多模态 · 私有化 · Agentic RAG

MMKB 多模态知识库

用户使用手册 —— 从零部署到上传问答的完整图文指南

1. 这是什么

MMKB 是一套**本地私有化**的多模态知识库系统：上传 PDF、Office、图片、音频，系统自动解析、向量化并索引；用户用自然语言提问，系统结合工作区文档**边检索边推理**，给出有出处、可溯源、图文并茂的答案。系统内置多租户隔离，并提供 OpenAI 兼容 API。

三类账户角色：

角色	能做什么
平台管理员	管理所有租户、后台 worker；可见全部工作区（运维角色）
租户	管理自己的工作区、上传文档、创建普通用户；可搜索与问答
普通用户	登录后直达聊天界面，仅能问答本租户工作区内的资料

2. 在自己的服务器上部署（少量命令）

前置：一台 Linux 服务器，已安装 `git`、`conda`（Anaconda/Miniconda）、`Docker`、`Node.js`。

① 拉取代码与环境

```
# 克隆仓库
git clone <your-repo-url> mmkb && cd mmkb

# 创建并激活 conda 环境（Python 3.10 + 全部依赖）
conda env create -f environment.runtime.yml
conda activate mmkb

#  （可选）重新构建前端，产物已随仓库提供，通常无需此步
cd frontend && npm install && npm run build && cd ..
```

② 配置密钥 `.env`

系统会用到**三类相互独立的外部服务**，密钥各不相同。先 `cp .env.example .env`，再按下表填写。

用途	相关变量	怎么填

① **嵌入模型** 文档/查询向量化	`MULTIMODAL_EMBED_API_KEY`	**必填。**默认走火山引擎 Ark 的多模态嵌入，地址已在代码里预置为 Ark，所以**通常只需填这个 key**。

② **对话/RAG 模型**
生成回答
	`CHAT_COMPLETION_API_KEY`
	`CHAT_COMPLETION_BASE_URL`
	`CHAT_COMPLETION_MODEL`
		默认 base_url 同样预置为 Ark：用 Ark 时可**只填 key**（留空则自动复用①的 key）。**若要换别家**（DeepSeek / 通义千问 / OpenAI 等任意 OpenAI 兼容服务），必须同时改三项：`BASE_URL`=该服务地址、`API_KEY`=该服务 key、`MODEL`=其模型名。

③ **文档解析** PDF/Office/图片 → 文本	`PADDLEX_API_TOKEN`	这是 **PaddleX 文档解析服务**的令牌，**不是 LLM 的 key**，是另一个独立服务的密钥。不配则无法解析上传的文件（纯对话/已入库问答不受影响）。

> **为什么看起来"只填 key、不填 URL"？**任何模型服务本质都需要「地址(base_url)+密钥(key)+模型名」三要素。默认地址已预置为火山引擎 Ark，所以用 Ark 时只补 key 即可；一旦换厂商，就必须连同 `BASE_URL` 一起改。也就是说——**并非只能用 Ark 的 key**，只是 Ark 是开箱即用的默认值。

> **思考型模型：**用 DeepSeek 这类"思考模型"跑 RAG 时，请设 `STRUCTURED_OUTPUT_METHOD=json_schema`（默认 `function_calling` 适配豆包，思考模型会因强制工具调用报错）。

③ 启动向量库、迁移数据库、创建管理员

```
# 启动 Weaviate 向量库（Docker，监听 8081）
./scripts/run_weaviate.sh

# 初始化数据库 + 创建平台管理员
python manage.py migrate
python manage.py createsuperuser
```

④ 启动服务

```
# 一键启动（tmux 中拉起 Django + 三个后台 worker）
./dev_up.sh

# 或分别启动:
./scripts/run_django.sh                 # Web 服务，监听 0.0.0.0:8000
./scripts/run_worker.sh preprocess      # 预处理 worker
./scripts/run_worker.sh parse           # 解析 worker
./scripts/run_worker.sh index           # 索引 worker
```

完成后浏览器访问 `http://<服务器IP>:8000/login` 即可。如需 HTTPS 域名访问，可在前置 Nginx 反代到 8000 端口。

> **内存提示：**低内存服务器请勿启用 `agent`（DeerFlow 超级体）模式，普通对话/知识库问答不受影响。

3. 登录与测试账户

网站首页即**产品展示页**，点「进入控制台」到达登录页 `/login`。登录框下方内置了「租户」「普通用户」两个**一键登录**的测试账户，点一下即可直接进入；也可手动输入账号（如平台管理员）登录。底部另有「返回首页」「查看用户手册（PDF）」两个链接。

