ModelGate 用户使用手册

ModelGate 是团队自建的统一大模型网关：一个 API Key 即可访问所有已接入的模型（GLM、DeepSeek、Kimi、Qwen 等），支持 OpenAI / Anthropic 双协议、流式输出、思考档位与多模态。本手册面向**使用者**（拿到 Key 的同学），不涉及管理端操作。

> **网关地址（记这两个就够了）**
>
> | 用途 | 地址 |
> | --- | --- |
> | 网关首页 / 用户门户 | `https://leturx.cc/modelgate` |
> | OpenAI 兼容 Base URL | `https://leturx.cc/modelgate/v1` |

---

1. 拿到你的 API Key

方式	说明
管理员分配	管理员在后台创建 Key 后把 `sk-...` 发给你
自助注册（若管理员开启了邮箱验证码登录）	门户登录页填邮箱 → 收验证码 → 系统为你分配一个 Key

Key 使用须知

- 形如 `sk-` 开头的长字符串，等同于密码，**不要提交进代码仓库、不要发到群里。**
- 怀疑泄露：登录用户门户执行「重置 API 密钥」（旧 Key 立即失效），或联系管理员禁用。
- Key 可能带有并发数、每日额度、可用模型范围、IP 白名单等限制（由管理员配置），超限时调用会返回明确错误信息。

---

2. 一分钟接入（任何 OpenAI 兼容客户端）

填三个字段即可：

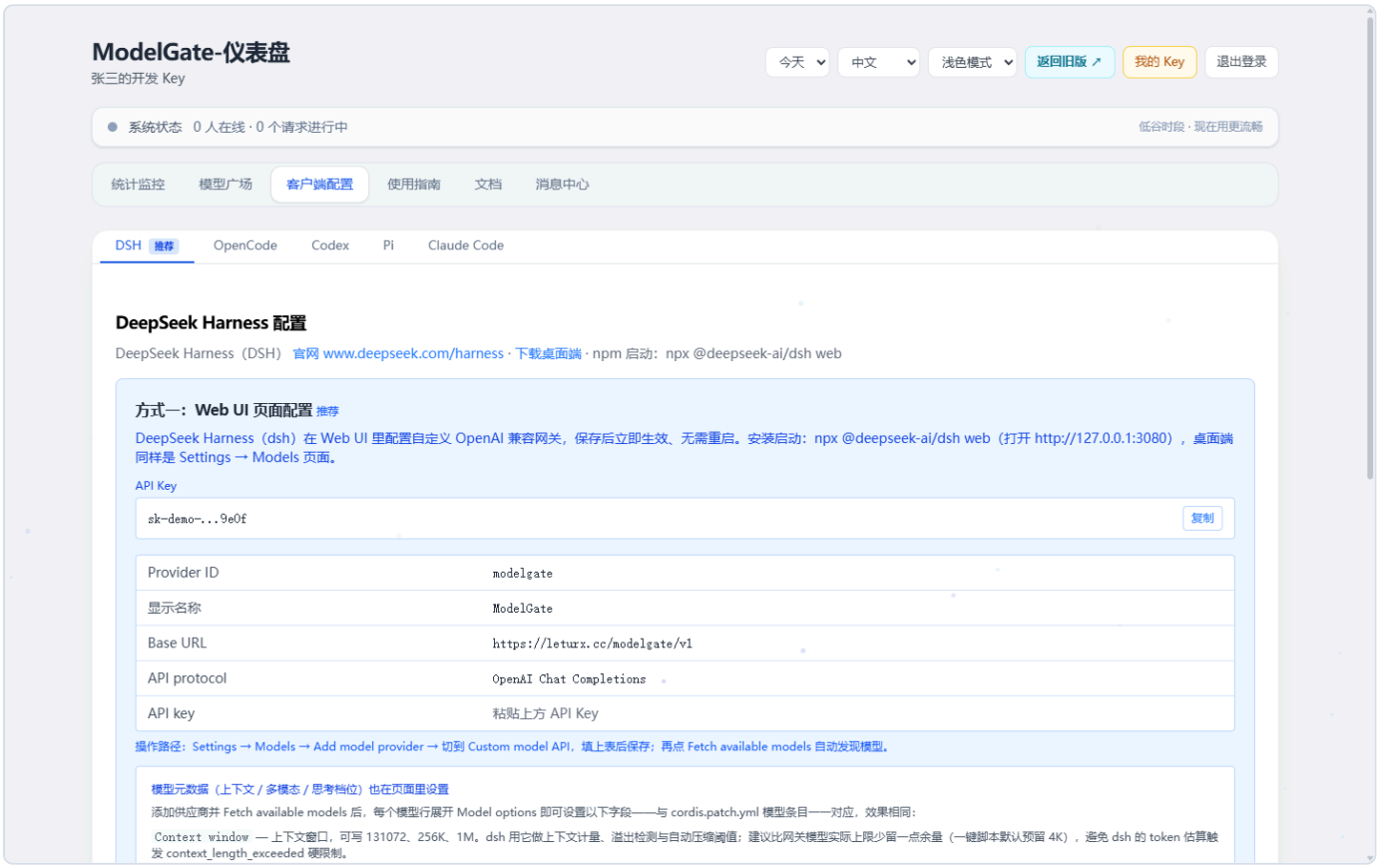
字段	值
Base URL	`https://leturx.cc/modelgate/v1`
API Key	你的 `sk-...`
模型名	从模型列表选（见第 5 节）

curl 验证（能返回 JSON 就通了）：

```
curl https://leturx.cc/modelgate/v1/chat/completions \
  -H "Authorization: Bearer sk-你的KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "model": "glm-5.3",
    "messages": [{"role": "user", "content": "你好"}],
    "stream": true
  }'
```

认证方式两种都支持：`Authorization: Bearer sk-...`（推荐）或 `x-api-key: sk-...`。个别工具不填 Base URL、只填一个接口地址的，可用**带 Key 的路径形式**：`https://leturx.cc/modelgate/v1/sk-你的KEY/chat/completions`。

