

AI绘本创作系统

企业后端集成指南 V4.0

V4.0 | 2026-04-06 | 数值状态驱动 + 5步创作状态机 + 取消dataVersion

| | |
|------|--|
| 文档版本 | V4.0 (替代V3.1) |
| 发布日期 | 2026-04-06 |
| 适用场景 | 企业融合乐读派C端 (H5+Android) 进行AI绘本创作 |
| 核心变更 | 取消dataVersion/completionStep, 改为纯数值status(1-5)驱动同步 |
| 创作流程 | A6角色提取 + A3图片创作 + 编目 + 配音, 5步状态机 |
| 同步机制 | 数值比较 (remote > local) + PROCESSING强制更新 + Webhook + B3兜底 |
| 认证方式 | Session Token (H5生产推荐) + HMAC-SHA256 (开发/Android) + config.js运行时配置 |

V3.1 → V4.0 主要变更

| 变更项 | V3.1 (旧) | V4.0 (新) |
|-----------|---------------------------------------|---|
| 状态字段 | status(VARCHAR) + completionStep(INT) | status(INT) 单字段, 值1-5 |
| 同步判断 | dataVersion 门卫 (remote > local) | status 数值比较 (remote > local) |
| 完成子状态 | completionStep: 0/1/2 | status: 3=图片完成, 4=编目完成, 5=配音完成 |
| Webhook事件 | 9种事件类型 | 2种: work.status_changed + work.progress |
| 终态定义 | COMPLETED + step=2 | status=5 (配音完成), 企业可从>=6自维护 |
| 编目/配音 | 可选环节 | 必须环节, 5步标准流程 |

目录

- 一、方案概述与架构
- 二、C端接入认证 (Session Token)
- 三、5步数值状态机 (核心)
- 四、企业同步规则
- 五、Webhook 回调详解
- 六、接口参考
- 七、企业端实现指南
- 八、H5 快速接入指南 (企业必读)
- 九、iframe 嵌入模式 (postMessage 通信)
- 十、附录：错误码与 FAQ

一、方案概述与架构

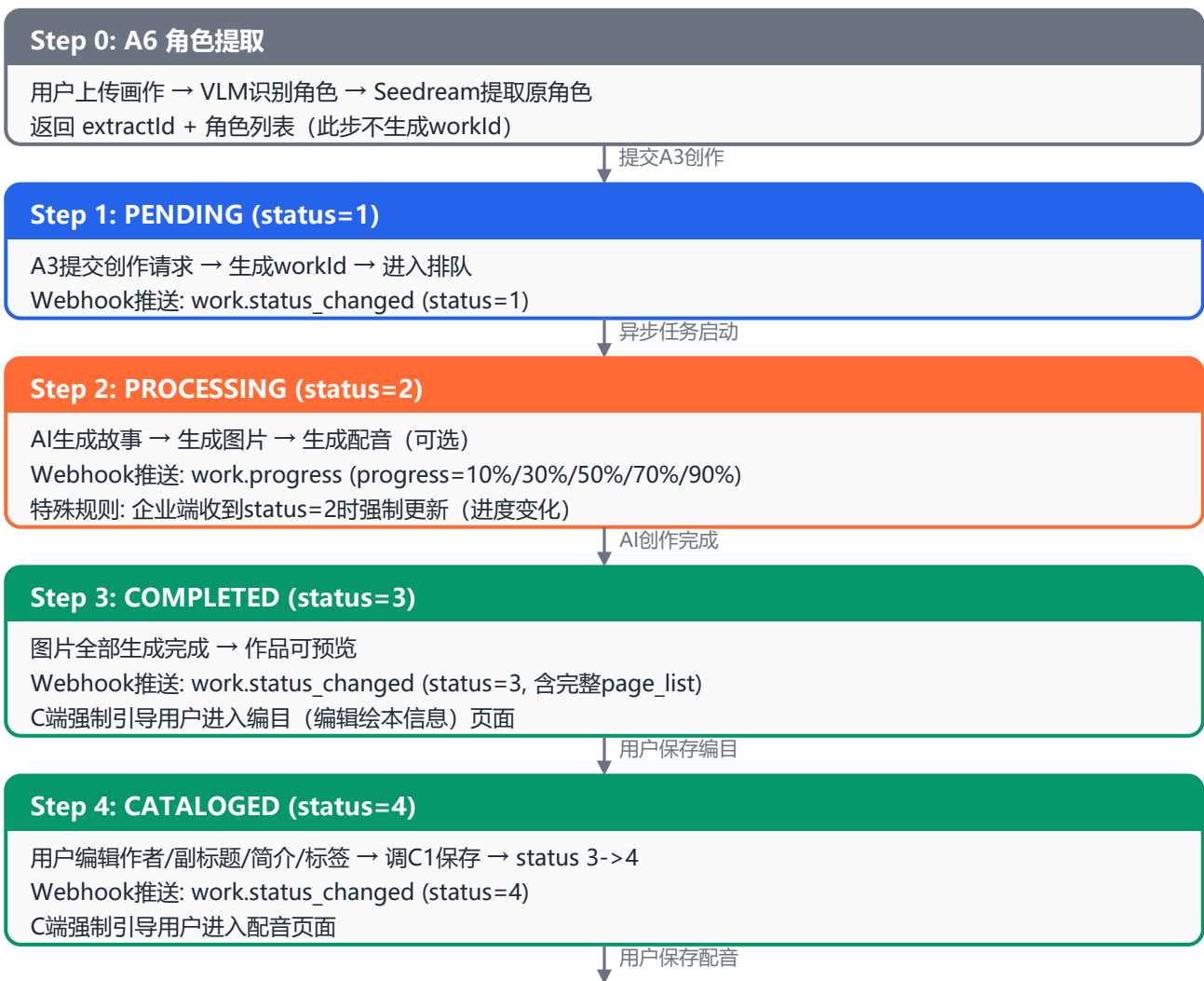
本方案适用于企业融合乐读派C端（H5网页端 + Android客户端）进行AI绘本创作的场景。乐读派提供完整的创作C端和后台服务，企业负责Webhook接收、数据同步和用户管理。

1.1 三层架构

| 层级 | 负责方 | 职责 |
|-----------------|-------|--------------------------------------|
| C端 (H5/Android) | 乐读派提供 | 用户创作交互、角色提取(A6)、图文创作(A3)、编目、配音 |
| 后台服务 | 乐读派提供 | AI创作引擎、状态管理、Webhook推送、OSS存储 |
| 企业后端 | 企业自建 | Webhook接收、数据同步、用户管理、二次编辑 (status>=6) |

1.2 创作时序 (C端网页端)

