Click 数据契约文档

总体流程

```
epub文件 → 技术预处理 → 统一工作流(联网搜索+章节处理) → 数据库存储 → 前端渲染
```

阶段1：技术预处理

输入

- epub文件（来自books表的epub_url）

处理

- 解析epub为纯文本
- 按章节分割
- 生成chapter JSON文件

输出

```
{
  "book_id": "123",
  "chapters": [
    {
      "chapter_index": 1,
      "title": "第一章 初见",
      "content": "那是一个寒冷的冬日...",
      "word_count": 1500,
      "paragraphs": [
        {
          "paragraph_index": 0,
          "text": "那是一个寒冷的冬日，雪花纷纷扬扬地落下...",
          "char_start": 0,
          "char_end": 67
        }
      ]
    }
  ]
}
```

阶段2：统一工作流 - 联网搜索+章节处理

输入

```
{
  "book_id": "123",
  "metadata": {
    "title": "书名",
    "author": "作者",
    "genre": "科幻",
    "publication_year": "2023"
  },
  "chapter_data": {
    "chapter_index": 1,
    "title": "第一章 初见",
    "content": "章节完整文本",
    "paragraphs": ["段落数组"]
  }
}
```

工作流内部处理

1. **联网搜索阶段**:
    - 搜索书籍背景信息、评论、人物分析
    - 获取同类型书籍的风格参考
    - 生成提示词模板和摘抄规则
2. **章节分析阶段**:
    - 基于搜索结果分析当前章节
    - 应用摘抄规则选择圆点位置
    - 生成符合书籍风格的图片和音频

输出

```
{
  "book_id": "123",
  "chapter_index": 1,
  "book_context": {
    "style_tags": ["现代", "写实", "都市"],
    "main_characters": [
      {
        "name": "李明",
        "description": "男主角，程序员，内向",
        "appearance": "中等身材，戴眼镜"
      }
    ],
    "settings": [
      {
        "name": "咖啡馆",
        "description": "温暖的室内环境，现代装修"
      }
    ],
```

```
    "image_style_template": "现代都市风格，写实画风，暖色调，电影感构图，
{scene_description}",
    "audio_style_template": "现代都市环境音，{scene_type}氛围，轻柔背景音乐",
    "extraction_rules": {
      "scene_keywords": ["环境描述", "人物动作", "情感转折"],
      "min_text_length": 10,
      "max_text_length": 50,
      "priority_types": ["场景描述", "人物互动", "情节转折"]
    }
  },
  "hotspots": [
    {
      "hotspot_id": "uuid-1",
      "paragraph_index": 2,
      "char_start": 45,
      "char_end": 67,
      "selected_text": "雪花纷飞的咖啡馆",
      "scene_type": "环境描述",
      "image_prompt": "冬日咖啡馆，雪花飞舞，温暖室内，现代都市风格，写实画风，暖色调，
电影感构图",
      "image_url": "https://oss.../generated/book123_ch1_spot1.jpg",
      "audio_prompt": "现代都市环境音，咖啡馆氛围，轻柔背景音乐",
      "audio_url": "https://oss.../generated/book123_ch1_spot1.mp3",
      "confidence_score": 0.85
    }
  ],
  "search_results": {
    "book_summary": "网络搜索获取的书籍简介",
    "character_analysis": "人物分析结果",
    "style_references": ["参考作品1", "参考作品2"]
  },
  "processing_status": "completed",
  "error_log": []
}
```

数据库表结构

1. books（已存在）

```
CREATE TABLE books (
  id INT AUTO_INCREMENT PRIMARY KEY,
  title VARCHAR(255),
  author VARCHAR(255),
  language VARCHAR(2) DEFAULT 'zh',
  epub_url VARCHAR(2083) NOT NULL,
  cover_url MEDIUMTEXT,
  cover_data MEDIUMBLOB,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

2. book_contexts（新增）

```
CREATE TABLE book_contexts (
  id INT AUTO_INCREMENT PRIMARY KEY,
  book_id INT NOT NULL,
  style_tags JSON,
  main_characters JSON,
  settings JSON,
  image_style_template TEXT,
  audio_style_template TEXT,
  extraction_rules JSON,
  search_results JSON,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (book_id) REFERENCES books(id)
);
```

3. chapters（新增）