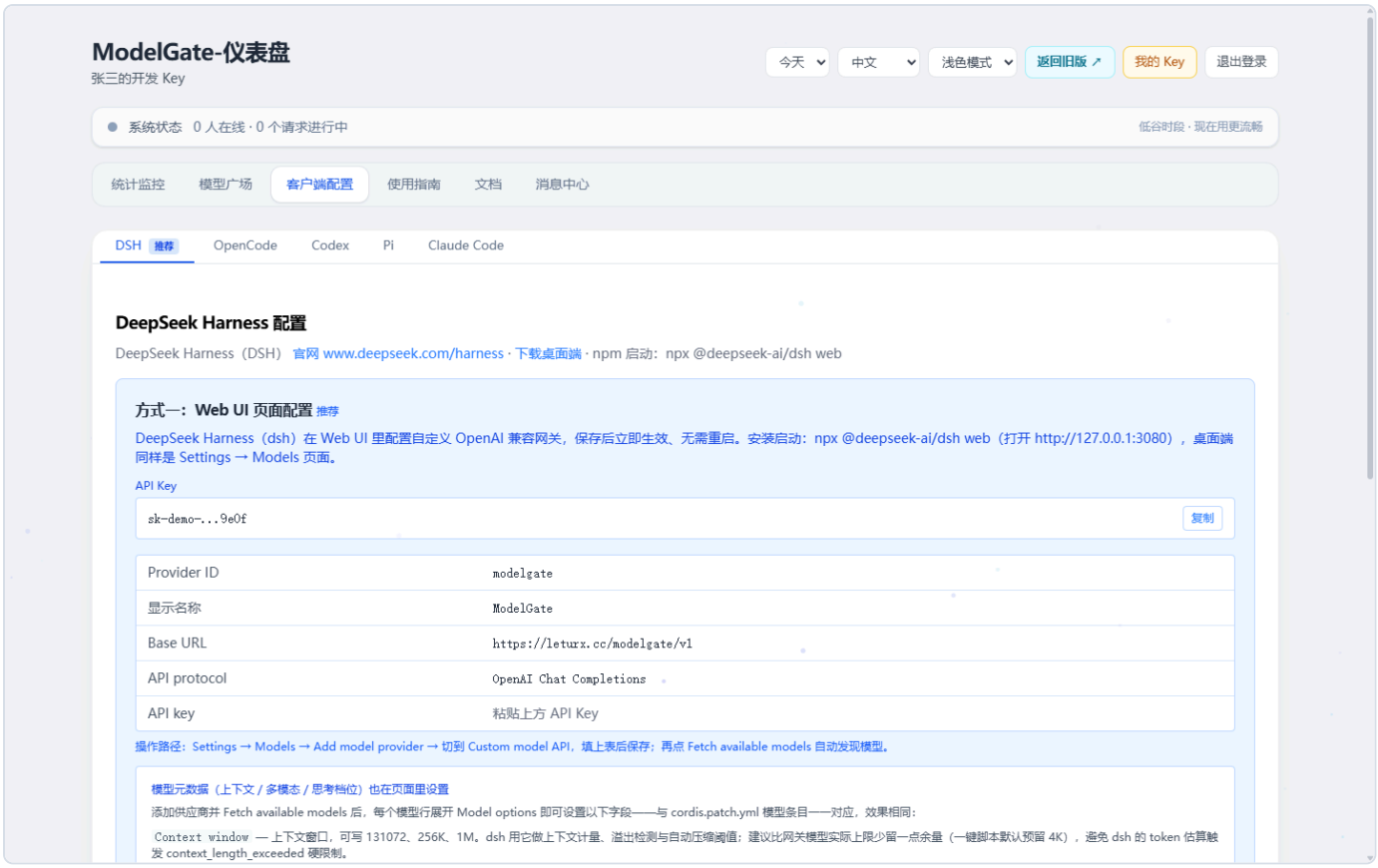
> 不想手填？打开用户门户的 **客户端配置** 标签页，选你的工具，页面会直接给出已填好 Key 和地址的表格与配置代码，复制即可。



---

3. 编程工具（AI Coding CLI）接入

最省事的方式：打开用户门户的「客户端配置」标签页，按工具分好了页签（DSH / OpenCode / Codex / Pi / Claude Code），每页都给出现成的一行命令 —— Key 已自动嵌入，复制粘贴回车即可。



每个工具页都有**方式一（一键脚本，推荐）** 和 **方式二（手动编辑配置文件）** 两种做法，脚本都会**先备份**再合并，不会覆盖你原有的其它配置。

> **参数名有个坑，务必分清**：脚本类端点用 `?key=`，文字说明类端点用 `?api_key=`。
>
> | 端点 | 参数 |
> | --- | --- |
> | `*.ps1` / `*.sh`（脚本） | `?key=sk-...` |
> | `*.md`（分步说明） | `?api_key=sk-...` |
>
> 写错的后果很隐蔽：用 `?api_key=` 请求 `.ps1` 仍会返回 **200**，但脚本里的 Key 是**空的**，跑完配置不会生效。从门户页面复制现成命令即可避开。

3.1 DeepSeek Harness（DSH）

方式 A • 插件自动同步（推荐）：安装 `dsh-modelgate` 插件后，模型目录（上下文窗口、输出上限、思考档位、多模态标记）每小时自动与网关保持一致，网关新增模型无需改配置。

方式 B • 一键脚本：

```
# Windows PowerShell
irm 'https://leturx.cc/modelgate/dsh/setup.ps1?key=sk-你的KEY' | iex
```

```
# macOS / Linux
curl -fsSL 'https://leturx.cc/modelgate/dsh/setup.sh?key=sk-你的KEY' | bash
```

脚本会写入 profile 补丁并设置环境变量。其它可用的生成端点：

打开	得到
`/dsh/setup.md?api_key=…`	分步配置说明（文字）
`/dsh/cordis.patch.yml?api_key=…`	可直接粘贴的 provider 配置块

手工配置：环境变量 `MODELGATE_API_KEY` + 在 profile 的 `cordis.patch.yml` 的 `llm-pi-ai` 条目下加 provider（`displayName` / `apiKeyEnv: MODELGATE_API_KEY` / `api: openai-completions` / `baseURL: https://leturx.cc/modelgate/v1` / `models` 列表）。最省事的做法：直接用 `/dsh/cordis.patch.yml` 渲染结果对照填写。

3.2 Claude Code

方式一 · 一键脚本（推荐）。脚本会备份现有配置、合并网关环境变量（`ANTHROPIC_BASE_URL` / `ANTHROPIC_AUTH_TOKEN` / `ANTHROPIC_MODEL`，其余设置全部保留），然后写入模型默认值：

```
irm 'https://leturx.cc/modelgate/claude/setup.ps1?key=sk-你的KEY' | iex
```

