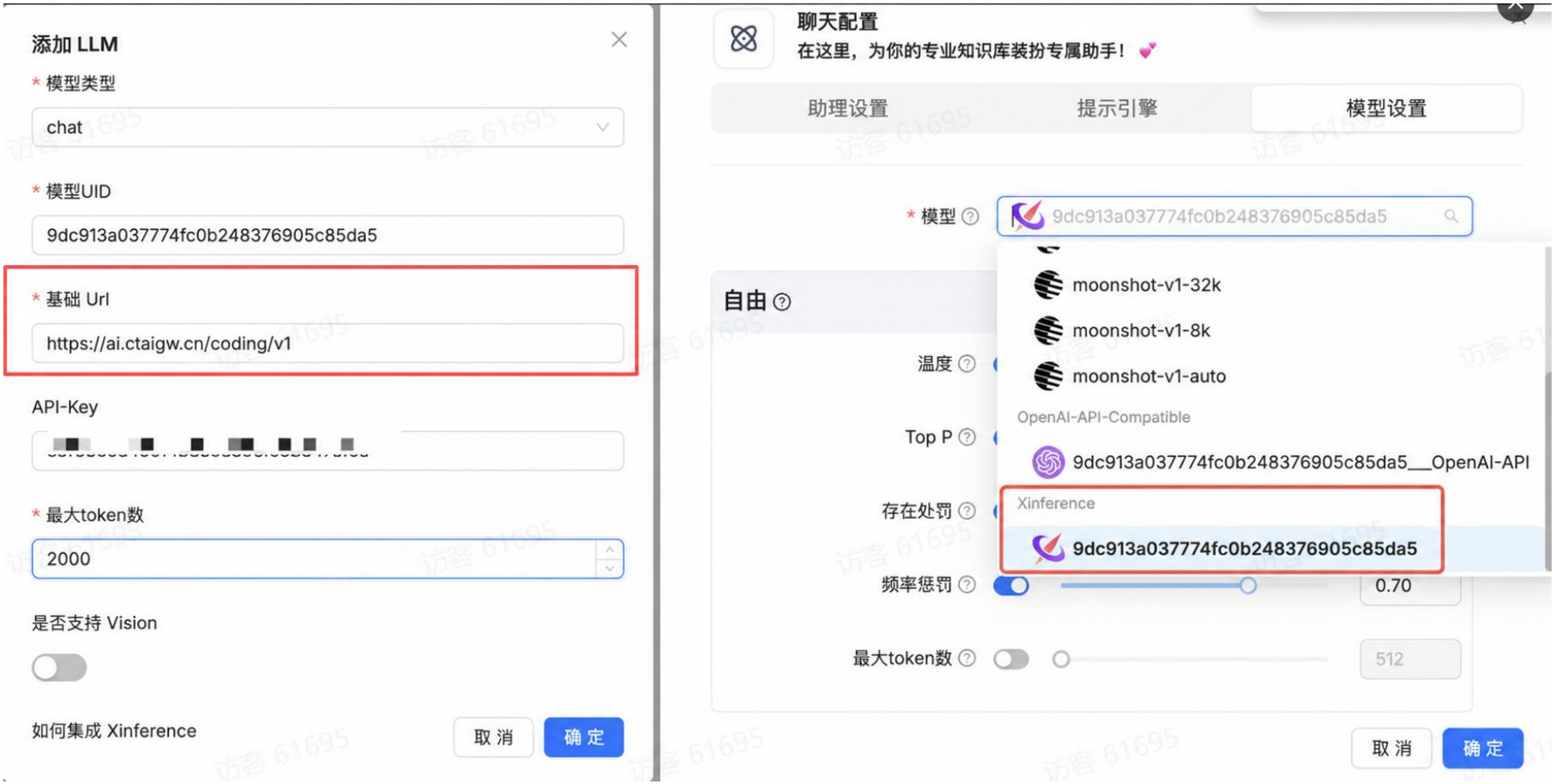
打开浏览器配置模型Xinference服务提供商。

在浏览器中输入服务器的 IP 地址访问 RAGFlow。默认情况下，RAGFlow 使用 HTTP 端口 80，因此无需输入端口号。

选择“Xinference”模型服务商（ Xinference是工厂配置，支持自定义模型名称）。



配置chatmodel的界面如下：



回到对话界面，配置聊天助手：

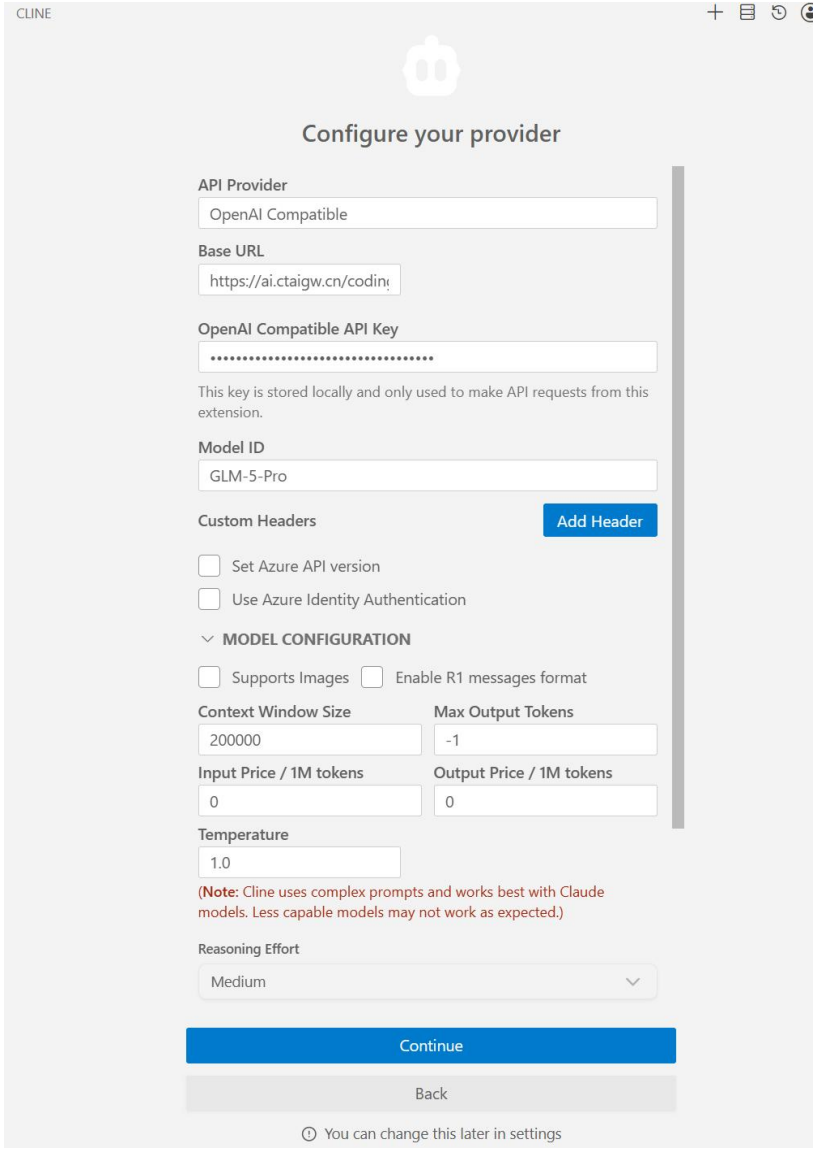

RAGFlow
知识库
聊天
搜索
Agent
文件管理
新建助理
聊天 0
jarvis3-openai
A helpful dialog
jarvi2
A helpful dialog
jarvis
A helpful dialog
聊天配置
在这里，为你的专业知识库装扮专属助手！
助理设置
提示引擎
模型设置
* 助理姓名
例如 贾维斯简历
助理描述
chat.descriptionPlaceholder
助理头像
+
上传
空回复
设置开场白
你好！ 我是你的助理，有什么可以帮到你的吗?
显示引文

开启对话：

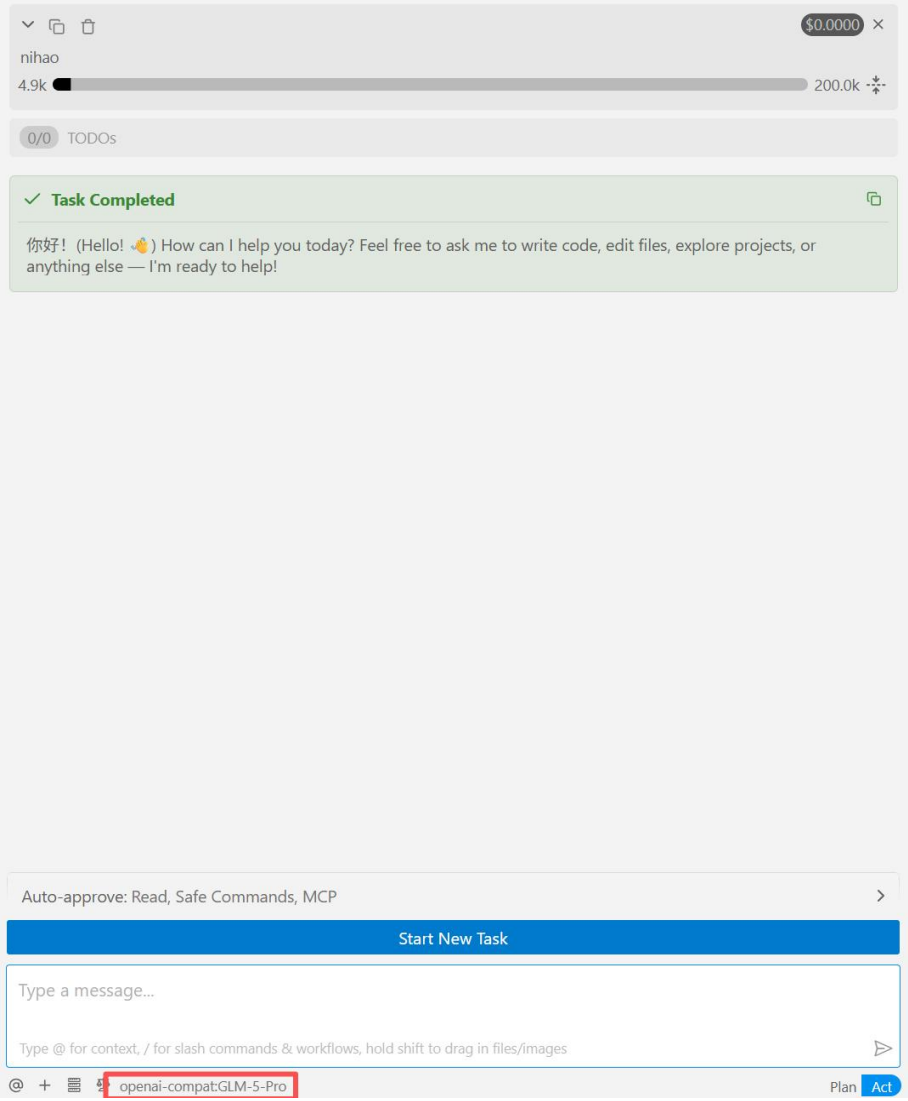


Cline

按照下图配置即可。注意baseUrl配置：https://ai.ctaigw.cn/v1


CLINE
Configure your provider
API Provider
OpenAI Compatible
Base URL
https://ai.ctaigw.cn/codin
OpenAI Compatible API Key
This key is stored locally and only used to make API requests from this extension.
Model ID
GLM-5-Pro
Custom Headers
Add Header
Set Azure API version
Use Azure Identity Authentication
MODEL CONFIGURATION
Supports Images
Enable R1 messages format
Context Window Size
200000
Max Output Tokens
-1
Input Price / 1M tokens
0
Output Price / 1M tokens
0
Temperature
1.0
(Note: Cline uses complex prompts and works best with Claude models. Less capable models may not work as expected.)
Reasoning Effort
Medium
Continue
Back
You can change this later in settings

配置完成后即可开启对话：



Maxkb

通过docker命令拉取部署

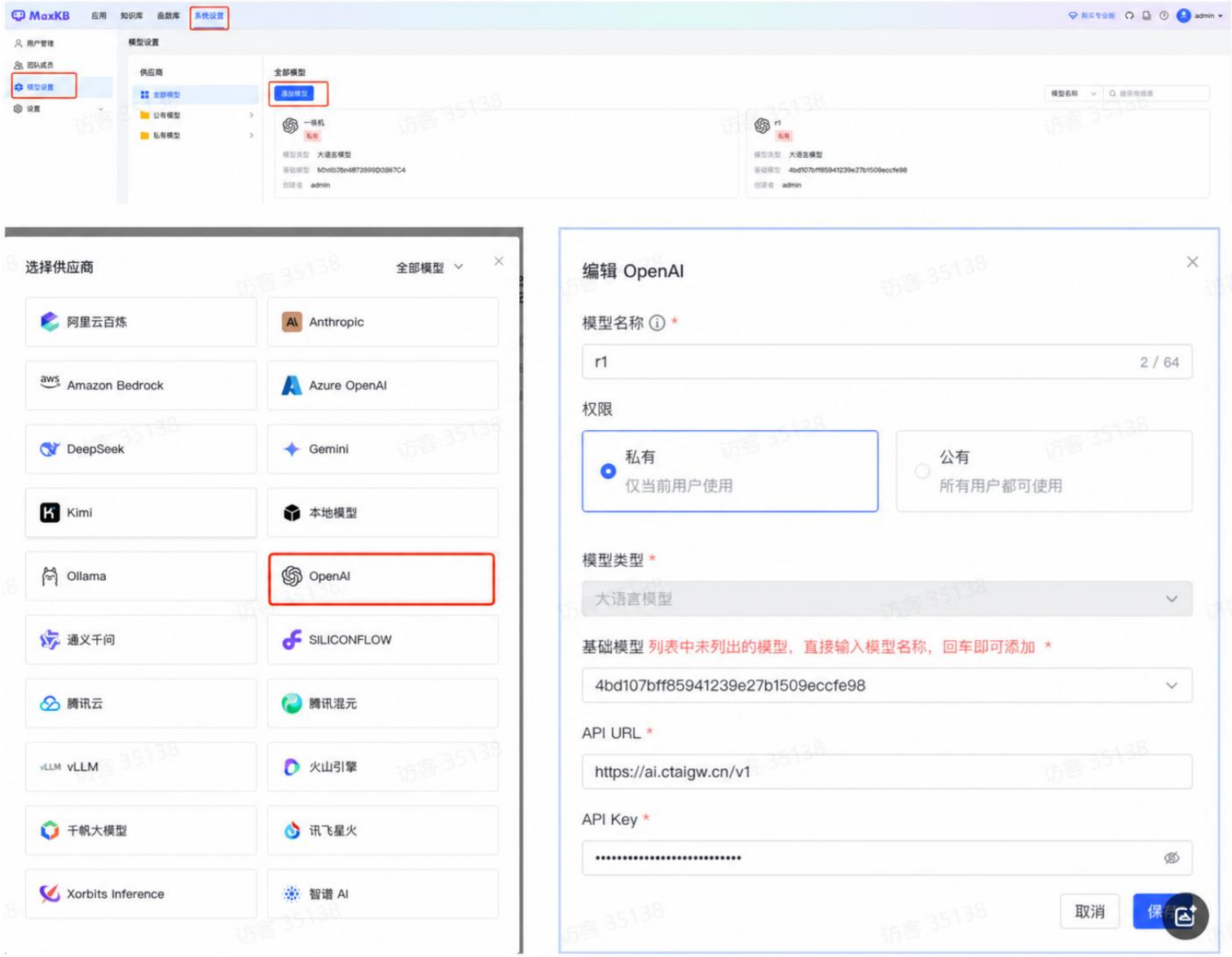
```
    docker run -d --name=maxkb --restart=always -p 8080:8080 -v
~/.maxkb:/var/lib/postgresql/data -v ~/.python-
packages:/opt/maxkb/app/sandbox/python-packages
https://registry.fit2cloud.com/maxkb/maxkb
```

启动后，浏览器访问http://localhost:8080

初始用户名：admin

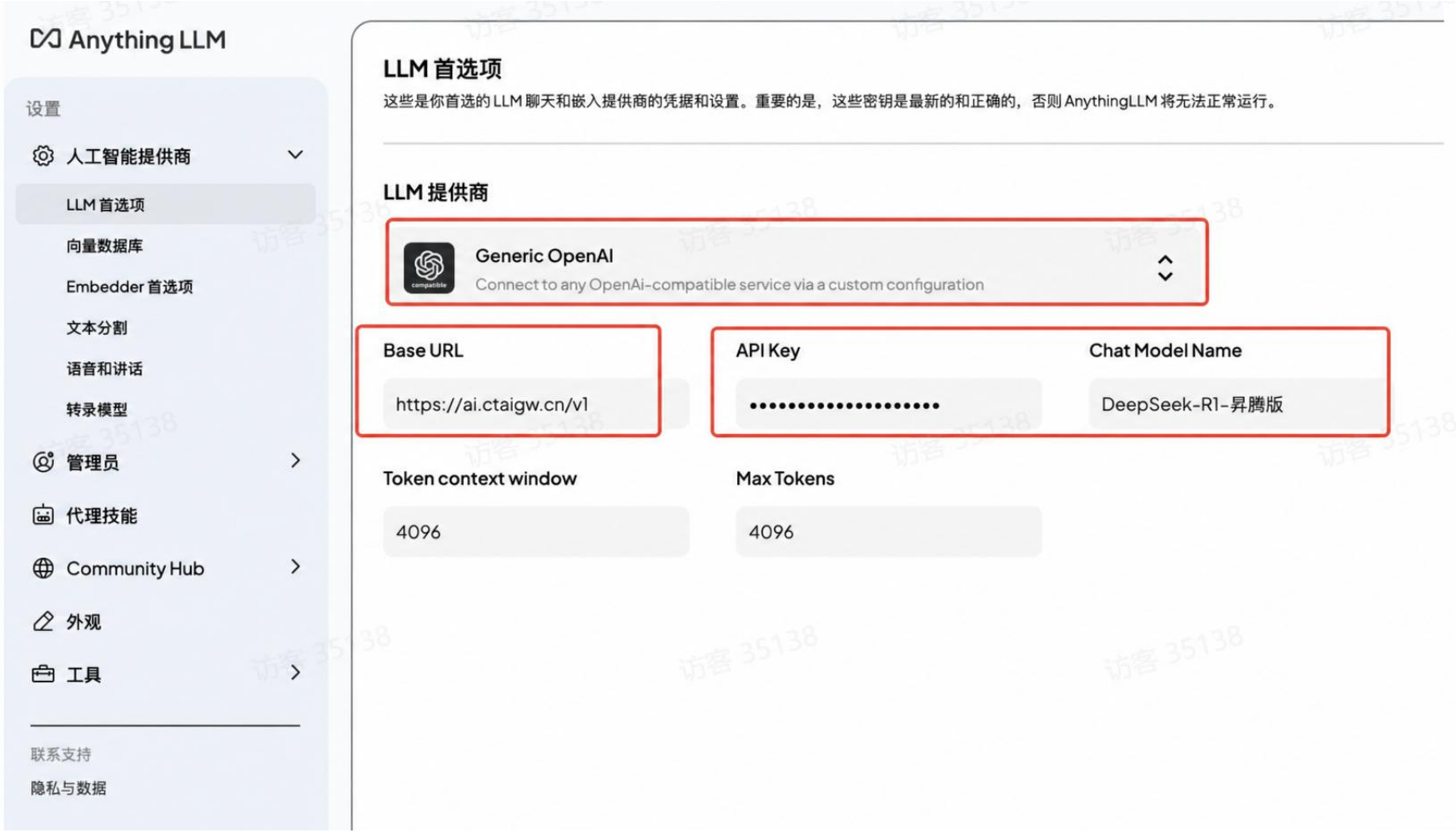
初始密码：见https://maxkb.cn/docs/installation/offline_installtion/#4

按照下图配置即可。注意baseUrl配置：https://ai.ctaigw.cn/v1


MaxKB
应用
知识库
函数库
系统设置
admin
用户管理
团队成员
模型设置
设置
模型设置
供应商
全部模型
公有模型
私有模型
全部模型
添加模型
一纸机
私有
大语言模型
admin
r1
私有
大语言模型
admin
选择供应商
全部模型
阿里云百炼
Anthropic
Amazon Bedrock
Azure OpenAI
DeepSeek
Gemini
Kimi
本地模型
Ollama
OpenAI
通义千问
SILICONFLOW
腾讯云
腾讯混元
vLLM
火山引擎
千帆大模型
讯飞星火
Xorbits Inference
智谱 AI
编辑 OpenAI
模型名称 *
r1
2 / 64
权限
私有
仅当前用户使用
公有
所有用户都可使用
模型类型 *
大语言模型
基础模型 列表中未列出的模型，直接输入模型名称，回车即可添加 *
4bd107bff85941239e27b1509eccfe98
API URL *
https://ai.ctaigw.cn/v1
API Key *
取消