标准创作流程从A6角色提取开始，经A3图文创作、编目、配音，共5步状态流转：



Step 5: DUBBED (status=5) — 终态

用户完成配音（或跳过）→ 调C2保存 → status 4->5

Webhook推送: work.status_changed (status=5)

乐读派系统终态，后续二次编辑由企业自维护(status>=6)

二、C端接入认证 (Session Token)

2.1 双模式认证

| 维度 | HMAC模式 | Session Token模式 |
|-------------|-----------------|-----------------|
| appSecret位置 | 浏览器localStorage | 仅企业服务端 |
| 安全性 | 低 (可被F12查看) | 高 (不到达浏览器) |
| 适用场景 | 开发调试 / Android | H5生产环境 |
| 令牌有效期 | 无过期 | 2小时 |

2.2 Session Token 流程 (H5生产推荐)

Step 1: 企业后端用appSecret换取令牌

企业后端 (服务端, 非浏览器) 调用乐读派后端接口, 用机构密钥换取一个2小时有效的会话令牌。此步骤在服务端完成, appSecret绝不到达浏览器。

```
POST https://{乐读派API域名}/api/v1/auth/session
Content-Type: application/json
```

请求:

```
{ "orgId": "ORG001", "appSecret": "管理后台分配的密钥", "phone": "13800001111" }
```

响应:

```
{ "code": 200, "data": { "sessionToken": "sess_09fa122b...", "expiresIn": 7200 } }
```

phone 参数来自企业自身的用户系统 (如企业登录后的用户手机号)。orgId 和 appSecret 在管理后台「机构管理」页面获取。

Step 2: 企业后端重定向用户浏览器到H5

拿到 sessionToken 后, 企业后端通过 HTTP 302 将用户浏览器重定向到乐读派H5页面。token、orgId、phone 通过 URL 参数传入。

重要: H5使用 hash 路由模式, 参数必须放在 #/? 之后, 而非直接 ? 之后。

正确的重定向URL格式:

```
https://{H5域名}/?token=sess_xxx&orgId=ORG001&phone=13800001111
    ↑↑↑
    注意 #/? 三个字符
```

错误示例 (缺少 #/):

```
https://{H5域名}/?token=sess_xxx ← H5无法读取参数!
```

H5页面 (Welcome.vue) 在加载时自动从URL中提取这三个参数:

| 参数 | 必填 | 来源 | 说明 |
|-------|----|-------------|----------------|
| token | 是 | Step 1 接口返回 | 会话令牌 (2小时有效) |
| orgId | 是 | 管理后台分配 | 机构唯一标识 |
| phone | 是 | 企业用户系统 | 用户手机号 (关联创作数据) |

| 参数 | 必填 | 来源 | 说明 |
|------------|----|-----------|---------------------------------|
| returnPath | 否 | token失效回调 | H5之前的页面路径（如 /creating），用于恢复创作进度 |

Step 3: H5使用令牌调用API

H5读取URL参数后，将token存入浏览器 sessionStorage，后续所有API请求自动带上 Bearer Token 认证头：

```
Authorization: Bearer sess_09fa122b...
```

token 有效期为2小时。过期后H5会自动触发失效处理流程（见2.3节）。

重要： appSecret 仅在企业服务端使用，绝不可暴露到浏览器前端。token 是短期凭证，即使泄露也只影响单个用户的2小时会话。

2.3 Token 失效自动处理（闭环）

当token过期或无效时，H5端会自动执行以下闭环流程，无需用户手动干预：



前提： 企业必须在管理后台「机构管理」中配置「认证回调URL」。未配置时，token失效后H5只能显示“请从企业入口进入”的文字提示。

企业认证回调页需要做什么？

企业的认证回调页（即「认证回调URL」指向的页面）需要完成以下逻辑：

1. 从URL参数获取 returnPath 和 orgId
2. 验证用户身份（企业自身的登录/session机制）
3. 获取用户手机号（从企业用户系统）

4. 调用 POST /api/v1/auth/session 换取新的 sessionToken

5. 302重定向回H5:

`https://{H5域名}/?token={新token}&orgId={orgId}&phone={phone}&returnPath={原returnPath}`

`returnPath` 参数要原样传回H5，不要修改。H5会用它恢复到用户之前的创作页面。

2.4 企业对接完整代码示例

Java (Spring Boot) — 令牌交换 + H5跳转 + 认证回调

```
@RestController
public class LeaiController {
    // 管理后台分配的凭证
    private static final String ORG_ID = "你的机构ID";
    private static final String APP_SECRET = "你的机构密钥";
    private static final String LEAI_API = "https://你的API域名";
    private static final String LEAI_H5 = "https://你的H5域名";

    /**
     * 入口：用户点击"开始创作"后调用
     * 1. 从企业session获取用户手机号
     * 2. 调乐读派auth/session换token
     * 3. 302重定向到H5
     */
    @GetMapping("/create")
    public void create(HttpServletRequest req, HttpServletResponse resp) throws Exception {
        String phone = getPhoneFromSession(req); // 企业自身的用户手机号

        // Step 1: 换取token
        String token = exchangeToken(phone);

        // Step 2: 重定向到H5 (注意 #/? 格式)
        String url = LEAI_H5 + "?token=" + token
            + "&orgId=" + URLEncoder.encode(ORG_ID, "UTF-8")
            + "□=" + URLEncoder.encode(phone, "UTF-8");
        resp.sendRedirect(url);
    }

    /**
     * 认证回调：H5 token失效后跳转回来
     * URL: /leai-auth?returnPath=/creating&orgId=ORG001
     */
    @GetMapping("/leai-auth")
    public void authCallback(HttpServletRequest req, HttpServletResponse resp) throws Exception {
        String returnPath = req.getParameter("returnPath"); // 原样传回
        String phone = getPhoneFromSession(req);

        // 重新换token
        String newToken = exchangeToken(phone);

        // 重定向回H5, 带上returnPath
        String url = LEAI_H5 + "?token=" + newToken
            + "&orgId=" + URLEncoder.encode(ORG_ID, "UTF-8")
            + "□=" + URLEncoder.encode(phone, "UTF-8")
            + (returnPath != null ? "&returnPath=" + URLEncoder.encode(returnPath, "UTF-8") : "");
        resp.sendRedirect(url);
    }

    private String exchangeToken(String phone) throws Exception {
        // POST /api/v1/auth/session
        String body = String.format(
            "{\"orgId\":\"%s\",\"appSecret\":\"%s\",\"phone\":\"%s\"}",
            ORG_ID, APP_SECRET, phone);
        HttpURLConnection conn = (HttpURLConnection)
            new URL(LEAI_API + "/api/v1/auth/session").openConnection();
    }
}
```

```
conn.setRequestMethod("POST");
conn.setRequestProperty("Content-Type", "application/json");
conn.setDoOutput(true);
conn.getOutputStream().write(body.getBytes("UTF-8"));
// 解析响应中的 data.sessionToken
String resp = new String(conn.getInputStream().readAllBytes());
// 用你的JSON库解析, 返回 sessionToken
return parseSessionToken(resp);
}
}
```