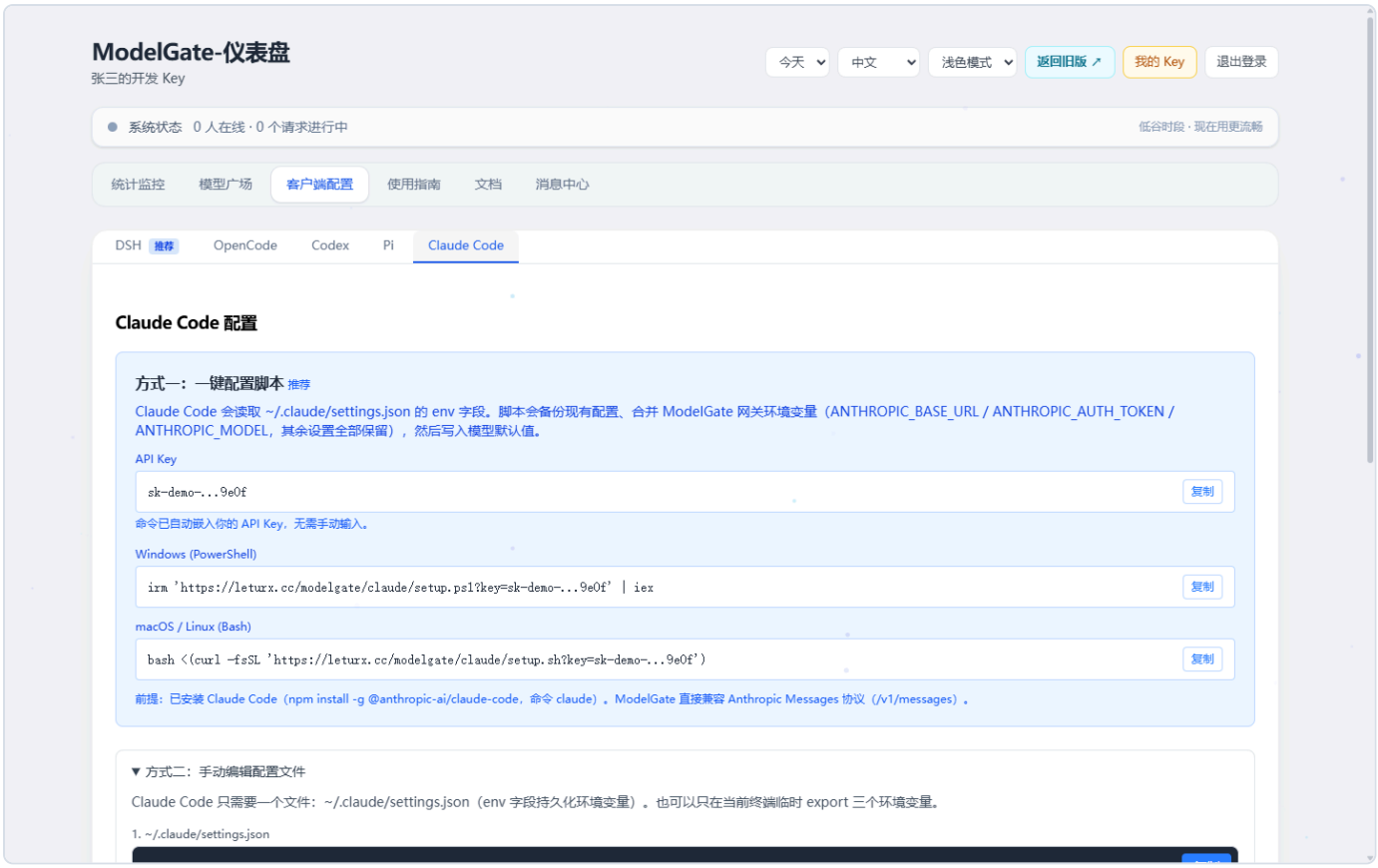
```
bash <(curl -fsSL 'https://leturx.cc/modelgate/claude/setup.sh?key=sk-你的KEY')
```

前提：已安装 Claude Code（`npm install -g @anthropic-ai/claude-code`，命令 `claude`）。

方式二 · 手动编辑 `~/.claude/settings.json`（Windows：`%USERPROFILE%\.claude\settings.json`）。Claude Code 读该文件的 `env` 字段持久化环境变量：

```
{
  "env": {
    "ANTHROPIC_BASE_URL": "https://leturx.cc/modelgate",
    "ANTHROPIC_AUTH_TOKEN": "sk-你的KEY",
    "ANTHROPIC_MODEL": "glm-5.3"
  }
}
```

注意 **Base URL 不带** `/v1`（SDK 会自己拼 `/v1/messages`）。ModelGate 原生支持 Anthropic Messages 协议，**无需任何代理。** 也可以在门户下载现成的 `/claude/settings.json?api_key=…`，或看图文说明 `/claude/setup.md?api_key=…`。



3.3 Codex CLI

方式一 • 一键脚本（推荐）。Codex CLI、ChatGPT 桌面端、VS Code Codex 插件**共用同一份配置**，配置一次全部生效。脚本会备份现有 `config.toml`、写入模型目录 `models.json`、合并 ModelGate 配置（MCP 服务器等原有配置全部保留）：

```
irm 'https://leturx.cc/modelgate/codex/setup.ps1?key=sk-你的KEY' | iex
```

```
bash <(curl -fsSL 'https://leturx.cc/modelgate/codex/setup.sh?key=sk-你的KEY')
```

