

# EaziFit 健身 Agent 标准化产品说明书

OpenGYMAgent Team

项目参与人：刘鸿彬、董佩杰

2026 年 3 月 16 日

## 目录

|                                 |          |
|---------------------------------|----------|
| 文档信息                            | 2        |
| 编制说明                            | 3        |
| <b>1 产品概述</b>                   | <b>3</b> |
| 1.1 产品定位                        | 3        |
| 1.2 当前版本一句话价值主张                 | 3        |
| 1.3 当前版本交付范围                    | 3        |
| <b>2 用户定位与场景</b>                | <b>3</b> |
| 2.1 核心用户画像                      | 3        |
| 2.2 典型使用场景                      | 4        |
| 2.2.1 场景一：第一次使用                 | 4        |
| 2.2.2 场景二：今天想开始训练               | 4        |
| 2.2.3 场景三：练完之后看结果               | 4        |
| 2.3 当前版本不承诺的场景                  | 4        |
| <b>3 产品目标与信息架构</b>              | <b>4</b> |
| 3.1 产品目标                        | 4        |
| 3.2 前端信息架构                      | 4        |
| 3.3 后端能力域                       | 5        |
| <b>4 核心模块说明（一）：账号、档案与 AI 对话</b> | <b>5</b> |
| 4.1 账号与会话                       | 5        |
| 4.2 首次欢迎与聊天建档                   | 5        |
| 4.3 WebSocket 流式聊天              | 5        |
| 4.4 个人档案                        | 6        |
| <b>5 核心模块说明（二）：训练工作台与动作能力</b>   | <b>6</b> |
| 5.1 首页 Dashboard                | 6        |
| 5.2 训练计划构建器                     | 6        |
| 5.3 模板库                         | 6        |
| 5.4 动作浏览器                       | 6        |

|                                  |          |
|----------------------------------|----------|
| <b>6 核心模块说明 (三): 执行、统计、健康与设置</b> | <b>6</b> |
| 6.1 训练执行与训练日志                    | 7        |
| 6.2 统计页                          | 7        |
| 6.3 健康总览与 Apple Health           | 7        |
| 6.4 设置与数据管理                      | 7        |
| <b>7 关键业务流程</b>                  | <b>7</b> |
| 7.1 新用户冷启动流程                     | 7        |
| 7.2 今日训练执行流程                     | 8        |
| 7.3 动作库到计划的流转流程                  | 8        |
| 7.4 健康数据导入流程                     | 8        |
| 7.5 当前闭环特征总结                     | 8        |
| <b>8 技术架构与数据架构</b>               | <b>8</b> |
| 8.1 总体技术架构                       | 8        |
| 8.2 存储模式说明                       | 8        |
| 8.3 关键组件关系                       | 8        |
| 8.4 数据与安全边界                      | 9        |
| <b>9 测试、验收与交付状态</b>              | <b>9</b> |
| 9.1 当前验收状态                       | 9        |
| 9.2 验证机制                         | 9        |
| 9.3 质量策略                         | 9        |
| <b>10 当前边界、风险与后续路线</b>           | <b>9</b> |
| 10.1 当前边界                        | 9        |
| 10.2 主要风险                        | 10       |
| 10.3 推荐后续路线                      | 10       |
| 10.4 结论                          | 10       |

## 文档信息

| 项目   | 内容  |
|------|---|
| 产品名称 | EaziFit 健身 Agent  |
| 仓库名称 | OpenGYMAgent  |
| 文档版本 | V1.0  |
| 文档日期 | 2026-03-16  |
| 适用对象 | 产品、研发、设计、测试、运营、投标与项目申报  |
| 编写依据 | README.md、docs/DEVELOPMENT.md、<br>docs/TESTING.md、feature_list.json、前后端<br>当前实现 |
| 编写原则 | 若旧版概念稿与代码实现冲突，以当前实现为准   |
| 篇幅控制 | 按 A4 导出约 10 页内容组织，建议导出时使用<br>12pt 正文字号  |