以上为核心对接代码片段。完整可运行的 Spring Boot Demo (含Webhook接收+签名验证、B3定时对账、认证回调、HMAC签名工具等全部功能, 约600行Java代码) 请参考随附的 [enterprise-demo/java-demo/](#) 目录。替换顶部4个配置常量即可直接运行。

管理后台需要配置的2项

| 配置项 | 位置 | 填写内容 | 说明 |
|-------------|------|---------------------------|-------------------|
| Webhook URL | 机构管理 | https://你的域名/webhook/leai | 接收作品状态变更推送 |
| 认证回调URL | 机构管理 | https://你的域名/leai-auth | H5 token失效后跳回重新认证 |

三、5步数值状态机（核心）

V4.0的核心变更：将status字段从VARCHAR改为INT，用1-5的数值表示创作流程的5个阶段。企业端仅需要通过数值大小比较即可判断是否需要同步。

3.1 状态定义

| 数值 | 状态名 | 含义 | 企业端动作 |
|----|------------|-------------|--------------------|
| -1 | FAILED | 创作失败（异常状态） | 记录失败原因，不阻塞正常流程 |
| 1 | PENDING | 已提交，排队等待 | 创建本地记录 |
| 2 | PROCESSING | AI创作中（进度变化） | 强制更新进度（特殊规则） |
| 3 | COMPLETED | 图片完成，待编目 | 存储完整pageList，作品可展示 |
| 4 | CATALOGED | 编目完成，待配音 | 更新元数据（作者/标签等） |
| 5 | DUBBED | 配音完成（乐读派终态） | 标记完成，启用企业二次编辑 |

3.2 状态流转图

正常流程: [1:PENDING] → [2:PROCESSING] → [3:COMPLETED] → [4:CATALOGED] → [5:DUBBED]
异常流程: [1/2] → [-1:FAILED]
失败恢复: [-1:FAILED] → [1:PENDING]（重试）

3.3 关键规则

规则1: 严格单向递增。状态只能从小到大前进（1→2→3→4→5），不可回退。

规则2: 编目和配音必须经过。图片完成(3)后C端强制引导用户编目，编目保存后(4)强制引导配音。

规则3: 配音可跳过内容。用户点击"保存"即通过配音环节(4→5)，无论是否实际录音。

规则4: status=5是乐读派终态。达到5后，乐读派API拒绝任何修改。企业可在本地从status >= 6开始维护自己的二次编辑状态。

规则5: FAILED独立处理。FAILED(-1)不参与数值递增比较，通过Webhook强制推送，企业端收到后无条件处理。

V3.1对比: 原来需要同时判断 status + completionStep + dataVersion 三个字段，V4.0简化为仅判断 status 一个INT字段。企业端逻辑从~30行降至~10行。

四、企业同步规则

V4.0同步规则极其简洁：remote_status

>

local_status

则同步，辅以PROCESSING强制更新和FAILED特殊处理。

4.1 核心同步判断表

| 场景 | 条件 | 企业端操作 | 说明 |
|--------|--|----------|------------------------|
| 新作品 | 本地无记录 | INSERT | 首次收到该workId |
| 状态前进 | remote > local | 全量UPDATE | 主流程推进，覆盖所有字段 |
| 进度更新 | remote == 2 | 强制UPDATE | PROCESSING期间进度变化，无条件更新 |
| 创作失败 | remote == -1 | 强制UPDATE | FAILED通知，无条件更新 |
| 重复/旧数据 | remote <= local 且 remote != 2 且 remote != -1 | SKIP | 已处理过，直接忽略 |

4.2 企业端同步伪代码 (Java)

```
@Transactional
public void onWebhookReceived(JSONObject payload) {
    String workId = payload.getString("work_id");
    int remoteStatus = payload.getIntValue("status");

    WorkRecord local = workRecordMapper.selectById(workId);

    if (local == null) {
        // 新作品，直接入库
        workRecordMapper.insert(buildFromPayload(payload));
        return;
    }

    if (remoteStatus == -1) {
        // FAILED: 强制更新
        local.setStatus(-1);
        local.setFailReason(payload.getString("fail_reason"));
        workRecordMapper.updateById(local);
        return;
    }

    if (remoteStatus == 2) {
        // PROCESSING: 强制更新进度
        local.setStatus(2);
        local.setProgress(payload.getIntValue("progress"));
        workRecordMapper.updateById(local);
        return;
    }

    if (remoteStatus > local.getStatus()) {
        // 状态前进: 全量覆盖
        updateAllFields(local, payload);
        workRecordMapper.updateById(local);
    }
}
```

```
    return;
}

// remoteStatus <= local.status: 忽略旧数据
log.info("skip: remote={} <= local={}", remoteStatus, local.getStatus());
}
```

4.3 企业端状态扩展 (≥ 6 自维护)

乐读派系统的status最大值为5（配音完成）。企业可在本地扩展：

乐读派维护: 1 → 2 → 3 → 4 → 5 (终态)

企业自维护: 5 → 6 (改标题) → 7 (改简介) → 8 (改配音) → ...

规则: 企业本地status从5开始, 每次二次编辑自增。

乐读派API永远返回 $\text{status} \in \{-1, 1, 2, 3, 4, 5\}$ 。

企业 ≥ 6 的状态永远不回传给乐读派。

企业可用 $\text{status} \geq 5$ 作为"二次编辑功能可用"的判断条件。

五、Webhook 回调详解

5.1 事件类型（精简为2种）

| 事件 | 触发时机 | 频率 | 说明 |
|---------------------|-----------------------------------|----------|---------------------------|
| work.status_changed | 状态变更 (1→2, 2→3, 3→4, 4→5, any→-1) | 每次状态变更1次 | 核心事件, 携带完整作品数据 |
| work.progress | PROCESSING阶段进度变化 | 多次/作品 | 进度里程碑推送 (10/30/50/70/90%) |