AnythingLLM

按照下图配置即可。注意baseUrl配置：https://ai.ctaigw.cn/v1



配置完成后即可开启对话：



常见报错

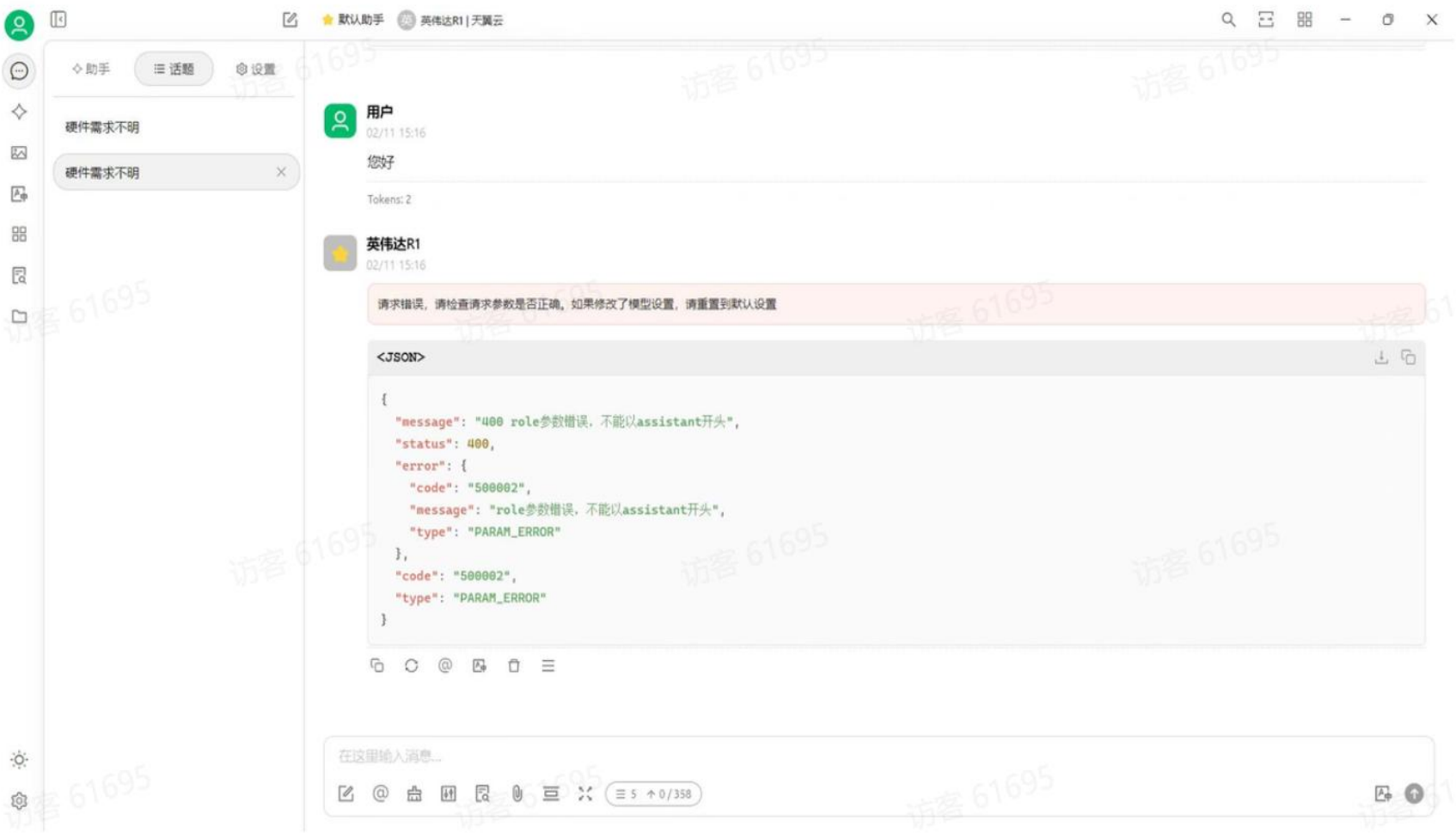
role参数错误，user后应为assistant



问题原因：模型setting首次配置错误，更改配置后，因本次对话包含历史错误信息，仍然会报错。

解决方案：开启一个新的对话即可。

400 role参数错误，不能以assistant开头



问题原因：模型setting首次配置错误，更改配置后，因为本次对话包含历史错误信息，导致仍然会报错。

解决方案：开启一个新的对话即可。

API statuscode429：DEEP_SEEK_MODEL_CONCURRENCY_LIMIT

```
    报错信息为：
[openai_api_compatible]Error:APIrequestfailedwithstatuscode429:{"code"
:700007,"detail":"Deepseek模型并发已达上限
","message":"DEEP_SEEK_MODEL_CONCURRENCY_LIMIT","error":{"code":"70000
7","message":"Deepseek模型并发已达上限
","type":"DEEP_SEEK_MODEL_CONCURRENCY_LIMIT"}}
```

问题原因：大模型并发是大模型的并发能力，和用户、tokens是否用完都无关，只和机器负载能力有关。目前单个模型有固定的并发度，这是所有ds模型共享的并发度，只能支持固定个请求同时访问。

解决方案：等一段时间再调用，或增加重试机制。

API 直连调用耗时长

问题原因：大模型API同步耗时长是因为其推理过程复杂，需要逐个生成token并拼接结果，且在处理长文本或复杂问题时，等待完整结果返回的时间过长。而流式API可以在推理过程中逐步返回结果，用户无需等待全部结果生成，从而提升交互效率。

解决方案：改为流式调用。

5.3 DeepSeek模型调用-专业版

本章节是关于开发者以 OpenAI-API-compatible 接口方式集成调用模型的实践。

1 天翼云官网获取模型调用API key等信息

模型API终端请求地址

请求地址

API终端请求地址是https://ai.ctaigw.cn/v1/chat/completions

通信协议

接口通过 HTTPS 进行通信，保护用户数据的机密性和完整性，确保网络通信的安全性。

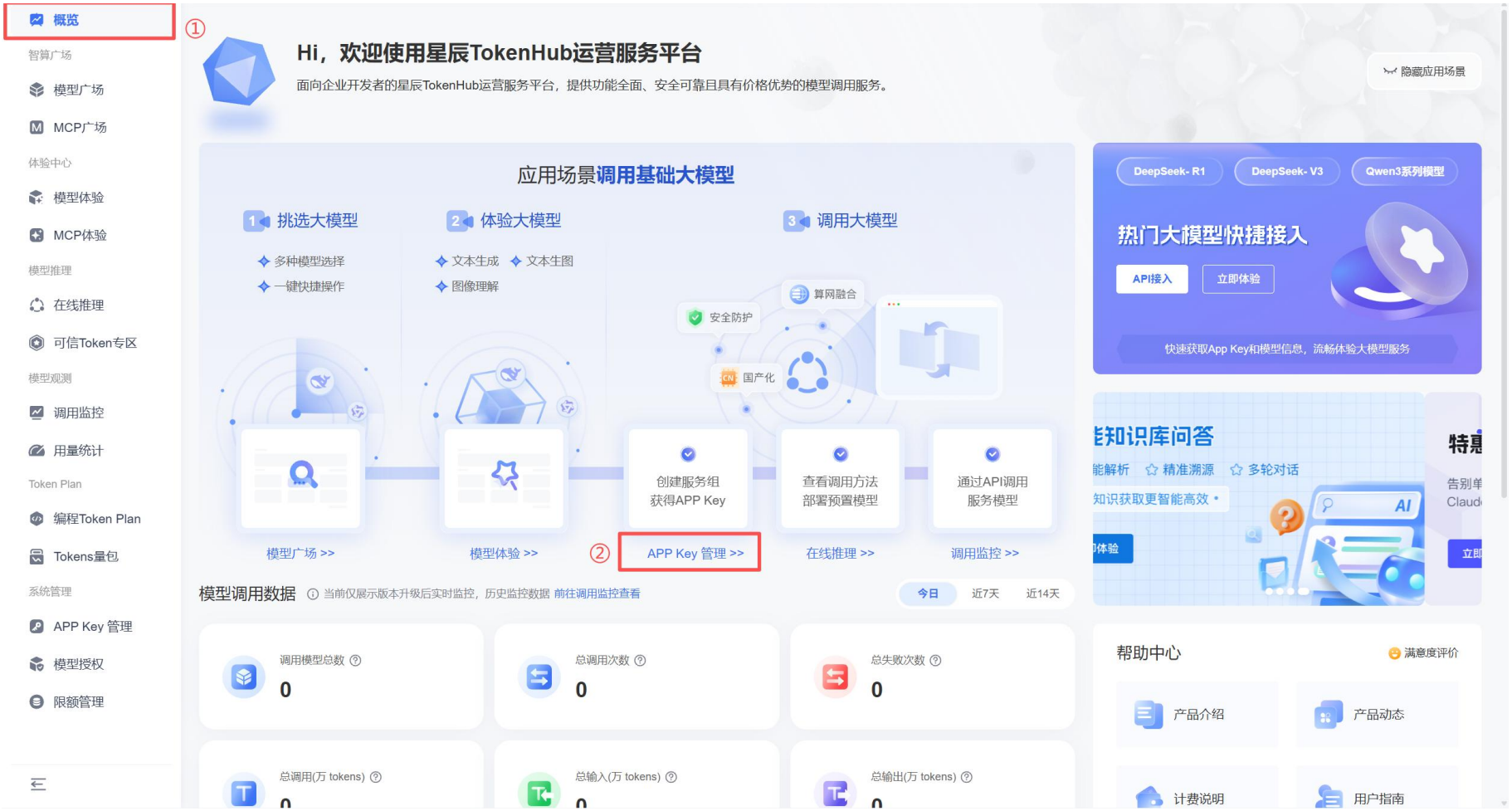
请求方法

支持的 HTTP 请求方法：POST。

POST 请求支持的 Content-Type 类型：application/json。

获取模型调用APP key

访问星辰TokenHub运营服务平台地址：https://ctxirang.ctyun.cn/maas/home，在左侧菜单栏进入“系统管理-APP Key管理”页。

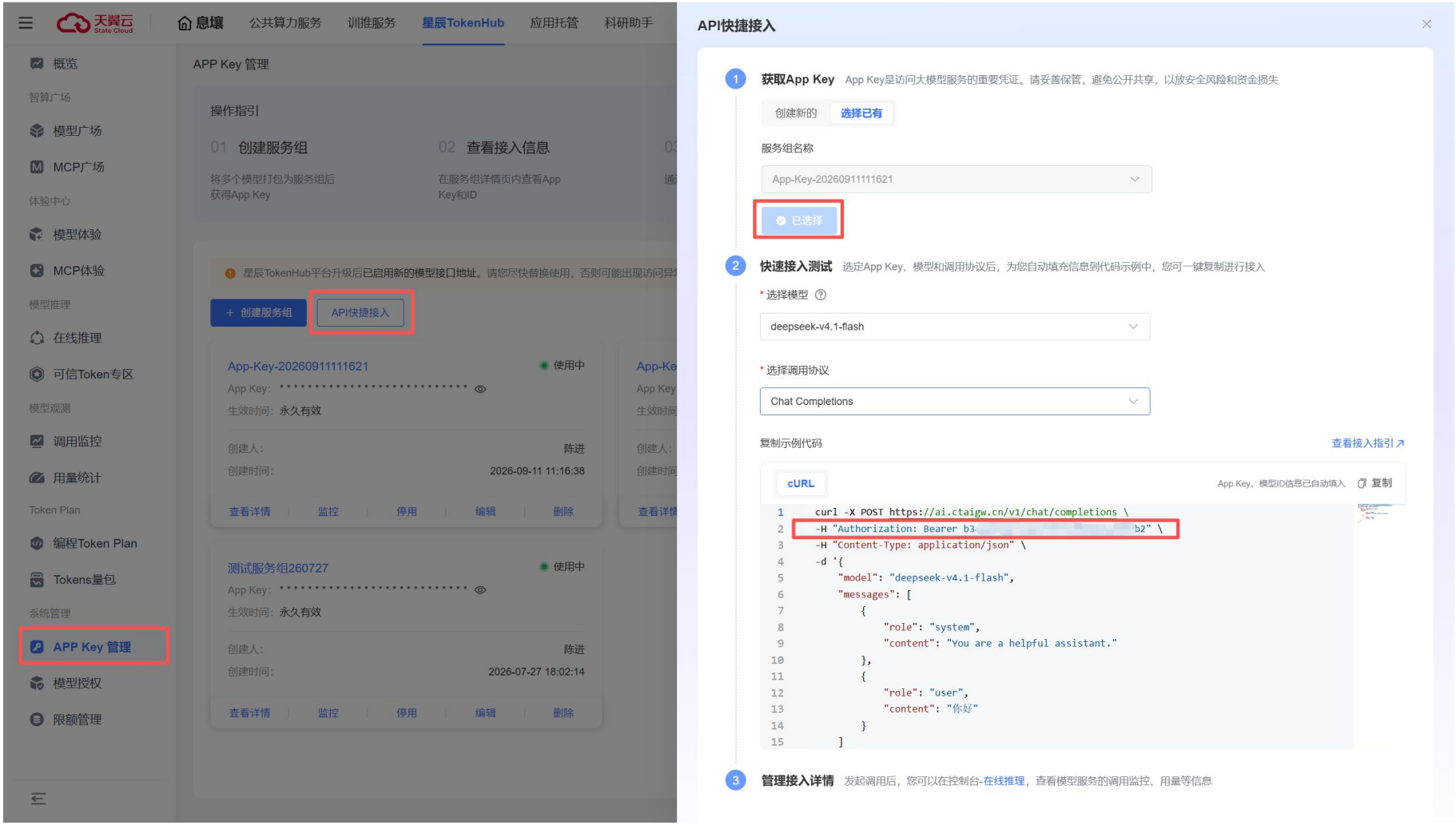


在“APP Key 管理”页，点击“API 快捷接入”拉起右侧抽屉弹窗，在 API 快捷接入页面：

选择服务组：从下拉菜单中选择第一步创建的服务组。如果尚未创建，可点击“确认创建并选择”快速创建默认服务组。

选择模型：从模型下拉菜单中选择需要接入的模型。

选择完成后，页面下方将自动展示该模型支持的所有编程语言的示例代码。



可进入左侧菜单栏的“系统管理-APP Key 管理”页，查看所创建的服务组及对应 APP Key。

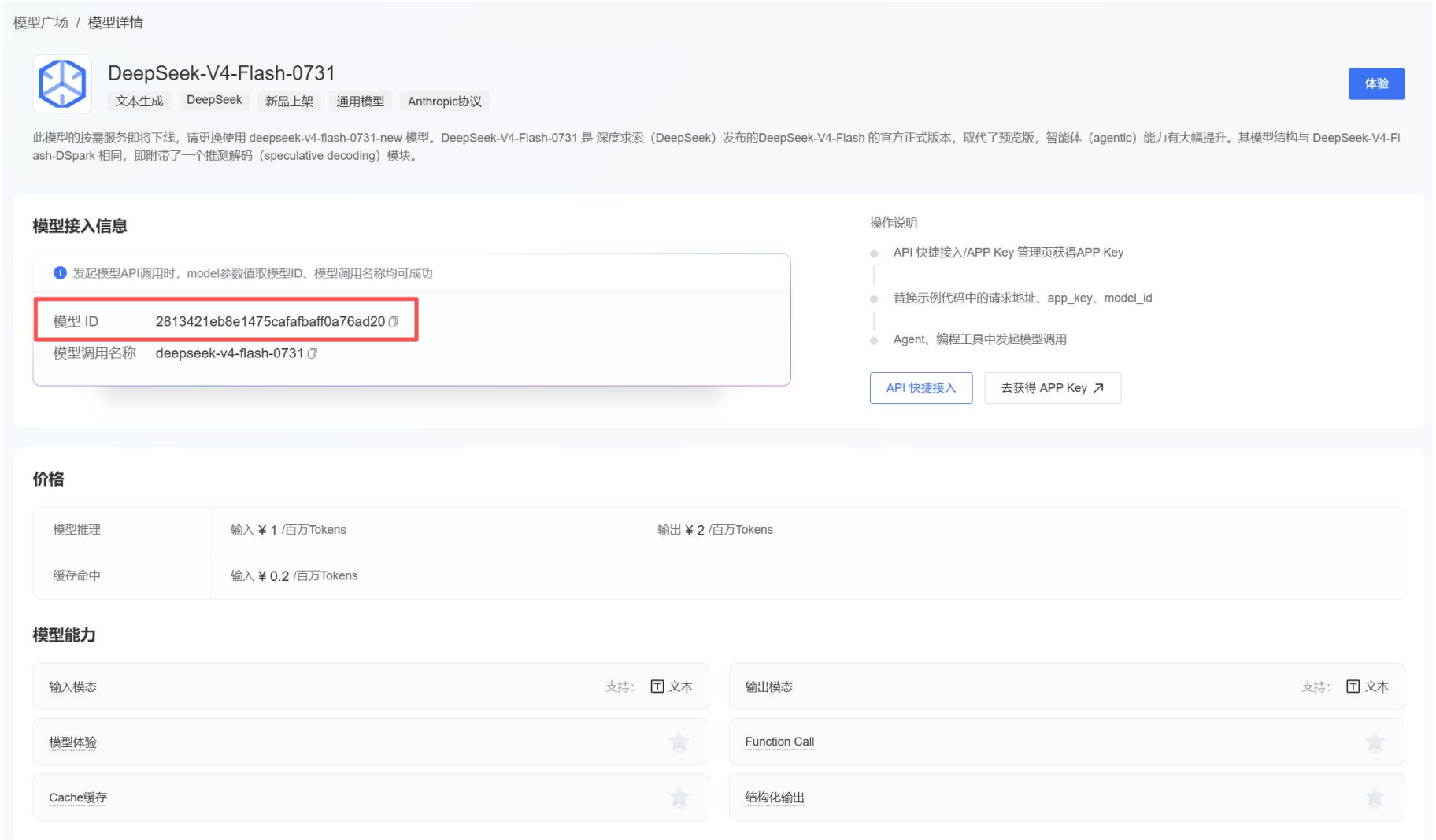


获取调用的模型名称

左侧导航栏切换到”模型广场“，选择对应想要调用的模型。



获取模型ID作为模型名称，例如下图中模型ID为：2813421eb8e1475cafafbaff0a76ad20。



DeepSeek模型版本选择参考：

DeepSeek-R1：适合需要高级推理能力和高精度的复杂任务，如科研、金融建模等。响应时间相对长。

DeepSeek-V3：适合需要快速响应和高性价比的通用自然语言处理任务，如智能客服、内容生成等。响应时间相对短。

DeepSeek-V4：适合高并发企业级 Agent 部署，海量数据结构化与清洗，日常代码辅助与工具调用，实时流式翻译与交互。

2.手动验证下key的可用性

下列请求为DeepSeek-R1-昇腾版请求样例，如果正常返回deepseek回答，代表没有问题。

注意：下面curl请求中替换成👆自己的API KEY后类似“--header 'Authorization: Bearer abcdefghxxssssss' \ ”

```
curl --location 'https://ai.ctaigw.cn/v1/chat/completions' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer 替换成自己的APP KEY' \
--data '{
    "messages": [
        {
            "role": "system",
            "content": "You are a test assistant."
        },
        {
            "role": "user",
            "content": "Testing. Just say hi and nothing else."
        }
    ],
    "model": "4bd107bff85941239e27b1509eccfe98"
}'
```

3 调用代码示例

Python

同步调用

```
    import os
    import openai
    from openai import OpenAI

    # 需要补充一下属性
    baseUrl = "https://ai.ctaigw.cn/v1"
    # 从环境变量获取API密钥，如果没有设置，也可以直接终端执行export
XIRANG_app_key="xxx"
    appKey=os.getenv("XIRANG_app_key")
    DeepSeek_R1_model_id = "4bd107bff85941239e27b1509eccfe98"
    DeepSeek_V3_model_id = "9dc913a037774fc0b248376905c85da5"
    prompt="你好，3.01+103.1等于多少" #对话问题

    def main():
        client = OpenAI(base_url=baseUrl, api_key=appKey)
        messages = [
            {"role": "user", "content": prompt}
        ]
        try:
            res = client.chat.completions.create(
                model=DeepSeek_R1_model_id,
                messages=messages,
                stream=False
            )
            print(res.choices[0].message.content or "", end="",
flush=True)

        except openai.APIStatusError as e:
```

流式调用

```
    import os
    import openai
    from openai import OpenAI

    # 需要补充一下属性
    baseUrl = "https://ai.ctaigw.cn/v1"
    # 从环境变量获取API密钥，如果没有设置，也可以直接终端执行export
XIRANG_app_key="xxx"
    appKey=os.getenv("XIRANG_app_key")
    DeepSeek_R1_model_id = "4bd107bff85941239e27b1509eccfe98"
    DeepSeek_V3_model_id = "9dc913a037774fc0b248376905c85da5"
    prompt= "你是谁"

    def main():
        client = OpenAI(base_url=baseUrl, api_key=appKey)
        messages = [
            {"role": "user", "content": prompt}
        ]
        try:
            stream = client.chat.completions.create(
                model=DeepSeek_R1_model_id,
                messages=messages,
                stream=True
            )
            # 流式
        for chunk in stream:
            print(chunk.choices[0].delta.content or "", end="",
flush=True)
```

Java

建议与Spring AI框架直接集成，而非使用openai-java版本库。

参考第四章「开源框架集成」Spring AI章节实现。

Golang

注意:

1、golang版本要求1.18以上

2、go.mod引用相关类库 go get github.com/openai/openai-go

同步调用

```
package main

import (
        "context"
        "fmt"
        "github.com/openai/openai-go"
        "github.com/openai/openai-go/option"
        "os"
)

func main() {
        baseUrl:="https://ai.ctaigw.cn/v1" //api前缀地址
        //从环境变量获取API密钥，如果没有设置，也可以直接终端执行export
XIRANG_app_key="xxx"
        appkey:=os.Getenv("XIRANG_app_key")
        modelId:="xxxxxx" //替换成自己要用的模型
        prompt:="你是谁" //对话问题

        client := openai.NewClient(
             option.WithAPIKey(appkey),
             option.WithBaseURL(baseUrl),
        )

        completion, err :=
client.Chat.Completions.New(context.Background(),
openai.ChatCompletionNewParams{
                      Messages:
openai.F([]openai.ChatCompletionMessageParamUnion{
```

流式调用

```
package main

import (
    "context"
    "fmt"
    "github.com/openai/openai-go"
    "github.com/openai/openai-go/option"
    "os"
)

func main() {
    baseUrl := "https://ai.ctaigw.cn:1443/v1" //api前缀地址
    //从环境变量获取API密钥，如果没有设置，也可以直接终端执行export
XIRANG_app_key="xxx"
    appkey := os.Getenv("XIRANG_app_key")
    modelId := "xxxxxx" //模型id
    prompt := "你好啊"      //对话问题

    client := openai.NewClient(
       option.WithAPIKey(appkey),
       option.WithBaseURL(baseUrl),
    )

    stream :=
client.Chat.Completions.NewStreaming(context.Background(),
openai.ChatCompletionNewParams{
        Messages:
openai.F([]openai.ChatCompletionMessageParamUnion{
```

PHP

注意：

1、php版本要求8.3.13以上

2、安装composer组件

3、运行以下命令

```
composer require openai-php/client
composer dump-autoload
```

同步调用

```
<?php
require __DIR__ . '/vendor/autoload.php';

use OpenAI\Client;

$baseUrl="https://ai.ctaigw.cn/v1"; //api前缀地址
//从环境变量获取API密钥，如果没有设置，也可以直接终端执行export
XIRANG_app_key="xxx"
$appkey = getenv("XIRANG_app_key");
$modelId="xxxxx"; //替换成自己要用的模型
$prompt="你好，介绍一下自己"; //#对话问题

$client = OpenAI::factory()
    ->withApiKey($appkey)
    ->withBaseUri($baseUrl)
    ->make();

$response = $client->chat()->create([
    'model' => $modelId,
    'messages' => [
        ['role' => 'user', 'content' => $prompt],
    ],
]);


foreach ($response->choices as $result) {
    echo $result->message->content;
}
```

