# AI文档管理系统 (aidocs) - 项目说明文档

## 📋 项目概述

AI文档管理系统是一个智能化的个人文档管理平台，集成了AI技术来提升文档处理和管理的效率。系统采用前后端分离架构，支持桌面GUI应用和Web API服务。

### 🎯 核心功能

* Markdown转Word: 高质量的文档格式转换
* AI文档分析: 智能内容分析和建议
* 文档管理: 分类、标签、搜索等完整管理功能
* 桌面客户端: PyQt6图形界面应用
* API服务: FastAPI后端服务

## 🏗️ 项目架构

### 📁 目录结构分析

src/

├── api/ # FastAPI后端API服务

│ ├── main.py # FastAPI主应用入口

│ ├── models/ # 数据模型

│ ├── routers/ # API路由模块

│ ├── schemas/ # Pydantic数据验证模式

│ └── dependencies/ # 依赖注入

├── core/ # 核心功能模块

│ ├── ai\_service.py # AI服务（DeepSeek集成）

│ ├── database.py # 数据库连接和配置

│ ├── word\_parser.py # Word文档解析器

│ ├── enhanced\_word\_parser.py # 增强版Word解析器

│ └── template\_manager.py # 模板管理器

├── gui/ # PyQt6桌面客户端

│ ├── main\_window.py # 主窗口

│ ├── document\_editor.py # 文档编辑器

│ ├── document\_outline\_navigator.py # 大纲导航器

│ ├── template\_manager\_gui.py # 模板管理界面

│ ├── widgets/ # 自定义组件

│ ├── dialogs/ # 对话框

│ └── resources/ # 资源文件

├── md2doc/ # Markdown转Word核心引擎

│ ├── core/ # 核心转换逻辑

│ ├── engines/ # 图表和图形引擎

│ ├── templates/ # 文档模板

│ ├── utils/ # 工具函数

│ └── cli/ # 命令行接口

├── templates/ # 文档模板

├── utils/ # 通用工具

└── data/ # 数据文件

## 🔧 技术栈

### 后端技术

* FastAPI: 现代Python Web框架，支持异步处理
* SQLAlchemy 2.0: ORM框架，支持异步数据库操作
* Pydantic: 数据验证和序列化
* SQLite/PostgreSQL: 数据存储

### 前端技术

* PyQt6: 跨平台桌面GUI框架
* Qt Designer: 界面设计工具

### AI服务

* DeepSeek API: 大语言模型服务
* OpenAI兼容接口: 支持多种AI模型

### 文档处理

* python-docx: Word文档操作
* markdown: Markdown解析
* PlantUML: UML图表生成
* Mermaid: 流程图和图表

## 🚀 核心模块详解

### 1. API服务模块 (src/api/)

#### 主要组件

* main.py: FastAPI应用入口，包含中间件、路由注册、异常处理
* routers/: RESTful API路由
* documents.py: 文档CRUD操作
* categories.py: 分类管理
* tags.py: 标签管理
* templates.py: 模板管理
* search.py: 搜索功能

#### 功能特性

* 异步请求处理
* CORS跨域支持
* 全局异常处理
* 数据库连接池
* API文档自动生成

### 2. 桌面客户端 (src/gui/)

#### 主要组件

* main\_window.py: 主窗口，集成所有功能模块
* document\_editor.py: 富文本编辑器
* document\_outline\_navigator.py: 文档大纲树形导航
* template\_manager\_gui.py: 模板管理界面

#### 功能特性

* 多标签页文档编辑
* 实时大纲导航
* 模板管理
* API服务集成
* 文件拖拽支持

### 3. AI服务模块 (src/core/ai\_service.py)

#### 功能特性

* 文档分析: 自动生成文档摘要和关键词
* 大纲生成: 智能生成文档结构
* 内容建议: 基于上下文的写作建议
* 文档分类: 自动分类和标签推荐

#### DeepSeek集成

语言: python

class AIService:

def \_\_init\_\_(self, api\_key: Optional[str] = None):

self.api\_key = api\_key or os.getenv('DEEPSEEK\_API\_KEY')

self.base\_url = "https://api.deepseek.com/v1"

self.model = "deepseek-chat"

### 4. Markdown转Word引擎 (src/md2doc/)

#### 核心组件

* converter.py: 主转换器，统一转换接口
* parser.py: Markdown解析器
* generator.py: Word文档生成器
* engines/: 图表和图形处理引擎

