

2025/1/25

- 后端仓库: A1 → A2 → C1
- 前端仓库: B1 → B2 (此时后端至少要有 /api/auth/me 基础可用)
- 后端仓库: C2 (detect 富响应) → C3 (parse-files)
- 前端仓库: B3 (Scan 对接) → B4 (MultiUpload)
- 后端仓库: C4 (credits) → C5 (history) → C6 (billing/config/contact)
- 前端仓库: B5 (history 对接) → B6 (billing/config/contact 对接)
- 前端仓库: D1、D3; 后端仓库: D2 (对齐自动化)

对齐前后端, 先补齐后端的数据库, 目前在链接前后端的登录注册, 注册功能正常, 获取用户信息 api 以及登录暂时不能使用, 因为所提交的 message 和后端所需要的不符。
后端已完成 C3 的重构, 前端已完成 B1,B2 重构, 正在实现 B3

A. 后端仓库 Prompts (先把契约定死 + 对齐基础设施)

【后端仓库 Prompt A1】生成 OpenAPI 契约基线 + Gap List (单一事实来源)

目标: 基于前端期望与后端现状, 产出第一版 contract/openapi.yaml 和 contract/gap-list.md, 作为唯一标准。

必须遵循的事实

现状: /detect 仅接收 JSON text; 不支持 UploadFile; /auth/login 仅返回 token。

后端清单

目标: 前端期望 /api/scan/detect (functions[] + AnalysisResponse 富结构)、/api/scan/parse-files (FormData files[])、Billing/Config/Contact 等。

前端清单

前端清单

决策: simulateAnalysis 迁移后端; camelCase 输出; 路由别名; 扣费后端事务化。

后端适配与架构决策方案

后端适配与架构决策方案

要做的改动 (文件级)

1. 新增目录 contract/:
 - contract/openapi.yaml
 - contract/gap-list.md (表格: 模块/接口/差异/建议/优先级/风险/影响前端页面)
2. OpenAPI 必须覆盖 (至少):
 - Auth: POST /api/auth/login、POST /api/auth/register、GET /api/auth/me
 - Scan: POST /api/scan/detect、GET /api/scan/examples、GET /api/scan/history、GET /api/scan/history/{id}、POST /api/scan/parse-files
 - Billing: POST /api/orders、POST /api/orders/{id}/pay、GET /api/subscriptions/preview
 - Config/Support: GET /api/config/faq、GET /api/config/quotes、POST /api/contact
3. Schema 要严格对齐前端:

AnalysisResponse 包含 summary/sentences/polish/translation/citations

前端清单

Detect 请求: { text: string, functions?: string[] }

前端清单

1. 在 gap-list.md 明确标注:
 - login 响应差异 (后端仅 token vs 前端 {token,user})
 - parse-files (后端无 UploadFile vs 前端需要)
 - detect 富响应差异

验收标准

- contract/openapi.yaml 能被 Swagger/Redoc 正常解析。

- gap-list.md 能回答“前端每个页面需要哪些接口，后端现在缺什么”。

【后端仓库 Prompt A2】契约治理：CI 门禁 + 变更流程

目标：把“先改契约再改实现”做成强制规则。

要做的改动

1. 新增 CONTRACT.md：命名规则（camelCase）、错误结构、分页字段、鉴权 header、时间格式。
2. 加 CI：校验 contract/openapi.yaml 格式与一致性（openapi-validator 或 swagger-cli）。
3. PR 模板增加勾选项：契约是否更新、是否有对应测试。
4. 新增 contract/changelog.md：记录破坏性变更与兼容期。

验收标准

- 故意破坏 OpenAPI 格式，CI 必须 fail。
- PR 没改契约但改了 API 结构，应在 Review/检查项中被阻断（至少流程上强制）。

【后端仓库 Prompt C1】路由前缀 /api + 旧路径 alias + 全局 camelCase 输出

目标：让前端能统一请求 /api/...，同时不破坏现有 /detect、/detections；并统一 camelCase 输出。

后端适配与架构决策方案

要做的改动

1. 新增路由前缀：/api 或 /api/v1（以契约为准）
2. 加 alias：
 - /api/scan/detect 指向现有 detect handler
 - /api/scan/history 对齐到 detections 列表（但字段结构要按契约返回）