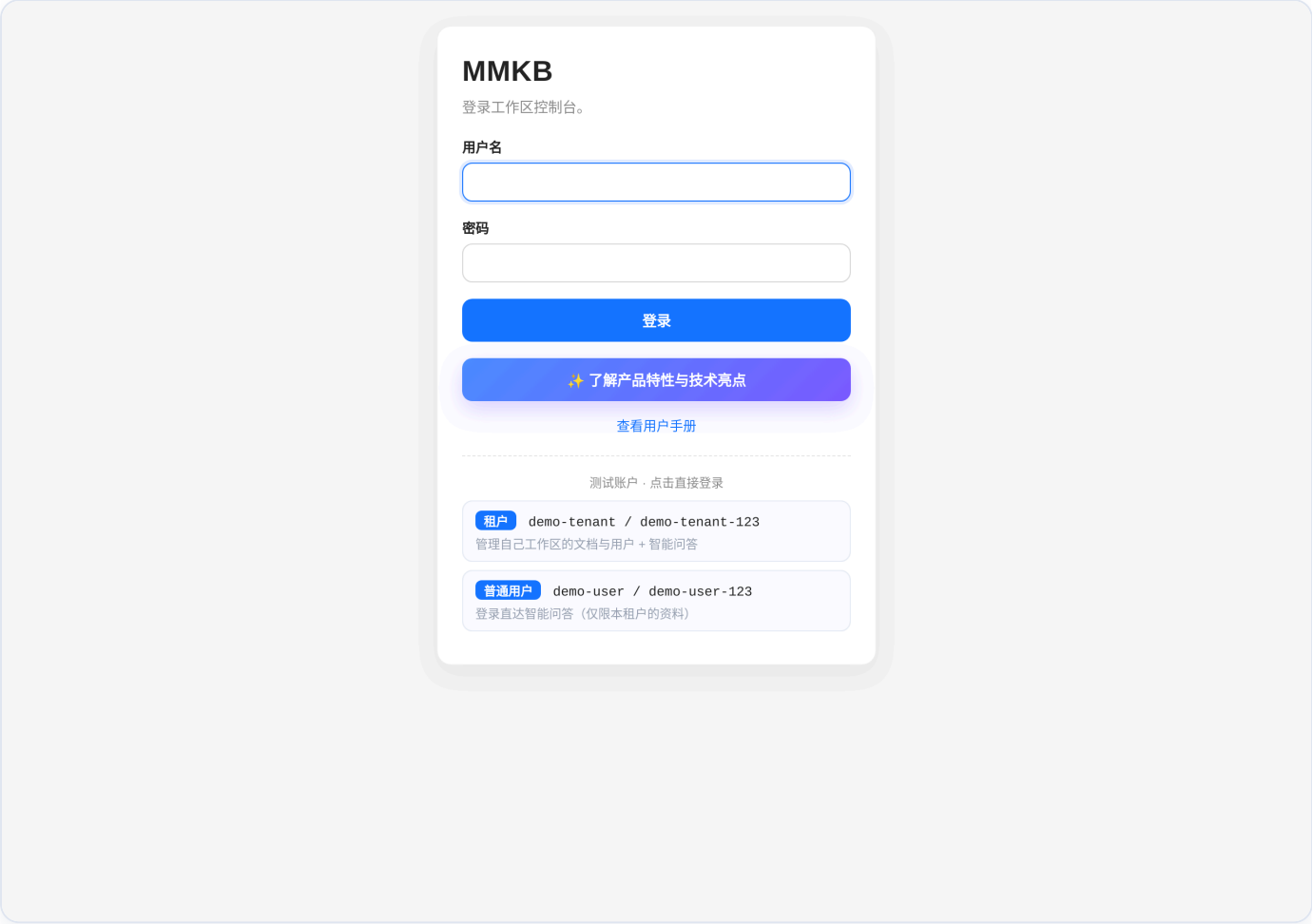

图 3-1　登录页：内置测试账户一键登录 + 产品特性入口

4. 租户：上传与管理文档

用租户账户登录后进入「文档」页。左上角可切换当前工作区；把文件**拖入上传区**或点击上传，支持 PDF、Word/PPT/Excel、图片、音频。上传后系统自动经「预处理 → 解析 → 索引」流水线处理，状态变为 `ready` 即可被检索。

图 4-1　文档管理页：拖拽上传区 + 已就绪文档列表与处理进度

> **填什么：**把文件拖入顶部上传区即可，上传时可填写文档标题（留空则用文件名）。无需其它配置，解析与向量化全自动完成；状态从 upload → parsing → indexing → `ready`。

5. 检索：验证知识库可用

进入「搜索」页，输入问题并选择检索方式，左侧为命中结果与相关度分数，右侧为原文内容与关联图片。三种模式的区别与适用场景如下：

模式	原理	适用场景
语义检索 (semantic)	把查询与文档都编码成向量，按**含义相近**召回。	问"意思"、换了说法也想找到时。例如"怎么请假"能匹配"休假流程"。对**专有名词/编号**可能漏召回。
稀疏检索 (sparse)	基于**关键词/字面**匹配（类似关键词搜索）。	要**精确命中**某个词、机构名、编号、术语、代号时。对同义改写不敏感。
混合检索 (hybrid)	同时跑语义+稀疏并**融合排序**，取两者之长。	**默认推荐**。既要理解语义又要精确命中术语/机构名时用它，绝大多数情况选它即可。