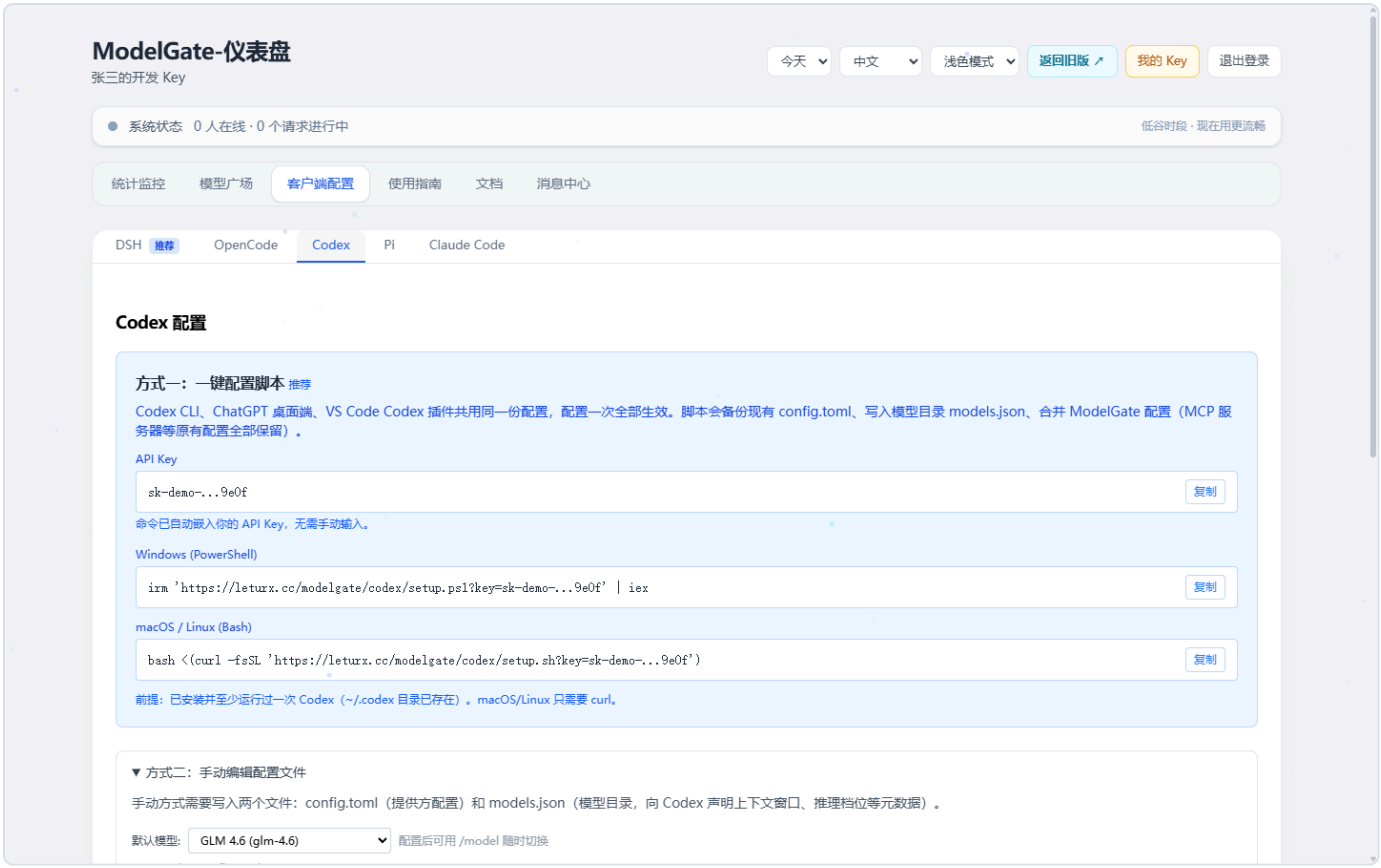
前提：已安装并**至少运行过一次** Codex（确保 `~/.codex` 目录已存在）。

方式二 • 手动编辑 `~/.codex/config.toml`：

```
model = "glm-5.3"
model_provider = "modelgate"
preferred_auth_method = "apikey"
forced_login_method = "api"
model_reasoning_effort = "high"
model_catalog_json = "~/.codex/models.json"

[model_providers.modelgate]
name = "ModelGate"
base_url = "https://leturx.cc/modelgate/v1"
wire_api = "responses"
experimental_bearer_token = "sk-你的KEY"
```

同时需要模型目录 `~/.codex/models.json`（向 Codex 声明每个模型的上下文窗口与推理档位），从门户下载：`/codex/models.json?api_key=…`。 配置完成后可用 `/model` 随时切换模型。



3.4 OpenCode

OpenCode 只需**一行命令**：脚本会备份现有配置 → 把 ModelGate 作为 provider 合并进去 → 打印可用模型列表。配置文件位于 `~/.config/opencode/opencode.jsonc`（项目内也可放 `opencode.jsonc`）。

在用户门户「客户端配置 → OpenCode」页签里，命令是**现成生成好的**（已嵌入你的 Key），复制粘贴即可：

Windows（PowerShell）：

```
irm 'https://leturx.cc/modelgate/opencode/setup.ps1?key=sk-你的KEY' | iex
```

macOS / Linux（Bash）：

```
curl -fsSL 'https://leturx.cc/modelgate/opencode/setup.sh?key=sk-你的KEY' | bash
```

执行完**重启 OpenCode**（配置只在启动时读一次），运行 `opencode models` 验证：带 `modelgate/` 前缀的条目即由本站提供。看到 `ModelGate provider configured` 字样即表示配置成功。

> ⚠️ 这三个端点的参数名不一样，抄错会拿到不含 Key 的脚本：`setup.ps1` / `setup.sh` 用 `?key=`；`setup.md`（分步文字说明）用 `?api_key=`。
>
> 没装 OpenCode？先去 opencode.ai 按官方指引安装，再回来运行上面的命令。



3.5 其它 Agent / SDK（Python / Node）

任何支持自定义 Base URL 的 OpenAI SDK 都能直接用：

```
from openai import OpenAI

client = OpenAI(
    base_url="https://leturx.cc/modelgate/v1",
    api_key="sk-你的KEY",
)
resp = client.chat.completions.create(
    model="glm-5.3",
    messages=[{"role": "user", "content": "你好"}],
)
print(resp.choices[0].message.content)
```