流式调用

```
    <?php
    require __DIR__ . '/vendor/autoload.php';

    use OpenAI\Client;

    $baseUrl="https://ai.ctaigw.cn/v1"; //api前缀地址
    //从环境变量获取API密钥，如果没有设置，也可以直接终端执行export
XIRANG_app_key="xxx"
    $appkey = getenv("XIRANG_app_key");
    $modelId="xxxxx"; //替换成自己要用的模型
    $prompt="你好，介绍一下自己"; //#对话问题

    $client = OpenAI::factory()
        ->withApiKey($appKey)
        ->withBaseUri($baseUrl)
        ->make();

    $stream = $client->chat()->createStreamed([
        'model' => $modelId,
        'messages' => [
            ['role' => 'user', 'content' => $prompt],
        ],
    ]);

    foreach($stream as $response){
        $data=$response->choices[0];
        echo json_encode($data,
JSON_UNESCAPED_UNICODE,JSON_PRETTY_PRINT).PHP_EOL;
```

Nodejs

注意：

1、测试用的node版本>=v17.9.1,npm版本>=8.11.0

2、运行前需执行一下命令

```
npm init -y
npm install openai
```

3、在生成的package.json文件中添加"type":"module"

4、替换baseUrl,appkey,modelId,prompt等字段的值

同步调用

```
    import OpenAI from 'openai';

    const baseUrl="https://ai.ctaigw.cn/v1" //api前缀地址
    //从环境变量获取API密钥，如果没有设置，也可以直接终端执行export
XIRANG_app_key="xxx"
    const appkey= process.env.XIRANG_app_key
    const modelId="xxxxx" //模型id
    const prompt="你好啊,给一篇100字左右的作文" //对话问题


    const client = new OpenAI({
        apiKey: appkey,
        baseURL:baseUrl
    });

    async function chatCompletion() {
        const completion = await client.chat.completions.create({
            model: modelId,
            messages: [{ role: 'user', content: prompt }],
        });
        console.log(completion.choices[0]?.message?.content);
    }

    chatCompletion();
```

流式调用

```
    import OpenAI from 'openai';

    const baseUrl="https://ai.ctaigw.cn/v1" //api前缀地址
    //从环境变量获取API密钥，如果没有设置，也可以直接终端执行export
XIRANG_app_key="xxx"
    const appkey= process.env.XIRANG_app_key
    const modelId="xxxxx" //模型id
    const prompt="你好啊,给一篇100字左右的作文" //对话问题


    const client = new OpenAI({
        apiKey: appkey,
        baseURL:baseUrl
    });

    async function stream_chatCompletion() {
        const stream = await client.chat.completions.create({
            model: modelId,
            messages: [{ role: 'user', content: prompt }],
            stream: true,
        });
        for await (const chunk of stream) {
            process.stdout.write(chunk.choices[0]?.delta?.content ||
'');
        }
    }

    stream_chatCompletion();
```

4.开源框架集成

说明

部分第三方工具调用可能存在接口不适配问题，如涉及工具调用时，建议优先使用下文提及的工具。

涉及第三方工具配置时，API key 等同于上文中创建的服务组 APP key。

Langchain调用

参考langchain的自定义模型文档通过OpenAI协议调用即可。

```
    from langchain.chat_models import ChatOpenAI
    from langchain.schema import HumanMessage

    def main():
        # 初始化LangChain模型，配置自定义API端点
        llm = ChatOpenAI(
            model_name="4bd107bff85941239e27b1509eccfe98",  # 自定义模型
名称
           # 从环境变量获取API密钥，如果没有设置，也可以直接终端执行export
XIRANG_app_key="xxx"
            openai_app_key=os.getenv("XIRANG_APP_KEY")
            openai_api_base="https://ai.ctaigw.cn/v1",  # API基础URL
            temperature=0.7
        )

        # 创建一个聊天消息
        message = HumanMessage(content="你是谁")

        # 使用LangChain进行对话
        response = llm([message])
        print(response.content)


    if __name__ == "__main__":
        main()
```

Dify

安装条件：本地下载并安装docker。（Docker 官网下载页面：访问 Docker 官方网站。在页面中点击“Download Docker Desktop ”按钮，即可选择适合 Windows、macOS 或 Linux 的版本进行下载。）

拉取dify镜像到本地

```
# 如果本地已安装dify可以忽略
# 假设当前最新版本为 0.15.3
git clone https://github.com/langgenius/dify.git --branch 0.15.3
```

修改配置参数

```
cd docker
mv .env.example .env
```

启动并验证

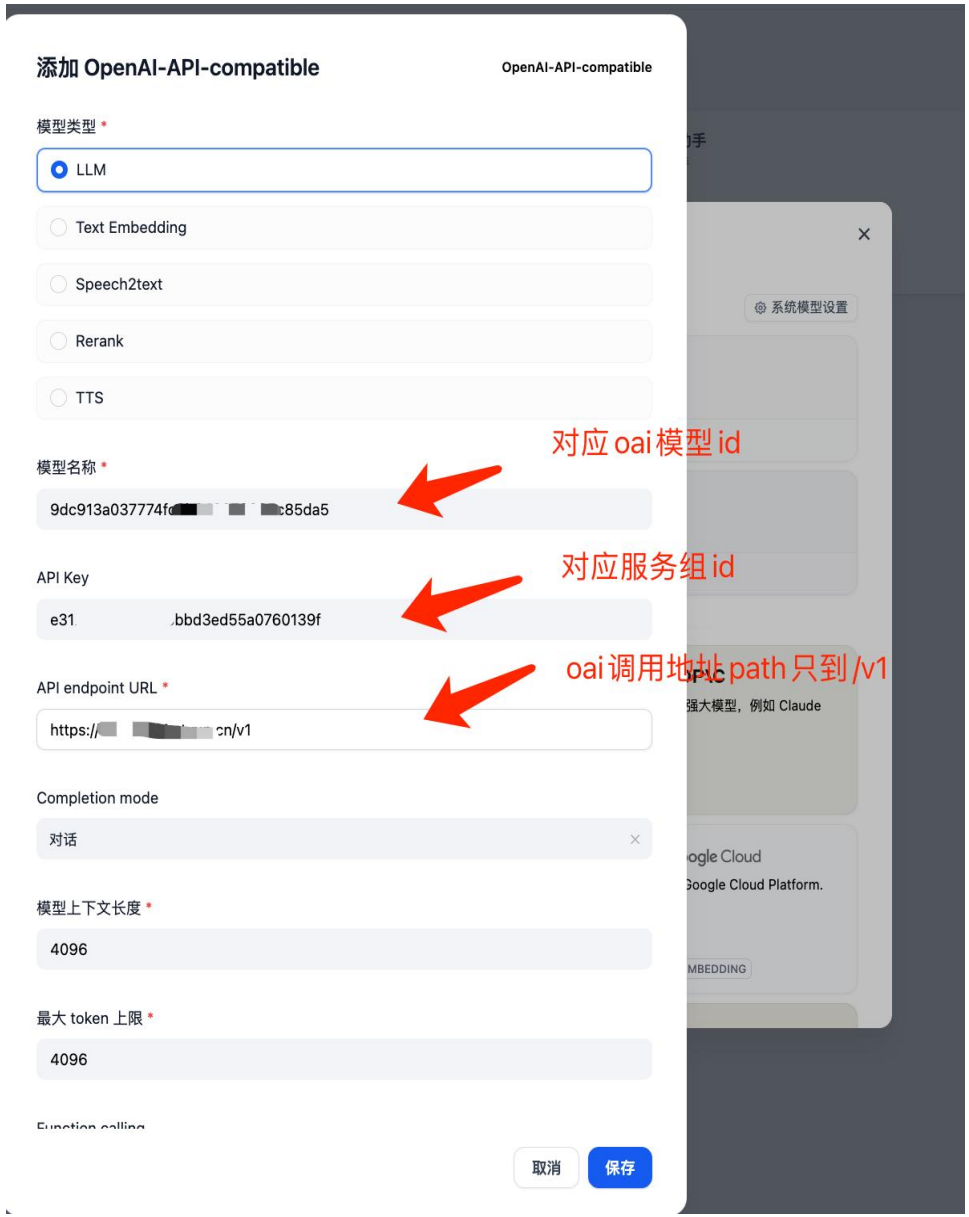
```
#执行：
docker compose up -d
# 等服务启动访问
http://localhost/apps
```

配置天翼云息壤模型

配置方式一：直接配置方式（所有版本均可使用）

通过直接配置OpenAI-API-compatible支持，配置方式如下：


设置
工作空间
模型供应商
成员
数据来源
API 扩展
通用
语言
模型供应商
模型列表
系统模型设置
Ollama
LLM
TEXT EMBEDDING
显示模型
OpenAI-API-compatible
LLM
TEXT EMBEDDING
SPEECH2TEXT
RERANK
TTS
显示模型
添加模型
+ 添加更多模型提供商
OpenAI
OpenAI 提供的模型，例如 GPT-3.5-Turbo 和 GPT-4。
deepseek
深度求索提供的模型，例如 deepseek-chat、deepseek-coder 。
ANTHROP\C
Anthropic 的强大模型，例如 Claude 3。


添加 OpenAI-API-compatible
OpenAI-API-compatible
模型类型 *
LLM
Text Embedding
Speech2text
Rerank
TTS
模型名称 *
对应 oai 模型 id
9dc913a037774f
c85da5
API Key
对应服务组 id
e31
bbd3ed55a0760139f
API endpoint URL *
oai 调用地址 path 只到 /v1
https://
cn/v1
Completion mode
对话
模型上下文长度 *
4096
最大 token 上限 *
4096
取消
保存
系统模型设置
强大模型，例如 Claude
Google Cloud Platform.

配置方式二：使用息壤镜像配置方式（仅支持0.15.3版本）

修改env配置

```
cd docker
#在.env文件添加
CUSTOM_MODEL_ENABLED=true
OLLAMA_API_BASE_URL=host.docker.internal:11434
```

修改下api和work的镜像地址，改为增加了xirang模型供应商的dify 0.15.3版本。

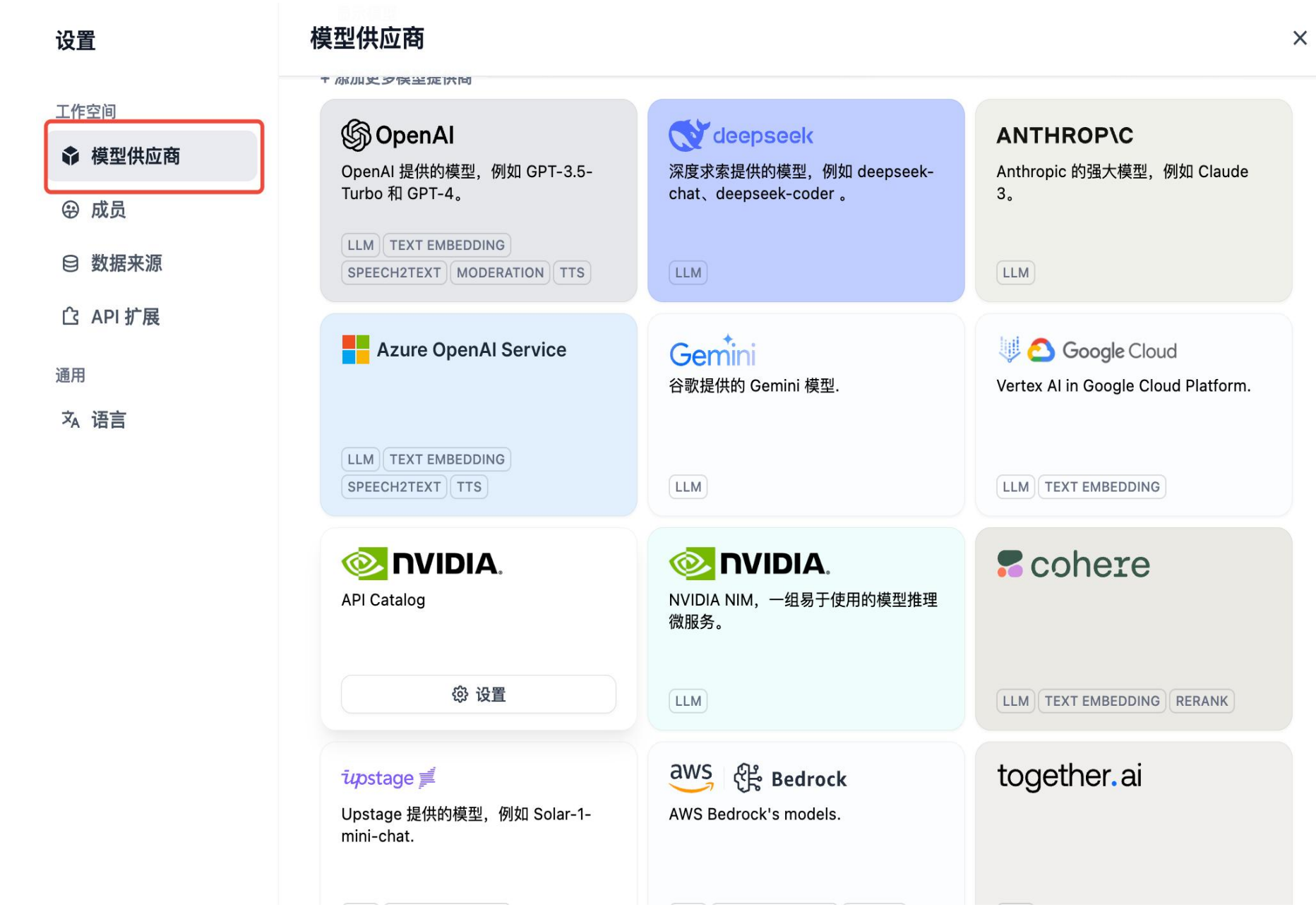
修改后需要重启dify。

```
    services:
      # API service with Xirang model provider
      api:
        image: telecom-eci-huadong1-crs-registry-
huadong1.crs.ctyun.cn/xirang/dify-with-xirang:0.15.3 # 原来为
langgenius/dify-api:0.15.3


       # worker service
      # The Celery worker for processing the queue.
      worker:
        image: telecom-eci-huadong1-crs-registry-
huadong1.crs.ctyun.cn/xirang/dify-with-xirang:0.15.3 # 原来为
langgenius/dify-api:0.15.3
```

在个人设置里面点击模型供应商，选择天翼云模型。


localhost/apps
百度翻译
百度搜搜
私有库
内部开发系统
天翼云
迅捷Markdown -...
内部办公网址
部署文档
在线工具
大模型
慧聚
Dify
Star 66,534
探索
工作室
知识库
工具
全部
聊天助手
Agent
工作流
我创建的
全部标签
创建应用
创建空白应用
从应用模版创建
导入 DSL 文件
测试
聊天助手
测试模型
聊天助手
销售助手
聊天助手
WX
聊天助手
工作空间
账户
设置
用户反馈
社区
帮助文档
路线图
关于
0.15.3
登出


设置
工作空间
模型供应商
成员
数据来源
API 扩展
通用
语言
模型供应商
OpenAI
OpenAI 提供的模型，例如 GPT-3.5-Turbo 和 GPT-4。
LLM TEXT EMBEDDING SPEECH2TEXT MODERATION TTS
deepseek
深度求索提供的模型，例如 deepseek-chat、deepseek-coder 。
LLM
ANTHROP\C
Anthropic 的强大模型，例如 Claude 3。
LLM
Azure OpenAI Service
LLM TEXT EMBEDDING SPEECH2TEXT TTS
Gemini
谷歌提供的 Gemini 模型.
LLM
Google Cloud
Vertex AI in Google Cloud Platform.
LLM TEXT EMBEDDING
NVIDIA
API Catalog
设置
NVIDIA
NVIDIA NIM，一组易于使用的模型推理微服务。
LLM
cohere
LLM TEXT EMBEDDING RERANK
upstage
Upstage 提供的模型，例如 Solar-1-mini-chat.
aws Bedrock
AWS Bedrock's models.
together.ai

模型列表

系统模型设置

中国电信 CHINA TELECOM | 天翼云 State Cloud

LLM

API-KEY

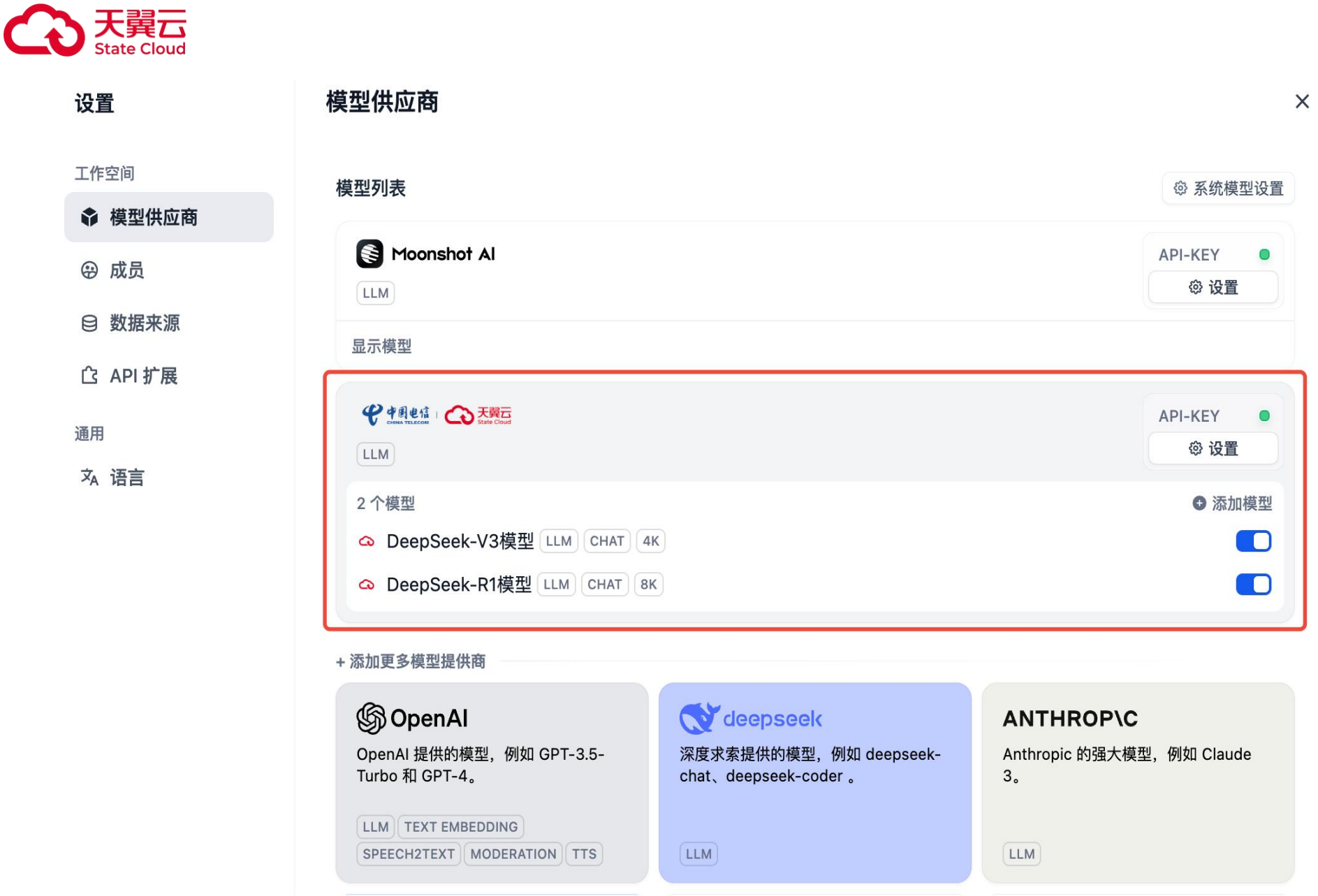
设置

显示模型

+ 添加更多模型提供商



点击保存后，可以看到模型的配置。


天翼云
State Cloud
设置
工作空间
模型供应商
成员
数据来源
API 扩展
通用
语言
模型供应商
模型列表
系统模型设置
Moonshot AI
LLM
API-KEY
设置
显示模型
中国电信
天翼云
LLM
API-KEY
设置
2 个模型
添加模型
DeepSeek-V3模型 LLM CHAT 4K
DeepSeek-R1模型 LLM CHAT 8K
+ 添加更多模型提供商
OpenAI
OpenAI 提供的模型，例如 GPT-3.5-Turbo 和 GPT-4。
LLM TEXT EMBEDDING SPEECH2TEXT MODERATION TTS
deepseek
深度求索提供的模型，例如 deepseek-chat、deepseek-coder 。
LLM
ANTHROP\C
Anthropic 的强大模型，例如 Claude 3。
LLM