> **一句话选择：**拿不准就用**混合**；只找某个确切词/编号用**稀疏**；只按意思找、不在乎用词用**语义**。

图 5-1　上传文档后搜索：命中文本块、相关度分数与原文/配图详情

6. 创建普通用户（发给终端使用者）

进入「用户管理」→「创建用户」。填写显示用户名；**若填写登录密码**，该用户即可用「用户名 + 密码」登录网页聊天助手；留空则仅作为 OpenAI 兼容 API 用户。系统会为每个用户签发一个 API 密钥。

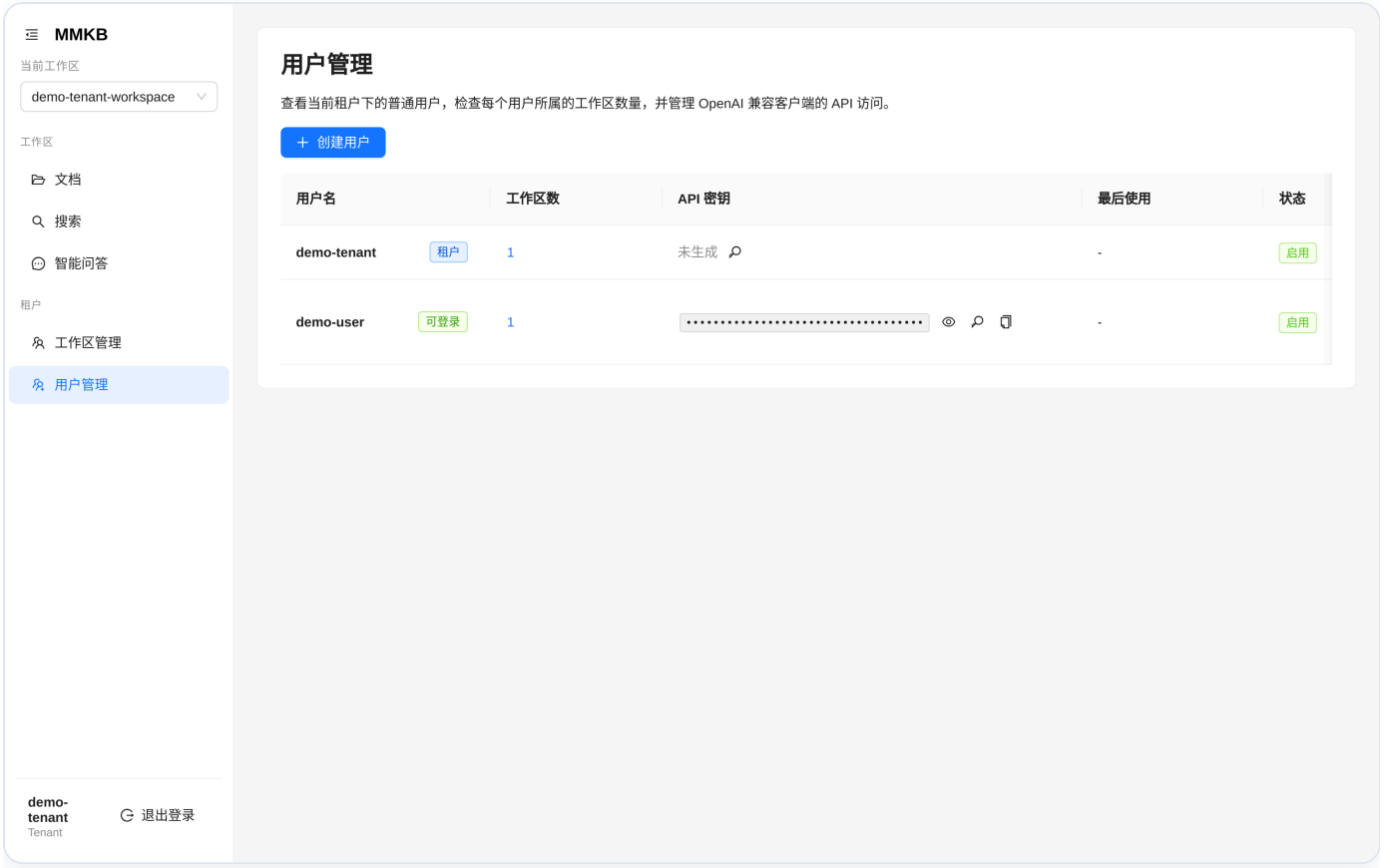