Anthropic SDK 同理：`base_url="https://leturx.cc/modelgate"`，`api_key="sk-你的KEY"`。

---

4. 接口一览

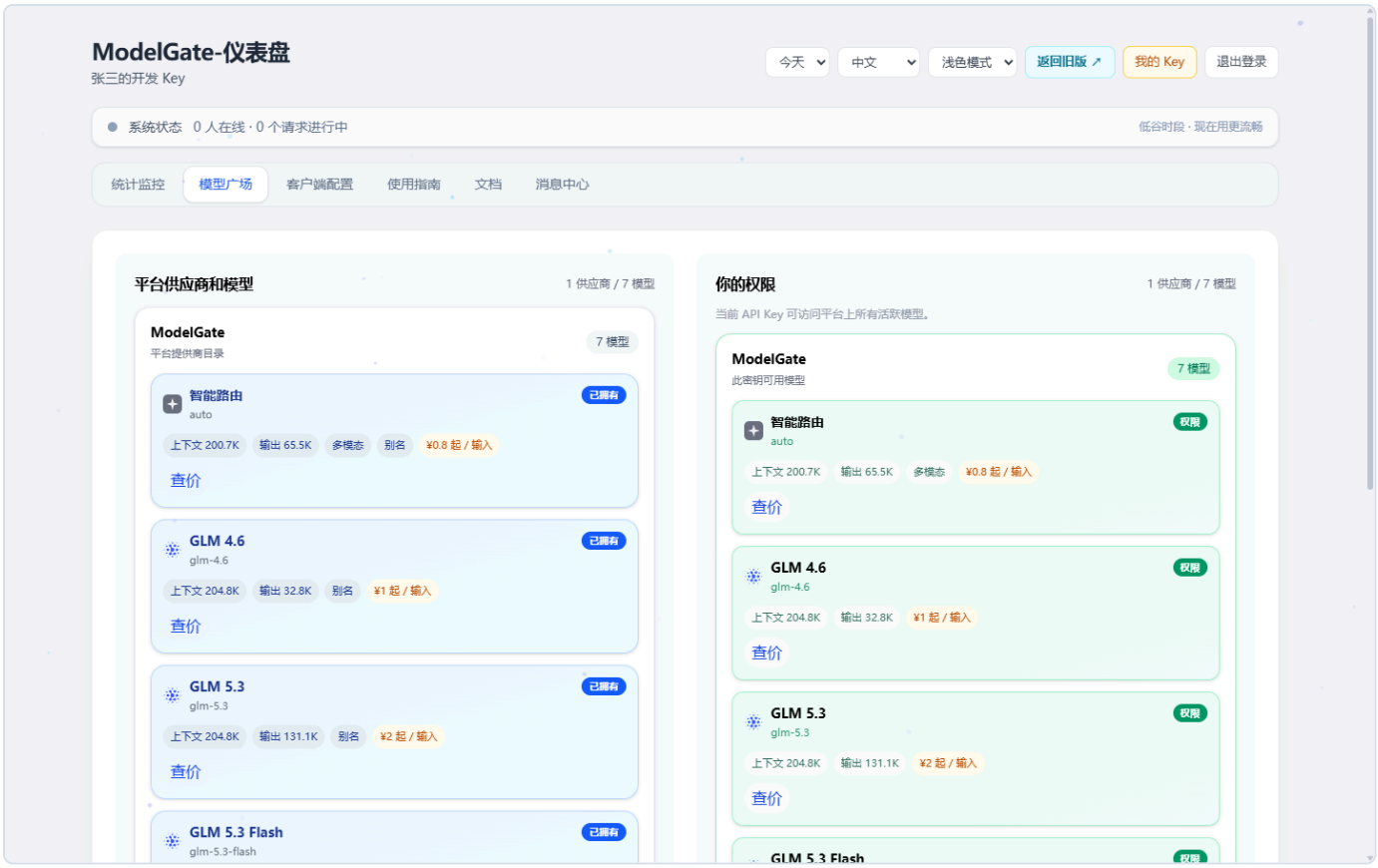
端点	方法	说明
`/v1/chat/completions`	POST	OpenAI Chat（支持 `stream: true`）
`/v1/models`	GET	当前 Key 可用的模型列表
`/v1/embeddings`	POST	向量嵌入
`/v1/responses`	POST/GET/DELETE	OpenAI Responses API（Codex 同款）
`/v1/messages`（或 `/anthropic/v1/messages`）	POST	Anthropic Messages 协议（Claude 系客户端）
`/v1/messages/count_tokens`	POST	Anthropic token 计数
`/v1/videos`	POST	视频生成任务（创建 / 查询 / 取结果，视开通情况）

以上均支持 Bearer / x-api-key 认证；chat/completions、models、videos 另有 `/v1/{你的KEY}/…` 路径形式。

---

5. 模型与能力

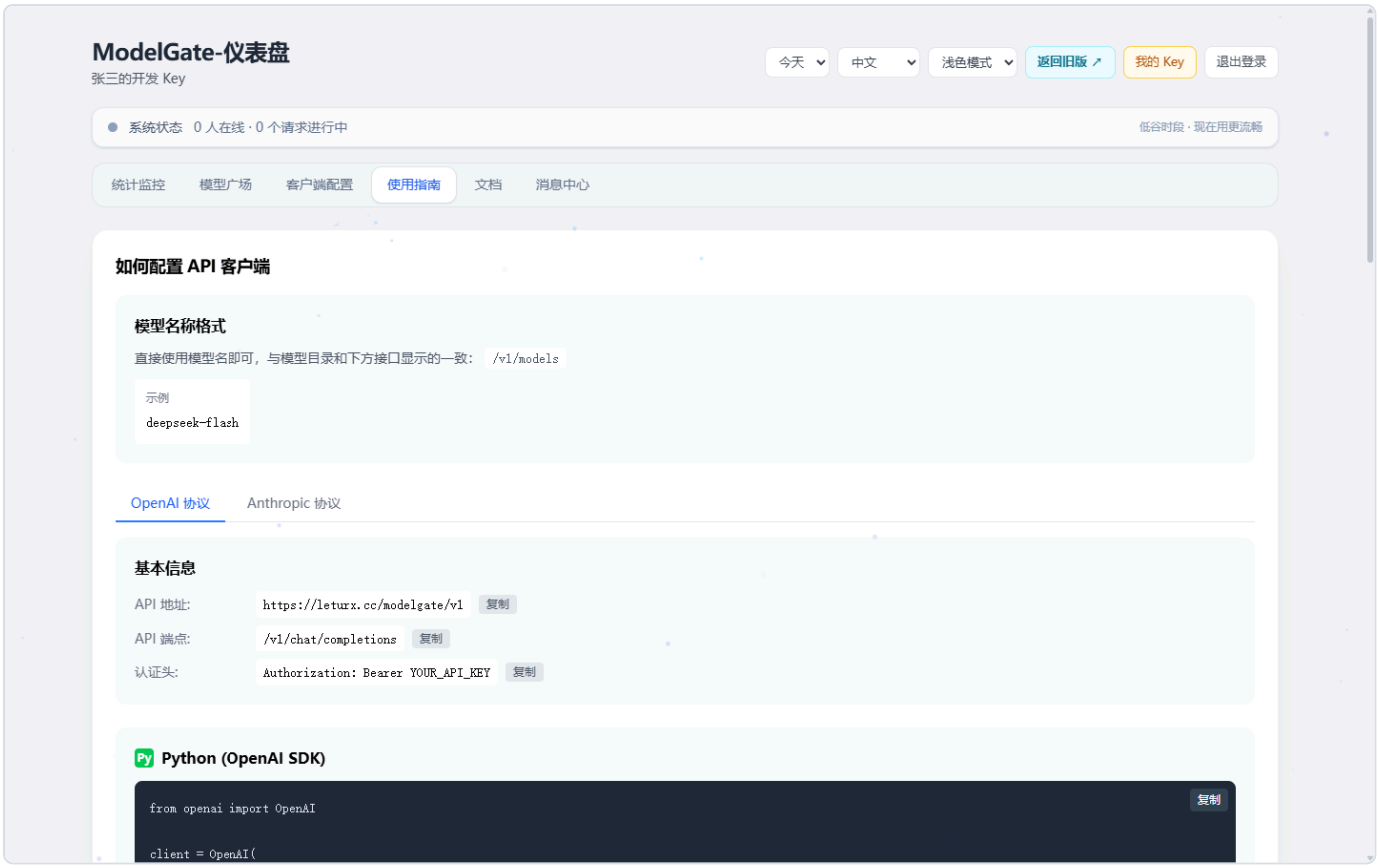
- **查模型**：`GET /v1/models`，或用户门户的「模型广场」页（含实时可用状态、单价、上下文窗口、思考档位、是否多模态）。
- **模型名**直接填网关模型 ID（如 `glm-5.3`、`glm-5.3-flash`）；`auto` 为智能路由虚拟模型（按任务复杂度自动选模型，省心且更省钱）。
- **思考档位**：支持思考的模型可传 `reasoning_effort: low | medium | high`（Anthropic 协议为 `thinking` 参数）；标了 `off` 的模型也可强制关思考。
- **多模态**：支持图片的模型在 messages 里直接用 image_url / base64 图片即可。
- 各模型实际上下文窗口以用户门户「模型广场」页为准——客户端里配置的窗口数不对会导致长文被截断，建议用各向导端点生成的配置（自动带正确元数据）。