Pydantic Response 统一 alias_generator 输出 camelCase。

后端适配与架构决策方案

1. 统一错误结构：{code, message, detail}（所有接口一致）

验收标准

- /api/... 路由可通。
- 返回字段为 camelCase（不再出现 created_at/page_size）。

【后端仓库 Prompt C2】Detect 富响应：functions[] + AnalysisResponse

目标：把前端 simulateAnalysis 迁移到后端；POST /api/scan/detect 返回完整 AnalysisResponse。

后端适配与架构决策方案

要做的改动

Request：新增 functions?: string[]（默认至少包含 scan）

前端清单

Response：新增 AnalysisResponse（summary/sentences/polish/translation/citations）

前端清单

1. 兼容：旧 /detect 可保留旧结构，但新 /api/scan/detect 必须按契约返回富结构。
2. 可先 stub：polish/translation/citations 可先返回占位结果，但结构必须正确，后续替换实现。

验收标准

- 前端 Scan 页面只靠后端返回即可渲染（为后续前端 Prompt B3 做准备）。
- functions 开关生效（未请求的模块字段为 null/缺省但符合契约）。

【后端仓库 Prompt C3】parse-files: UploadFile 多文件解析接口

目标：实现 /api/scan/parse-files 支持 pdf/docx/txt，输出 results[{fileName, content}]。

前端清单

要做的改动

1. 新增 endpoint: POST /api/scan/parse-files 接受 multipart/form-datafiles[]
2. 解析实现: pdf (pypdf/PyPDF2)、docx (python-docx)、txt (utf-8)
3. 安全: 限制文件大小/数量; 对不支持类型返回明确错误 (每文件级别)

验收标准

- 用 curl/PowerShell 上传多文件返回结构符合契约。
- 单文件失败不致全体 500 (返回 per-file error 或在 detail 中体现, 契约同步)。

【后端仓库 Prompt C4】Credits: 后端事务扣费 + /me 返回余额 + detect 返回 currentCredits

目标: 扣费必须后端事务化, 前端不做本地扣减。

后端适配与架构决策方案

要做的改动

1. 数据库迁移: user 增加 credits/quota 字段 (含 alembic)

/api/auth/me 返回 credits (total/remaining)

前端清单

1. detect: 同一事务中创建 detection + 扣 credits + 返回 currentCredits

验收标准

- 并发 N 次 detect 后 remaining 正确递减, 不出现负数。
- 前端可用 /me 和 detect 响应更新余额。

【后端仓库 Prompt C5】History: /api/scan/history 列表与 /api/scan/history/{id} 详情

目标: 替换前端 LocalStorage 历史, 返回字段按契约。

前端清单

要做的改动

1. /api/scan/history: 支持分页参数 (按契约)
2. 列表返回 {list,total}, list 项含 id,title,createdAt,summary
3. /api/scan/history/{id}: 返回完整 AnalysisResponse (与 detect 同结构)

验收标准

- 前端能用这两个接口做历史列表与详情回显。

【后端仓库 Prompt C6】Billing/Config/Contact: 最小可用闭环

目标: 补齐前端显性调用的订单/支付/订阅预览, 以及 FAQ/quotes/contact。

前端清单

要做的改动

1. orders: 创建订单、支付订单 (状态机)
2. subscriptions preview: 返回 plan + credits.total
3. config: faq/quotes
4. contact: 入库或邮件 (先入库即可)

验收标准

- 前端对应页面不再依赖硬编码数据; 失败有可解释错误。

B. 前端仓库 Prompts (只做展示与对接, 不做核心业务)

【前端仓库 Prompt B1】API Client 基建 (统一请求、错误、鉴权、类型)

目标: 建立 src/api/, 让所有请求集中管理, 避免散落导致对齐失败。

要做的改动

1. 新建 src/api/client.ts: 统一 baseUrl、超时、错误结构、自动带 token
2. src/api/modules/*: auth/scan/billing/config/contact
3. 页面与 store 中的 fetch 全部迁移