## 编制说明

本说明书基于仓库当前已经落地的产品能力编写，不再沿用早期草稿中尚未实现的功能承诺。旧稿中关于“懒人友好、低门槛启动、训练与健康一体化”的定位仍然有效，但诸如 `openclaw` 独立提醒系统、摄像头动作纠错、健康社区、补剂智能推荐、强运营化内容体系等内容，当前版本尚未形成完整交付链路，因此本文件仅将其作为后续演进方向，不作为现阶段标准交付能力。

当前系统的真实产品边界可以概括为一句话：**EaziFit** 是一个面向个人用户的 AI 健身工作台，围绕“建档、对话、计划、执行、记录、分析、健康摘要、数据管理”形成闭环，前端由 `React + Vite` 实现，后端由 `FastAPI` 提供多用户、分用户隔离的数据与 `Agent` 服务。

## 1 产品概述

### 1.1 产品定位

**EaziFit** 的定位不是传统课程型健身 App，也不是单纯的训练记录工具，而是一个以 AI 交互为核心入口的个人健身管理系统。它面向三类最核心的真实用户需求：

1. 不想研究复杂计划，但希望快速开始训练的人。
2. 已经开始训练，但缺少结构化执行和复盘工具的人。
3. 希望把训练记录、统计分析与健康数据放在一个工作台里统一查看的人。

从当前实现来看，产品的核心价值不在“海量课程内容”，而在“把分散的训练决策前移为一个连续的数字流程”：先通过登录和建档明确个人背景，再通过聊天与计划工具生成可执行方案，随后进入训练执行和训练记录，最后通过统计、健康总览和 `Apple Health` 摘要完成复盘。

### 1.2 当前版本一句话价值主张

**EaziFit** 帮助用户用更低的决策成本完成从“我今天练什么”到“我练完之后如何看结果”的完整链路。

### 1.3 当前版本交付范围

基于 `frontend/src/App.tsx`、`frontend/src/services/api.ts` 与 `backend/app/main.py`，当前版本已形成如下交付范围：

- 多用户注册、登录、鉴权与会话保持。
- 首次欢迎语、`WebSocket` 流式聊天、聊天内建档和日计划引导。
- 个人档案管理，包括基础身体信息、目标、训练环境、周频率、伤病与慢性病信息。
- 仪表盘、训练计划构建器、训练模板库、动作库、训练日志、统计页、健康总览、`Apple Health` 页面、设置页。
- 动作收藏、训练会话管理、计划版本保存、训练数据导出与清空。
- 后端高级能力接口，包括 `evidence`、`prediction`、`digital twin`、`privacy` 等能力底座。

## 2 用户定位与场景

### 2.1 核心用户画像

| 用户类型   | 典型特征                | 当前版本可满足的价值  |
|--------|---------------------|---|
| 轻量入门用户 | 缺少系统训练经验，不知道怎么开始    | 通过欢迎语、建档卡、聊天和计划构建器降低启动门槛                          |
| 回归训练用户 | 断练后回归，想快速恢复节奏       | 通过个人档案、计划版本、训练日志和统计页恢复训练结构                        |
| 数据驱动用户 | 已有训练习惯，关注 PR、训练量与趋势 | 通过统计页、动作统计、日历热力图和 <code>Apple Health</code> 摘要做复盘 |

## 2.2 典型使用场景

### 2.2.1 场景一：第一次使用

用户完成注册登录后进入聊天页，系统自动拉取欢迎语。如果判定为新用户且尚未建档，前端会展示 ChatOnboardingCard，用户可补充目标、频率、训练环境、时长、伤病史等信息。信息保存后，系统可继续在聊天中给出更贴近个人背景的建议，并根据训练日/恢复日给出对应引导卡片。

### 2.2.2 场景二：今天想开始训练

用户可以从首页查看本周训练概况、今日推荐动作、最近训练和当前计划，也可以直接进入训练计划构建器按器械与肌群生成训练方案。若已有模板、已有 AI 计划版本或已有历史训练，也可以直接一键转成训练会话，进入记录页面开始执行。