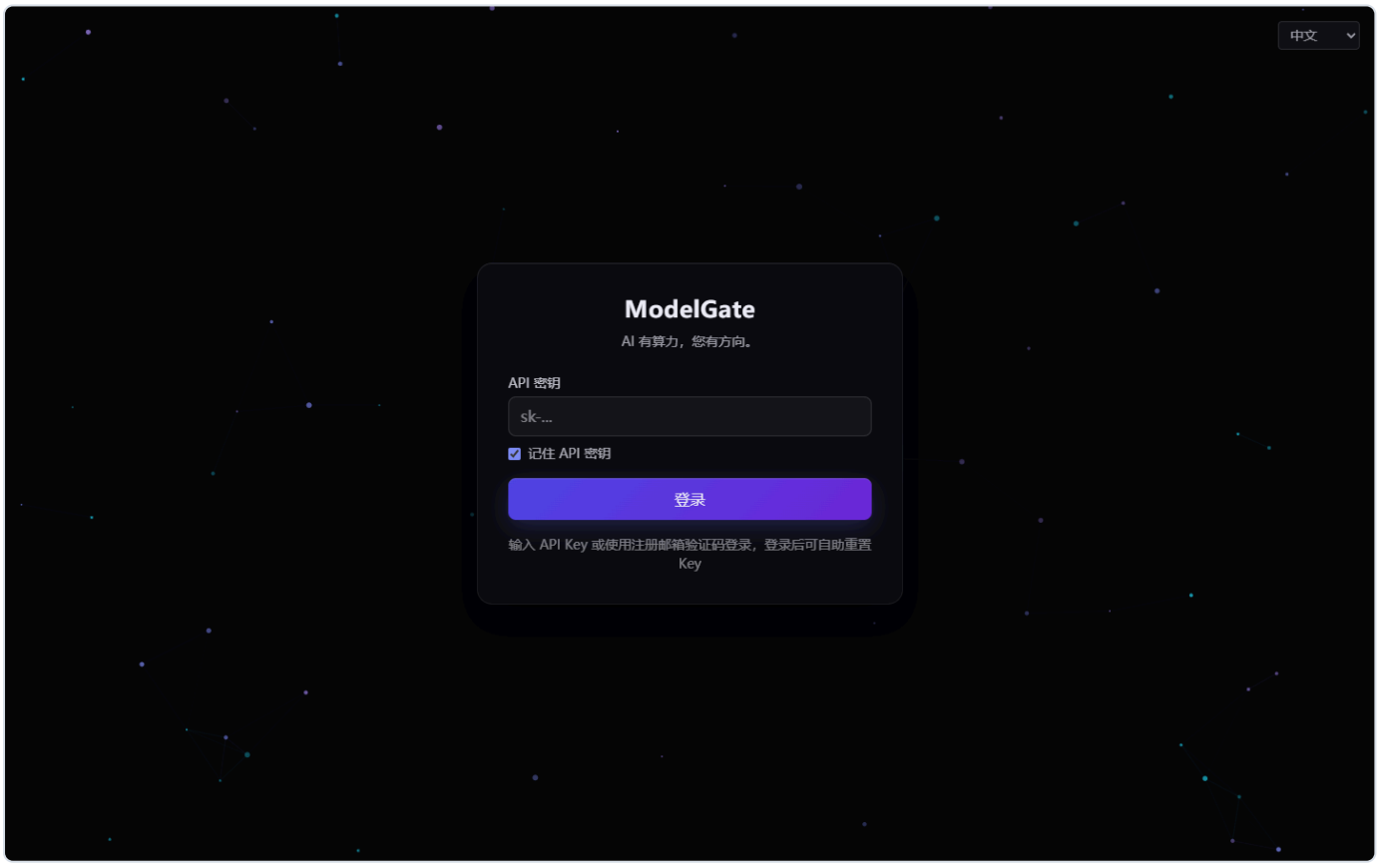
6. 用户门户（用量 / 费用 / 明细）

登录 `https://leturx.cc/modelgate/user/login` 后，顶部有以下标签页：

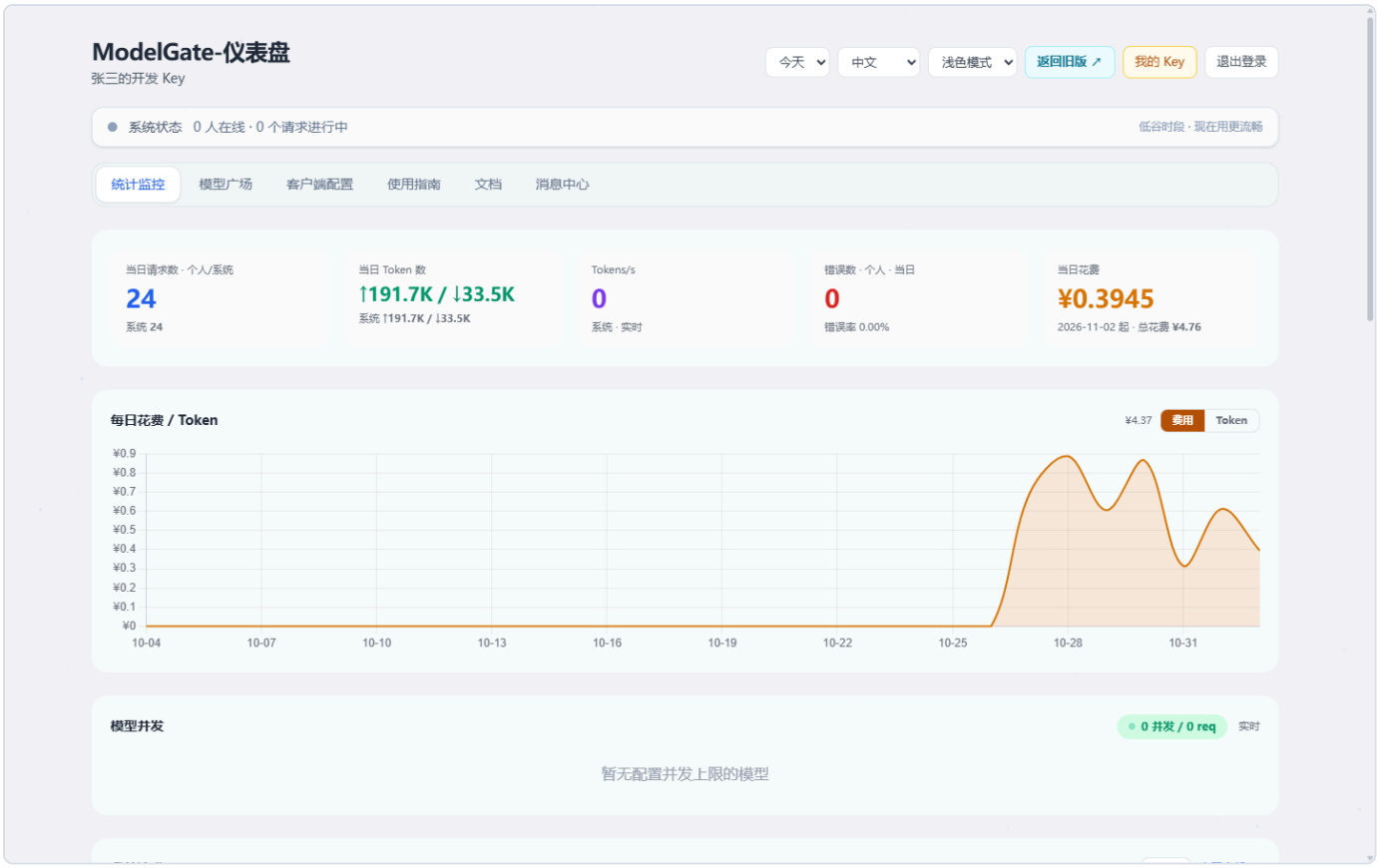
标签页	内容
统计监控	当日 / 本周 / 本月的请求数、Token 数、错误数、花费；每日花费趋势图、模型并发、最近请求明细
模型广场	平台全部可用模型：上下文窗口、输出上限、单价、多模态 / 思考标记、是否已对你开放