#### 支持特性

* 标准Markdown语法
* 表格和列表
* 代码块语法高亮
* 图片和链接
* PlantUML/Mermaid图表
* 自定义样式模板

### 5. 数据库模块 (src/core/database.py)

#### 特性

* SQLAlchemy 2.0异步支持
* 连接池管理
* 自动重连机制
* 开发/生产环境配置

## 📊 数据模型

### 文档模型

语言: python

# 文档实体

- id: 主键

- title: 标题

- content: 内容

- file\_path: 文件路径

- category\_id: 分类ID

- tags: 标签列表

- created\_at: 创建时间

- updated\_at: 更新时间

### 分类和标签

语言: python

# 分类层次结构

- parent\_id: 父分类

- depth: 层级深度

- path: 路径

# 标签系统

- name: 标签名

- color: 颜色

- usage\_count: 使用次数

## 🔌 API接口

### 文档管理

* GET /api/v1/documents: 获取文档列表
* POST /api/v1/documents: 创建文档
* PUT /api/v1/documents/{id}: 更新文档
* DELETE /api/v1/documents/{id}: 删除文档

### 搜索功能

* GET /api/v1/search/documents: 文档搜索
* GET /api/v1/search/suggest: 搜索建议

### AI功能

* POST /api/v1/ai/analyze: 文档分析
* POST /api/v1/ai/outline: 大纲生成
* POST /api/v1/ai/suggest: 内容建议

## 🛠️ 配置管理

### 应用配置 (config/settings.py)

语言: python

class Settings(BaseSettings):

# 应用基础配置

app\_name: str = "AI文档管理系统"

app\_version: str = "1.0.0"

debug: bool = True

# FastAPI 配置

api\_host: str = "127.0.0.1"

api\_port: int = 8000

cors\_origins: List[str] = ["\*"]

# 数据库配置

database\_url: str = "sqlite:///./data/database/aidocs.db"

async\_database\_url: str = "sqlite+aiosqlite:///./data/database/aidocs.db"

# DeepSeek AI 配置

deepseek\_api\_key: Optional[str] = None

deepseek\_api\_base: str = "https://api.deepseek.com/v1"

deepseek\_model: str = "deepseek-chat"

### 环境变量 (.env)

语言: bash

# AI服务配置

DEEPSEEK\_API\_KEY=your\_deepseek\_api\_key

# 数据库配置（可选，默认使用SQLite）

DATABASE\_URL=postgresql://user:pass@localhost/aidocs

# API服务配置

API\_HOST=127.0.0.1

API\_PORT=8000

DEBUG=true

# 日志级别

LOG\_LEVEL=INFO

### 专业维度配置

语言: python

PROFESSIONAL\_DIMENSIONS = {

"人工智能": {

"subcategories": ["机器学习", "深度学习", "自然语言处理", "计算机视觉"],

"color": "#FF6B6B",

"icon": "🤖"

},

"项目管理": {

"subcategories": ["敏捷开发", "需求管理", "风险管理", "团队协作"],

"color": "#4ECDC4",

"icon": "📊"

},

"DevOps": {

"subcategories": ["持续集成", "监控运维", "质量工程", "容器化"],

"color": "#45B7D1",

"icon": "🔧"

}

}

### 时间维度配置

语言: python

TIME\_DIMENSIONS = {

"year": {"format": "%Y", "display": "年度"},

"month": {"format": "%Y-%m", "display": "月度"},

"week": {"format": "%Y-W%W", "display": "周度"},

"day": {"format": "%Y-%m-%d", "display": "日期"}

}

## 🚀 启动方式

### 1. 环境准备

语言: bash

# 激活虚拟环境

cd d:\Programs\aidocs

venv\Scripts\Activate.ps1

# 安装依赖（如果需要）

pip install -r requirements.txt

# 设置DeepSeek API密钥

set DEEPSEEK\_API\_KEY=your\_api\_key\_here

### 2. API服务启动

语言: bash

# 方式1：直接运行主文件

python src/api/main.py

# 方式2：使用uvicorn启动

uvicorn src.api.main:app --host 127.0.0.1 --port 8000 --reload

# 服务启动后访问:

# API文档: http://localhost:8000/docs

# 健康检查: http://localhost:8000/health

### 3. 桌面客户端启动

语言: bash

# 启动GUI应用（会自动启动API服务）

python src/gui/main\_window.py

### 4. 命令行工具使用

语言: bash

# Markdown转Word（单文件）

python md2docx\_optimized.py input.md -o output.docx

# 批量转换

python md2docx\_optimized.py \*.md --output-dir ./output/

# 使用自定义模板

python md2docx\_optimized.py input.md -t custom\_template.docx

### 5. VS Code任务（已配置）

* MD转DOCX：当前文件: 转换当前打开的Markdown文件
* MD转DOCX：选择文件转换: 选择文件进行转换

## 📦 依赖包

### GUI桌面客户端

* PyQt6==6.6.1: 跨平台GUI框架
* PyQt6-Qt6==6.6.1: Qt6核心库
* PyQt6-sip==13.6.0: Python-Qt绑定

### Web后端服务

* fastapi==0.104.1: 现代异步Web框架
* uvicorn[standard]==0.24.0: ASGI服务器
* pydantic==2.5.0: 数据验证和序列化
* pydantic-settings==2.1.0: 配置管理

### 数据库支持

* sqlalchemy==2.0.23: ORM框架（支持异步）
* alembic==1.13.1: 数据库迁移工具
* aiosqlite==0.19.0: SQLite异步支持
* psycopg2-binary==2.9.9: PostgreSQL驱动