其他非预置模型：



模型列表

系统模型设置

中国电信 CHINA TELECOM | 天翼云 State Cloud

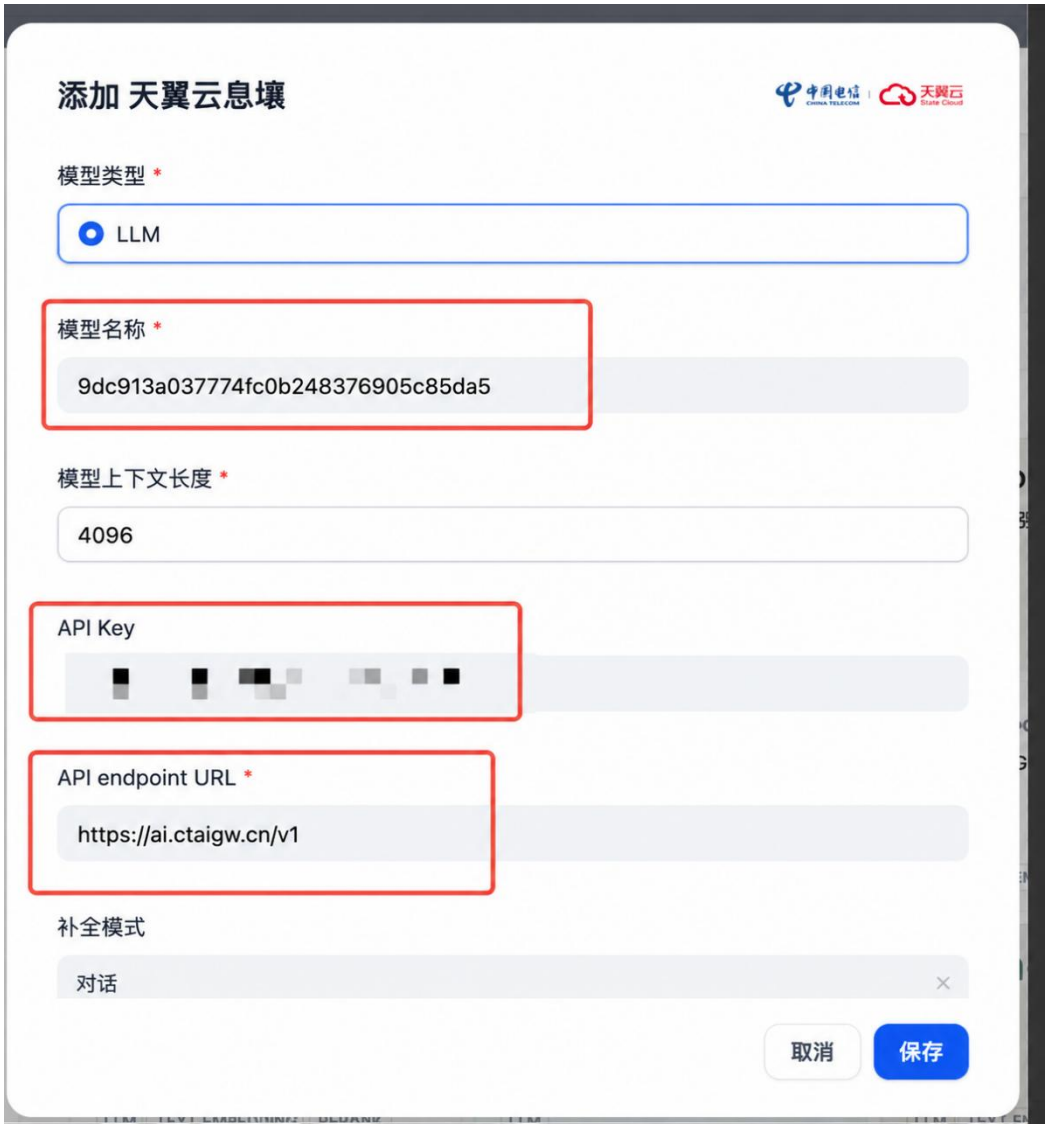
LLM

API-KEY

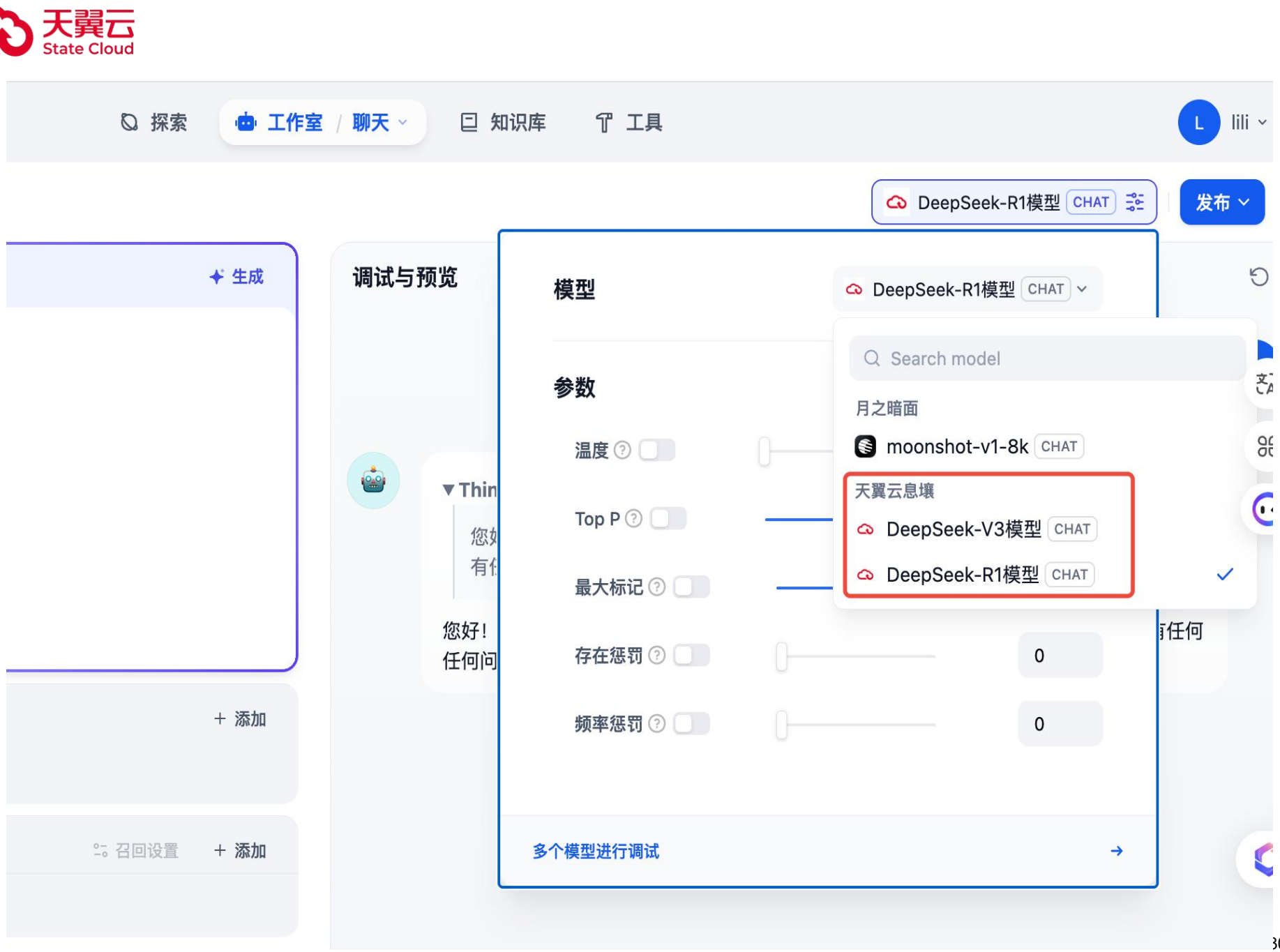
设置

显示模型

添加模型


添加 天翼云息壤
模型类型 *
LLM
模型名称 *
9dc913a037774fc0b248376905c85da5
模型上下文长度 *
4096
API Key
API endpoint URL *
https://ai.ctaigw.cn/v1
补全模式
对话
取消
保存

配置对话助手


天翼云
State Cloud
探索
工作室 / 聊天
知识库
工具
L
lili
DeepSeek-R1模型 CHAT
发布
生成
调试与预览
模型
DeepSeek-R1模型 CHAT
参数
Search model
月之暗面
moonshot-v1-8k CHAT
天翼云息壤
DeepSeek-V3模型 CHAT
DeepSeek-R1模型 CHAT
温度
Top P
最大标记
存在惩罚
0
频率惩罚
0
多个模型进行调试
添加
召回设置
添加


DeepSeek-R1模型
CHAT
发布
调试与预览
你是谁
Thinking
您好！我是由中国的深度求索（DeepSeek）公司开发的智能助手DeepSeek-R1。如您有任何任何问题，我会尽我所能为您提供帮助。
您好！我是由中国的深度求索（DeepSeek）公司开发的智能助手DeepSeek-R1。如您有任何任何问题，我会尽我所能为您提供帮助。

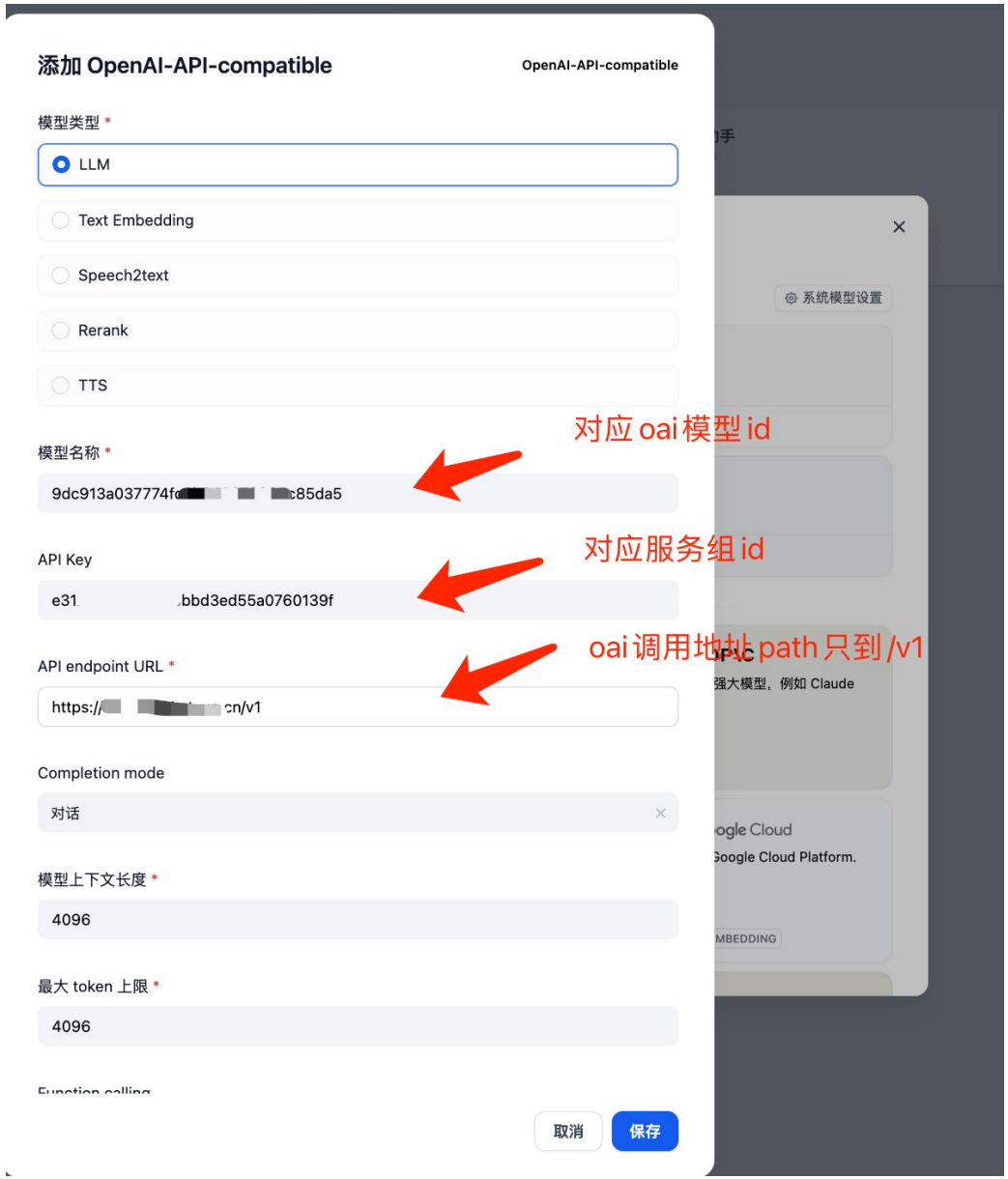
常见问题（0.15.3版本）

如遇到无法展示问题，可能是因为您的dify版本为最新版，我们目前还不支持，可执行以下操作切换到0.15.3版本并重启服务。

```
dify插件需要切换版本重新启动服务才可以达成兼容
git clone  https://gitcode.com/gh_mirrors/di/dify.git
cd dify
git checkout 7796984444191c639bd3c541a44e832b17ad1cae
```

如遇到息壤镜像失败问题，可切换到dify原始镜像，通过配置直接配置OpenAI-API-compatible支持，配置方式如下：


设置
工作空间
模型供应商
成员
数据来源
API 扩展
通用
语言
模型供应商
模型列表
系统模型设置
Ollama
LLM
TEXT EMBEDDING
显示模型
OpenAI-API-compatible
LLM
TEXT EMBEDDING
SPEECH2TEXT
RERANK
TTS
显示模型
添加模型
+ 添加更多模型提供商
OpenAI
OpenAI 提供的模型，例如 GPT-3.5-Turbo 和 GPT-4。
deepseek
深度求索提供的模型，例如 deepseek-chat、deepseek-coder 。
ANTHROP\C
Anthropic 的强大模型，例如 Claude 3。


添加 OpenAI-API-compatible
OpenAI-API-compatible
模型类型 *
LLM
Text Embedding
Speech2text
Rerank
TTS
对应 oai 模型 id
模型名称 *
9dc913a037774f
c85da5
对应服务组 id
API Key
e31
bbd3ed55a0760139f
oai 调用地址 path 只到 /v1
API endpoint URL *
https://
cn/v1
Completion mode
对话
模型上下文长度 *
4096
最大 token 上限 *
4096
取消
保存
系统模型设置
强大模型，例如 Claude
Google Cloud Platform.

Spring AI

说明：

Spring AI要求使用JDK 17或更高版本；Spring AI需要在Spring Boot 3.2.x或更高版本的环境中运行。

核心代码如下：

OpenAIConfig配置类：

```
package com.demo.aitest;

import jakarta.annotation.PostConstruct;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.ai.openai.OpenAiChatClient;
import org.springframework.ai.openai.OpenAiChatOptions;
import org.springframework.ai.openai.api.OpenAiApi;

@Configuration
public class OpenAIConfig {

    @Value("${spring.ai.openai.api-key}")
    private String apiKey;

    @Value("${spring.ai.openai.base-url}")
    private String baseUrl;

    @Value("${spring.ai.openai.chat.options.model}")
    private String model;

    @Bean
    public OpenAiChatClient openAiChatClient() {
        OpenAiApi openAiApi = new OpenAiApi(baseUrl, apiKey);
        OpenAiChatOptions options = new OpenAiChatOptions();
        options.setModel(model);
```

```
    package com.lili.aitest;

    import org.springframework.ai.openai.OpenAiChatClient;
    import org.springframework.ai.chat.messages.UserMessage;
    import org.springframework.beans.factory.annotation.Autowired;
    import org.springframework.beans.factory.annotation.Value;
    import org.springframework.ai.chat.ChatResponse;
    import org.springframework.ai.chat.prompt.Prompt;
    import org.springframework.web.bind.annotation.*;
    import
org.springframework.web.servlet.mvc.method.annotation.SseEmitter;
    import org.slf4j.Logger;
    import org.slf4j.LoggerFactory;
    import java.io.IOException;


    @RestController
    public class ChatController {
        private static final Logger logger =
LoggerFactory.getLogger(ChatController.class);

        @Autowired
        private OpenAiChatClient chatClient;

        @GetMapping("/chat")
        public String chat(@RequestParam String message) {
            try {
                ChatResponse response = chatClient.call(new
Prompt(message));
```

application.yml配置如下：

```
    spring:
      ai:
        openai:
          api-key: Your_APP_KEY    #替换成👆自己的API KEY
          base-url: https://ai.ctaigw.cn
          chat:
            options:
              model: 4bd107bff85941239e27b1509eccfe98. #替换成👆自己要用
的模型
    server:
      servlet:
        encoding:
          charset: UTF-8
          force: true
          enabled: true
```

pom.xml依赖如下

```
    <?xml version="1.0" encoding="UTF-8"?>
    <project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <parent>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-parent</artifactId>
            <version>3.2.2</version>
            <relativePath/> <!-- lookup parent from repository -->
        </parent>
        <groupId>com.demo</groupId>
        <artifactId>aitest</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <name>aitest</name>
        <description>aitest</description>

        <properties>
            <java.version>17</java.version>
            <spring-ai.version>0.8.1</spring-ai.version>
            <spring-cloud.version>2023.0.0</spring-cloud.version>
        </properties>
        <dependencies>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-web</artifactId>
            </dependency>
            <dependency>
```

访问地址：

同步：http://localhost:8080/chat?message=Hello

流式访问：http://localhost:8080/chat/stream?message=Hello

Ragflow

> **注意：**
>
> 目前息壤model不支持embedding model能力，因此只能作为chat model提供能力。

下载镜像git clone https://github.com/infiniflow/ragflow.git

通过docker安装并启动：

```
cd ragflow/docker
docker compose -f docker-compose-CN.yml up -d
```

检查 RAGFlow 服务是否正常启动：

```
docker logs -f ragflow-server
```

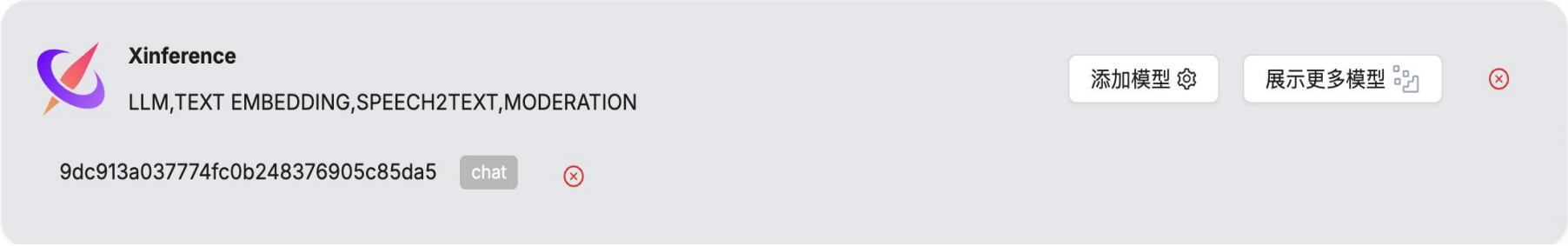
如果看到类似以下输出，则表示启动成功：

```
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:9380
* Running on http://x.x.x.x:9380
```

打开浏览器配置模型Xinference服务提供商

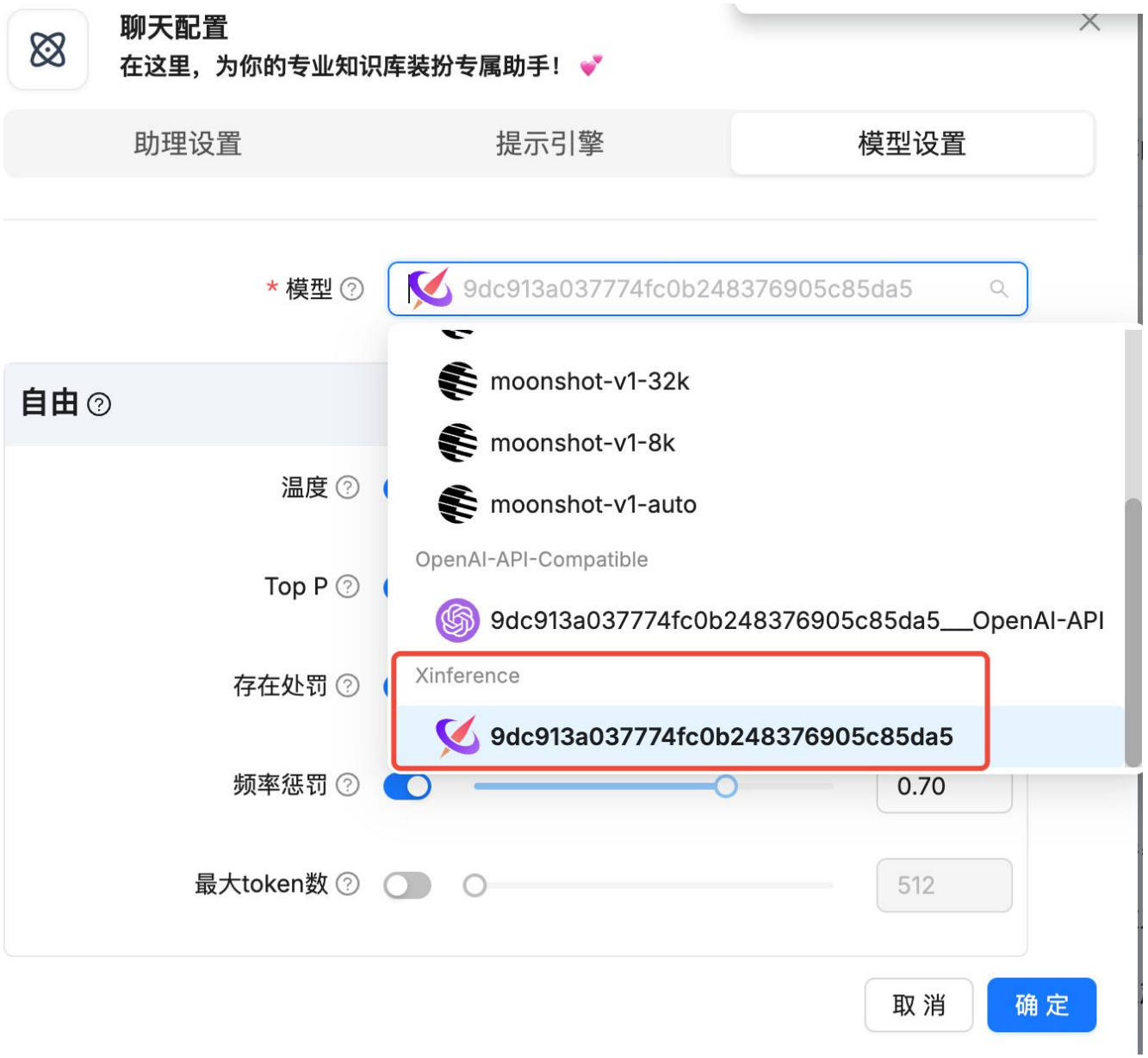
在浏览器中输入服务器的 IP 地址访问 RAGFlow。默认情况下，RAGFlow 使用 HTTP 端口 80，因此无需输入端口号。

选择“Xinference”模型服务商(Xinference是工厂配置，支持自定义模型名称)



配置chatmodel的界面如下：

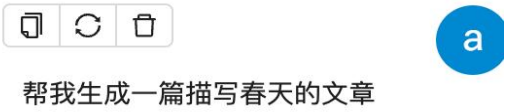

添加 LLM
* 模型类型
chat
* 模型UID
9dc913a037774fc0b248376905c85da5
* 基础 Url
https://ai.ctaigw.cn/coding/v1
API-Key
* 最大token数
2000
是否支持 Vision
如何集成 Xinference
取 消
确 定