图 6-1　用户管理：查看成员、登录能力与 API 密钥

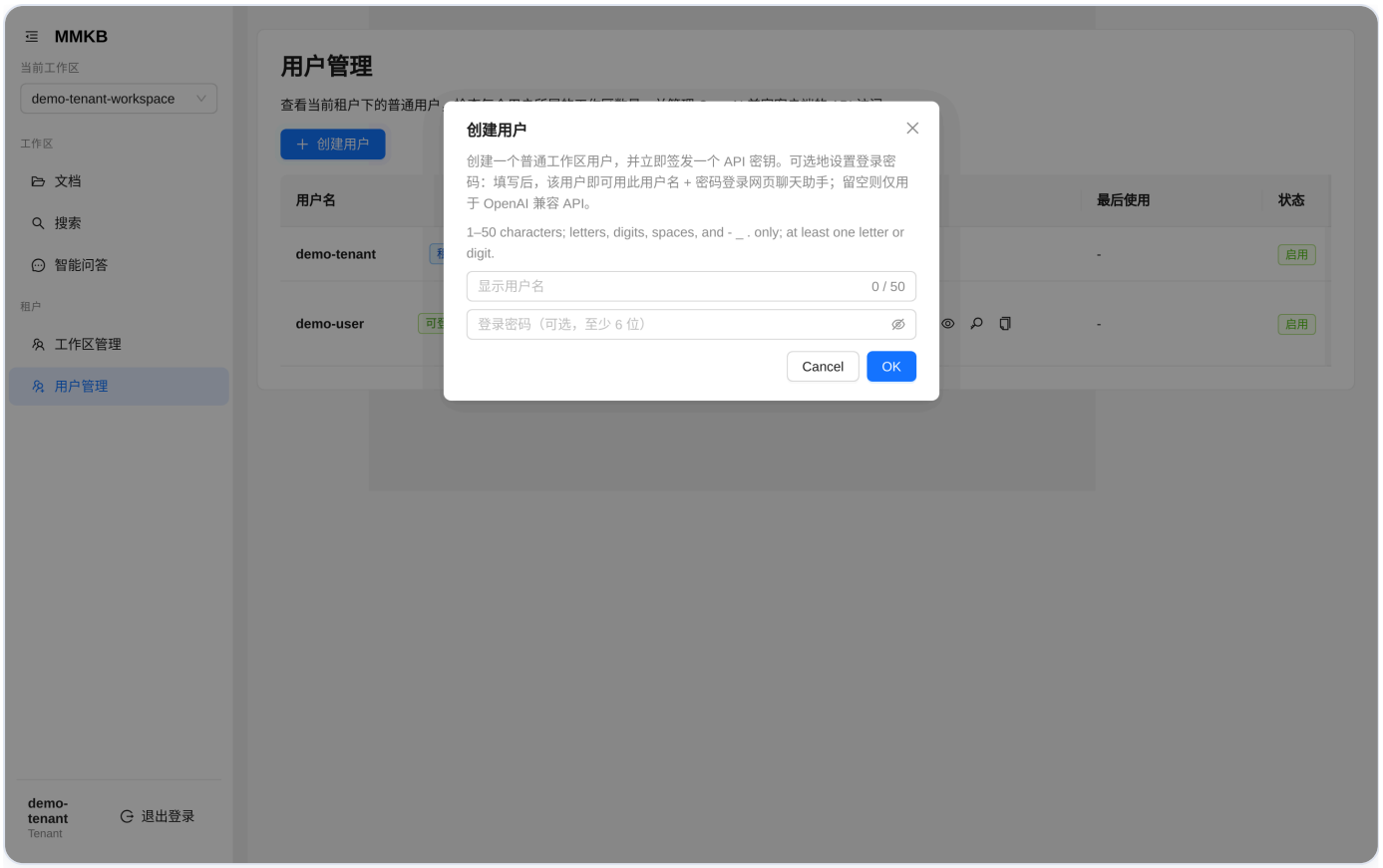

图 6-2　创建用户：填写用户名与可选登录密码

7. 普通用户：智能问答

普通用户登录后**直达一个干净的聊天界面**（不会看到文档/管理页）。在「知识库问答」模式下提问，系统会实时展示**检索与思考过程**（问题路由 → 规范化 → 多轮证据检索 → 进度判断），随后给出带引用编号与配图的答案；切换「直接对话」则不检索知识库。