客户端配置	按工具（DSH / OpenCode / Codex / Pi / Claude Code）给出可直接复制的配置
使用指南	模型名格式、OpenAI / Anthropic 两种协议的调用示例
文档	管理员发布的共享文章
消息中心	系统公告与通知



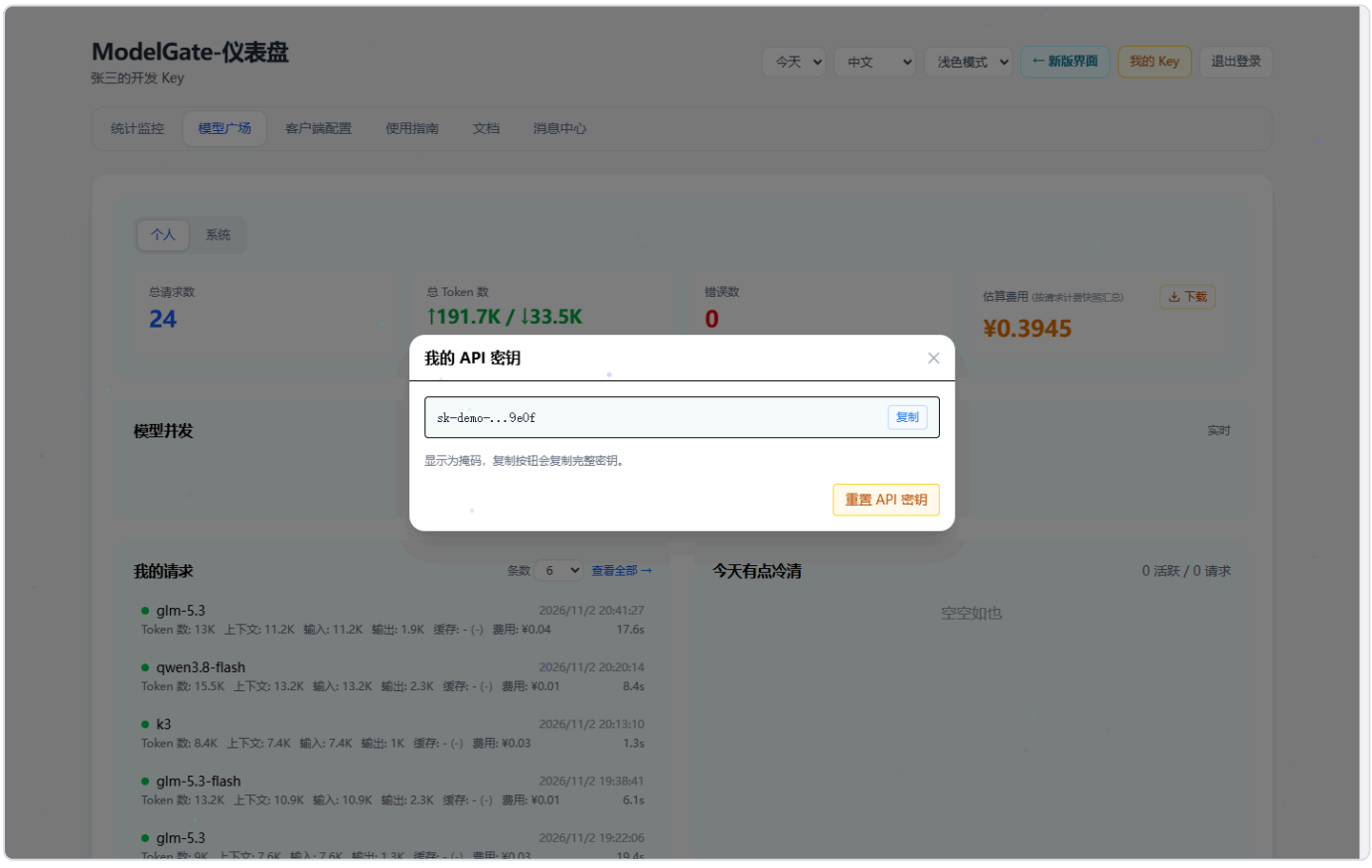
登录方式：在登录页填入你的 `sk-...`（没有账号密码），勾选「记住 API 密钥」可免重复输入。