验收标准

- 业务代码中不再出现散落 fetch。

- 401/403/5xx 统一行为一致。

【前端仓库 Prompt B2】Auth 去 Mock: login/register/me 全链路

目标: 替换本地 seed 用户逻辑; 以 /api/auth/me 为用户事实来源。

前端清单

要做的改动

1. auth store:
 - login → api.auth.login() 保存 token
 - 随后调用 api.auth.me() 填充 user/credits
2. 删除/隔离 mock: staticUsers/seedHistoryRecords

验收标准

- 刷新页面 token 有效则恢复用户, 否则回登录。
- 不再出现本地生成用户状态的逻辑。

【前端仓库 Prompt B3】Scan 去 simulateAnalysis: 只渲染后端 AnalysisResponse

目标: Scan 页面只渲染后端返回, 不再本地“生成检测结果”。

后端适配与架构决策方案

要做的改动

scan store: submit(text, functions) 调 POST /api/scan/detect

前端清单

UI 渲染: summary/sentences/polish/translation/citations 均来自 response

前端清单

1. Loading: 真实 loading; 若后端异步则实现轮询 (按契约)

验收标准

- simulateAnalysis 不参与生产逻辑 (删除或 dev-only) 。
- 同一文本检测结果只可能来自后端。

【前端仓库 Prompt B4】MultiUpload: 上传→后端 parse-files→合并→detect

目标: 前端不解析 pdf/docx; 统一使用 /api/scan/parse-files。

前端清单

要做的改动

1. MultiUploadPage: 上传 files → parse-files
2. 合并 results[].content → detect
3. 错误处理: 明确提示哪个文件失败

验收标准

- pdf/docx 上传后可进入 detect; 前端不依赖解析库。

【前端仓库 Prompt B5】History: 从后端拉取历史列表与详情 (替换 LocalStorage)

目标: 使用 /api/scan/history 与 /api/scan/history/{id} 做历史模块。

前端清单

要做的改动

1. history store: 分页拉取 list + total
2. 点击历史项 → 拉详情 → 回显到 Scan UI (复用 AnalysisResponse 渲染组件)

验收标准

- 左侧历史来自后端; 刷新后仍存在。
- 详情回显字段与 detect 一致。

【前端仓库 Prompt B6】Billing/Config/Contact: 真实请求 + 统一降级

目标: 订阅/配置/联系页改为真实 API; 失败态统一可重试。

前端清单

要做的改动

1. billing: orders/pay/subscriptions preview
2. config: faq/quotes
3. contact: 提交表单
4. UI: 统一错误态组件

验收标准

- 页面网络请求真实发生；失败可解释且可重试。

C. 对齐自动化 Prompts（分别在前端/后端仓库实现）

【前端仓库 Prompt D1】 从后端 OpenAPI 自动生成 TS Types（杜绝猜字段）

目标：任何契约变更自动触发前端类型报错。

要做的改动

1. 增加脚本 `pnpm gen:api`：从后端 `/openapi.json` 生成 types（`openapi-typescript`）
2. 输出到 `src/api/generated/`
3. 业务 types 改为 `re-export generated`

验收标准

- 改 OpenAPI 字段名，前端编译立刻报错。

【后端仓库 Prompt D2】Contract Tests：响应必须符合 OpenAPI

目标：后端返回不符合契约时测试失败，避免线上才发现对不齐。

要做的改动

1. 新增测试：对关键接口（`me/detect/history/parse-files`）做 schema 校验
2. CI 必须运行

验收标准

- 漏字段/类型不对 → test fail。

【前端仓库 Prompt D3】E2E（Playwright/Cypress）主链路门禁

目标：固定主链路，任何接口跑偏都会立刻失败。

要做的改动

1. E2E：注册→登录→detect→history→详情→订阅预览
2. CI 集成（可用 `docker-compose` 启服务）

验收标准

- 主链路一旦接口变动，E2E 第一时间暴露。