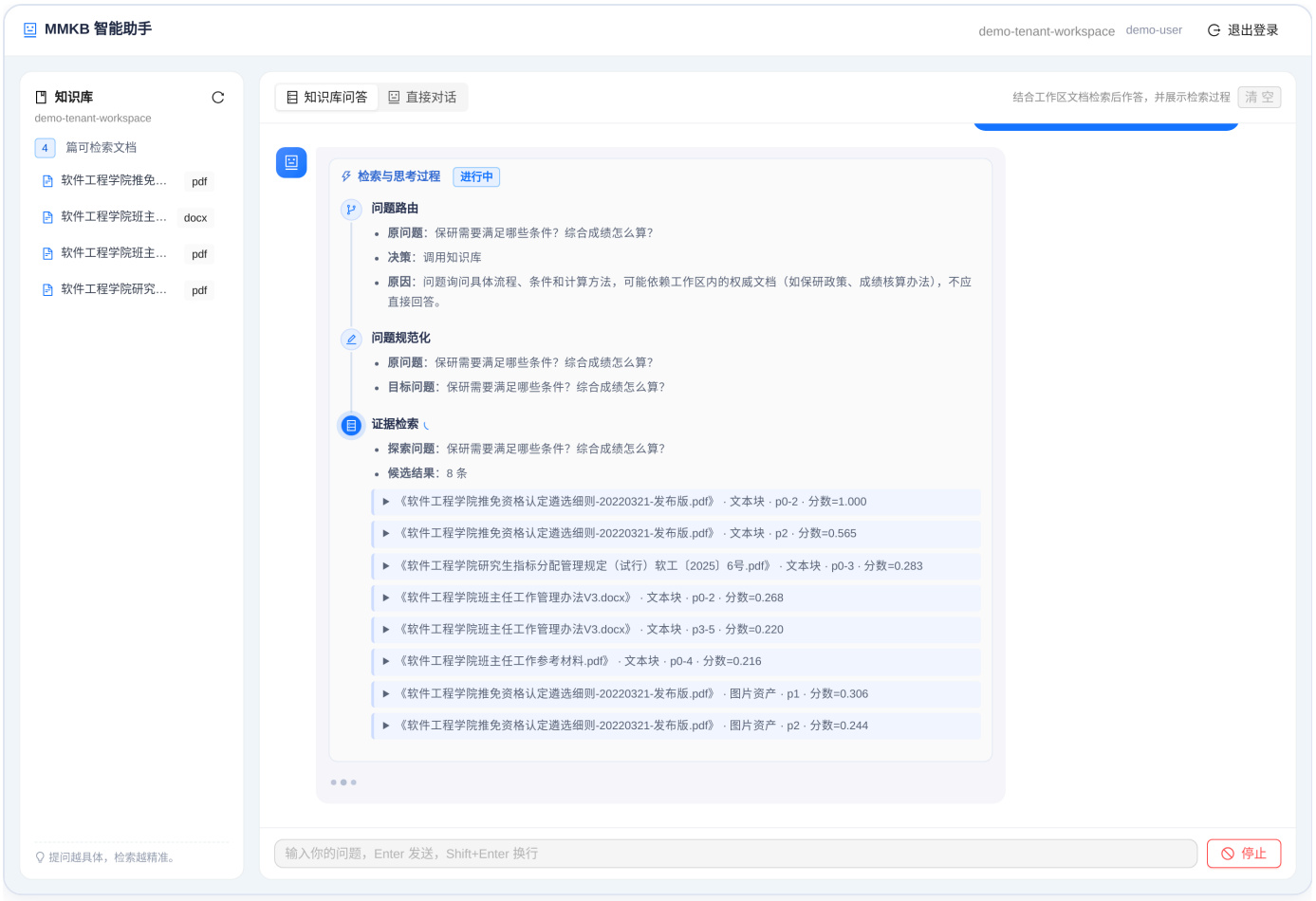

图 7-1　提问后实时展示检索与推理过程（探索链可视化）

图 7-2　最终答案：引用原文证据并插入知识库配图

8. 租户：工作区管理

工作区（workspace）是数据隔离的基本单位：文档、向量索引、成员、会话都归属于某个工作区，不同工作区之间互相不可见。一个租户可拥有多个工作区（例如把"人事库""产品库"分开管理）。

用租户账户登录后，左侧菜单「租户 → 工作区管理」即可 **新建 / 重命名 / 删除工作区**，并查看每个工作区的文档数、集合数、成员数。删除工作区会一并清除其文档、向量与文件，请谨慎操作。

切换"当前工作区"：在左上角「当前工作区」下拉框选择——之后的上传、搜索、创建用户等操作都作用于该工作区。给普通用户建账号（第 6 节）时，用户会被加入你当时**选中的那个工作区**。