### 2.2.3 场景三：练完之后看结果

训练完成后，系统会将训练会话沉淀为结构化训练日志，并在统计页呈现连续打卡、月度训练趋势、训练热力图、单动作进展、肌群热度和 PR 信息。同时，健康总览和 Apple Health 页面可以补充睡眠、活动三环、心率、步数等健康维度摘要。

## 2.3 当前版本不承诺的场景

以下内容虽可作为产品愿景保留，但当前版本不应写入正式交付承诺：

- 摄像头动作识别与动作纠错。
- 独立的提醒助手与外部消息推送编排。
- 健身社区、圈层互动、内容运营 feed。
- 自动化饮食计划、补剂推荐闭环 UI。
- 医疗级诊断或疾病风险结论。

## 3 产品目标与信息架构

### 3.1 产品目标

当前版本的产品目标不是覆盖全部健身服务，而是优先验证以下四个目标：

1. 建立一个稳定可用的多用户 AI 健身工作台。
2. 把“聊天式辅助”和“结构化训练执行”真正接起来。
3. 让训练记录、统计分析和 Apple Health 摘要形成一个可持续复盘闭环。
4. 为后续更强的个性化推荐、数字孪生和健康推理能力预留接口与数据基础。

### 3.2 前端信息架构

依据 `frontend/src/App.tsx`，当前主要页面包括：

| 页面              | 作用                           |
|-----------------|------------------------------|
| chat            | 欢迎语、AI 对话、聊天建档、计划问答、流式回复     |
| dashboard       | 首页工作台，展示周训练概况、今日动作、最近记录与快捷入口 |
| builder         | 训练计划构建器，按器械与肌群生成计划并保存版本      |
| templates       | 模板库，管理官方模板、个人模板和 AI 计划入口     |
| log             | 训练执行与训练日志记录页                 |
| stats           | 训练统计分析页                      |
| history         | 历史训练日历视图                     |
| health_overview | 综合健康总览                       |

| 页面           | 作用                     |
|--------------|------------------------|
| apple_health | Apple Health 数据摘要与同步状态 |
| profile      | 个人主页与健身档案编辑            |
| settings     | 偏好、导出、清空数据与账户设置        |
| exercises    | 动作浏览器、动作详情、收藏与加入计划     |

### 3.3 后端能力域

依据 `backend/app/main.py`，后端已形成以下能力域：

- 认证域： `/api/auth/*`
- 对话域： `/api/greeting`、`/api/ws/{session_id}`、`/api/chat`
- 档案与训练域： `/api/profile`、`/api/workouts`、`/api/workout-sessions`
- 动作与计划域： `/api/exercises`、`/api/workout-builder/*`
- 进度域： `/api/progress/*`
- 健康导入域： `/api/import/*`、`/api/apple-health/*`
- 设置与数据域： `/api/settings`、`/api/data/*`
- 高阶研究能力域： `/api/biomarkers/*`、`/api/twin/*`、`/api/evidence/*`、`/api/prediction/*`、`/api/privacy/*`

## 4 核心模块说明（一）：账号、档案与 AI 对话

### 4.1 账号与会话

系统支持用户注册、登录、获取当前用户信息和登录态保持。前端通过本地保存 token 并在请求头中附带 `Authorization` 完成鉴权；后端按用户隔离数据目录和 Agent 上下文，确保不同用户的会话、档案、训练记录和记忆空间互相独立。

### 4.2 首次欢迎与聊天建档

聊天页是当前产品的第一入口。`useChat` 会在页面首次挂载时自动调用 `/api/greeting`，返回欢迎语、是否为新用户、是否已建档，以及可选的训练日/恢复日引导信息。若用户尚未完成建档，前端展示 `ChatOnboardingCard`，字段覆盖：

- 性别、年龄、身高、体重。
- 训练目标、训练水平。
- 训练环境、每周频次、单次时长、近期状态。
- 伤病历史与慢性病信息。

这个设计承接了旧稿中“降低冷启动成本”的思路，但当前实现更务实：不是完全靠对话逐步问答，而是采用“欢迎语 + 建档卡 + 自动生成补充提示”的混合式流程，以减少新用户第一次进入时的理解成本。