5.2 Webhook Payload 格式

5.2.1 work.status_changed 事件

```
POST {企业webhook_url}
Content-Type: application/json
X-Webhook-Id: evt_190368671438289
X-Webhook-Event: work.status_changed
X-Webhook-Timestamp: 1712000000000
X-Webhook-Signature: HMAC-SHA256=a3f8c2d1...
```

```
{
  "id": "evt_190368671438289",
  "event": "work.status_changed",
  "created_at": 1712000000000,
  "data": {
    "work_id": "1903686714382889000",
    "org_id": "ORG001",
    "status": 3,
    "previous_status": 2,
    "phone": "13800001111",
    "title": "小璃的森林冒险",
    "author": null,
    "style": "watercolor",
    "original_image_url": "https://oss.../ai-upload/v1.0/ORG001/138xxx/original.png",
    "pages": 6,
    "page_list": [
      {"page_num": 0, "text": "封面", "image_url": "https://oss.../cover.png", "audio_url": null},
      {"page_num": 1, "text": "故事正文...", "image_url": "https://oss.../p1.png", "audio_url": null}
    ],
    "fail_reason": null,
    "created_at": "2026-04-06T10:30:00Z",
    "completed_at": "2026-04-06T10:32:15Z"
  }
}
```

5.2.2 work.progress 事件

```
{
  "id": "evt_190368671438290",
  "event": "work.progress",
  "created_at": 1712000010000,
  "data": {
    "work_id": "1903686714382889000",
```

```

"org_id": "ORG001",
"status": 2,
"progress": 30,
"progress_message": "故事完成, 开始绘图...",
"phone": "13800001111"
}
}

```

5.3 各状态 Webhook 推送内容

| 状态 | 事件 | page_list | 关键字段 |
|----------------|---------------------|-----------|---|
| 1 (PENDING) | work.status_changed | 无 | work_id, phone, style, original_image_url |
| 2 (PROCESSING) | work.progress | 无 | progress, progress_message |
| 3 (COMPLETED) | work.status_changed | 完整列表 | page_list, original_image_url |
| 4 (CATALOGED) | work.status_changed | 完整列表 | title, author, tags, original_image_url |
| 5 (DUBBED) | work.status_changed | 完整列表 | page_list (含audio_url) , original_image_url |
| -1 (FAILED) | work.status_changed | 无 | fail_reason |

5.3.1 企业收到事件后如何处理 (关键)

企业收到 Webhook 推送后, 需要按以下规则处理。核心原则: remote_status > local_status 则同步。

```

// == 企业端处理逻辑 (Java伪代码) ==
@Transactional
public void handleWebhook(JSONObject data) {
    String workId = data.getString("work_id");
    int remoteStatus = data.getIntValue("status");

    // 1. 幂等去重 (用event_id)
    if (processedEvents.contains(eventId)) return;

    // 2. 查本地记录
    WorkRecord local = db.selectForUpdate(workId); // SELECT ... FOR UPDATE 行锁

    // 3. 新作品
    if (local == null) {
        db.insert(buildRecord(data)); // 创建本地记录
        return;
    }

    // 4. FAILED(-1): 强制更新
    if (remoteStatus == -1) {
        local.setStatus(-1);
        local.setFailReason(data.getString("fail_reason"));
        db.update(local);
        return;
    }

    // 5. PROCESSING(2): 强制更新进度
    if (remoteStatus == 2) {
        local.setProgress(data.getIntValue("progress"));
        local.setProgressMessage(data.getString("progress_message"));
    }
}

```

```

    db.update(local);
    return;
}

// 6. 状态前进: 全量覆盖
if (remoteStatus > local.getStatus()) {
    updateAllFields(local, data); // 更新title/author/page_list等
    db.update(local);
    return;
}

// 7. 旧数据/重复: 忽略
}

```

5.3.2 各状态下企业应保存的关键数据

| 状态 | 企业需保存的字段 | 用途 |
|--------------|--------------------------------------|------------------|
| 1 PENDING | work_id, phone, org_id, style | 创建本地记录, 关联企业用户 |
| 2 PROCESSING | progress, progress_message | 显示创作进度 (可选) |
| 3 COMPLETED | title, pages, page_list(含image_url) | 作品图片已生成, 可预览展示 |
| 4 CATALOGED | title, author, subtitle, intro, tags | 用户填写的编目信息 |
| 5 DUBBED | page_list(含audio_url) | 配音URL已填充, 作品完整可用 |
| -1 FAILED | fail_reason | 记录失败原因, 通知用户 |

5.3.3 并发安全 (Webhook + B3 同时触发)

Webhook实时推送和B3定时对账可能同时更新同一个作品。推荐两种方案防止并发冲突：

```

// 方案1: 行锁 (推荐, 简单可靠)
SELECT * FROM your_table WHERE work_id = ? FOR UPDATE;
// 然后按同步规则判断和更新

// 方案2: CAS乐观锁 (无锁, 适合高并发)
UPDATE your_table
SET status = #{remoteStatus}, title = #{title}, page_list = #{pageList}
WHERE work_id = #{workId}
AND (status < #{remoteStatus} OR #{remoteStatus} = 2 OR #{remoteStatus} = -1);
// rows=0 表示被其他线程抢先更新了, 安全忽略

// 幂等去重: 用 webhook_event_id 避免重复处理
// 建议在企业DB中增加 webhook_event_id 字段

```

重要: PROCESSING(2)和FAILED(-1)必须强制更新, 不参与数值比较。企业切勿用简单的 remote > local 判断, 必须特殊处理这两个状态。

5.4 签名验证

```

签名体 = "{X-Webhook-Id}.{X-Webhook-Timestamp}.{请求body原文}"
签名值 = HMAC-SHA256(签名体, app_secret).toHex()
Header: X-Webhook-Signature: HMAC-SHA256={签名值}

```

5.5 重试策略

首次发送 + 5次重试 = 共6次尝试。延迟：10s → 30s → 2min → 10min → 30min → 30min。全部失败后需通过B3批量拉取兜底。

六、接口参考

6.1 企业后端常用接口

| 编号 | 接口 | 方法 | 路径 | 认证 | 说明 |
|------|--------|------|-----------------------------|----------------|-------------------|
| AUTH | 会话令牌交换 | POST | /api/v1/auth/session | 密钥 | appSecret换取H5会话令牌 |
| B2 | 作品详情查询 | GET | /api/v1/query/work/{workId} | HMAC/T oken | 单个作品完整数据 |
| B3 | 批量作品查询 | GET | /api/v1/query/works | HMAC | 按时间范围批量查询（兜底同步） |
| B1 | 额度校验 | POST | /api/v1/query/validate | HMAC | 检查剩余创作额度 |

6.2 B2 作品详情返回格式

GET /api/v1/query/work/{workId}

```
{
  "code": 200,
  "data": {
    "workId": "1903686714382889000",
    "status": 3, // INT: -1/1/2/3/4/5
    "originalImageUrl": "https://oss.../original.png", // 用户原创作品
    "progress": 100, // 0-100, 仅status=1/2时有意义
    "progressMessage": "创作完成",
    "failReason": null, // 仅status=-1时有值
    "pageList": [
      {
        "pageNum": 0,
        "text": "封面文字",
        "imageUrl": "https://oss.../cover.png",
        "audioUrl": null
      }
    ]
  }
}
```