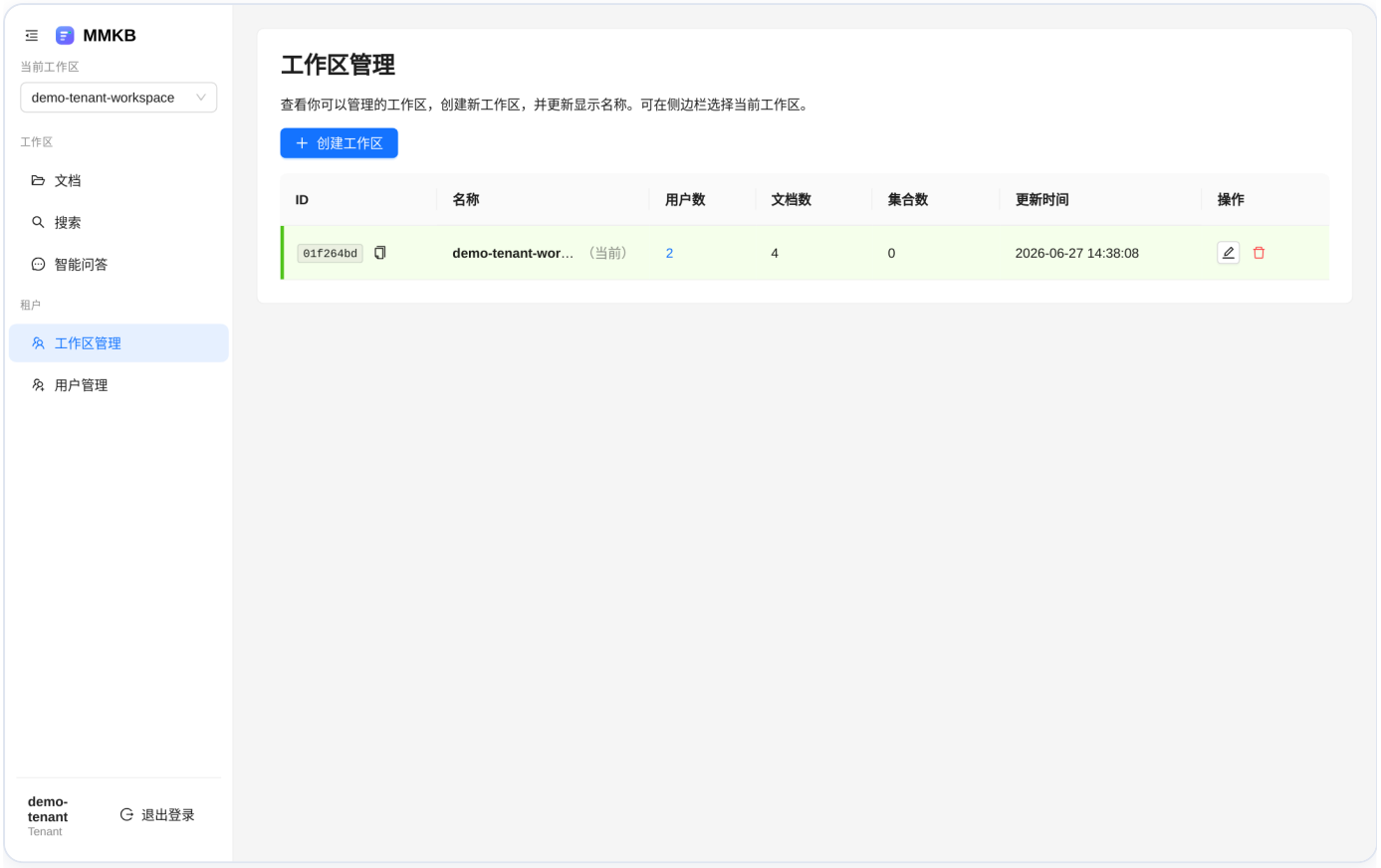

图 8-1　工作区管理：新建/重命名/删除，查看文档与成员数

> **什么时候用得到：**需要按主题/部门隔离资料时用它建多个工作区；若只有一个库、无需隔离，直接用默认工作区即可，不必进此页。

9. 平台管理员

平台管理员是 Django 超级用户（用 `python manage.py createsuperuser` 创建，或具备 `is_staff / is_superuser`），位于所有租户**之上**，用于平台运维。它能看到全部工作区，并比租户多出「系统」菜单：

- **租户管理**（`/tenant-management`）：查看并 **新建租户** 账号，查看每个租户的工作区数、用户数、加入时间。新租户创建后会自动获得一个初始工作区。
- **后台**（`/backend`）：查看并 **启停三个后台 worker**（预处理 / 解析 / 索引）、查看 worker 与 Django 的实时日志——这是排查"文档一直卡在处理中"等问题的入口。

