PDF 沉浸式翻译

技术说明

低侵入集成 · 语义级分段 · 表格结构识别 · 本地缓存复用

Foxit PDF SDK　Vue 3 + TypeScript　IndexedDB Cache

技术摘要

本方案在福昕 PDF 在线预览能力之上，通过独立 Adapter 抽取文本与坐标，以右侧译文轨道、原文高亮和连接线提供双语对照阅读体验。整体架构强调低侵入、可维护、可扩展与稳定阅读体验。

生成日期：2026-06-04

目录

技术定位

PDF 沉浸式翻译是在既有福昕 PDF 在线预览能力之上构建的阅读增强模块。系统不改变 PDF 原文内容，不重排 PDF 页面，不替换福昕渲染链路，而是通过独立的前端适配层抽取文本与坐标，再以右侧译文轨道、原文高亮和连接线的方式提供双语对照阅读体验。

该方案的核心思路是“低侵入集成、结构化抽取、语义级分段、异步翻译、本地缓存、浮层式呈现”。它既保留原 PDF 预览和批注能力，又为工程类 PDF、技术规范、设计文件和表格密集文档提供更稳定的翻译阅读体验。

> **设计原则**
>
> 翻译能力作为阅读增强层存在，不侵入 PDF 原始渲染链路；所有文本抽取、分段、翻译和展示都在独立模块内完成，确保 PDF 原始功能稳定可用。

总体架构

系统架构将 PDF 引擎、翻译调度、缓存策略和 UI 展示解耦。福昕相关逻辑集中在 Adapter 层，翻译队列只依赖统一的 `PdfAdapter` 接口。后续替换 PDF 引擎或调整翻译后端时，不需要大规模改动 UI 和业务调度代码。

```
PDF 预览页面
  -> 福昕 Foxit PDF SDK 渲染 PDF
  -> Foxit Adapter 抽取文本、页码、坐标和上下文
  -> 前端 NLP 分段与表格结构识别
  -> IndexedDB 本地缓存查询
  -> 翻译调度队列
  -> 同源后端翻译接口
  -> 百炼/模型服务
  -> 译文缓存写入
  -> 右侧沉浸式译文浮层渲染
```

架构分层

表现层

负责右侧译文轨道、译文卡片、原文高亮、连接线、加载态、错误态和用户偏好展示。

调度层

负责滚动扫描、防抖、缓冲区管理、队列并发、请求取消、重试和状态发布。

语义层

负责自然段续行合并、句子边界判断、表格结构识别、标题和列表保护。

适配层

负责对接 Foxit PDF SDK，完成文本抽取、页码识别、坐标转换和锚点刷新。

技术栈

类型	技术选型	作用
前端框架	Vue 3、Composition API、TypeScript	构建响应式翻译浮层、队列状态和用户偏好。
PDF 引擎	Foxit PDF SDK for Web	提供 PDF 在线预览、文本层、坐标体系和批注能力。
UI 集成	Ant Design Vue、既有 ADF Vue 框架	复用项目既有按钮、布局和页面结构。
文本处理	前端轻量 NLP、句子边界识别、规则化表格分析	优化自然段、列表、标题、表格单元格和技术编号分段。
异步控制	AbortController、滚动防抖、队列并发控制、退避重试	保证滚动顺滑、请求可取消、模型调用稳定。
本地缓存	IndexedDB、稳定 hash、TTL、LRU 裁剪	复用成功译文，减少重复请求和等待时间。
后端接口	同源 `POST /idoc/ai/pdf/translate`	前端统一调用入口，隐藏模型密钥与外部服务细节。
模型服务	阿里云百炼兼容接口或翻译模型	提供段落级翻译能力。

核心技术点

低侵入福昕集成

沉浸式翻译没有改造福昕 SDK 内部实现，也没有替换 PDF 渲染流程。系统在 PDF 渲染完成后创建独立 Adapter，通过福昕实例读取文本层、页码和页面坐标，再将结果转换为业务可用的标准段落结构。