### AI服务集成

* openai==1.3.7: AI API客户端（兼容DeepSeek）
* httpx==0.25.2: 异步HTTP客户端
* tiktoken==0.5.2: Token计算工具

### 文档处理

* python-docx==0.8.11: Word文档读写
* docx2txt==0.8: Word文本提取
* Markdown==3.5.1: Markdown解析
* lxml==4.9.3: XML解析增强

### 搜索和索引

* whoosh==2.7.4: 全文检索引擎
* jieba==0.42.1: 中文分词工具

### 工具库

* Pillow==10.1.0: 图像处理
* PyYAML==6.0.1: YAML配置文件支持
* python-dotenv==1.0.0: 环境变量管理
* loguru==0.7.2: 高级日志系统
* python-dateutil==2.8.2: 日期时间处理

### 开发工具

* pytest==7.4.3: 测试框架
* pytest-asyncio==0.21.1: 异步测试支持
* black==23.11.0: 代码格式化
* isort==5.12.0: 导入语句排序
* flake8==6.1.0: 代码检查

## 🔄 工作流程

### 文档处理流程

1. 文档导入: 支持MD/DOCX/TXT格式
2. AI分析: 自动提取摘要、关键词、分类
3. 存储管理: 分类存储，标签标记
4. 搜索检索: 全文搜索，语义搜索
5. 格式转换: MD↔DOCX互转

### AI增强流程

1. 内容分析: 文档结构分析
2. 智能建议: 写作改进建议
3. 自动分类: 基于内容的智能分类
4. 关联推荐: 相关文档推荐

## 🎯 使用场景

### 个人文档管理

* 知识库构建
* 笔记整理
* 学习资料管理

### 团队协作

* 文档标准化
* 模板共享
* 版本管理

### 内容创作

* 写作辅助
* 格式转换
* 自动排版

## 🔧 扩展性

### 插件系统

* 自定义AI模型接口
* 第三方服务集成（如GitHub、Notion）
* 格式转换器扩展（支持更多文档格式）
* 主题和样式定制

### 部署选项

* 单机部署: 直接运行Python应用
* Docker容器: 容器化部署（配置文件已准备）
* 云服务部署: 支持AWS、Azure、腾讯云等

## 🐛 故障排除

### 常见问题

#### 1. API服务启动失败

语言: bash

# 检查端口占用

netstat -ano | findstr :8000

# 检查数据库连接

python -c "from src.core.database import test\_async\_db\_connection; print(test\_async\_db\_connection())"

#### 2. DeepSeek API调用失败

语言: bash

# 检查API密钥设置

echo $DEEPSEEK\_API\_KEY

# 测试API连接

python -c "from src.core.ai\_service import AIService; ai = AIService(); print(ai.is\_available())"

#### 3. PyQt6界面显示异常

* 确保安装了正确版本的PyQt6
* 检查系统字体和主题设置
* 查看控制台错误信息

#### 4. 文档转换失败

* 检查输入文件格式是否支持
* 确认Word模板文件存在
* 查看转换日志输出

### 日志调试

语言: bash

# 查看应用日志

tail -f logs/aidocs.log

# 设置调试模式

export DEBUG=true

## 👥 开发指南

### 代码结构原则

* 模块化设计: 每个功能模块独立开发
* 接口分离: GUI、API、核心逻辑分离
* 配置驱动: 通过配置文件控制行为
* 异步优先: 所有IO操作使用异步模式

### 开发环境设置

语言: bash

# 安装开发依赖

pip install -r requirements.txt

# 代码格式化

black src/

isort src/

# 代码检查

flake8 src/

# 运行测试

pytest tests/

### 新功能开发流程

1. 需求分析: 明确功能需求和接口设计
2. 模块开发: 按模块独立开发和测试
3. 接口集成: 集成到API和GUI中
4. 测试验证: 单元测试和集成测试
5. 文档更新: 更新API文档和用户文档

### 贡献指南

* Fork项目并创建特性分支
* 遵循代码规范和测试要求
* 提交Pull Request并描述变更内容
* 等待代码审查和合并

## 📈 项目路线图

### 已完成功能 ✅

* [x] Markdown到Word高质量转换
* [x] PyQt6桌面客户端
* [x] FastAPI后端服务
* [x] DeepSeek AI集成
* [x] 基础文档管理功能
* [x] 双维度索引系统

### 进行中功能 🚧

* [ ] 全文检索优化
* [ ] 批量文档处理
* [ ] 模板管理增强
* [ ] 用户权限管理

### 计划功能 📅

* [ ] Web前端界面
* [ ] 多人协作功能
* [ ] 云端同步支持
* [ ] 移动端适配
* [ ] 插件市场
* [ ] 企业版功能

---

本文档最后更新：2025年9月12日

项目维护者：AI文档工作空间团队

如有问题请提交Issue或联系开发团队