### 4.3 WebSocket 流式聊天

系统已支持真正的流式聊天输出。后端通过 OpenAI 兼容接口的流式能力逐步回传 `text_delta`，前端在 `useChat` 中进行增量拼接，同时支持：

- 中途停止生成。
- 工具调用提示 `tool_hint`。
- 对流式污染文本进行防御性清洗。
- 在训练日或恢复日场景下追加结构化行动建议卡片。

这意味着当前聊天模块不是单纯的“问答窗口”，而是连接档案、计划、训练和健康建议的智能入口。

#### 4.4 个人档案

档案模块既可在聊天中完成首次录入，也可在个人主页中继续编辑。当前档案信息可用于：

- 展示姓名、头像、目标、训练环境与节奏。
- 计算 BMI、TDEE 等基础指标。
- 为训练计划生成、健康评分和建议文案提供上下文。
- 支撑训练频次与单次训练时长的默认判断。

从产品视角看，档案模块已经不是附属设置，而是整个系统个性化的起点。

## 5 核心模块说明（二）：训练工作台与动作能力

### 5.1 首页 Dashboard

首页当前承担“训练工作台”的职责。它聚合了：

- 本周训练次数、累计组数、最近训练记录。
- 今日推荐动作与今日计划预览。
- 继续进行中的训练会话。
- 一键跳转至聊天、训练计划构建器或训练日志。

这一页的产品意义是把原本分散在多个页面里的“今天练什么、是否有历史、我该从哪开始”收敛为一个入口。

### 5.2 训练计划构建器

WorkoutBuilder 是当前版本最强的结构化训练能力之一。其流程为三步：

1. 选择器械。
2. 选择目标肌群。
3. 生成训练计划并支持替换、补热身、保存版本、激活计划、直接开始训练。

计划构建器支持读取动作库、热身动作、历史计划草稿和版本化保存。相比旧稿中的“完全由 AI 自动生成日计划”，当前实现更强调“用户主动选择 + 系统智能匹配”的可控性，更适合作为正式交付方案。

### 5.3 模板库

模板页分为“个人模板”和“个人计划（AI）”两个方向：

- 个人模板：加载官方模板与用户模板，支持展开分类、快速开始、删除自定义模板。
- AI 计划：作为聊天或计划版本的入口，承接更偏个性化的生成场景。

这让产品既保留了可复制的标准模板，也保留了个性化计划的入口。

### 5.4 动作浏览器

动作浏览器提供当前版本最完整的动作数据能力：

- 关键词搜索。
- 按肌群、器械、类型、热身区域筛选。
- 动作详情展示，包括图片/视频、描述、关键点。
- 收藏与取消收藏。
- 加入今日训练计划。
- 直接向 AI 询问该动作。

当前实现已经把“动作库”从静态列表升级为可被计划构建、聊天和训练执行共同复用的基础设施。

## 6 核心模块说明（三）：执行、统计、健康与设置

## 6.1 训练执行与训练日志

LogPage 同时承担“训练执行器”和“训练日志录入器”的角色。它支持：

- 从 Dashboard、模板、计划版本或历史会话进入执行态。
- 为每个动作记录组数、重量、次数。
- 训练计时和组间休息倒计时。
- 会话草稿自动同步。
- 训练完成后沉淀为结构化训练记录。

这部分能力让 EaziFit 不止能“给建议”，还能把建议真正落到执行结果。

## 6.2 统计页

统计页当前是训练复盘的核心看板，已落地的展示包括：

- 连续训练天数。
- 月度训练趋势。
- 个人 PR。
- 训练热力图与日历。
- 肌群热度与训练分布。
- 单动作强度和训练量统计。

对有训练习惯的用户而言，这一页是产品留存的关键，因为它将“做过什么”转化为“看得懂的进展”。

## 6.3 健康总览与 Apple Health

健康相关能力分成两层：

- **HealthOverview**：综合训练、睡眠、补水、目标等形成统一的健康摘要。
- **AppleHealthDashboard**：展示 Apple Health 日摘要、活动三环、训练记录、数据来源模式、同步状态与可信度提示。