技术优势

- 保留福昕原生预览、下载、打印、缩放、全屏和批注能力。
- 翻译能力可以独立开启或关闭。
- PDF 引擎相关耦合集中，后续维护边界清晰。

文本坐标标准化

PDF 文本坐标、浏览器视口坐标和浮层坐标属于不同坐标体系。模块在 Adapter 层完成坐标映射，将 PDF 内部矩形转换为相对 `.fuxin-container` 的浮层坐标。

系统维护以下关键坐标：

- 原文文本矩形：用于 hover 热区和高亮框。
- 段落锚点：用于译文卡片垂直定位。
- 页面右边界：用于计算译文轨道和连接线。
- 视口可见性：用于区分当前可见段落和缓冲区段落。

技术优势

- 原文高亮与译文卡片位置稳定对应。
- 滚动、缩放、窗口变化后可重新刷新锚点。
- UI 浮层不干扰 PDF 原文渲染。

前端语义分段

PDF 文本抽取往往不是天然段落结构，尤其在表格、双栏、列表、页眉页脚和跨行文本中容易出现断句或串段。系统在前端增加轻量 NLP 与规则化结构分析，对抽取结果进行语义级整理。

策略	说明
自然段续行合并	对同列、间距接近、未到句末的连续文本进行合并。
句子边界识别	保护英文缩写、小节号、编号和俄文常见缩写。
标题保护	章节标题、短标题、编号标题不与正文误合并。

列表保护	列表项、冒号后的说明块保持独立。
长段拆分	长文本优先在完整句子边界处拆分，避免请求过长。

分段收益

译文单位更贴近阅读语义，而不是机械文本行；可以减少“一句话拆成两张卡片”和“两个语义段落合成一个翻译块”的问题。

表格结构识别与单元格优先策略

工程 PDF 中大量内容以表格形式存在。表格抽取常见问题是同一单元格被拆成多行，或整行多个单元格被融合为一段。系统针对表格场景做了专门处理：

- 识别常见表头组合，例如 `Reference / Title`。
- 根据列边界、文本位置和编号模式拆解融合行。
- 同一单元格的多行文本优先合并为一个语义块。
- 跨单元格关系作为第二优先级处理，避免错误串联。
- 标准号、规范号、图纸号保持独立。

原文结构	分段结果
`API SPEC 5CRA`	独立编号段，原文显示。
`Specification for Corrosion-resistant Alloy Seamless Tubes for Use as Casing, Tubing, and Coupling Stock`	同一标题单元格合并为一个翻译块。
`API SPEC 5CT`	独立编号段，原文显示。
`Casing and Tubing`	独立标题段。
`API TR 5C3`	独立编号段，原文显示。

技术编号原文直出

系统对强技术标识符进行识别，例如：

```
API SPEC 5CT
API TR 5C3
CR-OT-DW-410
GS-OT-COR-004
```

这类内容属于标准号、图纸号、规范号或文件编号，不适合交给模型自由翻译。系统会直接以原文作为译文显示，不进入模型调用流程。

技术优势

- 避免 `API SPEC` 被误译为“API 规范”。
- 避免编号大小写、连字符和顺序被模型改写。

- 减少不必要的模型调用。

异步翻译队列

翻译调度采用队列化设计。系统优先处理当前可见段落，其次处理上下缓冲区段落。默认低并发执行，并设置请求启动间隔、退避重试和滚动防抖。

关键机制：

- 滚动停止后再重新扫描文本，减少滚动中访问 PDF SDK 的频率。
- 当前视口上下预扫缓冲区，提升继续阅读时的译文出现速度。
- 关闭翻译、切换文件、离开缓冲区时取消未完成请求。
- 网络波动或限流时进行有限重试。

