

Prd Output 3.0

📦 核心架构演进图

在 SOP 3.0 中，PRD Viewer 不再只是个“阅读器”，而是一个**“解析渲染引擎 + 资产提取工具”**。

- **数据源剥离：**UI 代码 (`ui.html`) + 业务逻辑 (`rules.md`) + 埋点宽表 (`tracking.csv/json`) + 多语言字典 (`i18n.csv/json`)。
- **知识库同源：**`.md` 文件既是前端渲染的数据源，也是直接无缝 Copy 或 API 同步到 Confluence 的文档资产。

📝 一、Demo HTML 标注军规升级 (SOP 3.0)

为了让 PRD Viewer 能够精准提取你需要的所有资产，前端 Demo 的标签不仅要绑定业务区域，还要独立剥离“埋点”和“多语言”属性。

每个核心可交互元素需要打上三种标签（可根据功能按需添加）：

1. **逻辑锚点 (绑定 MD 文档):** `data-zone-id="模块_页面_元素"`
2. **多语言锚点 (绑定翻译表):** `data-i18n-key="页面_元素_文案"`
3. **埋点锚点 (绑定神策/数据表):** `data-track-id="event_name"` 及 `data-track-params='{"key": "val"}'`

代码示例：

HTML

Code block

```
1 <div class="card"
2   data-zone-id="intent_course_card"
3   data-track-id="click_intent_course"
4   data-track-params='{"source": "splash", "user_type": "new"}'><h3 data-
i18n-key="intent_course_title">Find a Course</h3></div>
```

📝 二、业务逻辑文档的 Markdown 规范 (适配 Confluence)

为了废弃 `.json` 并完美适配 Confluence，我们需要制定一套对人类阅读极度友好，同时又能被 **PRD Viewer 正则解析** 的 Markdown 书写格式。

核心规则：通过标题中的中括号 [zone_id] 来建立程序映射。

rules.md 模板示例：

Markdown

Code block

 优势：任何不懂代码的产品经理都可以直接在 Confluence 里用大白话写这篇 MD 文档，写完后 PRD Viewer 就能自动抓取 [xxx] 里的 ID，与左侧的 `data-zone-id` 绑定。

三、 PRD Viewer 3.0 引擎改造需求清单

为了支撑这套体系，我们需要对你本地的 `prd_view.html` 外壳进行四项重大引擎升级：

能力 1：双向高亮引擎 (Bidirectional Sync)

- **实现逻辑：**
 - **正向 (UI -> 文档)：**监听左侧 iframe 内带有 `data-zone-id` 元素的点击/悬浮事件，拿到 ID 后，右侧的 Markdown 渲染区自动平滑滚动 (smooth scroll) 到对应的 `### [zone_id]` 标题，并加上短暂的黄色背景高亮。
 - **反向 (文档 -> UI)：**监听右侧渲染后 Markdown 标题的悬浮事件，向 iframe 发送 `postMessage`，让左侧对应的 HTML 元素出现呼吸灯光圈 (Box-shadow glow) 或红框。
 - **价值：**真正实现“一目了然，指哪打哪”，极大降低评审沟通成本。

能力 2：埋点提取器 (Tracking Extractor)

- 实现逻辑：
 - 在 PRD Viewer 顶部增加一个 “ 埋点清单” Tab。
 - 切换到该 Tab 时，外壳脚本自动遍历 iframe DOM 树，提取所有 `data-track-id` 和 `data-track-params`，以及对应的节点文本/结构。
 - 动态生成一个完整的 HTML Table（甚至加上“一键复制为 CSV”按钮）。
- 价值：需求写完，埋点表自动生成，研发和测试直接复制到飞书/Excel，无缝衔接数据团队。

能力 3：多语言动态注入字典 (i18n Engine)

- 实现逻辑：
 - PRD Viewer 接入一个公共的字典数据源（可以是根目录下的 `locales.json`，未来也可以是通过 API 接入公有表格服务）。
 - 顶部增加一个 “ 语言切换” Dropdown（如 EN / ZH / ES）。
 - 切换语言时，引擎扫描所有 `data-i18n-key`，从字典中匹配对应语种的 Value，动态替换 iframe 内部的文本节点。
- 价值：出海团队无需看两份文档，直接在原型上验收不同语言的 UI 溢出和适配情况。

能力 4：知识库热更新 (Live Markdown Editor)

- 实现逻辑：
 - 既然右侧是 Markdown，我们只需在右侧集成一个轻量级的开源 MD 编辑器（如 SimpleMDE 或直接用 textarea + 实时预览）。
 - 发现错误，原地修改文本，点击“保存为 .md 文件”，然后把更新后的文件上传到 Git 或直接粘贴回 Confluence。