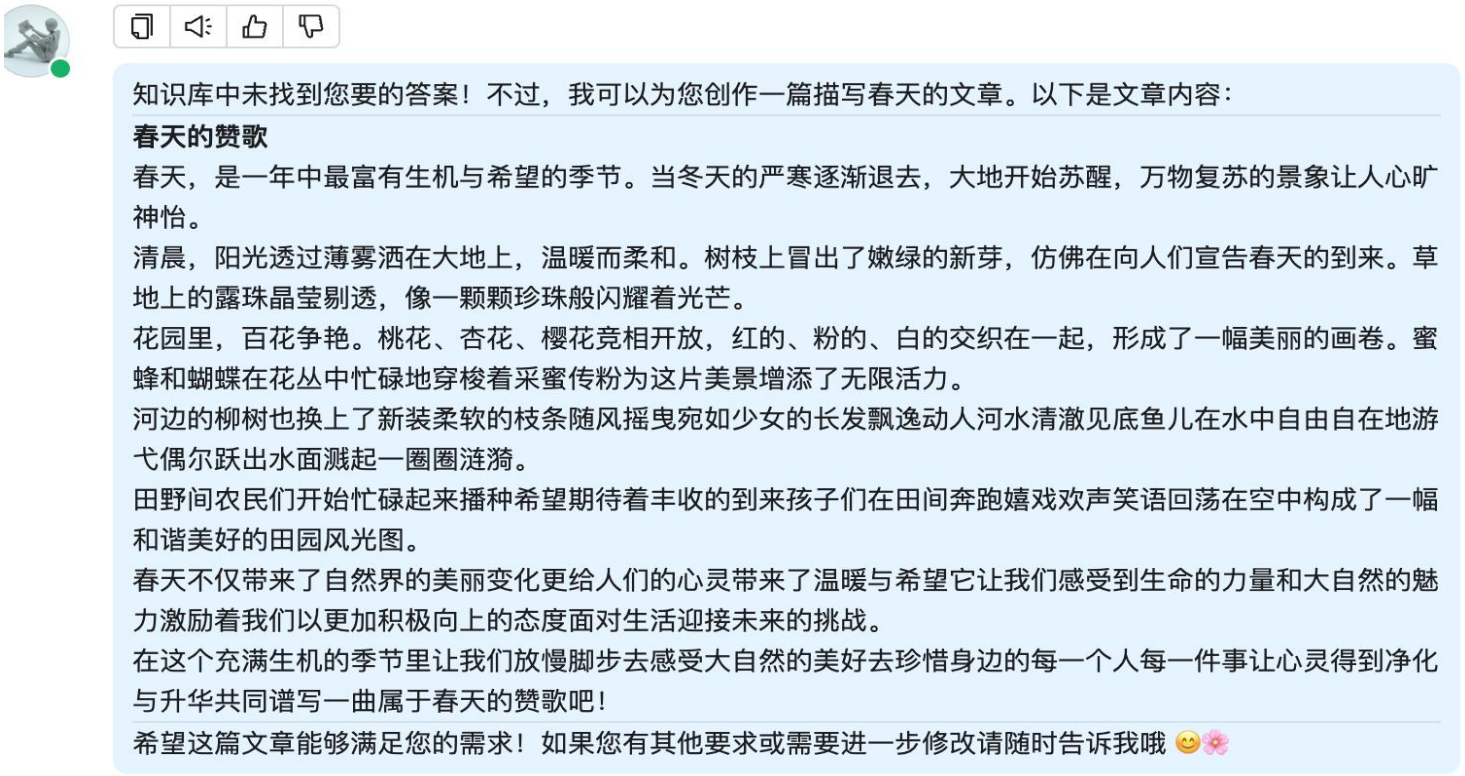

聊天配置
在这里，为你的专业知识库装扮专属助手！
助理设置
提示引擎
模型设置
* 模型
9dc913a037774fc0b248376905c85da5
moonshot-v1-32k
moonshot-v1-8k
moonshot-v1-auto
OpenAI-API-Compatible
9dc913a037774fc0b248376905c85da5___OpenAI-API
Xinference
9dc913a037774fc0b248376905c85da5
自由
温度
Top P
存在处罚
频率惩罚
0.70
最大token数
512
取 消
确 定

回到对话界面，配置聊天助手：


RAGFlow
知识库
聊天
搜索
Agent
文件管理
新建助理
聊天 0
jarvis3-openai
A helpful dialog
jarvi2
A helpful dialog
jarvis
A helpful dialog
聊天配置
在这里，为你的专业知识库装扮专属助手！
助理设置
提示引擎
模型设置
* 助理姓名
例如 贾维斯简历
助理描述
chat.descriptionPlaceholder
助理头像
+
上传
空回复
设置开场白
你好！ 我是你的助理，有什么可以帮到你的吗?
显示引文

开启对话：


a
帮我生成一篇描写春天的文章


知识库中未找到您要的答案！不过，我可以为您创作一篇描写春天的文章。以下是文章内容：
春天的赞歌
春天，是一年中最富有生机与希望的季节。当冬天的严寒逐渐退去，大地开始苏醒，万物复苏的景象让人心旷神怡。
清晨，阳光透过薄雾洒在大地上，温暖而柔和。树枝上冒出了嫩绿的新芽，仿佛在向人们宣告春天的到来。草地上的露珠晶莹剔透，像一颗颗珍珠般闪耀着光芒。
花园里，百花争艳。桃花、杏花、樱花竞相开放，红的、粉的、白的交织在一起，形成了一幅美丽的画卷。蜜蜂和蝴蝶在花丛中忙碌地穿梭着采蜜传粉为这片美景增添了无限活力。
河边的柳树也换上了新装柔软的枝条随风摇曳宛如少女的长发飘逸动人河水清澈见底鱼儿在水中自由自在地游弋偶尔跃出水面溅起一圈圈涟漪。
田野间农民们开始忙碌起来播种希望期待着丰收的到来孩子们在田间奔跑嬉戏欢声笑语回荡在空中构成了一幅和谐美好的田园风光图。
春天不仅带来了自然界的美丽变化更给人们的心灵带来了温暖与希望它让我们感受到生命的力量和大自然的魅力激励着我们以更加积极向上的态度面对生活迎接未来的挑战。
在这个充满生机的季节里让我们放慢脚步去感受大自然的美好去珍惜身边的每一个人每一件事让心灵得到净化与升华共同谱写一曲属于春天的赞歌吧！
希望这篇文章能够满足您的需求！如果您有其他要求或需要进一步修改请随时告诉我哦

5.常见报错

触发限流

```
报错信息为：{
openai_api_compatible]Error:APIrequestfailedwithstatuscode429:{
"code":700007,"detail":"Deepseek 模型并发已达上限",
"message":"DEEP_SEEK_MODEL_CONCURRENCY_LIMIT",
"error":{"code":"700007","message":"Deepseek 模型并发已达上限",
"type":"DEEP_SEEK_MODEL_CONCURRENCY_LIMIT"
}
    }
```

问题原因：大模型并发是大模型的并发能力，和用户、tokens是否用完都无关，只和机器负载能力有关。目前单个模型有固定的并发度，这是所有ds模型共享的并发度，只能支持固定个请求同时访问。

解决方案：等一段时间再调用，或增加重试机制。

同步调用响应耗时长

问题原因：大模型API同步耗时长是因为其推理过程复杂，需要逐个生成token并拼接结果，且在处理长文本或复杂问题时，等待完整结果返回的时间过长。而流式API可以在推理过程中逐步返回结果，用户无需等待全部结果生成，从而提升交互效率。

解决方案：改为流式调用。

5.4 基于息壤的企业级RAG知识库构建指南

5.4.1 概述

本文档基于天翼云息壤星辰 TokenHub 运营服务平台，为企业用户提供构建 RAG（检索增强生成）知识库问答系统的最佳实践指导。企业用户可在自建 RAG 系统时参考本文档进行合理的模型选型，从而构建高效、准确的企业知识库智能问答系统。

文档目标

本最佳实践案例旨在帮助企业技术团队：

理解 RAG 系统的核心架构与工作流程

掌握知识入库环节涉及的模型选型与配置

掌握知识问答环节涉及的模型选型与配置

了解基于天翼云星辰TokenHub运营服务平台的模型部署实践

适用范围

本指南适用于以下场景：

企业自建知识库智能问答系统

基于大模型的企业知识管理平台

需要定制化 RAG 能力的业务场景

利用天翼云息壤星辰TokenHub运营服务平台进行模型推理服务的企业

5.4.2 RAG系统架构概述

本文档基于天翼云息壤星辰 TokenHub 运营服务平台，为企业用户提供构建 RAG（检索增强生成）知识库问答系统的最佳实践指导。企业用户可在自建 RAG 系统时参考本文档进行合理的模型选型，从而构建高效、准确的企业知识库智能问答系统。

RAG（Retrieval-Augmented Generation，检索增强生成）是一种结合信息检索与大语言模型生成的技术架构。RAG 系统的工作流程主要分为两个阶段：知识入库阶段和知识问答阶段。

RAG 系统工作流程

RAG 系统的完整工作流程如下：

知识入库阶段	知识问答阶段
1. 文档上传与解析	1. 用户问题输入
2. 文本切块（Chunking）	2. Query 向量化
3. Embedding 向量化	3. 向量相似度检索
4. 向量数据库存储	4. LLM 生成回答

核心组件说明

组件类别	功能描述	关键模型
文档解析	处理 PDF、Word、Markdown 等格式文档	文档解析模型
文本切块	将长文档分割成语义完整的小块	切块策略/模型
向量化	将文本转换为高维稠密向量	Embedding 模型
向量存储	存储和索引向量，支持高效检索	向量数据库
问答生成	基于检索结果生成回答	LLM 模型
结果优化	对检索结果进行相关性排序	Reranker 模型

5.4.3 知识入库环节模型选型

知识入库是 RAG 系统的基础环节，负责将企业文档转化为可检索的向量数据。该环节主要涉及文本切块模型和 Embedding 向量化模型。

文本切块模型（Chunking）

文本切块是将长文档分割成适合检索和理解的小文本块的过程。合理的切块策略直接影响检索质量和生成效果。

常见切块策略

切块策略	描述	适用场景
固定长度切块	按固定字符数或 token 数切分	通用场景，文档结构简单
滑动窗口切块	带重叠区域的固定长度切分	需要保留上下文连贯性
语义切块	基于语义边界（段落、句子）切分	文档结构清晰，内容关联性强
递归切块	按层级结构递归切分	复杂文档，如 Markdown、HTML
父子块切块	大块+小块的层级结构	需要细粒度检索又保留上下文

切块策略选择建议

在选择切块策略时，需要考虑以下因素：

文档类型：结构化文档（Markdown、HTML）适合递归切块；非结构化文档（PDF、Word）可使用语义切块

块大小：一般建议单个块在 256-512 tokens 之间，过大会导致语义稀释，过小会丢失上下文

重叠度：使用滑动窗口时，建议 10-20% 的重叠度以保持上下文连续性

Embedding 模型适配：不同的 Embedding 模型对输入长度有不同的限制和要求

Embedding 向量化模型

Embedding 模型是 RAG 系统的核心组件，负责将文本转换为高维稠密向量，使语义相似的内容在向量空间中距离相近。

BGE-M3 模型详解

BGE-M3（BAAI General Embedding-M3）是智谱 AI 开源的文本嵌入模型，在 MTEB（Massive Text Embedding Benchmark）评测中表现优异：

多语言支持：支持 100+ 语言的文本嵌入

高密度检索：支持 ColBERT 式的多向量检索模式

长文本支持：最大输入长度可达 8192 tokens

精准语义捕捉：能够精准捕捉文本语义，有效提高向量检索的准确性

强大泛化能力：在大规模中文数据上充分训练，具备出色的泛化能力

5.4.4 知识问答环节模型选型

知识问答是 RAG 系统的核心环节，负责将用户问题转化为回答。该环节主要涉及 Query 向量化模型、大语言模型（LLM）和重排模型（Reranker）。

Query 向量化模型

Query 向量化与知识入库使用相同的 Embedding 模型，确保查询向量与文档向量的语义空间一致。

大语言模型（LLM）选型

LLM 是 RAG 系统的「大脑」，负责基于检索到的内容生成最终回答。选择合适的 LLM 对回答质量和用户体验至关重要。

主流 LLM 模型对比

模型名称	特点	中文能力	适用场景
通义千问	阿里自研，开源可商用	优秀	企业知识库
DeepSeek	高性能，低成本	优秀	复杂推理场景
GLM-5	智谱 AI，中文优化	优秀	学术/专业问答

LLM 选型建议

中文企业场景：推荐使用通义千问或DeepSeek，对中文语义理解更精准

成本敏感场景：DeepSeek系列在保持高性能的同时具有成本优势

长文档理解：选择支持长上下文的模型

需要精确引用：选择支持工具调用和引用溯源的模型

LLM 深度思考模式

模型开启深度思考可在生成回复前先进行推理，从而在逻辑推理、数学计算等复杂任务中提升准确性。星辰 TokenHub 运营服务平台支持深度思考的模型分为两种：

混合推理模式：通过enable_thinking参数控制思考开关

设为true：模型先思考再回复；

设为false：模型直接回复；

仅深度思考模式：模型始终在回复前进行思考，无法关闭，即无需设置enable_thinking参数。

启用思考模式可提升回复质量，但会增加响应延迟和 Token 消耗。使用混合推理的模型时，可根据问题复杂度动态切换，建议简单任务（日常聊天、简单问答）将 enable_thinking 设为 false 关闭思考；复杂任务（逻辑推理、代码生成、数学解答）将 enable_thinking 设为 true 开启思考。

说明：部分模型支持混合推理模式，具体可通过模型卡片 API 文档查看是否支持 enable_thinking 参数。

LLM 流式输出

在实时聊天或长文本生成应用中，可开启流式输出通过持续返回模型生成的文本片段，解决长时间等待全部输出内容和触发服务端超时导致任务失败的问题。

配置方式：通过stream参数控制是否以流式接口的形式返回数据

设为true：模型先思考再回复；

设为false：模型直接回复；

查看Token消耗：OpenAI协议默认不返回Token消耗量，若需要统计流式输出模式下token数目，需将stream_options参数配置为stream_options={"include_usage":True}。

Reranker 重排模型

Reranker（重排模型）用于对初步检索结果进行二次排序，提高相关性。典型的两阶段检索架构是：向量检索 + Reranker 重排。

Reranker 模型

模型名称	模型提供方	特点	适用场景	模型服务请求路径

BGE-Reranker-V2-m3	北京智源研究院（BAAI）	支持 100+ 种语言，具备强大的跨语言检索能力（如用中文搜英文），且对长文本支持更好，整体运行效率更优。	出海/国际化业务、存在多语种混合的知识库，或文档切块（Chunk）较长的综合性 RAG 场景。	https://ai.ctaigw.cn/v1/rerank
BGE-Reranker-Large	北京智源研究院（BAAI）	参数量大，具备极深度的中英文语义理解能力，重排打分的精准度极高。	对检索准确率要求极苛刻的严肃场景（如政务、金融、法务），主要处理纯中文或纯英文文档。	

Reranker 使用建议

对于检索质量要求高的场景，建议使用 Reranker 进行结果优化。

一般流程：向量检索返回 Top 20-50 条结果 → Reranker 重排 → 取 Top 5-10 条送入 LLM。

Reranker 会增加响应延迟，需根据业务需求权衡。

5.4.5 天翼云星辰TokenHub运营服务平台模型部署实践

天翼云息壤星辰 TokenHub 运营服务平台提供强大的模型推理服务能力，支持企业快速部署和管理各类模型。模型调用步骤：

获取 App Key

方式一：

在控制台左侧菜单栏进入“概览”，点击“API 接入”

在 API 快捷接入页面：

选择服务组：从下拉菜单中选择第一步创建的服务组。如果尚未创建，可点击“确认创建并选择”快速创建默认服务组

选择模型：从模型下拉菜单中选择需要接入的模型

选择完成后，页面下方将自动展示该模型支持的所有编程语言的示例代码

方式二：

在左侧菜单栏进入“系统管理-APP Key 管理”，点击“+创建服务组”，填写服务组名称、服务组描述、生效时间、配置服务等信息。

创建服务组后，从服务组上面获取 APP KEY。

5.2 调用模型 API

Embedding 向量化模型、LLM 大语言模型、Reranker（重排模型）的请求路径后缀不同：

功能分类	支持模型类别	请求路径后缀	请求完整路径	功能描述
chat	文本生成、图像理解	/chat/completions	https://ai.ctaigw.cn/v1/chat/completions	针对描述会话的消息列表，模型将返回响应。
Embeddings	文本向量化	/embeddings	https://ai.ctaigw.cn/v1/embeddings	创建表示输入文本的嵌入向量。
Reranker	重排序	/rerank	https://ai.ctaigw.cn/v1/rerank	通常用于计算查询和文档之间的相关性分数。

模型详细 API 文档可查看对应模型卡片-API 文档。

5.4.6 RAG系统最佳实践建议

知识入库最佳实践

文档预处理：上传前对文档进行清洗和格式化，去除无关干扰信息

合理切块：根据文档结构选择合适的切块策略，建议块大小在 256-512 tokens

模型选择：中文企业知识库推荐使用BGE-M3

增量更新：支持文档增量入库，避免全量重建索引

知识问答最佳实践

Query 优化：引导用户使用清晰、具体的问法，提高检索准确性

检索参数调优：根据场景调整 Top-K 检索数量，平衡召回率和精确率

Prompt 工程：设计高质量的 Prompt，引导 LLM 准确引用检索内容

结果溯源：为回答提供引用来源，增强用户信任度

性能优化建议

缓存策略：对频繁查询的 Embedding 结果进行缓存，减少重复计算

异步处理：知识入库采用异步处理模式，不阻塞主流程

向量化加速：使用 GPU 加速 Embedding 模型推理

负载均衡：部署多实例服务，均衡请求负载

5.5 缓存命中

本文为您介绍星辰TokenHub运营服务平台缓存命中能力。

缓存命中概念

在大模型API的实际调用中，许多场景（如长文档问答、多轮对话、带有大量 System Prompt 的角色扮演）会反复向模型发送相同的前置内容。缓存命中是一项旨在优化此类场景的高效计算机制，系统会将您请求中的重复前置内容进行缓存，当您后续的请求携带相同的前置内容时，模型无需重新计算这些 Token。从而降低使用成本，命中缓存的输入Token将享受折扣价。

支持的模型

支持缓存命中能力的模型见各模型卡片，或前往按需计费模式-Tokens查看。

如何提升缓存命中率

在分布式集群架构下，为了提升缓存命中率，您可以在调用模型服务时传入特定的自定义参数。

1. 提升命中率的参数接入说明

在调用支持缓存命中能力的模型时，您可以在请求体Body中携带上下文会话标识参数。

参数名称：prompt_cache_key

参数限制：最大长度不超过64个字符。

参数要求：建议传入具有唯一性的字符串（如用户的 UserID、会话的 SessionID、文档的 DocumentID 等）。属于同一长文档阅读或同一用户连续对话的请求，建议保持该参数值一致。

2.调用请求示例 (cURL)

```
curl --location --request POST
'https://ai.ctaigw.cn/v1/chat/completions' \
--header 'Authorization: Bearer xxx' \
--header 'Content-Type: application/json' \
--header 'Accept: */*' \
--data-raw '{
    "model": "xxx",
    "prompt_cache_key": "my-prefix-cache-key-001",
    "messages": [
        {
            "role": "user",
            "content": "请对北京与上海这两座中国顶级一线城市进行全面、深度、多
角度、客观中立的对比分析。"
        }
    ],
    "max_tokens": 200,
    "temperature": 0.6,
    "top_p": 0.7,
    "repetition_penalty": 1.0,
    "stream": true,
    "enable_thinking": true,
    "stream_options": {
        "include_usage": true
    }
}'
```

3. 如何查看缓存命中情况

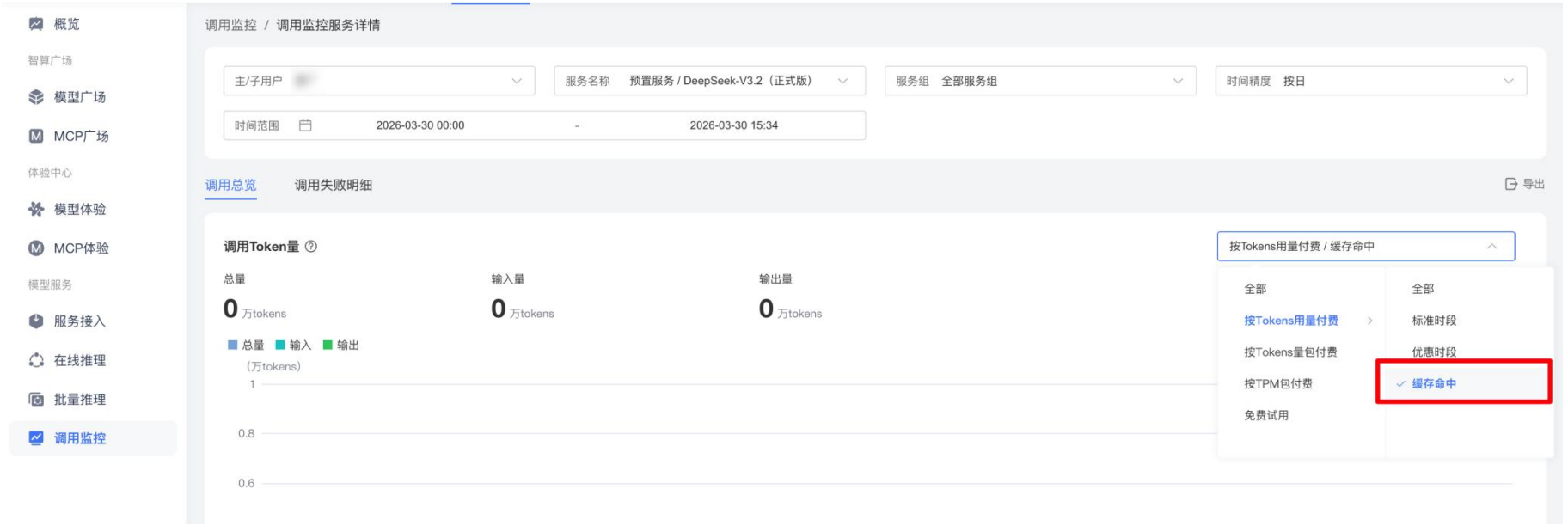
说明

缓存命中属于“尽力而为”的底层优化机制，系统为每个会话维持保留缓存的时间是有限的，如果调用时间过长，或超出了系统承载能力，缓存将被释放。

您可以通过以下两种方式查看您的缓存命中数据：

方式一：通过平台“调用监控”查看

在星辰TokenHub运营服务平台“调用监控服务详情”页面。在调用Token量图表筛选条件中选择“按Token用量付费”-“缓存命中”，即可直观地按时段查看缓存命中的Token量。