需要特别说明的是，当前版本强调的是“健康数据摘要与可信度展示”，而不是“医疗诊断”。相关页面会标明数据来自真实设备、测试导入或演示数据，并在低可信度场景给出提示。

## 6.4 设置与数据管理

设置页支持：

- 主题、语言、单位、时区等偏好设置。
- Agent 参数如 `temperature`。
- 数据导出。
- 删除全部数据。
- 退出登录。

这保证了产品不仅有训练功能，也具备基础的数据自治能力，适合正式交付时强调“用户可管理自己的数据”。

# 7 关键业务流程

## 7.1 新用户冷启动流程

1. 用户注册并登录。
2. 系统自动获取欢迎语与用户状态。
3. 若未建档，聊天页展示建档卡。
4. 用户提交关键资料后，前端调用 `/api/profile` 保存。
5. 系统根据档案信息和训练日类型返回引导内容。
6. 用户可继续聊天、生成计划或直接进入首页工作台。

## 7.2 今日训练执行流程

1. 用户从首页、模板页或计划构建器选择训练方案。
2. 前端创建训练会话并进入日志页。
3. 用户记录每个动作的组数、重量、次数，系统自动保存草稿。
4. 用户完成训练后提交会话。
5. 后端将会话转为训练日志。
6. 统计页与健康总览在后续查询中自动体现新数据。

## 7.3 动作库到计划的流转流程

1. 用户在动作浏览器搜索、筛选、收藏动作。
2. 在计划构建器中按器械和肌群生成方案。
3. 用户可对动作做替换、补充热身或保存为计划版本。
4. 计划版本可再次激活或直接转成训练执行入口。

## 7.4 健康数据导入流程

1. 用户进入 Apple Health 页面或通过导入接口同步数据。
2. 后端写入时序数据与汇总信息。
3. 页面显示日摘要、周摘要、数据源类型与可信度状态。
4. 健康总览将训练与健康维度进行组合展示。

## 7.5 当前闭环特征总结

当前产品已经具备一个完整的“建档 -> 计划 -> 执行 -> 记录 -> 统计 -> 健康复盘”闭环，这是本项目现阶段最值得强调的产品成就。

# 8 技术架构与数据架构

## 8.1 总体技术架构

| 层级    | 当前实现   |
|-------|--|
| 前端    | React 18 + Vite + TypeScript                     |
| 后端    | FastAPI + Python 3.11+                           |
| 模型接入  | OpenAI 兼容接口                                      |
| 流式通信  | WebSocket  |
| 文件存储  | backend/data/users/<user_id>/ 下的用户级 JSON / JSONL |
| 结构化存储 | backend/data/openlife.db、backend/data/users.db   |

## 8.2 存储模式说明

当前系统采用“混合存储”而非单一数据库：

- 用户档案、训练记录、会话记忆等数据按用户目录隔离，便于导出、删除与多用户隔离。
- 动作库、收藏、训练会话、时序数据等结构化能力使用 SQLite 存储，默认数据库文件为 backend/data/openlife.db。
- 用户认证信息独立保存在 backend/data/users.db。

这种设计的优点是开发效率高、调试直观、可在本地快速落地；同时也为后续迁移到更强的数据库方案保留了接口边界。

## 8.3 关键组件关系

- frontend/src/App.tsx 负责主工作台路由切换。

- `frontend/src/services/api.ts` 定义前端对后端的主要访问契约。
- `frontend/src/hooks/useChat.ts` 与 `useWebSocket.ts` 管理聊天流式链路。
- `backend/app/main.py` 是主要 API 入口。
- `backend/app/agent/` 负责 Agent 循环、工具调度与会话管理。
- `backend/app/storage/` 负责动作、收藏、训练会话、计划版本、时序数据存储。