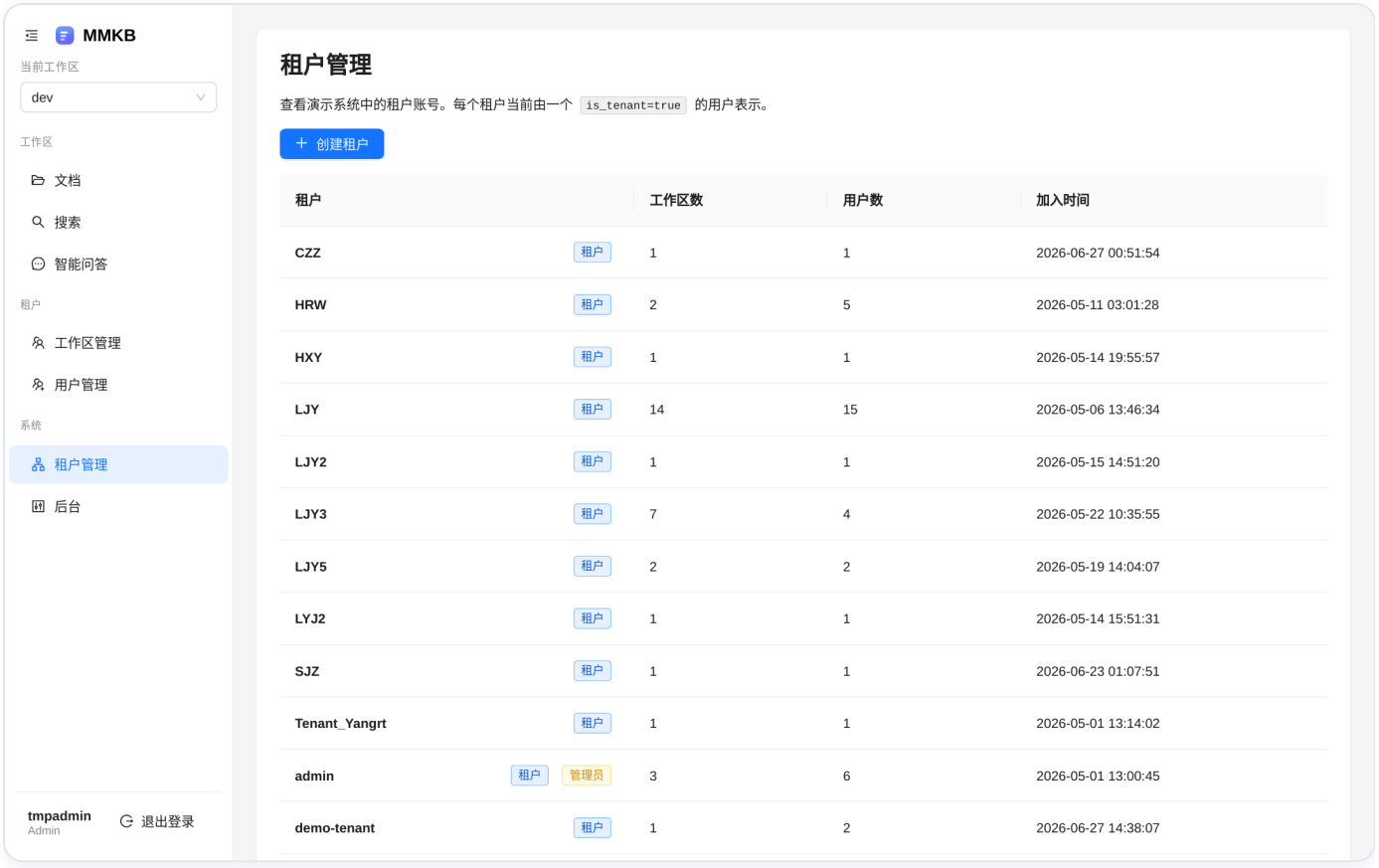

图 9-1　租户管理（平台管理员）：创建与查看各租户

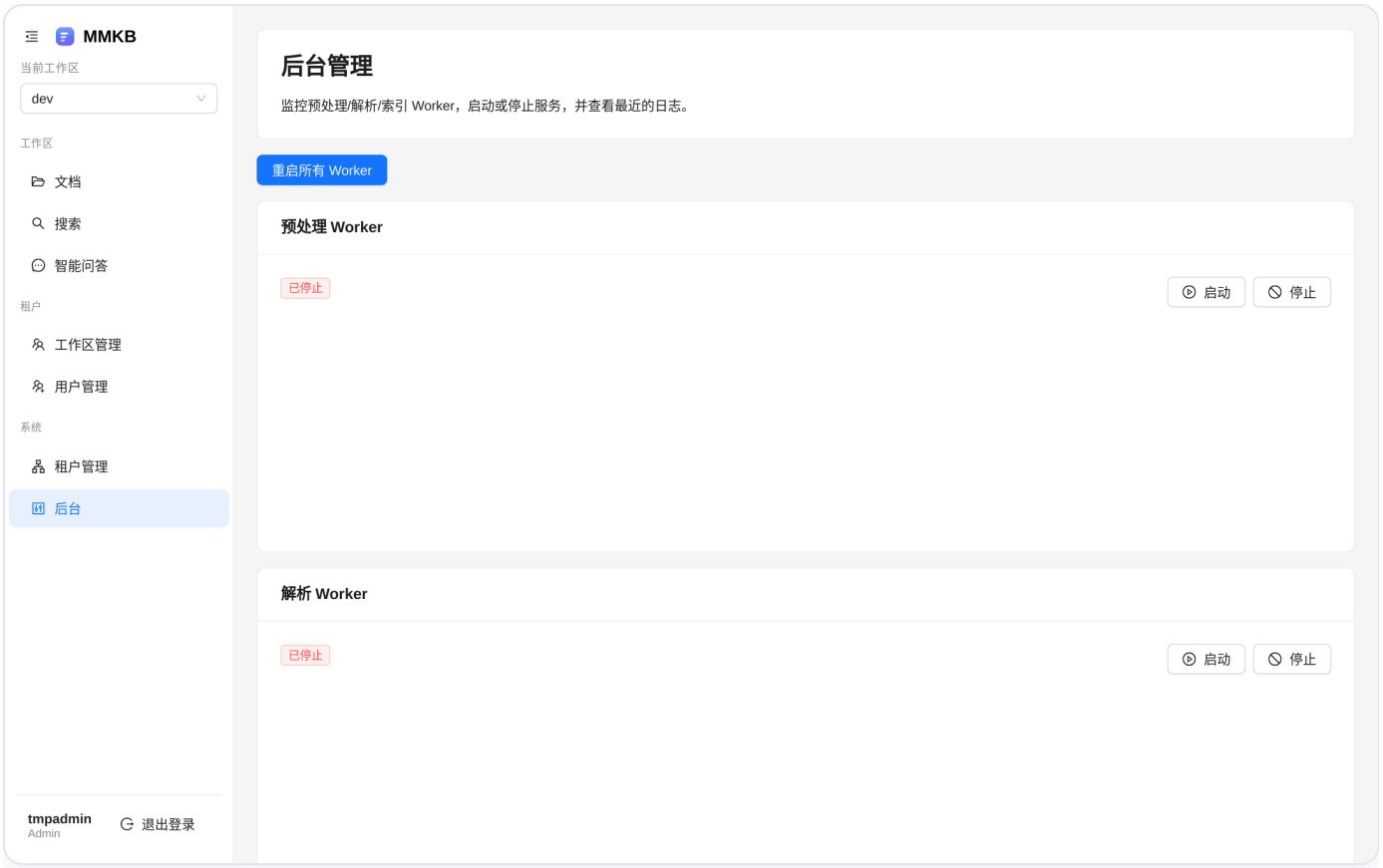

图 9-2　后台：worker 启停与实时日志

安全提示：平台管理员权限很大，账号密码切勿对外分发；对外演示只发放"租户 / 普通用户"账号即可。

附录：环境变量速查表

变量	说明	是否必填
`MULTIMODAL_EMBED_API_KEY`	① 嵌入模型密钥（文档/查询向量化，默认火山引擎 Ark 多模态）	**必填**
`CHAT_COMPLETION_API_KEY`	② 对话/RAG 模型密钥；留空则复用①的嵌入密钥	建议
`CHAT_COMPLETION_BASE_URL`	② 对话模型的 OpenAI 兼容地址；默认 Ark；**换厂商时必改**	可选
`CHAT_COMPLETION_MODEL`	② 对话模型名，如 `doubao-seed-1-6-250615` / `deepseek-v4-flash`	可选
`AGENTIC_RAG_MODEL`	② RAG 内部节点（路由/评估）所用模型；留空＝同 `CHAT_COMPLETION_MODEL`	可选
`STRUCTURED_OUTPUT_METHOD`	② 结构化输出：`function_calling`（默认，豆包）/ `json_schema`（DeepSeek 等思考模型）	可选