> **体验收益**
>
> 翻译任务在后台渐进执行，不阻塞 PDF 阅读主流程；无效请求可以及时取消，减少资源浪费。

IndexedDB 本地持久缓存

系统使用 IndexedDB 保存成功译文。缓存 key 不依赖页码、坐标或 DOM ID，而是由规范化原文、上下文、语言和翻译策略生成。

缓存策略	说明
写入条件	仅缓存成功且非空译文。
排除内容	不缓存失败、超时、取消或错误态结果。
有效期	默认 90 天。
容量控制	最大记录数 12000 条，超出后按最久未访问记录裁剪。
策略隔离	切换模型、prompt 或术语策略时可通过缓存 profile 隔离旧结果。

优势：

- 同一文档重复打开时译文展示更快。
- 坐标变化或滚动刷新不会影响缓存命中。
- 本地缓存不会无界增长。
- 失败结果不会污染后续翻译。

译文浮层与交互联动

译文 UI 采用独立浮层轨道，不改变 PDF 页面内容。卡片按原文锚点位置排列，并支持 hover、focus、click 多种激活方式。

交互特性：

- 卡片激活时显示原文高亮。
- 卡片与原文之间显示连接线。
- 鼠标移动到原文热区时反向激活卡片。
- 非激活卡片自动弱化。
- 字号和背景可即时调整。

安全设计

系统遵循前后端密钥隔离原则：

- 前端不保存百炼 API Key。
- 前端不直接请求百炼公网地址。
- 前端只调用同源翻译接口。
- 模型密钥、模型 ID、Base URL、prompt 和限流策略由后端管理。
- 关闭翻译或切换文件时取消未完成请求。
- IndexedDB 缓存保存译文和 hash 化 key，不保存原文全文。

该设计便于统一鉴权、审计、限流、脱敏和模型策略管理。

性能设计

场景	优化策略
PDF 滚动	滚动防抖，停止后扫描。
视口翻译	可见段落优先，上下缓冲区预加载。
模型调用	低并发、启动间隔、退避重试。
请求生命周期	AbortController 取消无效请求。
内存使用	仅保留缓冲区条目，离开后裁剪。
重复阅读	IndexedDB 命中后直接展示。
样式调整	字号和背景纯前端生效，不重新翻译。

整体目标是让翻译能力在后台渐进加载，不阻塞 PDF 阅读主流程。

可扩展性

PDF 引擎可替换

翻译调度只依赖 `PdfAdapter` 接口。如需支持 PDF.js 或其他 PDF 引擎，可新增 Adapter 实现。

翻译服务可替换

前端只依赖统一的 `translatePdfParagraph` 方法。后端可根据环境切换百炼模型、翻译模型或企业内部模型。

分段策略可增强

表格规则、标题识别、术语保护和语言识别可以继续独立演进。

缓存策略可灰度

可通过缓存 profile 隔离不同模型、prompt 或术语策略产生的译文。

技术优势总结

优势	说明
低侵入	不重构福昕预览主链路，沉浸翻译作为独立能力挂载，风险边界清晰。
强对照	通过右侧卡片、原文高亮和连接线实现段落级双语对应，比普通全文翻译更适合 PDF 阅读。
分段准确	前端 NLP 与表格规则结合，针对工程 PDF 的表格、编号和跨行文本做专门优化。
调用可控	可见优先、低并发、请求取消和 IndexedDB 缓存共同减少无效调用和重复调用。
安全可控	前端不暴露模型密钥，翻译服务通过同源后端统一管理。
可维护	Adapter、队列、缓存、UI、接口封装边界明确，后续替换模型或扩展 PDF 引擎成本较低。
体验稳定	翻译失败不影响 PDF 阅读，关闭或切换文件会清理状态，滚动和缩放后可自动刷新锚点。

结论

PDF 沉浸式翻译方案以较低集成成本补充了 PDF 阅读中的双语对照能力。其优势不只在于调用翻译模型，而在于围绕 PDF 文本抽取、语义分段、表格结构、坐标联动和缓存复用形成了完整工程化闭环。