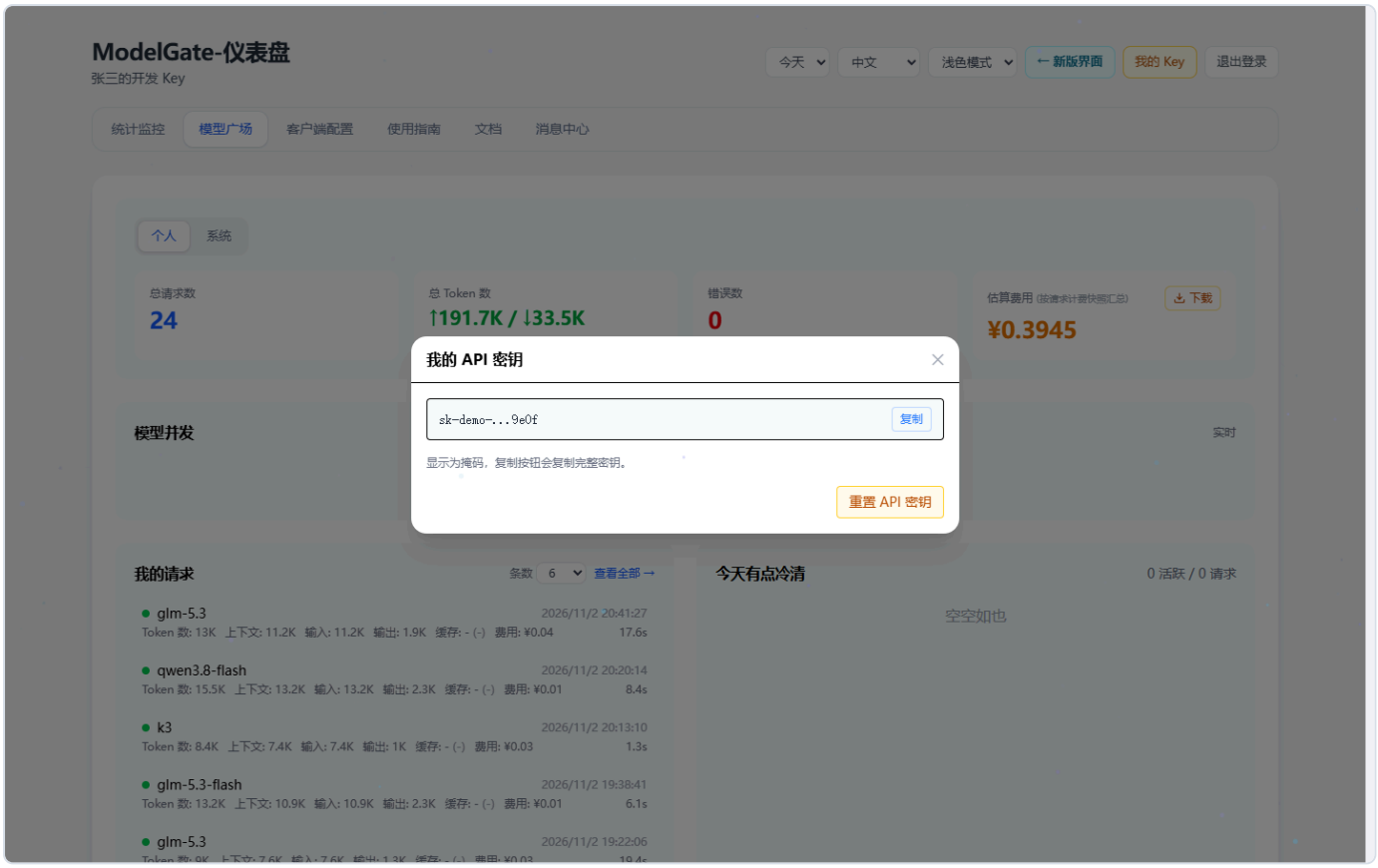
登录后进入仪表盘，第一屏就是你的用量与花费：



点右上角「我的 Key」可查看掩码后的 Key 并复制完整值，或执行「重置 API 密钥」：



「返回旧版」可切到旧版统计视图（含请求趋势、模型使用排行、请求明细）：



> 想让别人也能看实时系统状态？`/api/public/stats` 是免登录的公开状态接口。

---

7. 常见问题

现象	原因与处理
`401 Invalid API Key`	Key 填错 / 已被重新生成 / 已禁用。门户里重新查看 Key，或联系管理员
`400 No models available`	Key 的模型范围为空或全部被禁用，联系管理员开通
`429` 并发 / 额度超限	Key 有并发数或每日额度限制；到门户看用量，需要扩容找管理员
部分模型时好时坏	网关繁忙时会对重模型降级（保轻模型可用）；稍后重试，或换 flash 档模型 / 联系管理员给你的 Key 开繁忙豁免
长对话中途报错 / 内容被截	检查客户端配置的上下文窗口是否偏大（应与门户「系统模型」页一致）；用向导生成的配置可避免
流式输出卡住后断开	上游超时或网络抖动，客户端重试即可；明细可在「我的请求」里查状态
想换新 Key	门户「我的 Key」→「重置 API 密钥」——旧 Key 立即失效并自动登出，**先改好配置再重置**，重置后立刻更新所有客户端

---

8. 安全须知

1. Key 等同密码：不进 git、不进截图、不发群。
2. 本机长期使用建议放环境变量（如 `MODELGATE_API_KEY`），客户端用 `apiKeyEnv` 引用而非明文。
3. 公共电脑不要勾「记住 API 密钥」；用完在门户重置 API 密钥。
4. 发现异常调用（门户请求明细里出现不是你发起的请求）→ 立即重置 Key 并联系管理员。

问题反馈：联系 ModelGate 管理员；本手册的最新版本以用户门户「文档」区为准。