`CHAT_DEFAULT_TEMPERATURE`	② 对话默认温度（0~1）；聊天页「高级」里可临时覆盖	可选
`PADDLEX_API_TOKEN`	③ PaddleX 文档解析服务令牌（**非 LLM key**）；解析上传文件必需	解析必填
`AUDIO_ASR_API_KEY`	音频转写（ASR）密钥；仅上传音频文件时需要	音频必填
`DJANGO_ALLOWED_HOSTS`	允许访问的域名/IP，逗号分隔；`*`＝任意（内网演示可用）	可选
`DJANGO_SECRET_KEY`	Django 会话/签名密钥；生产环境务必设一个随机值	生产必填
`DEER_FLOW_INTERNAL_AUTH_TOKEN`	仅启用 `agent`（DeerFlow 超级体）模式时需要；普通部署留空即可	可选

> **关于 `.env.example` 里那些"不知道怎么填"的行：**凡是以 `#` 开头（被注释）的都是**可选、且已有内置默认值**的参数——不改就用默认，需要时删掉行首 `#` 再填即可。真正必须自己填的只有上表标「必填」的几项。

附录：常见问题

Q：文档一直不变成 ready?

检查三个 worker 是否在运行、Weaviate 是否启动、嵌入与解析密钥是否已配置。

Q：问答回答“内容审核拒绝”?

这是上游大模型的内容审核拦截（并非检索失败），可更换内容审核更宽松的模型/服务。

Q：HTTPS 页面图片不显示?

站内问答的图片使用相对地址，经反向代理（设置 `Host` 与 `X-Forwarded-Proto`）即可正常显示。