方式二：通过 API 返回的 usage 字段实时查看

在模型服务调用返回的信息中，usage字段会透出本次请求Token的消耗明细：

```
{
    "code": 0,
    "id": "chatcmpl-177460325585943900000da17b1bf",
    "choices": [],
    "created": 1774603256,
    "model": "DeepSeek-V3.2",
    "object": "chat.completion.chunk",
    "usage": {
        "completion_tokens": 200,
        "prompt_tokens": 414,
        "total_tokens": 614,
        "prompt_tokens_details": {
            "cached_tokens": 256
        }
    }
}
```


计费说明

关于支持缓存命中的模型，Input (输入) Token 的计费将被自动拆分为两部分：

未命中缓存：按照模型输入单价的标准价格进行计费。

命中缓存：按照模型“缓存命中”的输入价格进行计费。

Output (输出) Token：无论是否命中缓存，输出部分的计费标准保持不变。

注意

缓存命中价格全天时段统一，不参与优惠时段（00:00-08:00）折扣。

推理服务 API

如何调用API

本文为您介绍推理服务 API 调用说明。

6.1.1 终端节点

主要作用：用户信息的发送和接收，信令信息的控制处理、安全保护等作用。

终端节点：https://ai.ctaigw.cn/

6.1.2 接口构造

请求域名

终端请求地址：https://ai.ctaigw.cn（具体模型终端请求地址详见单个模型的 API 文档）。

通信协议

接口通过 HTTPS 进行通信，保护用户数据的机密性和完整性，确保网络通信的安全性。

版本管理

为区分接口版本，在 http 请求路径中加入版本信息，目前版本为 v1，因而请求的 url 前缀为：**https://ai.ctaigw.cn/v1**

请求鉴权

请求 header 中需要填入 Authorization 鉴权信息，Authorization 对应值应为 Bearer + 平台上获取的服务组 App Key

AppKey 获取方法：登录平台门户，星辰 TokenHub 运营服务平台->系统管理->APP Key 管理->创建服务组->绑定服务->获取 appKey

为避免被安全护栏拦截，建议在 http 请求 header 中填入 User-Agent 信息

浏览器或客户端标识：Chrome/58.0.3029.110、Mozilla/5.0、AppleWebKit/537.36、Safari/537.36、Windows NT 10.0、

PostmanRuntime-ApipostRuntime/1.1.0 等

请求示例

```
    请求路径：https://ai.ctaigw.cn/v1/xxx/yyy，其中/xxx/yyy 为具体的功能路径，如
/chat/completions
    请求方式：POST
    请求header必填项：
    Authorization: Bearer AppKey
    Content-Type: application/json
    其他header：
    User-Agent: PostmanRuntime-ApipostRuntime/1.1.0
```

错误码（新版）

本文介绍星辰 TokenHub 平台新版错误码。 适用范围：接入域名使用 https://ai.ctaigw.cn 或 https://ai.ctaigw.com 的请求。

文档概述

本文档适用于平台所有模型 API 接口，包括 OpenAI 兼容接口、Anthropic 兼容 Messages 接口。

错误分为三层：HTTP 状态码 + 协议层错误类型 + 平台自定义业务错误码。

流式接口重要特性：HTTP 返回 200 不代表业务调用成功；推理过程异常会在 SSE 流内部返回错误事件。

不同兼容协议对外返回的 JSON 结构不一样，但底层平台业务错误码语义统一。

6.2.1 错误响应通用约定

非流式错误响应格式

OpenAI 兼容格式示例

```
{
    "error": {
        "type:": "Invalid_request",
        "code": "missing_parameter",
        "message": "Missing required parameters"
    }
}
```

Anthropic 兼容格式示例

```
{
    "type:": "Invalid_request",
    "error": {
        "code": "missing_parameter",
        "message": "Missing required parameters"
    }
}
```

流式 SSE 错误响应格式

OpenAI 兼容格式示例

```
data: {"error": {...}}\n\n
```

Anthropic 兼容格式示例

```
data: {"type":"error","error":{...}}\n\n
```

备注：HTTP Status=200，SSE 流中返回错误事件，不会走到正常结束事件，随后返回 data: [DONE]关闭流。

6.2.2 公共返回字段说明

字段	说明
type	错误类型
code	错误码
message	人类可读错误描述，文案会迭代变更，不要用于代码判断

6.2.3 http状态码总览表

HTTP 状态码	含义	说明
400	Bad Request	请求体格式错误、参数校验失败、参数非法
401	Unauthorized	APIKEY 认证失败
403	Forbidden	权限不足，无空间 / 模型调用权限
404	Not Found	接口地址不存在、模型不存在
422	Unprocessable Entity	参数语义校验失败，模型不支持该能力
429	Too Many Requests	限流、RPM/TPM 超限、配额耗尽
500	Internal Server Error	平台内部未知错误

503	Service Unavailable	模型服务过载、实例不可用
504	Gateway Timeout	超时

6.2.4 业务错误码对照表

请求参数类错误（400状态码）

错误 type	错误 code	错误 message
Invalid_request	json_parse_error	请求体 JSON 解析失败
Invalid_request	missing_parameter	请求缺少必要的参数
Invalid_request	invalid_parameter_value	请求参数取值有问题
Invalid_request	invalid_endpoint	请求 url 错误
Invalid_request	unsupported_api_provider	当前模型不支持所请求的协议

Invalid_request	invalid_http_method	请求 method 错误
content_policy_violation	input_content_violation	输入内容触发安全策略
content_policy_violation	output_content_violation	输出内容触发安全策略
Invalid_request	missing_parameter_model	请求缺少必填参数：模型名称

认证失败（401状态码）

错误 type	错误 code	错误 message
authentication_error	api_key_not_found	ApiKey 不正确，请使用正确的 ApiKey
authentication_error	invalid_api_key	ApiKey 被禁用或者被限额
authentication_error	no_authentication_credential	未携带 Authorization 请求头
authentication_error	api_key_expired	ApiKey 过期
authentication_error	apikey_manual_disabled	APIKEY 被人工禁用

authentication_error	apikey_unavailable	appkey 已被禁用

无权限（403状态码）

错误 type	错误 code	错误 message
account_permission	product_not_activated	产品未开通
account_permission	subaccount_manual_disabled	子账号被手动禁用
account_permission	subaccount_unavailable	账号已被禁用，请联系管理员
account_permission	user_manual_disabled	账号被手动禁用
account_permission	user_unavailable	用户已被禁用，请联系管理员
account_permission	account_frozen	用户被冻结
account_permission	account_quota	账号下的模型被禁用
account_permission	account_abnormal	账号异常

order_permission	order_not_found	订单不存在
model_permission	model_not_activated	模型未开通，请联系管理员
model_permission	model_unavailable_in_region	该区域暂时不支持该模型
model_permission	model_no_authorization	模型未授权，请联系管理员
account_permission	ip_not_in_whitelist	IP 未加入白名单，禁止访问

错误 **type**	错误 **code**	错误 **message**

找不到模型/请求内容（404状态码）

错误 type	**错误 code**	**错误 message**
task_not_found	task_not_found	任务不存在，请确认任务 ID 是否正确
model_offline	model_offline	模型已下线，请联系管理员
model_not_found	model_not_found	模型不存在

限流/限额（429状态码）

错误 type	错误 code	错误 message
quota_error	master_account_quota_exhausted	主账号已触发限额
quota_error	subaccount_token_quota	子账号已触发限额
quota_error	apikey_quota_exhausted	ApiKey 已触发限额
quota_error	user_trial_quota_exhausted	账号试用额度已用完
quota_error	subaccount_trial_quota_exhausted	子账号试用额度已用完
quota_error	apikey_trial_quota_exhausted	APIKEY 试用额度已用完
quota_error	package_quota_exceeded	量包额度已用完
quota_error	model_trial_quota_exhausted	模型免费额度已用完

rate_limit_error	apikey_tpm_limit	ApiKey 请求 TPM 超限，请减少 tokens 后重试
rate_limit_error	apikey_rpm_limit	ApiKey 请求 RPM 超限，请稍后重试
rate_limit_error	apikey_ipm_limit	ApiKey 请求 IPM 超限，请稍后重试
rate_limit_error	apikey_qps_limit	ApiKey 触发 QPS 限流
rate_limit_error	apikey_conc_limit	ApiKey 并发已达上限，请稍后重试
rate_limit_error	model_tpm_limit	模型请求 TPM 超限，请减少 tokens 后重试
rate_limit_error	model_rpm_limit	模型请求 RPM 超限，请稍后重试
rate_limit_error	model_ipm_limit	模型请求 IPM 超限，请稍后重试
rate_limit_error	model_qps_limit	模型触发 QPS 限流

rate_limit_error	model_concurrency_limit	模型并发已达上限，请稍后重试
rate_limit_error	subaccount_concurrency_limit	子账号并发已达上限，请稍后重试
rate_limit_error	subaccount_tpm_limit	子账号请求 TPM 超限，请减少 tokens 后重试
rate_limit_error	subaccount_rpm_limit	子账号请求 RPM 超限，请稍后重试
rate_limit_error	subaccount_ipm_limit	子账号请求 IPM 超限，请稍后重试
rate_limit_error	subaccount_qps_limit	子账号触发 QPS 限流

内部软件异常（500状态码）

错误 type	错误 code	错误 message
server_error	database_error	内部数据查找失败
server_error	data_parse_error	内部数据解析异常

server_error	no_available_provider_group	无可用的模型服务商
server_error	no_available_provider	无可用的模型服务商
server_error	server_error	服务处理请求时发生未知内部错误

过载/超时（503/504状态码）

错误 type	错误 code	错误 message
server_error	model_overloaded	模型负载过高
server_error	model_call_failed	模型调用失败
server_error	model_request_timeout	模型请求超时

错误码（旧版）

本文是一些常见的错误情形及示例。

请求处理过程中出现异常时，服务会对外抛出非 200 的 http 状态码，表明当前请求无法正常完成。

对于异常的请求响应，请求体中会返回 error 结构，error 中返回具体的错误信息。

特殊地，在流式请求中：

如果在流式请求接收处理之前发生了异常，如鉴权、参数校验等问题，与普通的非流式一样返回 http code，并带有 error 结构。

如果在流式请求已经接收，会先对外返回流式请求连接建立的信息，此时 http code 为 200，而在后续模型流式返回过程中发生了异常，会在流失返回的 chunk 返回 error 结构，并终止当前的流式请求。

6.3.1 错误error结构

字段名称	二级字段	类型	必选	描述
error	-	object	是	错误信息
-	code	string	是	平台错误码
-	type	string	是	平台错误类型
-	message	string	是	平台错误详情

6.3.2 错误结果示例

```
{
    "error" : {
      "code" : "500001",
      "type" : "INVOKE_MODEL_ERROR",
      "message" : "服务接口异常，请联系管理员"
    }
}
```

6.3.3 错误码

如遇问题可通过提交工单联系平台管理员。

HTTP 状态码	错误码	错误信息	错误描述
500	INVOKE_MODEL_ERROR	服务接口异常，请联系管理员	推理接口异常
400	PARAM_ERROR	参数不正确，请联系管理员	参数校验失败
500	SYSTEM_ERROR	系统错误， 请联系管理员	系统内部错误
401	INCORRECT_API_KEY_PROVIDED	AppKey 不正确，请使用正确的 AppKey	鉴权信息错误

400	INVOKE_TEXT_AUDIT_ERRO R	调用审查服务接口发生异常，请联系管理员	内容审核失败
400	TEXT_AUDIT_QUESTION_NO T_PASS	很抱歉，关于这个问题我无法提供相应的信息。如果您有其他问题，我将很愿意为您回答。	问题未通过内容审核
400	TEXT_AUDIT_ANSWER_NOT _PASS	很抱歉，关于这个问题我无法提供相应的信息。如果您有其他问题，我将很愿意为您回答。	答案未通过内容审核
429	RESOURCES_TIPS	免费额度已用完或已到期，或账户金额不足。请开通付费	免费试用限制
429	USER_RPM_RATELIMITING	用户请求 RPM 超限，请稍后重试	RPM（Requests Per Minute）针对用户每分钟允许的请求次数超过限制
429	USER_TPM_RATELIMITING	用户请求 TPM 超限，请减少 tokens 后重试	TPM（Tokens Per Minute）针对用户每分钟消耗的 token 超出限制

429	AK_BANNED	您的请求 AK 已被封禁，请更换 AK 或联系后台人员解封后重试	App Key 被封禁
429	MODEL_RPM_RATELIMITING	模型请求 RPM 超限，请稍后重试	RPM（Requests Per Minute）针对模型侧每分钟允许的请求次数超过限制
429	MODEL_TPM_RATELIMITING	模型请求 TPM 超限，请减少 tokens 后重试	TPM（Tokens Per Minute）针对模型侧的每分钟消耗的 token 超出限制

API

6.4.1 chat 对话API

本文是关于 Chat 对话 API 的详情描述。

1 接口描述

-	描述
接口名称	对话
请求路径	https://ai.ctaigw.cn/v1/chat/completions
功能描述	针对描述会话的消息列表，模型将返回响应

2 请求参数

2.1 请求头参数

参数	示例值	描述
Authorization	Bearer AppKey	鉴权信息填入 AppKey。
Content-Type	application/json	

2.2 请求参数

备注：此参数为全平台模型通用，每个模型支持的参数、参数范围可能因模型不同而有所差异，详细可见模型广场内每个模型的 API 文档。

参数名称	二级参数	三级参数	四级参数	类型	必选	描述
model	-	-	-	string	是	模型 ID。
messages	-	-	-	array	是	用户当前输入的期望模型执行指令。一个列表内多个字典，支持多轮对话。 对话列表，每个列表项为一个 message object， message object 中包含用户 role 和 content 两部分信息： role 可选值为 user、assistant、system；

						role 为 system 时，不校验 content 空值，且 message 中 system 只能位于开头，即 messages[0]位置； role 为 user 时说明是用户提问，role 为 assistant 时说明是模型回答，而 content 为实际的对话内容； 单轮/多轮对话中，最后一个 message 中 role 必须为 user，content 为用户输入的最新问

						题，其余结果除 system 角色外都为历史信息拼接送入 messages 中，assistant 和 user 的 role 只能交替出现，assistant 后只能跟 user，user 后只能跟 assistant。
-	role	-	-	string	否	对话角色，role 类型枚举值：user、assistant、system。
-	content	-	-	string/array	是	对话内容，内容目前有两种格式：string,array。

						string 类型：表示文本对话内容。 array 类型：表示多个对话内容列表，每个列表项为一个 content object，每个 content object 包含 type、image_url、text 等信息。 type 可选值为 text、image_url。 type 为 text 时，取 text 字段作为对话内容。 type 为 image_url 时，取 image_url 字

						段作为对话内容。
-	-	type	-	string	否	对话内容类型，type 类型枚举值：text,image_url。
-	-	text	-	string	否	文本对话内容，type 为 text 时传入。
-	-	image_url		object	否	图片对话内容，type 为 image_url 时传入。
-	-	-	url	string	否	图片对话内容中的图片地址，目前可以为二进制数据的 base64 编码。

frequency_penalty	-	-	-	float	否	频率惩罚。它影响模型如何根据文本中词汇 token 的现有频率惩罚新词汇 token。值大于 0，会根据新标记在文本中的现有频率来惩罚新标记，从而降低模型逐字重复同一行的可能性。一般取值范围[-2, 2]，**具体取值范围、默认值需见对应模型。**
max_tokens	-	-	-	int	否	最大生成长度。控制最大生成长度，超过该值则截断。

						一般取值范围(0, 2048]，**具体取值范围、默认值需见对应模型。**
n	-	-	-	int	否	1-n 个 choices。
presence_penalty	-	-	-	float	否	存在惩罚。用户控制模型生成时整个序列中的重复度。 一般取值范围[-2.0, 2.0]，**具体取值范围、默认值需见对应模型。**
response_format	-	-	-	object	否	返回格式。
-	type	-	-	string	否	返回格式枚举值：

						text,json_object。
seed	-	-	-	int	否	随机种子。用于指定推理过程的随机种子，相同的 seed 值可以确保推理结果的可重现性，不同的 seed 值会提升推理结果的随机性。 一般取值范围(0,9223372036854775807]，**具体取值范围、默认值需见对应模型。**
stop				string/array	否	生成停止标识。当模型生成结果以 stop 中某个元

						素结尾时，停止文本生成。
stream	-	-	-	bool	否	是否以流式接口的形式返回数据。默认为 False，非流式。
stream_options	-	-	-	object	否	流式选项，stream 为 True 有效。
-	include_usage	-	-	bool	否	是否在返回中包含 usage，stream 为 True 有效。**取值为 True 时，会在流式返回的最后一个 chunk 里返回 usage 信息，并该 chunk 中**

						choices 列表为空。
temperature	-	-	-	float	否	温度采样。该值越高生成文本的多样性越高，该值越低生成文本的确定性越高。 一般取值范围(0, 2)，**具体取值范围、默认值需见对应模型。**
top_k	-	-	-	int	否	top_k 采样。取值越大，生成的随机性越高；取值越小，生成的确定性越高。 一般取值范围[1, 100]，**具体取值范围、默认值需见对应模型。**

top_p				float	否	top_p 采样。该值越高生成文本的多样性越高，该值越低生成文本的确定性越高。该值为 0 时没有随机性。一般取值范围(0, 1]，**具体取值范围、默认值需见对应模型**
user	-	-	-	string	否	用户唯一身份ID。

请求参数示例

```
{
  "model": "1234567890",    // 模型ID
  "messages": [
    {
      "role": "user",
      "content": "Hello!"
    }
  ]
}
```

3 请求返回

3.1非流式返回

3.1.1非流式正常返回

字段名称	二级字段	三级字段	字段类型	描述
id	-	-	string	唯一标识符
choices	-	-	string	choices 列表

-	index	-	int	choice 索引
-	message	-	object	模型生成的消息
-	-	role	string	对话角色
-	-	content	string	对话消息内容
-	finish_reason	-	string	模型停止生成标记的原因。 stop: 模型生成遇到自然停止点或提供的停止序列; length: 达到请求中指定的最大标记数; content_filter：如果由于内容过滤器中的标志而省略了内容 tool_calls/function_call：模型调用了函数。

created	-	-	int	Unix 时间戳（以秒为单位）。
model	-	-	string	调用的模型名称。
object	-	-	string	返回的对象类型。非流式返回始终为：chat.completion
usage	-	-	object	请求使用情况的统计信息。
-	completion_tokens	-	int	生成 token 数。
-	prompt_tokens	-	int	输入 token 数。
-	total_tokens	-	int	使用的 token 总数（prompt + completion）。

返回结果示例

```
{
  "id": "chatcmpl-123",
  "object": "chat.completion",
  "created": 1677652288,
  "model": "xxx-chat",
  "choices": [{
    "index": 0,
    "finish_reason": "stop",
    "message": {
      "role": "assistant",
      "content": "\n\nHello there, how may I assist you today?"
    }
  }],
  "usage": {
    "prompt_tokens": 9,
    "completion_tokens": 12,
    "total_tokens": 21
  }
}
```

3.1.2 非流式异常返回

非流式异常返回时：

http code 返回非 200。

http body 中返回 error 结构，error 结构中包含 code、type、message、param 等信息，具体可见【推理服务 API—错误处理】章节内容。

错误结果示例

```
{
  "error" : {
    "code" : "500001",
    "type" : "INVOKE_MODEL_ERROR",
    "message" : "服务接口异常，请联系管理员"
  }
}
```

3.2 流式返回

3.2.1 流式正常返回

字段名称	二级字段	三级字段	字段类型	描述
id	-	-	string	唯一标识符
choices	-	-	string	choices 列表
-	index	-	int	choice 索引
-	delta	-	object	模型生成的消息

-	-	role	string	对话角色
-	-	content	string	对话消息内容
-	finish_reason	-	string	模型停止生成标记的原因。 stop: 模型生成遇到自然停止点或提供的停止序列; length: 达到请求中指定的最大标记数; content_filter：如果由于内容过滤器中的标志而省略了内容 tool_calls/function_call： 模型调用了函数。
created	-	-	int	Unix 时间戳（以秒为单位）。
model	-	-	string	调用的模型名称。

object	-	-	string	返回的对象类型。流式返回始终为：chat.completion.chunk
usage	-	-	object	请求使用情况的统计信息。 仅在 stream_options: {"include_usage": true} 设置时显示。如果存在，则它包含一个 null 值，但最后一个块包含整个请求的 token 使用情况的统计信息。
-	completion_tokens	-	int	生成 token 数。
-	prompt_tokens	-	int	输入 token 数。
-	total_tokens		int	使用的 token 总数（prompt + completion）。

返回结果示例

```
    {"id":"chatcmpl-
123","object":"chat.completion.chunk","created":1694268190,"model":"xxx-chat",
"choices":[{"index":0,"delta":{"role":"assistant"},"finish_reason":null}]}
    {"id":"chatcmpl-
123","object":"chat.completion.chunk","created":1694268190,"model":"xxx-chat",
"choices":[{"index":0,"delta":{"content":"Hello"},"finish_reason":null}]}
    ....
    {"id":"chatcmpl-
123","object":"chat.completion.chunk","created":1694268190,"model":"xxx-chat",
"choices":[{"index":0,"delta":{},"finish_reason":"stop"}]}
```

stream_options.include_usage 为 True 时多返回一条包含 usage 流式消息。

```
    {"id":"chatcmpl-
123","object":"chat.completion.chunk","created":1694268199,"model":"xxx-chat",
"choices":[], "usage": {"prompt_tokens": 9,"completion_tokens": 120,"total_tokens":
129}}
```

3.2.2 流式异常返回

流式异常分为两种：

如果在流式请求接收处理之前发生了异常，如鉴权、参数校验等问题，与普通的非流式一样返回 http code，并带有 error 结构。

如果在流式请求已经接收，会先对外返回流式请求连接建立的信息，此时 http code 为 200，而在后续模型流式返回过程中发生了异常，会在流失返回的 chunk 返回 error 结构，并终止当前的流式请求。