```
CREATE TABLE chapters (
  id INT AUTO_INCREMENT PRIMARY KEY,
  book_id INT NOT NULL,
  chapter_index INT NOT NULL,
  title VARCHAR(255),
  content_url VARCHAR(512) NOT NULL,  -- 指向OSS的JSON文件
  word_count INT,
  processing_status ENUM('pending', 'processing', 'completed', 'failed') DEFAULT
'pending',
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (book_id) REFERENCES books(id),
  UNIQUE KEY unique_book_chapter (book_id, chapter_index)
);
```

4. hotspots（新增）

```
CREATE TABLE hotspots (
  id INT AUTO_INCREMENT PRIMARY KEY,
  hotspot_id VARCHAR(36) UNIQUE NOT NULL,  -- UUID
  book_id INT NOT NULL,
  chapter_id INT NOT NULL,
  paragraph_index INT NOT NULL,
  char_start INT NOT NULL,
  char_end INT NOT NULL,
  selected_text TEXT NOT NULL,
  scene_type VARCHAR(50),
  image_prompt TEXT,
  image_url VARCHAR(512),
  audio_prompt TEXT,
  audio_url VARCHAR(512),
```

```
  confidence_score DECIMAL(3,2),
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (book_id) REFERENCES books(id),
  FOREIGN KEY (chapter_id) REFERENCES chapters(id)
);
```

5. user_progress（新增）

```
CREATE TABLE user_progress (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  book_id INT NOT NULL,
  chapter_index INT NOT NULL,
  paragraph_index INT DEFAULT 0,
  percentage DECIMAL(5,2) DEFAULT 0.00,
  last_read_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (book_id) REFERENCES books(id),
  UNIQUE KEY unique_user_book (user_id, book_id)
);
```

6. ai_variations（新增）

```
CREATE TABLE ai_variations (
  id INT AUTO_INCREMENT PRIMARY KEY,
  hotspot_id VARCHAR(36) NOT NULL,
  user_id INT NOT NULL,
  variation_prompt TEXT,
  image_url VARCHAR(512),
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (hotspot_id) REFERENCES hotspots(hotspot_id)
);
```

API接口设计

1. 提交统一工作流任务

```
POST /api/workflow/process-chapter
{
  "book_id": "123",
  "metadata": {
    "title": "书名",
    "author": "作者",
    "genre": "科幻",
    "publication_year": "2023"
  },
  "chapter_data": {
```

```
    "chapter_index": 1,
    "title": "第一章 初见",
    "content": "章节完整文本",
    "paragraphs": ["段落数组"]
  }
}
```

2. 查询处理状态

```
GET /api/workflow/status/{task_id}
```

3. 获取书籍上下文

```
GET /api/books/{book_id}/context
```

4. 获取章节圆点

```
GET /api/books/{book_id}/chapters/{chapter_index}/hotspots
```

5. 批量获取章节数据

```
GET /api/books/{book_id}/chapters/{chapter_index}/full
# 返回章节内容+圆点+书籍上下文的完整数据
```

6. 工作流回调接口

```
POST /api/workflow/book-context
POST /api/workflow/chapter-hotspots

接收工作流输出，存入对应数据表
```

7. 用户交互接口

```
POST /api/hotspots/{hotspot_id}/variations
GET /api/users/{user_id}/progress/{book_id}
PUT /api/users/{user_id}/progress/{book_id}
```

文件存储规则

OSS路径结构

```
/books/{book_id}/
  ├── chapters/
  │   ├── chapter_1.json
  │   ├── chapter_2.json
  │   └── ...
  ├── generated/
  │   ├── images/
  │   │   ├── ch1_spot1.jpg
  │   │   └── ...
  │   └── audio/
  │       ├── ch1_spot1.mp3
  │       └── ...
  └── variations/
      ├── user123_spot1_var1.jpg
      └── ...
```

错误处理

工作流失败处理

1. **书籍分析失败**：标记book_contexts状态为failed，使用默认模板
2. **章节处理失败**：标记chapters状态为failed，跳过该章节
3. **部分圆点生成失败**：记录error_log，继续处理其他圆点

重试机制

- 工作流失败后24小时内自动重试3次
- 超过重试次数后人工介入
- 提供手动重新处理接口

版本控制

数据版本

- book_contexts表增加version字段
- 工作流算法更新时，增量处理新版本
- 保留旧版本数据，用户可选择使用

API版本

- 使用/api/v1/前缀
- 向后兼容至少2个版本
- 废弃接口提前3个月通知