注意: V4.0已移除 dataVersion 和 completionStep 字段。status为INT类型。

重要: originalImageUrl 是用户在A6角色提取时通过OSS直传的原作品图片。企业客户必须保存此URL，它是用户原始创作的唯一凭证。该URL长期有效。

6.3 B3 批量查询返回格式

GET /api/v1/query/works?orgId=ORG001&updatedAfter=2026-04-05T00:00:00

```
{
  "code": 200,
  "data": [
    {
      "workId": "1903686714382889000",
      "status": 5, // INT
      "title": "小璃的森林冒险",
      "originalImageUrl": "https://oss.../original.png",
    }
  ]
}
```

```
"createdAt": "2026-04-06T10:30:00",  
"updatedAt": "2026-04-06T10:35:00"
```

```
}  
]  
}
```

七、企业端实现指南

7.1 企业DB推荐表结构

```
CREATE TABLE `enterprise_work` (  
  `work_id` VARCHAR(32) PRIMARY KEY,  
  `status` INT NOT NULL DEFAULT 0 COMMENT '状态: 乐读派1-5, 企业>=6',  
  `progress` INT DEFAULT 0 COMMENT '创作进度0-100',  
  `title` VARCHAR(200),  
  `author` VARCHAR(50),  
  `tags` JSON,  
  `page_list` JSON COMMENT '页面数据 (含image_url/audio_url) ',  
  `original_image_url` VARCHAR(500) COMMENT '用户原创作品图片URL (A6提取时直传OSS) ',  
  `phone` VARCHAR(20) COMMENT '用户手机号 (关联企业用户) ',  
  `fail_reason` VARCHAR(500),  
  `webhook_event_id` VARCHAR(64) COMMENT '最近一次webhook事件ID (幂等去重) ',  
  `synced_at` DATETIME COMMENT '最近同步时间',  
  -- 企业扩展字段 (乐读派不碰)  
  `local_user_id` BIGINT COMMENT '企业用户ID',  
  `review_status` VARCHAR(20) COMMENT '审核状态',  
  `remark` TEXT COMMENT '企业备注',  
  `created_at` DATETIME DEFAULT CURRENT_TIMESTAMP,  
  `updated_at` DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
  INDEX `idx_status` (`status`),  
  INDEX `idx_phone` (`phone`)  
);
```

注意: 不再需要 data_version 和 completion_step 列。status 一个字段即可。

7.2 同步方案选择

| 方案 | 实现方式 | 实时性 | 复杂度 | 推荐场景 |
|--------------------|------------------------|-------|-----|------|
| 方案1: B3轮询 | 每5分钟调B3拉取变更 | 分钟级 | 最低 | 快速验证 |
| 方案2: Webhook | 接收Webhook回调 | 秒级 | 中等 | 生产推荐 |
| 方案3: Webhook+ B3兜底 | Webhook主通道 + B3每30分钟兜底 | 秒级+保障 | 较高 | 高可靠性 |

7.3 B3兜底同步伪代码

```
// 每30分钟执行一次  
@Scheduled(fixedRate = 30 * 60 * 1000)  
public void reconcile() {  
  String since = lastSyncTime.format(ISO_FORMAT);  
  List remoteList = callB3(since);  
  
  for (RemoteWork remote : remoteList) {  
    WorkRecord local = db.get(remote.workId);  
    if (local == null  
        || remote.status > local.getStatus()  
        || remote.status == 2  
        || remote.status == -1) {  
      db.upsert(remote);  
    }  
  }  
}
```

```
}  
lastSyncTime = now();  
}
```

7.4 企业端状态路由（C端页面导航参考）

如企业自建C端（非使用乐读派提供的C端），参考以下路由逻辑：

```
switch (work.status) {  
  case 1: case 2:           // 排队/创作中  
    navigate('/creating/' + workId); // → 进度等待页  
    break;  
  case 3:                 // 图片完成  
    navigate('/catalog/' + workId); // → 编目编辑页（强制）  
    break;  
  case 4:                 // 编目完成  
    navigate('/dubbing/' + workId); // → 配音编辑页（强制）  
    break;  
  case 5:                 // 配音完成  
    navigate('/reader/' + workId); // → 阅读页 或 二次编辑页  
    break;  
  case -1:                // 失败  
    showError(work.failReason);  
    break;  
}
```

八、H5 快速接入指南（企业必读）

8.1 配置文件说明

乐读派提供的C端H5代码（dist包）中包含一个运行时配置文件 config.js。企业上线只需修改此文件，无需重新编译前端代码。

config.js 完整结构

```
window.__LEAI_CONFIG__ = {
  // 认证模式
  // "hmac" = 开发调试 (HMAC签名 + 局域网地址)
  // "token" = 生产环境 (Bearer Token + 正式地址)
  authMode: "hmac",

  // 开发环境配置 (authMode=hmac 时生效)
  dev: {
    apiBaseUrl: "http://192.168.1.72:8080",
    wsBaseUrl: "ws://192.168.1.72:8080",
    orgId: "你的机构ID",
    appSecret: "你的机构密钥",
    phone: "测试手机号"
  },

  // 生产环境配置 (authMode=token 时生效)
  // orgId/phone 由企业重定向URL参数动态传入
  // appSecret 仅在企业服务端使用，不到达浏览器
  prod: {
    apiBaseUrl: "https://你的API域名",
    wsBaseUrl: "wss://你的API域名"
  },

  // 品牌定制 (两种模式通用)
  brand: {
    title: "你的品牌名",
    subtitle: "副标题",
    slogan: "标语",
    favicon: "/favicon.ico"
  }
}
```

8.2 5分钟上线步骤

Step 1: 部署 dist 包到 Web 服务器

Step 2: 修改 dist/config.js:

```
authMode: "token",
prod: {
  apiBaseUrl: "https://你的API域名",
  wsBaseUrl: "wss://你的API域名"
}
```

Step 3: 管理后台「机构管理」中填写「认证回调URL」

Step 4: 企业后端实现认证跳转（参考2.2节Session Token流程）

8.3 Token 失效闭环

V4.0的H5端已内置token失效自动处理机制。当API返回会话过期(20009/20010)时:

1. H5自动保存当前创作进度 (页面路径+workId等)
2. H5自动跳转到企业「认证回调URL」
3. 企业重新认证后 302重定向回H5:
`https://{H5域名}/?token=新token&orgId=xxx&phone=xxx&returnPath=/creating`
4. H5自动恢复到之前的页面, 继续创作

前提: 企业必须在管理后台填写「认证回调URL」, 否则token失效后H5只能显示提示文案。

8.4 开发调试

开发阶段保持 `authMode=hmac`, 修改 `dev` 块中的局域网地址和测试凭证即可。HMAC模式下 `orgId/appSecret/phone` 从 `config.js` 读取, 无需从URL传入。

```
dev: {  
  apiBaseUrl: "http://局域网IP:8080",  
  wsBaseUrl: "ws://局域网IP:8080",  
  orgId: "管理后台分配的机构ID",  
  appSecret: "管理后台分配的机构密钥",  
  phone: "测试手机号"  
}
```

8.5 品牌定制

修改 `config.js` 中 `brand` 块即可自定义页面标题、Hero文字和绘本封面品牌。无需修改任何前端代码。

| 配置项 | 影响位置 | 说明 |
|-----------------------------|----------------------|---------|
| <code>brand.title</code> | 页面标题 + 首页Hero + 绘本封面 | 品牌主名称 |
| <code>brand.subtitle</code> | 页面标题 + 首页副标题 | 品牌副标题 |
| <code>brand.slogan</code> | 绘本背封 | 品牌标语 |
| <code>brand.favicon</code> | 浏览器标签页图标 | ICO文件路径 |

8.6 随附的 Java Demo

本指南随附了完整的 Spring Boot 企业对接 Demo (`enterprise-demo/java-demo/`), 包含令牌交换、H5跳转、认证回调、Webhook签名验证、B3定时对账全部功能。约600行Java代码, JDK 1.8兼容。

使用方法: 修改 `LeaiDemoApplication.java` 顶部的4个配置常量 (`ORG_ID`、`APP_SECRET`、`LEAI_API_URL`、`LEAI_H5_URL`) → 放入 Spring Boot 项目 → 启动运行。

Demo 代码中每个方法都有详细注释, 可直接作为企业对接的代码参考或移植基础。

九、iframe 嵌入模式 (postMessage 通信)

9.1 架构总览

企业C端通过 iframe 嵌入乐读派H5创作页面。企业前端作为「父页面」，乐读派H5作为「子页面」在iframe中运行。父子页面通过 window.postMessage API 进行双向通信。



9.2 两种嵌入模式对比

| 模式 | 认证方式 | token失效处理 | 适用场景 |
|-------------------|------------------|-------------|----------|
| standalone (独立页面) | redirect跳转企业认证页 | 页面跳转, 有中断感 | 独立打开H5页面 |
| iframe (嵌入模式) | postMessage通知父页面 | 无感刷新, 创作不中断 | 嵌入企业C端 |

9.3 iframe 加载方式

企业通过设置 iframe 的 src URL 加载乐读派H5, 可直接指定目标页面:

```
// 新建创作 (进入Welcome首页)
iframe.src = "https://{H5域名}/?token=xxx&orgId=xxx&phone=xxx&embed=1"

// 续编目 (status=3的作品, 直接跳到编辑绘本信息页)
iframe.src = "https://{H5域名}/edit-info/{workId}?token=xxx&orgId=xxx&phone=xxx&embed=1&from=works"

// 续配音 (status=4的作品, 直接跳到配音页)
iframe.src = "https://{H5域名}/dubbing/{workId}?token=xxx&orgId=xxx&phone=xxx&embed=1&from=works"

// 查看已完成作品 (status=5)
iframe.src = "https://{H5域名}/read/{workId}?token=xxx&orgId=xxx&phone=xxx&embed=1&from=works"

// 查看创作进度 (status=1/2)
iframe.src = "https://{H5域名}/creating?token=xxx&orgId=xxx&phone=xxx&embed=1&workId={workId}"

iframe标签需要 allow="camera;microphone" 属性 (录音功能需要)。每次切换作品时重新设置iframe.src (自动获取最新token)。from=works参数: 从作品列表进入时传入, H5的返回按钮会通知父页面切回作品列表。
```

9.4 postMessage 消息协议 (核心)

所有消息采用统一的信封格式。企业前端通过 `window.addEventListener('message', handler)` 监听H5发来的消息。

```
// 消息信封格式
{
```

```

source: "leai-creation", // 固定标识, 用于过滤非本系统消息
version: 1, // 协议版本号
type: "消息类型", // 如 TOKEN_EXPIRED, WORK_COMPLETED
payload: { ... } // 消息体, 不同类型内容不同
}

```

9.4.1 H5 → 企业父页面 (6种消息, 企业需要监听)

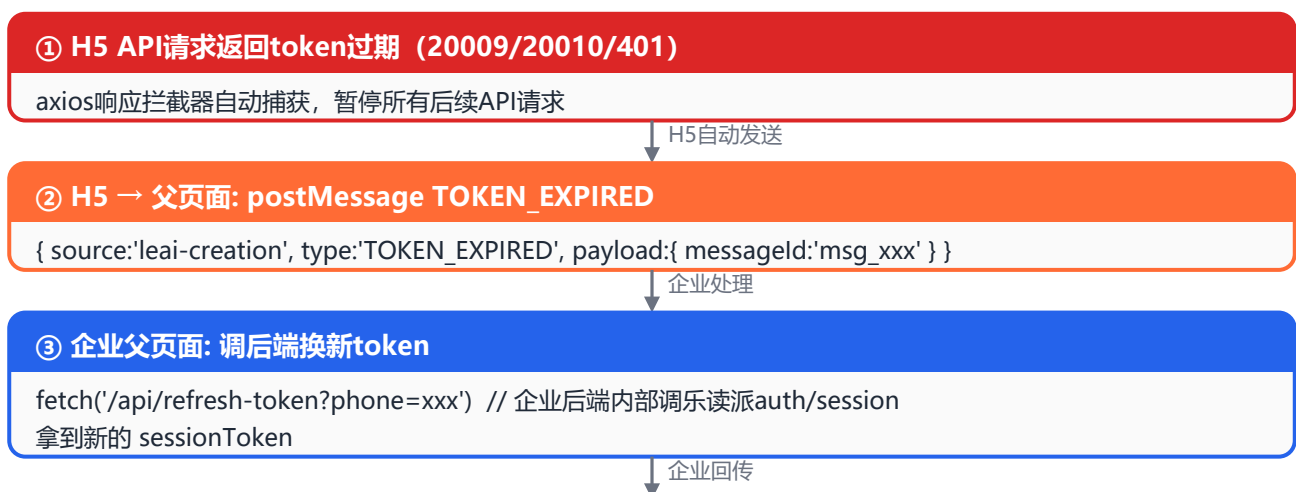
| type | 触发时机 | payload | 企业处理方式 |
|----------------|--------------------------|---------------|-------------------------------------|
| READY | H5加载完毕, 可接收消息 | {} | 可选: 记录就绪状态, 显示加载完成 |
| TOKEN_EXPIRED | token过期, API返回2009/20010 | { messageId } | 必须: 调企业后端换新token, 回传TOKEN_REFRESHED |
| WORK_CREATED | 用户提交创作, 获得workId | { workId } | 可选: 记录workId, 显示创作中 |
| WORK_COMPLETED | 创作完成 (图片生成完毕) | { workId } | 推荐: 刷新企业作品列表 |
| CREATION_ERROR | 创作失败 | { message } | 可选: 显示错误提示 |
| NAVIGATE_BACK | 用户点击返回按钮(从作品列表进入时) | {} | 推荐: 切换回企业的作品列表tab |

9.4.2 企业父页面 → H5 (1种消息, 企业需要发送)

| type | 场景 | payload | 说明 |
|-----------------|-----------------|------------------------------------|-------------------------------|
| TOKEN_REFRESHED | 响应TOKEN_EXPIRED | { messageId, token, orgId, phone } | messageId必须与TOKEN_EXPIRED中的一致 |

9.5 Token 无感刷新流程 (最关键的交互)

当H5的session token过期时, 以下流程自动执行, 用户完全无感知:



④ 父页面 → H5: postMessage TOKEN_REFRESHED

```
{ source:'leai-creation', type:'TOKEN_REFRESHED',  
  payload:{ msgid:'msg_xxx', token:'新token', orgId:'xxx', phone:'xxx' } }
```

注意: msgid 必须与步骤②中的一致!

↓ H5自动

⑤ H5 自动重试所有暂停的请求

用新token替换旧token → 队列中的请求自动重发 → 用户无感知

9.6 企业前端完整实现代码

以下是企业前端需要编写的完整JavaScript代码（约50行）。复制到你的HTML页面中即可使用：

```
// =====  
// 企业前端: iframe嵌入 + postMessage通信 完整实现  
// =====  
  
var iframe = document.getElementById('leai-iframe');  
var ENTERPRISE_API = 'https://你的企业后端'; // 企业自己的后端  
var LEAI_H5 = 'https://H5域名'; // 乐读派H5地址  
var ORG_ID = '你的机构ID';  
var phone = '当前登录用户手机号'; // 从企业用户系统获取  
  
// —— 1. 加载创作页面 ——  
function loadCreation() {  
  fetch(ENTERPRISE_API + '/api/refresh-token?phone=' + phone)  
    .then(function(r) { return r.json(); })  
    .then(function(data) {  
      iframe.src = LEAI_H5 + '/?token=' + data.token  
        + '&orgId=' + ORG_ID + '&phone=' + phone + '&embed=1';  
    });  
}  
  
// —— 2. 从作品列表跳转到编目/配音 ——  
function openWork(workId, status) {  
  var pathMap = { 3: '/edit-info/', 4: '/dubbing/', 5: '/read/' };  
  var path = pathMap[status] ? pathMap[status] + workId : '';  
  fetch(ENTERPRISE_API + '/api/refresh-token?phone=' + phone)  
    .then(function(r) { return r.json(); })  
    .then(function(data) {  
      iframe.src = LEAI_H5 + path + '?token=' + data.token  
        + '&orgId=' + ORG_ID + '&phone=' + phone  
        + '&embed=1&from=works';  
    });  
}  
  
// —— 3. 监听H5的postMessage ——  
window.addEventListener('message', function(event) {  
  var msg = event.data;  
  if (!msg || msg.source !== 'leai-creation') return;  
  
  switch (msg.type) {  
    case 'TOKEN_EXPIRED': // ★ 必须处理: token过期刷新  
      fetch(ENTERPRISE_API + '/api/refresh-token?phone=' + phone)  
        .then(function(r) { return r.json(); })  
        .then(function(data) {
```

```

iframe.contentWindow.postMessage({
  source: 'leai-creation', version: 1,
  type: 'TOKEN_REFRESHED',
  payload: {
    messageId: msg.payload.messageId, // ★ 必须原样传回
    token: data.token,
    orgId: ORG_ID,
    phone: phone
  }
}, '*');
});
break;
case 'WORK_COMPLETED': // 创作完成
  refreshWorkList(); // 刷新你的作品列表
  break;
case 'NAVIGATE_BACK': // H5请求返回
  switchToWorksTab(); // 切回作品tab
  break;
}
});

```

9.7 config.js 配置

乐读派H5的 dist/config.js 需要设置为iframe模式:

```

window._LEAI_CONFIG_ = {
  authMode: "token",
  prod: {
    apiBaseUrl: "https://乐读派API域名",
    wsBaseUrl: "wss://乐读派API域名"
  },
  embedMode: "iframe",
  parentOrigins: ["https://你的企业域名"], // 安全校验
  brand: {
    title: "你的品牌名",
    subtitle: "副标题"
  }
}

```

9.8 企业后端需要实现的接口

企业后端需要实现以下接口供前端JS调用:

```

// 1. Token刷新接口 (供前端JS在TOKEN_EXPIRED时调用)
GET /api/refresh-token?phone=xxx
→ 内部调用: POST {乐读派API}/api/v1/auth/session
  Body: { orgId, appSecret, phone }
→ 返回: { "token": "sess_xxx", "orgId": "ORG001" }

// 2. 认证入口 (用户首次进入 + token失效回调)
GET /leai-auth?returnPath=/creating&orgId=ORG001
→ 验证企业session → 换token → 302重定向:
  https://{H5域名}/?token=新token&orgId=xxx&phone=xxx&returnPath=/creating

```

9.9 数据同步: Webhook + B3对账

企业通过两个通道接收作品数据，确保数据一致：

| 通道 | 实时性 | 可靠性 | 说明 |
|-----------|------|------|------------------------|
| Webhook推送 | 秒级 | 可能丢失 | 主通道。作品状态变更时乐读派主动推送 |
| B3定时对账 | 30分钟 | 高可靠 | 兜底通道。定时批量查询补偿Webhook遗漏 |

重要：建议B3对账周期不低于30分钟，避免对乐读派API造成查询压力。查询范围建议覆盖2个对账周期（如30分钟对账则查最近2小时）。

9.10 注意事项

| 注意项 | 说明 |
|-------------------|---|
| iframe allow属性 | 需要 allow="camera;microphone"，否则录音功能不可用 |
| 不要频繁销毁iframe | 建议用显隐控制，避免丢失创作状态 |
| messageId必须原样传回 | TOKEN_REFRESHED中的messageId必须与TOKEN_EXPIRED中的一致，否则H5无法关联响应 |
| 每次切换作品重新获取token | 设置iframe.src时获取新token，确保token有效 |
| parentOrigins安全配置 | 生产环境建议在config.js中配置企业域名白名单 |
| B3对账周期 | 建议30分钟，不低于15分钟 |

完整可运行的企业对接Demo请参考随附的 [enterprise-demo/java-demo/](#) 目录。Demo包含上述所有接口的完整实现，替换4个配置后可直接运行。代码中的TODO注释指导你如何替换为自己的数据库操作。

8.7 两种集成方式

乐读派提供两种集成方式，企业可根据自身技术能力和定制需求选择：

方式一：dist 包交付（iframe 嵌入）

乐读派提供编译后的 dist 包，企业部署到自己的Web服务器，通过 iframe 嵌入创作功能。不提供源码。

| 项目 | 说明 |
|---------|---|
| 交付物 | dist/ 目录（编译产物，含 config.js） |
| 企业能否改UI | 不能修改创作页面UI，只能通过 config.js 改品牌名/标语 |
| 企业前端工作 | 编写iframe嵌入代码 + postMessage通信（约50行JS） |
| 企业后端工作 | 实现4个接口：/leai-auth、/api/refresh-token、/webhook/leai、B3对账 |
| 我方更新方式 | 直接提供新dist包，企业替换部署即可 |
| 对接周期 | 7-10个工作日 |
| 适合场景 | 快速上线、前端能力弱、不需要深度UI定制 |

方式二：源码交付（企业自集成）

乐读派提供完整的 Vue 3 源码，企业将创作模块集成到自己的前端项目中。无iframe，创作页面是企业应用的一部分。

| 项目 | 说明 |
|---------|--|
| 交付物 | src/ 源码目录 (23个文件, 约5000行Vue/JS代码) |
| 企业能否改UI | 可以修改任何页面样式和布局 |
| 企业前端工作 | 注册9个路由 + 设置config.js (约20行代码改动, 无需改创作逻辑) |
| 企业后端工作 | 与方式一相同 (4个接口 + Webhook + B3对账) |
| 我方更新方式 | 提供源码更新, 企业需合并代码 (git merge) |
| 对接周期 | 4-7个工作日 (需Vue前端开发能力) |
| 适合场景 | UI深度定制、追求无iframe无缝体验、有Vue开发团队 |

方式二源码集成步骤 (仅3步)

```
// Step 1: 将乐读派 src/ 复制到企业项目中
cp -r lesingle-aicreate-client/src/ your-project/src/modules/lesingle/

// Step 2: 修改 config.js (3个值)
authMode: "token",
prod: { apiBaseUrl: "https://你的API域名", wsBaseUrl: "wss://你的API域名" }

// Step 3: 在企业router中注册路由 (约10行)
import { routes as leaiRoutes } from '@modules/lesingle/router'
leaiRoutes.forEach(r => router.addRoute(r))

// 入口跳转 (企业页面中, 传入token和用户信息)
router.push('/?token=sess_xxx&orgId=ORG001□=138xxx')
```

方式二不需要改任何创作页面代码。企业只需改config.js和注册路由。所有创作交互 (上传-识别-画风-故事-创作-编目-配音-阅读) 原封不动使用。

技术依赖 (方式二)

源码仅依赖6个npm包, 无第三方UI库 (无Element Plus/Ant Design等), 不会与企业现有UI框架冲突:

```
vue: ^3.5 // 框架 (企业项目已有)
vue-router: ^4.5 // 路由 (企业项目已有)
axios: ^1.7 // HTTP客户端 (企业项目已有)
ali-oss: ^6.23 // 阿里云OSS (图片/音频直传)
@stomp/stompjs: ^7 // WebSocket (创作进度推送)
crypto-js: ^4.2 // HMAC签名 (开发模式用)
```

十、附录：错误码与 FAQ

10.1 状态相关错误码

| 错误码 | 含义 | 触发场景 |
|-------|--------|----------------------------------|
| 20001 | 参数错误 | 缺少必填参数 |
| 20003 | 作品不存在 | workId无效 |
| 20004 | 作品状态异常 | C1在status!=3时调用, C2在status!=4时调用 |
| 20009 | 会话令牌过期 | Session Token超过2小时 |
| 30010 | 额度不足 | 创作额度已用完 |

10.2 常见问题

Q1: 如何判断作品是否需要同步?

比较 `remote_status` 和 `local_status` 的数值大小。若 `remote` > `local`, 则需要同步。PROCESSING(2)和FAILED(-1)是例外, 始终强制同步。

Q2: status=5 之后企业如何维护二次编辑?

企业在本地DB中, `status`从5开始自增。例如修改标题后`status=6`, 修改简介后`status=7`。这些`>=6`的状态完全由企业自维护, 不回传给乐读派。

Q3: 用户没有配音就点了保存, status也会变5吗?

是的。配音环节的"保存"按钮无论是否实际录音, 都会将`status`从4推进到5。配音是可选的内容, 但保存操作是必须的环节。

Q4: PROCESSING期间进度更新太频繁怎么办?

Webhook只在关键里程碑 (10%/30%/50%/70%/90%) 推送 `work.progress` 事件, 不会每1%都推送。企业也可选择忽略进度事件, 仅处理 `work.status_changed`。

Q5: 从V3.1迁移到V4.0需要改什么?

1) 将本地 `data_version` 和 `completion_step` 字段删除或弃用; 2) 将同步逻辑从 `dataVersion` 门卫改为 `status` 数值比较; 3) 更新Webhook处理器, 适配新的事件类型和payload格式。

Q6: OSS图片/音频链接会过期吗?

不会。所有资源存储在阿里云OSS, 链接长期有效 (除非作品被删除)。

附录B：严禁事项（数据一致性生命线）

| 严禁操作 | 后果 | 正确做法 |
|--------------------|-----------------|------------------------|
| 不按数值比较判断同步 | 数据被错误覆盖或遗漏 | 严格执行 remote > local 规则 |
| 忽略 status=2 的强制更新 | 进度信息缺失 | PROCESSING期间无条件接受更新 |
| 忽略 status=-1 的失败通知 | 用户看不到失败信息 | FAILED通知无条件处理 |
| 修改乐读派同步字段后不通知 | 下次被覆盖导致数据丢失 | 企业扩展字段与乐读派同步字段严格隔离 |
| 不做B3对账（仅依赖Webhook） | Webhook丢失导致数据遗漏 | 至少每30分钟跑一次B3兜底 |
| 企业status>=6回传给乐读派 | 系统状态混乱 | >=6的状态仅在企业本地维护 |