流式请求建立后的异常返回示例

```
    {"id":"chatcmpl-
123","object":"chat.completion.chunk","created":1694268190,"model":"xxx-chat",
"choices":[{"index":0,"delta":{"role":"assistant"},"finish_reason":null}]}
    {"id":"chatcmpl-
123","object":"chat.completion.chunk","created":1694268190,"model":"xxx-chat",
"choices":[{"index":0,"delta":{"content":"Hello"},"finish_reason":null}]}
    ....
    {"error":{"code":"500001","type":"INVOKE_MODEL_ERROR","message":"服务接口异常，请联系管
理员"}}
```

4 请求示例代码

假设平台用户组AppKey=884c8fc4054548a7b1ca1123592f5b7，模型ID=96dcaaaaaaaaaaa5ff55ea377831a，以此为例进行说明。

4.1 curl方式请求

```
curl --request POST \
  --url https://ai.ctaigw.cn/v1/chat/completions \
  --header 'Accept: */*' \
  --header 'Accept-Encoding: gzip, deflate, br' \
  --header 'Authorization: Bearer 884c8fc4054548a7b1ca1123592f5b7' \
  --header 'Content-Type: application/json' \
  --data '{
        "model": "96dcaaaaaaaaaaa5ff55ea377831a",
        "messages": [
                {
                        "role": "user",
                        "content": "Hello"
                }
        ]
}'
```

4.2 python方式请求

```
import json
import requests
URL = "https://ai.ctaigw.cn/v1/chat/completions"
headers = {
    "Authorization": "Bearer 884c8fc4054548a7b1ca1123592f5b7",
    "Content-Type": "application/json"
}
data = {
    "model": "96dcaaaaaaaaaaa5ff55ea377831a",
    "messages": [
        {"role": "user", "content": "Hello"}
    ],
    "stream": True
}
try:
    response = requests.post(URL, headers=headers, json=data, stream=True)
    if response.status_code != 200:
        print(response.text)
    else
        for line in response.iter_lines(chunk_size=8192, decode_unicode=True):
            ## 处理请求
            if line :
                if "[DONE]" == line:
                     break
                # 去除data前缀:
                json_string = line.removeprefix("data:")
                # 转为json:
                jsonData = json.loads(json_string)
```

4.3 openai 客户端示例代码

```
    import openai
    from openai import OpenAI
    client = OpenAI(base_url="https://ai.ctaigw.cn/v1",
api_key="884c8fc4054548a7b1ca1123592f5b7")
    messages = [
        {"role": "user", "content": "Hello"}
    ]
    try:
        stream = client.chat.completions.create(
            model="96dcaaaaaaaaaaa5ff55ea377831a",
            messages=messages,
            stream=True
        )
        # 流式
        for chunk in stream:
            print(chunk.choices[0].delta.content or "", end="", flush=True)
    except openai.APIStatusError as e:
        print(f"APIStatusError: {e.status_code}, {e.message}, {e.body}")
    except openai.APIError as e:
        print(f"APIError: {e.body}")
    except Exception as e:
        print(f"Exception: {e}")
```

6.4.2 image 文本生图API

本文是关于 Image 文本生图 API 的详细描述。

1 接口描述

-	描述
接口名称	文本生图
请求路径	https://ai.ctaigw.cn/v1/images/generations
功能描述	基于提示创建图像

2 请求参数

2.1 请求头参数

参数	示例值	描述
Authorization	Bearer AppKey	鉴权信息填入 AppKey。

Content-Type	application/json	-

2.2 请求参数

备注：此参数为全平台模型通用，每个模型支持的参数、参数范围可能因模型不同而有所差异，详细可见模型广场内每个模型的 API 文档。

参数名称	参数类型	必选	描述
model	string	是	模型 ID。
prompt	string	是	所需图像的文本描述，最大长度 1000。
negative_prompt	string	否	避免生成负面图像的提示，最大长度 1000。
n	int	否	要生成的图像数。 一般取值为 1，**具体取值范围、默认值需见对应模型。**

num_inference_steps	int	否	去噪步骤的数量。去噪步骤越多，通常可以得到更高质量的图像，但推理速度会变慢。 一般取值范围 [1, 50]，**具体取值范围、默认值需见对应模型。**
quality	int	否	生成图像质量。openai 参数，平台未启用。
size	string	否	输出的图像尺寸（高 x 宽）。 一般取值范围 360x640,480x640,512x512,640x360,640x480,720x1280,1024x1024,1280x720，默认值为 512x512，**具体取值范围、默认值需见对应模型。**
response_format	string	否	返回图像的格式。 一般取值范围 url, b64_json，默认值为 b64_json，**具体取值范围、默认值需见对应模型。**

style	string	否	生成图像风格。 一般取值范围 standard,portrait,landscape,cartoon,technology,photography,concept，默认值为standard，**具体取值范围、默认值需见对应模型。**
user	string	否	用户唯一身份 ID。

请求参数示例

```
{
  "model": "bfetrcdggsdfsdf",
  "prompt": "A cute baby sea otter",
  "n": 1,
  "size": "512x512"
}
```

3 请求返回

3.1请求正常返回

字段名称	二级字段	字段类型	描述
created	-	string	Unix 时间戳（以秒为单位）。
data	-	array	图像列表
-	b64_json	string	b64_json 格式图片
-	url	string	url 格式图片
-	revised_prompt	string	实际使用的修改后的 prompt

返回结果示例

```
{
  "id": "1714459832",
  "data": [{
    "b64_json": "xxxxxxxxxxxxxxxxxx"
  }]
}
```

3.2异常返回

异常返回时：

http code 返回非 200。

http body 中返回 error 结构，error 结构中包含 code、type、message、param 等信息，具体可见【推理服务 API—错误处理】章节内容。

错误结果示例

```
{
  "error" : {
    "code" : "500001",
    "type" : "INVOKE_MODEL_ERROR",
    "message" : "服务接口异常，请联系管理员"
  }
}
```

4 请求示例代码

假设平台用户组AppKey=884c8fc4054548a7b1ca1123592f5b7，模型ID=96dcaaaaaaaaaaa5ff55ea377831a，以此为例进行说明。

4.1 curl方式请求

```
curl --request POST \
  --url https://ai.ctaigw.cn/v1/images/generations \
  --header 'Accept: */*' \
  --header 'Accept-Encoding: gzip, deflate, br' \
  --header 'Authorization: Bearer 884c8fc4054548a7b1ca1123592f5b7' \
  --header 'Content-Type: application/json' \
  --data '{
        "model": "96dcaaaaaaaaaaa5ff55ea377831a",
        "prompt" : "A cute baby sea otter",
        "n" : 1,
        "size" : "512x512"
}'
```

4.2 python方式请求

```
import json
import requests
URL = "https://ai.ctaigw.cn/v1/images/generations"
headers = {
    "Authorization": "Bearer 884c8fc4054548a7b1ca1123592f5b7",
    "Content-Type": "application/json"
}
data = {
        "model": "96dcaaaaaaaaaaa5ff55ea377831a",
        "prompt" : "A cute baby sea otter",
        "n" : 1,
        "size" : "512x512"
}
try:
    response = requests.post(URL, headers=headers, json=data)
    if response.status_code != 200:
        print(response.json())
    else:
        b64_file_json = response.json()["data"][0]["b64_json"]
except Exception as e:
    print(f"Exception: {e}")
```

4.3 openai 客户端示例代码

```
    import openai
    from openai import OpenAI
    client = OpenAI(base_url="https://ai.ctaigw.cn/v1",
api_key="884c8fc4054548a7b1ca1123592f5b7")
    try:
        response = client.images.generate(
            model="96dcaaaaaaaaaaa5ff55ea377831a",
            prompt="A cute baby sea otter",
        )
        print(f"{response.data[0].b64_json}")
    except openai.APIStatusError as e:
        print(f"APIStatusError: {e.status_code}, {e.message}, {e.body}")
    except openai.APIError as e:
        print(f"APIError: {e.body}")
    except Exception as e:
        print(f"Exception: {e}")
```

6.4.3 Embeddings文本向量化API

本文是关于 Embeddings 文本向量化 API 的详情描述。

1 接口描述

-	描述
接口名称	嵌入
请求路径	https://ai.ctaigw.cn/v1/embeddings
功能描述	创建表示输入文本的嵌入向量

2 请求参数

2.1 请求头参数

参数	示例值	描述
Authorization	Bearer 123456789	鉴权信息填入 AppKey
Content-Type	application/json	

2.2 请求参数

参数名称	参数类型	必选	描述

model	string	是	模型 ID。
input	object	是	字符串或数组类型，表示输入的文本，编码为字符串或令牌数组（整型数组、字符串数组、整型数组的数组）。要在单个请求中为多个输入获取嵌入，请传递字符串数组或令牌数组的数组。每个输入的长度不得超过 8192 个令牌。
encoding_format	string	否	编码格式，float 或 base64，默认为 float。
dimensions	int	否	输出的 embeddings 维度，仅支持 text-embedding-3 及其高版本。
user	string	否	用户唯一身份 ID。

3 请求返回

3.1 请求正常返回

字段名称	二级字段	三级字段	字段类型	描述
model	-	-	string	调用的模型名称。
object	-	-	string	返回的对象类型，默认为 list
data	-	-	array	embedding 列表
-	index	-	int	index 索引
-	embeddings	-	array/string	embedding 嵌入向量的 float 数组 或 base64 字符串
-	object	-	string	默认为 embedding
usage	-	-	object	请求使用情况的统计信息。
-	prompt_tokens	-	int	输入 token 数。

-	total_tokens	-	int	使用的 token 总数。

3.2 异常返回

异常返回时：

http code 返回非 200。

http body 中返回 error 结构，error 结构中包含 code、type、message、param 等信息，具体可见 OpenAPI 接口文档中的 error 结构描述及错误码部分介绍。

错误结果示例

```
{
  "error" : {
    "code" : "500001",
    "type" : "INVOKE_MODEL_ERROR",
    "message" : "服务接口异常，请联系管理员"
  }
}
```

4 请求示例代码

假设星辰 TokenHub 运营服务平台用户组 AppKey=884c8fc4054548a7b1ca1123592f5b7，模型 ID=96dcaaaaaaaaaaa5ff55ea377831a，以此为例进行说明。

4.1 curl方式请求

```
curl --request POST \
  --url https://ai.ctaigw.cn/v1/embeddings \
  --header 'Accept: */*' \
  --header 'Accept-Encoding: gzip, deflate, br' \
  --header 'Authorization: Bearer 884c8fc4054548a7b1ca1123592f5b7' \
  --header 'Content-Type: application/json' \
  --data '{
        "model": "4b38a36cabe247c8a996206984c9e2a61",
        "input" : "A cute baby sea otter"
}'
```

4.2 python方式请求

```
import json
import requests
URL = "https://ai.ctaigw.cn/v1/embeddings"
headers = {
    "Authorization": "Bearer 884c8fc4054548a7b1ca1123592f5b7",
    "Content-Type": "application/json"
}
data = {
        "model": "4b38a36cabe247c8a996206984c9e2a61",
        "input" : "A cute baby sea otter"
}
try:
    response = requests.post(URL, headers=headers, json=data)
    if response.status_code != 200:
        print(response.json())
    else:
        embeddings = response.json()["data"][0]["embeddings"]
except Exception as e:
    print(f"Exception: {e}")
```

4.3 openai 客户端示例代码

```
    import openai
    from openai import OpenAI
    client = OpenAI(base_url="https://ai.ctaigw.cn/v1",
api_key="884c8fc4054548a7b1ca1123592f5b7")
    try:
        response = client.embeddings(
            model="4b38a36cabe247c8a996206984c9e2a61",
            input="A cute baby sea otter",
        )
        print(f"{response}")
    except openai.APIStatusError as e:
        print(f"APIStatusError: {e.status_code}, {e.message}, {e.body}")
    except openai.APIError as e:
        print(f"APIError: {e.body}")
    except Exception as e:
        print(f"Exception: {e}")
```

6.4.4 Reranker 重排序API

1.接口描述

-	**描述**
接口名称	重排序
请求路径	https://ai.ctaigw.cn/v1/rerank
功能描述	通常用于计算查询和文档之间的相关性分数。

2.请求参数

2.1请求头header参数

参数	类型	必选	示例值	描述
Content-Type	String	是	application/json	json格式
Authorization	String	是	Bearer APP_KEY	请求鉴权的 App Key，由平台提供。Bearer 开头

2.2请求体body参数

名称	类型	必选	示例值	描述
model	String	是	96dcaaaaaaaaaaa5ff55ea377831a	模型id，请在模型详情页顶部区域查看，样例：96dcaaaaaaaaaaa5ff55ea377831a
query	String	是	人工智能的发展趋势	待重排序的查询文本（例如用户搜索词）
documents	Array	是	["人工智能正在快速发展","机器学习是AI的核心技术","大语言模型改变了AI领域"]	待排序的文档列表（每个元素为一个文档文本，例如搜索结果中的候选文本）
truncate_prompt_tokens	Int	否	8192	最大token值，取值范围（1, 8192）

top_n	Int	否	2	返回top_n个文档，取值范围大于0，且小于documents的数量
return_documents	Boolean	否	false	是否返回原文档

2.3请求代码示例

使用前请替换具体的${APP_KEY}和${MODEL_ID}值

```
curl --location 'https://ai.ctaigw.cn/v1/rerank' \
--header 'Authorization: Bearer ${APP_KEY}' \
--header 'Content-Type: application/json' \
--data '{
 "model": "${MODEL_ID}",
 "query": "人工智能的发展趋势",
 "documents": [
      "人工智能正在快速发展。",
      "机器学习是AI的核心技术。",
      "大语言模型改变了AI领域。",
      "自然语言是AI领域的里程碑"
  ],
  "top_n": 2,
  "return_documents": true
 }'
```

3.返回值说明

3.1请求成功返回响应参数

名称	类型	描述	示例值	下级对象

code	Int	接口状态码，返回状态，返回 0 表示成功	0	-
id	String	请求标识	endpoint_common_11684	-
results	Array	模型返回的结果	-	results
created	Int	回答的创建时间，由模型侧返回	1719904068	
model	String	模型名称	Bge-Rerank-Large	-

表results

名称	类型	描述	示例值	下级对象
index	Int	在原文档中对应索引	0	-
relevance_score	Float	分数	0.992	

document	Object	文档内容，当return_documents为true时，无此字段	-	document

表document

名称	类型	描述	示例值	下级对象
text	String	目标文档	人工智能正在快速发展	-

3.2请求失败返回响应参数

字段名	二级字段	类型	描述	示例值
error	-	Object	错误信息	-
-	code	String	错误码，返回错误代码参考下面的错误代码列表	500002

-	type	String	返回对应的错误码	PARAM_ERROR
-	message	String	返回对应的错误信息	modelId不正确，请重试

3.3错误码

如遇模型调用失败且返回报错信息，可访问官网错误处理页面，寻求解决方法。

3.4返回值示例

请求成功返回值示例

```
{
 "code": 0,
 "id": "34870e0590084e40800f856bc8e08201",
 "model": "b2d640f305f64c52ab5a04cd8565a233",
 "results": [
  {
   "index": 0,
   "document": {
    "text": "人工智能正在快速发展。"
   },
   "relevance_score": 0.992597
  },
  {
   "index": 2,
   "document": {
    "text": "大语言模型改变了AI领域。"
   },
   "relevance_score": 0.8079157
  }
 ]
}
```


请求失败返回值示例

```
{
 "error": {
 "code": "500001",
 "message": "算法服务接口异常，请联系管理员",
 "type": "INVOKE_MODEL_ERROR"
 }
}
```

6.4.5 模型列表API

本文是关于模型列表 API 的具体描述。

1.接口描述

-	**描述**
接口名称	模型列表
请求路径	https://ai.ctaigw.cn/v1/models

功能描述	用于获取用户模型列表

2.请求参数

2.1 请求头参数

参数	示例值	描述
Authorization	Bearer AppKey	鉴权信息填入 AppKey。
Content-Type	application/json	-

3.请求返回

3.1请求正常返回

字段名称	二级字段	字段类型	描述
object	-	string	返回的对象类型，默认为 list

data	-	array	模型列表
-	id	string	模型 id 或者模型名称
-	object	string	默认为 model
-	created	string	创建时间，默认为空
-	owned_by	string	组织 机构，默认为空

返回结果示例

```
{
    "object": "list",
    "data": [
        {
            "id": "测试模型名称1",
            "object": "model",
            "created": "",
            "owned_by": ""
        }
    ]
}
```

3.2异常返回

异常返回时：

http code 返回非 200。

http body 中返回 error 结构，error 结构中包含 code、type、message、param 等信息，具体可见【推理服务 API—错误处理】章节内容。

错误结果示例

```
{
"code": 500004,
"detail": "AppKey不正确，请使用正确的AppKey",
"message": "INCORRECT_API_KEY_PROVIDED",
"error": {
    "code": "500004",
    "message": "AppKey不正确，请使用正确的AppKey",
    "type": "INCORRECT_API_KEY_PROVIDED"
    }
}
```

4.请求示例代码

假设平台用户组AppKey=884c8fc4054548a7b1ca1123592f5b7，以此为例进行说明。

4.1 curl方式请求

```
curl --request get \
  --url https://ai.ctaigw.cn/v1/models \
  --header 'Accept: */*' \
  --header 'Accept-Encoding: gzip, deflate, br' \
  --header 'Authorization: Bearer 884c8fc4054548a7b1ca1123592f5b7' \
  --header 'Content-Type: application/json' \
```

4.2 python方式请求

```
import json
import requests
URL = "https://ai.ctaigw.cn/v1/models"
headers = {
    "Authorization": "Bearer 884c8fc4054548a7b1ca1123592f5b7",
    "Content-Type": "application/json"
}
try:
    response = requests.get(URL, headers=headers)
    if response.status_code != 200:
       print(response.text)
    else:
        data = response.json()["data"]
        print(data)
except Exception as e:
    print(f"Exception: {e}")
```

4.3 openai 客户端示例代码

```
    import openai
    from openai import OpenAI
    client = OpenAI(base_url="https://ai.ctaigw.cn/v1/models",
api_key="884c8fc4054548a7b1ca1123592f5b7")
    try:
        response = openai.models.list()
        print(f"{response}")
    except openai.APIStatusError as e:
        print(f"APIStatusError: {e.status_code}, {e.message}, {e.body}")
    except openai.APIError as e:
        print(f"APIError: {e.body}")
    except Exception as e:
        print(f"Exception: {e}")
```

平台功能 OpenAPI

平台功能API使用说明

OpenAPI 门户提供了产品的 API 文档、API 调试、SDK 中心等。

关于用户如何使用产品功能 API 的详细介绍，请参见 API 文档。您可以在 OpenAPI 门户了解到具体的调用前必知、API 概览、如何调用 API 以及具体的 API 的接口详细说明。

常见问题

计费类

本文向您介绍星辰 TokenHub 运营服务平台计费相关常见问题。

星辰TokenHub运营服务平台支持哪些计费方式?

星辰TokenHub运营服务平台支持包周期计费模式、按需计费模式-卡时和按需计费模式-按量三种计费方式。

预付费和后付费的账单分别是以何种维度结算?

星辰TokenHub运营服务平台后付费账单是按月结算，预付费账单是按小时结算。

账户欠费后如何充值?

请前往天翼云官网进行充值，详细操作参见在线充值。

为什么账户欠费后仍在出账?

欠费自账单出具起算，而账单出账通常存在 约1 小时的延迟。在此期间，系统按照实际使用量正常出账。

大模型免费token额度使用规则是什么?

每个模型提供一定额度且限期2周的免费调用量，限期计算以领取免费额度时间为起始时间，主/子账号共用免费额度及有效期周期额度。

免费额度用完或到期后会转入按量付费使用，可选在可开通付费服务继续付费使用，开通付费服务后仍优先消耗剩余免费额度，免费额度用完或到期后自动转入付费使用。具体模型对应免费额度可前往平台“模型推理--预置服务”详情页查看。

大模型免费额度用完后，不希望产生按需费用要怎么处理?

开启免费体验时，请主动勾选“免费额度耗尽或过期后，自动关闭服务”，如下图所示。若未勾选此选项，在免费额度失效后，服务仍然可以继续正常调用，将按实际用量和模型单价计费。


开启免费体验
[redacted]
开启后即可进行免费调用模型服务，开启免费体验后不可关闭
单模型免费权益
1.赠送 [redacted]万 tokens 免费额度可用于体验额度，有效期 14 天，从开启免费体验服务开始； 历史免费额度未用完/到期的用户叠加 [redacted] 万 tokens，有效期重置为 14 天；
2.仅主账号支持开启免费体验，开启后主子账号共享体验额度。
开启服务
不开启，无法调用
开启后，免费额度耗尽时会根据您的实际使用情况从账户中扣除费用，不使用则不产生费用
免费额度耗尽或到期后，自动关闭服务
计费单价
输入 ¥ 6.5 /百万Tokens
输出 /百万Tokens
取 消
确认开启

操作类

本文向您介绍星辰 TokenHub 运营服务平台操作类常见问题。

使用第三方工具对接API访问报错404：请求路径错误怎么解决?

修改配置的请求路径配置到/v1，不同工具的具体配置请参考最佳实践文档。

模型回答内容重复如何处理?

请求模型的重复惩罚度参数设置过高，会导致重复输出概率增大。建议将重复惩罚度参数设置为0。

API接口调用错误（错误码：500001）怎么解决?

大概率为客户端主动断开连接问题，请检查客户端设置的超时时间是否过短或者存在主动断开情况。

可信Token专区内模型与其他预置模型有何区别?

可信Token专区内模型采用专属智算集群，独立机柜、资源独享、网络隔离，所有模型基于可信计算（TEE）环境部署在中国电信自有机房，绝不调用转发互联网第三方模型服务。模型推理全流程数据采取VPN加密传输，安全可靠。

权限类

本文向您介绍星辰 TokenHub 运营服务平台用户权限类常见问题。

为什么在“模型广场”见到的模型在“模型推理”看不见且无法调用?

需确认前往账号中心您的账号属性是否为子账号。

子账号调用模型需要得到主账号的授权，请联系主账号在“控制台->模型授权”模块添加授权。

主账号如何管理子账号的App Key?

主账号自身可在“APP Key管理”模块，选择对应子账号的用户名称查询其创建的服务组，支持主账号对子账号创建服务组进行创建、编辑、停用、删除等操作。

如何申请开通可信Token专区服务?

