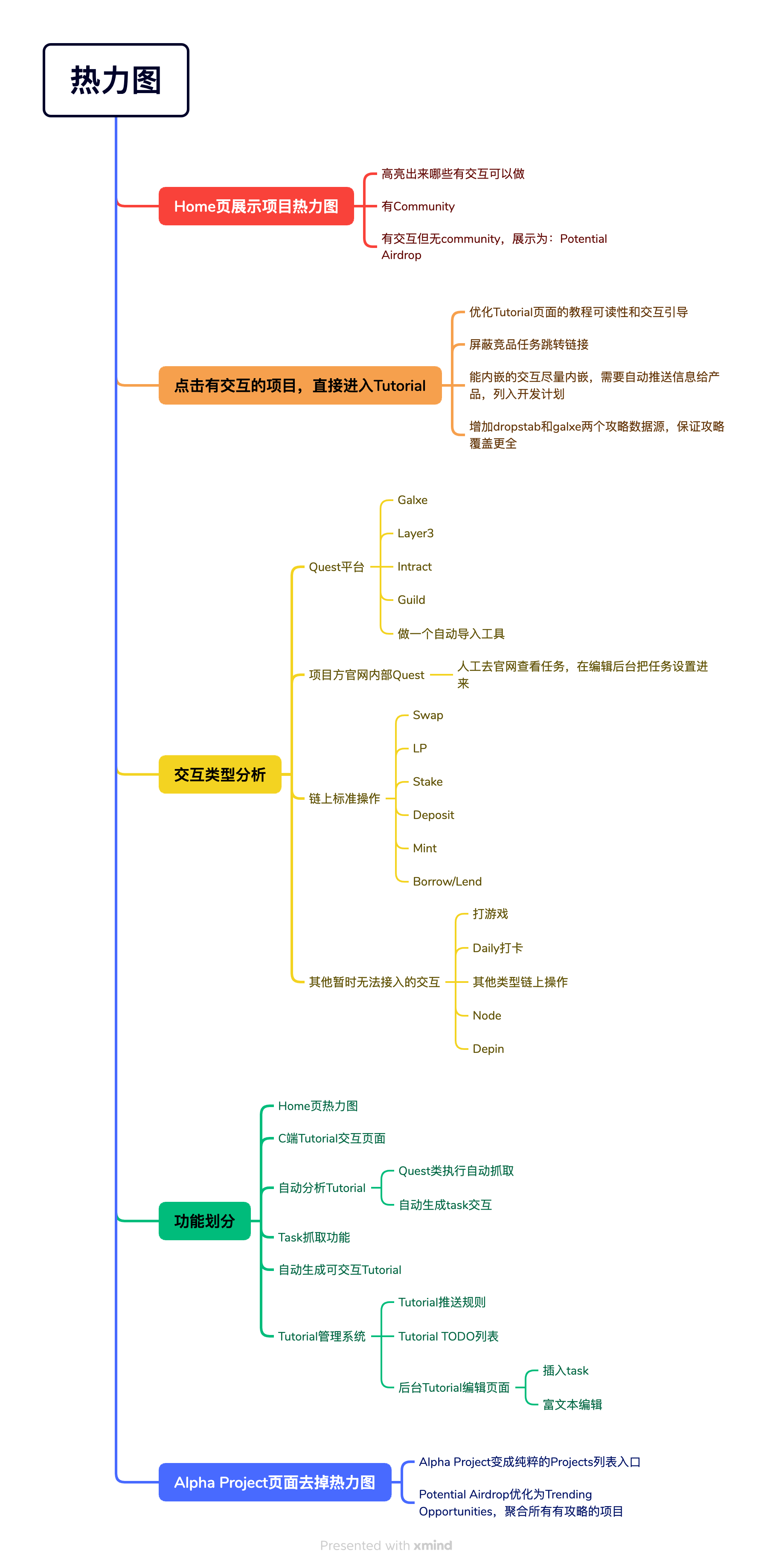
**热力图+可交互Tutorial需求文档**

**整体思路**



**用户流程**

**C端用户流程**

1. 在Home页看到热力图，有交互的项目会有高亮和标签；
2. 热力图的文字/数字等大小样式，要根据UI稿再优化一下；



1. 用户点击有交互的项目，直接进入项目详情页，并定位在Tutorial板块；
2. 用户查看项目信息和攻略，所有可执行的交互动作都有类似task的跳转链接和验证机制；
3. 用户执行某一个action，该task勾选变成已完成状态；

**平台工作流程**

1. 从三方网站抓取到Project的Tutorial；
2. 放入Tutorial自动分析流程，分析如何处理；
3. 分支一，Quest类Tutorial，进入Task自动抓取流程；
4. 分支二，其他类型Tutorial，进入人工处理流程；
5. 分支一：执行Task自动抓取流程，去Quest网站上把task抓取下来
6. 与我们的task模板匹配
7. 相同task去重
8. 进入自动生成可交互Tutorial流程
9. 自动生成可交互Tutorial流程
10. 自动生成标题
11. 自动生成时间范围
12. 自动生成简介
13. 自动为每一个task配置描述
14. 生成Tutorial并关联到Project
15. 针对有Community的情况，生成Tutorial引导用户参与Community
16. 分支二：人工处理流程
17. 热点项目优先处理
18. 非热点项目次要处理
19. 没有教程的项目，可以手动添加一个教程
20. 热点项目优先处理
21. 上了热力图的项目，如果有Tutorial尚未处理，推送加入后台的todo list
22. 非热点项目次要处理
23. 没有加入todo list的项目，运营可以在后台所有项目列表中看到哪些教程还没处理，有空慢慢处理
24. 在后台打开教程，阅读内容
25. 去项目官网做quest，需要打开项目官网，查看所有task，然后在后台把与taskon任务模板匹配的task手动加入到教程的相应位置
26. 链上任务，如果已经支持了，直接把task手动加入到教程中
27. 链上任务，如果没支持，提交给产品团队判断是否要支持
28. 运营需要优化教程文案，保证教程在加入task之后阅读流畅
29. 完成整个Tutorial的编辑，手动标记为Complete

**功能清单（1，2，3，4已完成，先交付）**

1. 首页增加热力图（P0）
2. