



# 同步队列页 · 交互与布局规格 V0.1

👤 创建者	Ⓜ z yx
🕒 创建时间	@2025年10月30日 21:30
☰ 类别	FrontEnd TechSpec
👤 上次编辑者	Ⓜ z yx
🕒 上次更新时间	@2025年10月30日 22:09
⚙️ 状态	完成

给出“同步队列视图”的成品规格。目标：列出所有待同步与失败项，支持单条重试与批量重试，覆盖手机竖屏与平板横竖屏，风格与现有 V0.1 文档一致。

## 1. 信息架构 / 版块

### SyncQueuePage

- └─ HeaderBar (返回 | 标题“同步队列” | 溢出菜单)
- └─ SummaryBar (计数: 待同步/成功/失败 | 一键“全部重试”)
- └─ QueueList (虚拟列表)
  - └─ QueueItem × N
    - └─ Meta (类型/来源/时间/大小)
    - └─ StatusBadge (Pending | Syncing | Success | Failed)
    - └─ ErrorRow (仅失败时显示: 错误码/简述/展开日志)
    - └─ Actions (重试 | 取消/撤回 | 查看)
- └─ FooterBar (批量: 全选/反选 | 重试所选 | 清除成功)
- └─ Toast/Confirm (清除/取消二级确认)

## 2. 响应式布局

- 手机竖屏: HeaderBar 56–64, 高度可滚; SummaryBar 固定; FooterBar 固定。

- 平板横屏：左右 7:5。左侧 QueueList，右侧 DetailsPane（选中项的错误日志/原文/Diff），可折叠。
- 网格：12 列，gutter 16/24，与学习页一致。

### 3. 交互逻辑

场景	行为
进入页面	触发 <code>queue.refresh</code> 拉取本地队列与最近一次同步结果，首屏 20 条。
单条重试	点击「重试」。状态切换 Failed→Retrying→Syncing→Success/Failed。失败留在原位并更新重试次数。
批量重试	SummaryBar「全部重试」或 FooterBar「重试所选」。并行度默认 3，排队可视。
取消同步	对 Syncing 项点击「取消」，进入 Canceled，保持在列表（可二次重试）。
查看错误	点击 ErrorRow 展开日志面板（平板右侧显示，手机底部抽屉）。
清除成功	FooterBar「清除成功」→二级确认→仅移除 Success 项的可见性，不删历史日志。
自动重试	网络恢复/令牌刷新后，对 Pending/Retryable 自动拉起一次，遵循退避策略。
筛选	Header 溢出菜单：全部   仅失败   仅进行中。
空态	无队列项时显示空态插画与「返回学习页」「现在同步」按钮。

### 4. 组件与状态

#### 4.1 QueueItem

- Meta：图标（Sentence/Highlight/Pref），主标题（对象摘要），次信息（本地时间，大小/变更字段数）。
- StatusBadge： `Pending / Syncing / Success / Failed / Canceled / Skipped` 。
- Progress：Syncing 时显示进度条或「队列 X/总 N」。
- Actions：
  - `Failed`：重试 | 查看日志。
  - `Syncing`：取消。
  - `Success`：查看详情 | 从列表隐藏。

- 批选：左侧复选框。
- ErrorRow（失败态可展开）：
  - 错误码、简述、重试建议、最后一次尝试时间、已重试次数。
  - 展开查看更多：请求/响应片段、栈、Notion/本地对象 ID 对照。

## 4.2 SummaryBar

- Pill 统计：待同步、成功、失败。
- CTA：「全部重试」禁用条件 = 无失败或当前有进行中的重试。

## 4.3 DetailsPane（平板）

- Tab：信息 | 错误日志 | Diff。
- Diff：本地与远端字段差异，冲突字段高亮。

# 5. 视觉与主题

- 继承学习页字体与主题色。Badge 颜色示例：
  - Pending 灰 #BDBDBD
  - Syncing 蓝 #42A5F5
  - Success 绿 #66BB6A
  - Failed 红 #EF5350
  - Canceled 橙 #FFA726
- 触控区  $\geq 44 \times 44$ 。深色模式对比  $\geq 4.5:1$ 。

## 6. 数据结构（Notion/本地）

```
// SyncQueueItem
{
  "id": "sq_...",
  "entity_type": "sentence|highlight|pref",
  "entity_id": "xxx",
  "op": "create|update|delete",
  "payload": { /* 差异或全量 */ },
}
```

```

"status": "pending|syncing|success|failed|canceled|skipped",
"attempt": 0,
"max_attempt": 5,
"last_error": {
  "code": "HTTP_401|NET_TIMEOUT|CONFLICT|RATE_LIMIT|SCHEMA",
  "message": "string",
  "at": "ISO8601",
  "meta": { "requestId": "...", "endpoint": "/pages", "duration_ms": 532 }
},
"created_at": "ISO8601",
"updated_at": "ISO8601",
"priority": 0
}

```

## 7. 同步与重试策略

- 并发：默认 3，失败不阻塞其他项。
- 退避：指数退避 + 抖动，`delay = min(2^attempt * 1000, 60000) + random(0..500)`。
- 可重试错误：5xx、429、网络错误、401（在刷新令牌后重试一次）。不可重试：400/404/422/冲突未解决。
- 冲突处理：
  - 策略：`server_wins | client_wins | prompt_user`。默认 `server_wins`，保留本地副本进入 `Skipped` 并生成“需要处理”标签。
- 原子性：同一实体多项变更折叠合并；delete 吞并前序未同步 create/update。
- 顺序：按实体分片内序列一致，跨实体可乱序。

## 8. 事件与遥测

- `queue.refresh`、`queue.filter_change`
- `item.retry_click`、`item.cancel_click`、`item.expand_error`
- `bulk.retry_all`、`bulk.retry_selected`
- `auto.retry_triggered`（网络恢复/令牌刷新）
- `item.status_change`（from→to，耗时）

- `queue.clear_success`
- 指标：失败率、P50/P95 同步时延、平均尝试次数、顶层错误分布。

## 9. API 合约（示例）

- `POST /sync/batch`：请求体为若干 `SyncQueueItem` 的 payload 列表，返回每条 `id → {status, error?}`。
- `POST /sync/retry?id=...`：单条重试。
- `GET /sync/logs?id=...`：拉取错误日志与最近请求回放。
- 身份：401 时先 `POST /auth/refresh`，成功后自动重试一次。

## 10. 空态与异常文案

- 空态：`暂无待同步项`，副文 `所有内容已是最新`。按钮 `返回学习页`、`现在同步`。
- 失败条目：
  - `NET_TIMEOUT`：`网络超时，稍后可重试`
  - `HTTP_429`：`请求过于频繁，已为你稍后重试`
  - `CONFLICT`：`远端已有更新，已暂停此项`
- Toast：
  - `已开始重试 5 项`
  - `3 项成功，2 项仍失败`

## 11. Figma 搭建要点

1. Page 变体：`Filter=All/Failed/Syncing`，`RightPane=Hidden/Visible`。
2. QueueItem 组件变体：`Status` 六态、`Selected` 开关、`ErrorExpanded` 开关。
3. Badge、Button、Toast、Progress 复用系统库；保持与学习页组件体系一致。
4. 原型交互：
  - 点击「重试」→ 切换到 `Syncing` 变体，再依据定时器切到 `Success/Failed`。
  - 溢出菜单切换筛选变体。
  - 平板：选择项→右侧 Pane 弹出。

5. 适配：手机 360×800；平板 1280×800/1024×768；Auto Layout + Hug/Fill。

## 12. 开发落地建议

- 状态机：

```
Pending → Syncing → Success
    ↘ Failed → Retrying(=Syncing) / Canceled / Skipped
```

- 队列实现：本地持久化（SQLite/Notion 缓存表），应用前台与网络可用时驱动出队。
- 选择模型：List virtualize，带 `selectedIds:Set`。
- 性能：并发受限 + 请求合并；日志/响应大对象延迟加载。
- 测试：
  - 可重试与不可重试错误路径。
  - 令牌过期→刷新→恢复。
  - 批量重试并发上限。
  - 冲突策略与折叠合并。

## 13. 与“单词学习页”的数据对应

- `Sentence/Highlight/ReadingPrefs` 的 create/update/delete 均入队一条 `SyncQueueItem`。
- 删除句子时同时入队关联高亮的清理任务，前端合并为单条 delete。

需要我把这页做成可导入 Figma 的组件清单或导出一张合成 SVG，再给你。

同步队列视图 · 合成SVG（Phone + Tablet） V0.1