## 8.4 数据与安全边界

当前系统已实现以下基础安全边界：

- 认证用户后才能访问核心训练与健康数据。
- 数据按用户隔离目录和用户 ID 进行读写。
- 导出与删除数据通过显式接口触发。
- 测试策略要求外部模型与外部网络依赖尽量 mock，避免将测试建立在真实第三方可用性之上。

# 9 测试、验收与交付状态

## 9.1 当前验收状态

根据 `feature_list.json` 与 `docs/NEXT_TASK.md`，截至 2026-03-16：

- `feature_list.json` 共 97 / 97 条需求已通过验收。
- 当前不存在 `passes=false` 的待验收条目。
- 最近完成任务为 FEAT-97，聚焦前端统计页升级与回归验证。

## 9.2 验证机制

项目已经建立统一验证脚本：

- `./scripts/agent/verify.sh docs`
- `./scripts/agent/verify.sh backend`
- `./scripts/agent/verify.sh frontend`
- `./scripts/agent/verify.sh all`

其中：

- `docs` 负责 JSON、Shell、文档结构检查。
- `backend` 负责 `pytest`。
- `frontend` 负责 ESLint、Vitest 与生产构建。
- `e2e` 支持浏览器端到端链路，但在受限环境中可退化为语法和构建校验。

## 9.3 质量策略

当前项目的质量策略是“以高风险链路的自动化回归为核心”，主要覆盖：

- 认证与多用户隔离。
- 流式聊天与欢迎语链路。
- 档案、训练、进度、动作库等关键接口。
- 前端关键视图的展示与交互。
- Apple Health 导入与摘要数据处理。

这意味着本项目已具备较成熟的工程化交付基础，不是单纯的原型演示。

# 10 当前边界、风险与后续路线

## 10.1 当前边界

为了保证说明书与实现一致，以下边界需要明确写入：

- 当前版本是 AI 健身工作台，不是完整的内容型健身平台。

- 当前版本支持健康数据摘要，但不提供医疗结论。
- 当前版本已具备训练计划和训练执行闭环，但饮食管理仍以工具与接口能力为主，前台闭环较弱。
- 当前版本已有高级研究接口，但这些接口尚未全部形成前台稳定产品功能。

## 10.2 主要风险

| 风险点         | 说明  |
|-------------|---|
| 产品能力分层不均    | 训练能力最完整，饮食、提醒、社区等模块仍偏早期   |
| 高级接口未完全产品化  | <code>twins</code> 、 <code>prediction</code> 、 <code>privacy</code> 更像平台能力，前台场景需要继续收敛                                     |
| 文档与品牌命名历史混用 | 仓库、旧稿、前端品牌存在 <code>OpenGYMAgent</code> 、 <code>MOVA</code> 、 <code>EaziFit</code> 多套命名，当前应以用户可见品牌 <code>EaziFit</code> 为准 |

## 10.3 推荐后续路线

建议下一阶段按“先补闭环，再扩场景”的顺序推进：

1. 强化训练闭环：让计划版本、训练会话、今日动作、聊天建议之间形成更统一的用户路径。
2. 完成饮食闭环：将已有营养能力从工具层推进到可见 UI。
3. 完成提醒闭环：若要重新引入旧稿中的提醒概念，应先明确触发器、频率、通知渠道与用户授权模型。
4. 产品化高级能力：把 `digital twin`、`prediction`、`evidence` 收敛为用户能理解的摘要卡，而不是裸 API。
5. 统一品牌与对外文档：在不做大规模代码重命名的前提下，逐步清理对外资料中的历史命名残留。

## 10.4 结论

从当前实现出发，`EaziFit` 已经不是单点功能应用，而是一个具备多用户、AI 对话、结构化训练、健康摘要、数据导出和自动化测试基础的健身产品平台。它最适合对外表达为“以 AI 为入口的个人健身与健康工作台”，而不是夸大为一个已经完成全部健康生态闭环的超级应用。这样的表述既符合代码现实，也更有利于后续迭代与正式交付。