用户需通过天翼云客户经理提交业务需求单，经审核通过后，即可获得专区访问权限。

平台升级与用户割接说明

本文向您介绍星辰TokenHub运营服务平台平台升级与用户割接说明。

8.4.1 升级背景

平台已升级至 v2.0 版本，本次升级实现了底层架构、产品能力与使用体验的全方位升级迭代，将提供更流畅的交互、更丰富的功能以及更完善的数据安全机制。

由于新版本采用了全新的数据架构与存储规范，旧版本产生的部分数据无法被新版本系统自动兼容，需要进行数据割接（即把旧版本系统的数据按新规范迁移至新版本系统），割接完成之后无法回滚。为充分保障您的数据权益与知情权，本次割接需由您主动阅读并同意后方可执行。

升级后您将获得

赠送免费额度：支持免费体验的模型赠送 50 万 tokens 免费推理额度（自领取之日起 14 天有效）。若您此前有未用完且未到期的免费额度，将在剩余额度基础上叠加 50 万 tokens，有效期重置为 14 天。

一键开通全部模型按量付费：完成本次升级后，您将获得全量模型及后续新增模型的按量计费使用权限，无需每个模型单独订购；同时可依托独立功能开关管控单款模型的调用启停状态，灵活配置各模型调用权限。

更丰富的模型选择：升级后，平台将开放更多类型、更多版本的模型供您调用，覆盖更广泛的业务场景与性能需求。

更多管控能力：升级后，您将获得（1）控制子用户调用模型范围能力；（2）按服务组粒度设置模型调用额度上限的能力。

优化主子账户权限体系：升级后，子用户将不再允许自助启用、订购模型服务。相关权限收敛至主用户。如存在历史模型服务订单，需要在升级后完成旧订单退订，才可释放全量用户的模型服务使用权限。历史存量子用户，近三个月内未调用模型服务的将无法继续调用模型。

闲时推理下线调整：升级后，GLM-5（正式版）、DeepSeek-V3.2（旗舰版）、Qwen3.5-397B-A17B（正式版）等模型原有优惠时段（0:00-8:00）折扣取消。

割接涉及数据范围

数据类别	是否割接	说明
账户基本信息	迁移	原样迁移至新系统，您的登录方式不变。
业务数据（您创建的任务/APP Key 等）	迁移	全量迁移，迁移后可在新版本查看与编辑。
子用户模型调用权限	迁移	历史主用户下存在子用户已开通服务，升级后系统将授权全部子用户全部模型的调用权限。如需调整可在割接升级后前往控制台“权限管理”操作。
模型调用监控数据	不迁移	新版本控制台仅支持查看割接后的调用数据；割接前的历史调用数据可在控制台「历

		史监控」入口查询。
已过期/失效的临时数据	**不迁移**	如草稿缓存等，将不再保留。
按量计费账单数据	**需重新开通按量计费服务**	原账单数据不变，升级后需开通 TokenHub 按量计费服务，费用中心依据新订单出具账单。过程中模型推理服务不中断。
OpenAPI 接口	**部分兼容**	查看服务、服务组接口文档兼容新版本；开通服务、退订服务仅支持旧版本接口；服务监控仅支持割接前模型调用数据查询。

数据安全说明：整个割接过程采用加密传输，迁移完成后无法回滚，请提前导出所需数据。您的数据不会被用于任何未经授权的用途。

割接操作指引

阅读割接须知：完整阅读数据割接说明，了解数据范围与权利义务。

确认并同意：勾选“我已阅读并同意”，点击“同意并继续”。同意后即视为您授权迁移所列数据。

等待割接完成：系统自动执行迁移，请勿关闭页面或退出登录。

新版本验证：割接完成后，使用同一账号登录新版本，核验您的数据是否完整。

注意事项

割接期间业务不会中断：旧版本的APPKey和请求地址可正常使用。**割接后建议尽快更换为新平台的请求地址**，否则可能出现访问异常等情况。

割接不可中途取消：一旦点击"同意并继续"并执行，迁移过程无法中断，请确认无误后再操作。

割接期间请勿多端操作：为避免数据冲突，割接过程中请勿在其他设备登录同一账号。

一次性操作：割接为一次性迁移，重复触发无额外影响，但割接后不建议在旧版本产生新数据。

同意即生效：您点击同意后，即视为您授权平台按本说明迁移相关数据。

割接成功后不可回退：数据割接完成后**不可逆、无法回滚**。此外，TokenHub 按量计费服务为模型调用依赖基础服务，若退订该服务订单，将导致**全部模型调用请求被拒绝**，请谨慎操作。

部分客户历史上可能存在由子用户订购的模型按需服务订单。在您完成升级后，您需要**手工退订**这部分存量订单，以开启全量用户使用模型的权利。否则可能导致部分用户无法正常使用对应模型服务等异常情况。

支持与联系

如在割接过程中遇到任何问题，可通过以下方式获取帮助：

提交工单：https://www.ctyun.cn/console/smartservice/ticket/workorder/list

专属支持：企业客户可联系您的客户经理咨询

8.4.2 常见问题

割接触发问题

Q：怎么触发开始升级和割接?

下述操作任选其一即可

概览页平台升级和割接说明提示

下单操作（除编程Token Plan外）

创建/修改/删除服务组、重置APPKey

使用体验中心（包含模型体验和MCP体验）

升级权益与功能变动问题

Q：升级后可以获得哪些专属权益和新能力?

本次升级用户可享受多项权益与功能升级，具体如下：

免费额度福利：平台赠送14天50万tokens免费推理额度；用户原有未用完/未到期的免费额度可叠加使用，且整体额度有效期重置为14天

。

按量付费便捷开通：支持一键开通全部模型按量付费服务，可通过开关灵活控制各模型调用权限。

模型类型扩展：开放更多类型、更多版本的模型，覆盖更广的业务场景与性能需求。

精细化管控能力：新增子用户模型调用权限控制、服务组粒度的模型调用流量/额度上限设置能力。

Q：升级后模型闲时折扣有什么调整?

本次升级后，正式版GLM-5（正式版）、DeepSeek-V3.2（旗舰版）、Qwen3.5-397B-A17B（正式版）等模型的0:00-8:00**闲时优惠折扣取消。**

数据割接范围问题

Q：本次数据割接具体包含哪些数据？哪些数据不迁移？

本次数据割接迁移、不迁移及特殊调整数据明细如下：

全额迁移数据：账户基本信息（登录方式保持不变）、所有业务数据（用户创建的订单、任务、APP Key等，迁移后可正常查看、编辑）。

不迁移数据：割接前的模型调用监控数据（可在新版本控制台「历史监控」入口查询）、已过期/失效的临时数据（如草稿缓存，不再保留）。

特殊调整数据：按量计费订单数据，原订单数据保留不变，但升级后需重新开通TokenHub按量计费服务，后续费用中心话单将依据新订单出具。

Q：OpenAPI接口兼容性如何?

接口为**部分兼容**状态：开通服务、查看服务、退订服务、服务组相关接口可正常兼容使用；服务监控接口仅支持查询割接前的历史数据。

Q：数据割接过程的数据安全如何保障?

本次割接全程采用加密传输方式迁移数据，平台严格遵守隐私合规要求，**用户数据不会被用于任何未经授权的用途。**同时割接操作不可逆、无法回滚，建议用户提前导出留存重要原始数据。

Q：为什么在割接完成后，我的部分子用户使用不了这个模型了？

您需要在**天翼云费用中心-订单管理-退订管理**中检查是否存在该模型的旧订单。如发现存量旧订单，请在保证本次割接产生的**新订单不退订**的前提下，将存量旧订单退订，即可正常使用该模型服务。

联系我们

本文为您介绍如何与我们取得联系。

产品使用方面的问题如何反馈?

您可通过天翼云官网工单或者客服电话【400-810-9889】进行反馈。

相关协议

9.1 天翼云星辰TokenHub运营服务平台服务协议

天翼云为您提供星辰 TokenHub 运营服务平台服务协议，请您点击查看。

天翼云星辰 TokenHub 运营服务平台服务协议

9.2 天翼云星辰TokenHub运营服务平台用户信息处理规则

天翼云为您提供星辰 TokenHub 运营服务平台用户信息处理规则，请您点击查看。

天翼云星辰 TokenHub 运营服务平台用户信息处理规则

9.3 生成式人工智能服务备案信息参考

天翼云星辰 TokenHub 运营服务平台为用户提供相关开源大模型推理能力，本文提供第三方服务算法备案信息参考，和生成式人工智能服务备案信息参考。

服务算法备案信息

关于**服务算法备案信息**，可点击查询下载网信办每批次公告的**境内深度合成服务算法备案清单。**

或通过 https://beian.cac.gov.cn/#/searchResult 查询备案记录。



生成式人工智能服务备案信息

关于**生成式人工智能服务备案信息**，可点击查询下载网信办每批次发布的**生成式人工智能服务已备案信息。**

以下星辰 TokenHub 运营服务平台所提供的部分开源大模型**生成式人工智能服务备案信息**，本列表根据平台模型情况定期更新，不再另行公告。

属地	模型名称	备案单位	备案号	备案时间
北京市	求索对话 DeepSeek Chat	北京深度求索人工智能基础技术研究有限公司	Beijing-DeepseekChat-202404280016	2024/5/13
浙江省	通义千问大模型	阿里巴巴达摩院（杭州）科技有限公司	ZheJiang-TongYiQianWen-20230901	2023/9/12

9.4 人工智能大模型备案指南

本文为您详细介绍如何进行人工智能大模型备案。

一、备案介绍

大模型备案是指利用生成式人工智能技术，向中国境内公众提供具有舆论属性或社会动员能力的生成文本、图片、音频、视频等内容服务的模型，依据《生成式人工智能服务管理暂行办法》等法律法规履行的备案手续。其核心特征在于“主动监管”，要求服务提供者对模型的安全性、数据合法性和内容可控性进行全面评估。

政策链接：https://www.cac.gov.cn/2023-07/13/c_1690898327029107.htm

国家互联网信息办公室

中华人民共和国国家发展和改革委员会

中华人民共和国教育部

中华人民共和国科学技术部

中华人民共和国工业和信息化部

中华人民共和国公安部

国家广播电视总局

令

第15号

《生成式人工智能服务管理暂行办法》已经2023年5月23日国家互联网信息办公室2023年第12次室务会会议审议通过，并经国家发展和改革委员会、教育部、科学技术部、工业和信息化部、公安部、国家广播电视总局同意，现予公布，自2023年8月15日起施行。

国家互联网信息办公室主任 庄荣文

国家发展和改革委员会主任 郑栅洁

教育部部长 怀进鹏

科学技术部部长 王志刚

生成式人工智能服务管理暂行办法

第一章 总　则

第一条 为了促进生成式人工智能健康发展和规范应用，维护国家安全和社会公共利益，保护公民、法人和其他组织的合法权益，根据《中华人民共和国网络安全法》、《中华人民共和国数据安全法》、《中华人民共和国个人信息保护法》、《中华人民共和国科学技术进步法》等法律、行政法规，制定本办法。

第二条 利用生成式人工智能技术向中华人民共和国境内公众提供生成文本、图片、音频、视频等内容的服务（以下称生成式人工智能服务），适用本办法。

国家对利用生成式人工智能服务从事新闻出版、影视制作、文艺创作等活动另有规定的，从其规定。

行业组织、企业、教育和科研机构、公共文化机构、有关专业机构等研发、应用生成式人工智能技术，未向境内公众提供生成式人工智能服务的，不适用本办法的规定。

第三条 国家坚持发展和安全并重、促进创新和依法治理相结合的原则，采取有效措施鼓励生成式人工智能创新发展，对生成式人工智能服务实行包容审慎和分类分级监管。

第四条 提供和使用生成式人工智能服务，应当遵守法律、行政法规，尊重社会公德和伦理道德，遵守以下规定：

（一）坚持社会主义核心价值观，不得生成煽动颠覆国家政权、推翻社会主义制度，危害国家安全和利益、损害国家形象，煽动分裂国家、破坏国家统一和社会稳定，宣扬恐怖主义、极端主义，宣扬民族仇恨、民族歧视，暴力、淫秽色情，以及虚假有害信息等法律、行政法规禁止的内容；

二、大模型备案流程与材料

第一步：向属地网信办申请备案

申请单位需向属地网信办提交备案申请，领取并填写《生成式人工智能（大语言模型）备案信息采集表》。

第二步：材料准备与安全自评估

《生成式人工智能上线备案表》：正式提交模型上线备案的申请文件，明确模型基本信息、备案主体及合规承诺。

《安全评估报告》：全面评估模型在数据安全、内容合规、算法偏见及系统防护等方面的风险控制能力。

《模型服务协议》：界定用户与开发者之间的权利义务，明确数据使用范围、服务条款及法律责任划分。

《语料标注规则》：规范训练数据的标注流程与标准，确保数据质量并避免伦理偏见。

《拦截关键词列表》：列出需过滤的违法、敏感及虚假信息关键词，作为内容安全的基础拦截规则。

《评估测试题》：通过设计合规性、边界及性能测试场景，验证模型的实际表现与风险抵御能力。

其他备案必需材料。

第三步：提交属地网信办初审测试

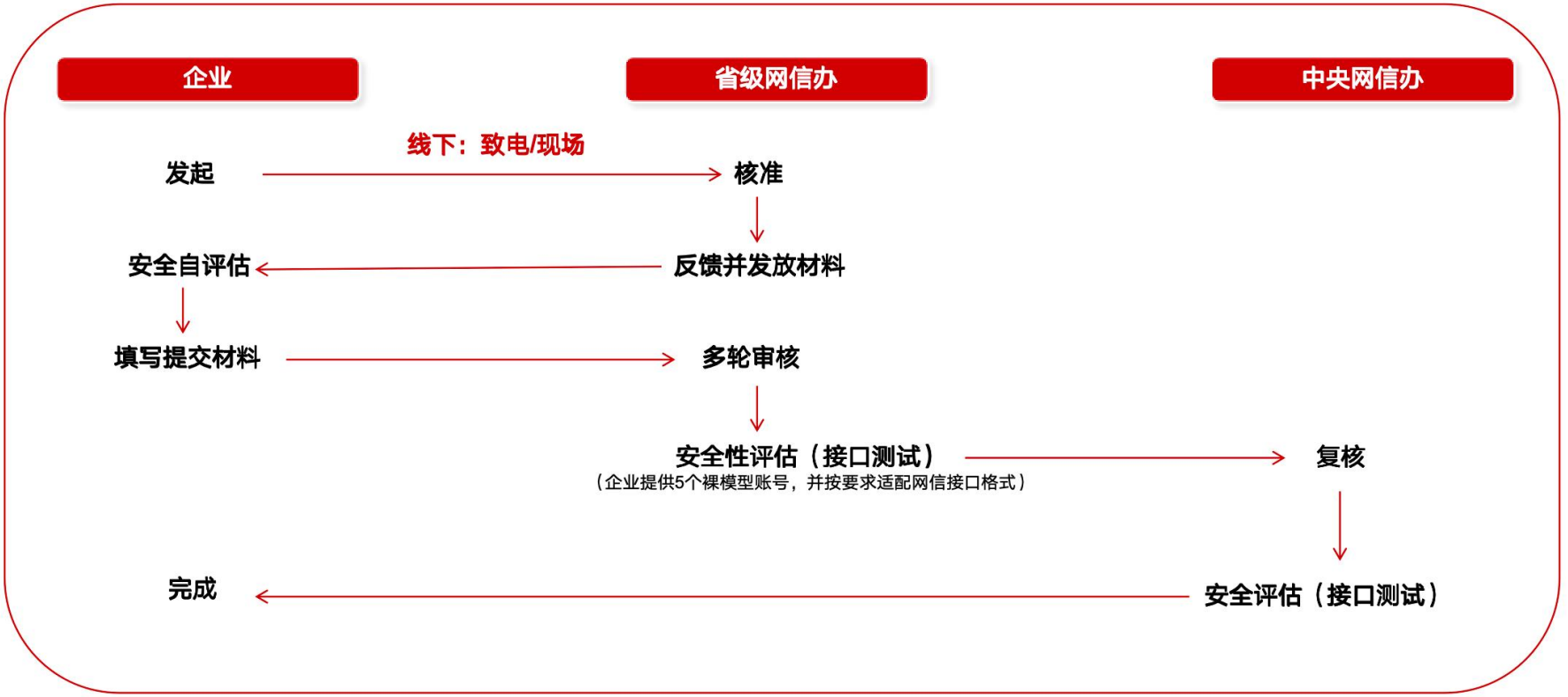
申请单位将材料附件及 5 个实名认证测试账号提交给属地网信办进行初审。审核过程中，属地网信办会对材料的完整性、准确性进行核查，并对大模型进行技术测试。申请单位应配合属地网信办完成多轮审核，根据审核意见修改完善材料。

第四步：等待中央网信办复审

属地网信办将初审通过的材料上报给中央网信办，由中央网信办进行材料复审和大模型技术评审。

第五步：获取备案编号

中央网信办复审通过后，申请单位将获得备案号和官方网站公示，若未通过，则需要根据反馈进行整改，并重新提交审核。



三、注意事项

在大模型备案过程中，存在一些重点和难点问题，需要申请单位和开发者特别注意：

语料标注的规范性：语料标注是构建和训练大模型的关键环节。为了确保语料的合法性和标注的规范性，企业需要明确标注团队的资质、质量控制机制以及数据来源的合法性等。同时，还需要对标注结果进行定期的评估和优化，以确保标注质量。

拦截关键词列表的全面性：拦截关键词列表是大模型内容安全过滤机制的重要组成部分。为了确保内容的安全性和合规性，企业需要准备一份全面且精确的拦截关键词列表。这份列表应覆盖可能涉及的风险领域，如政治敏感、色情、暴力等，并定期进行更新和优化。

安全评估报告的编制：安全评估报告是大模型备案过程中的关键材料之一。这份报告需要由具备相应资质的专业机构或团队负责编制，全面评估大模型在各个关键环节的安全性。报告内容应包括数据隐私保护、算法偏见识别与纠正、有害信息过滤机制等方面。

模型服务协议的制定：模型服务协议是确保用户权益和企业利益得到充分保护的关键法律文件。企业需要制定一份详细且公平的模型服务协议，明确服务范围、双方权利义务、数据使用与保护等内容。这份协议需要经过法务团队的审阅和确认，以确保其合法合规且公平合理。

四、大模型备案补贴

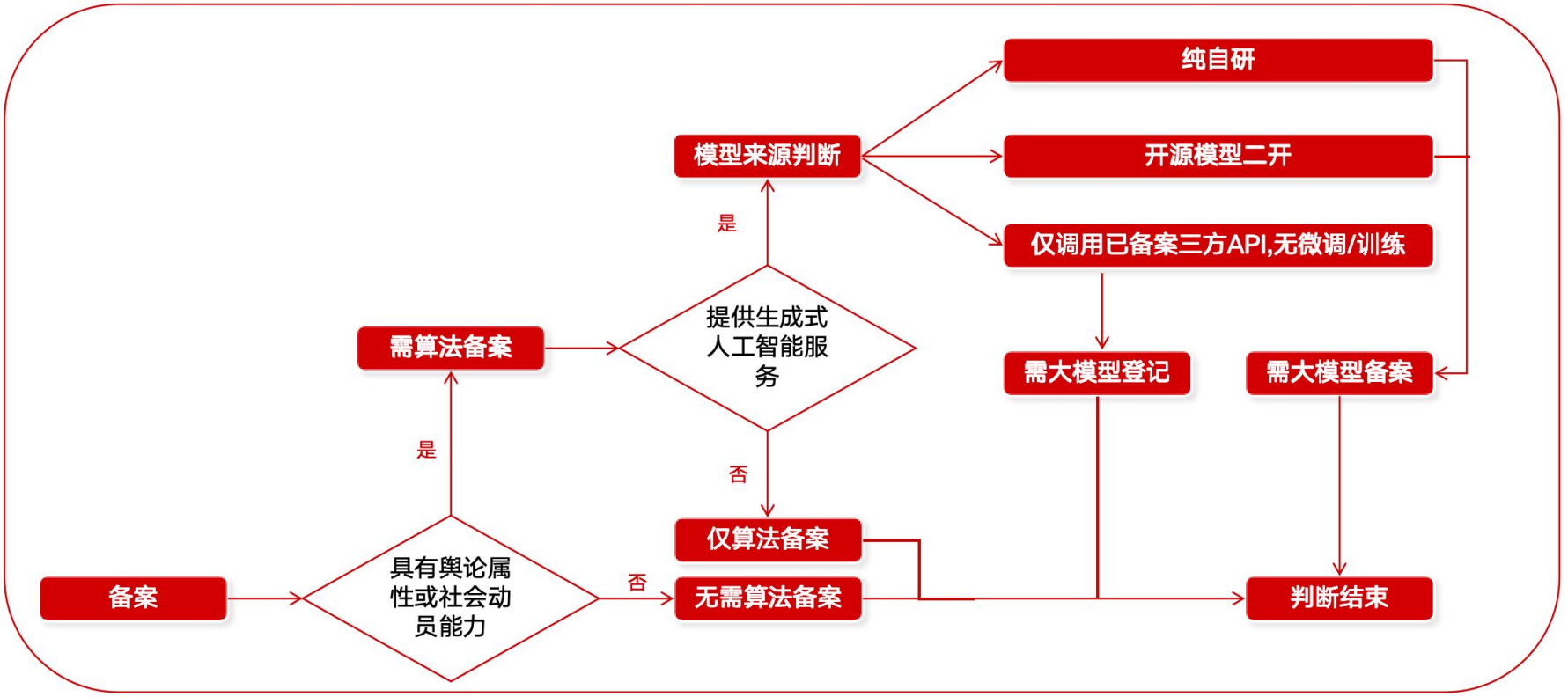
为推动人工智能（AI）行业规范化发展，鼓励企业主动落实合规备案要求，全国多个重点 AI 产业集聚区已陆续出台针对性备案补贴政策。相关政策需要查看当地官方网站公布的 AI 大模型备案政策。

五、常见问题

1. 大模型备案和登记有什么区别?

大模型登记适用于通过 API 接口或其他方式直接调用已备案大模型能力，且面向中国境内公众提供具有舆论属性或社会动员能力的生成式人工智能服务的企业。登记流程相对简化，企业需向属地网信部门提交登记申请及相关材料，经审核通过后授予上线编号并公示。

2. 算法备案、大模型备案、大模型登记三者有何联系?



3. 为什么要做大模型备案?

大模型备案侧重于事前风险管控，确保 AI 大模型在正式投入市场前已满足国家安全标准及合规要求，有效预防潜在危害，如误导用户或传播非法信息，从而构建安全可靠的人工智能生态。

4. 哪些企业要做大模型备案?

向中国境内公众提供具有舆论属性或社会动员能力的生成文本、图片、音频、视频等内容服务的模型，都需要做大模型备案。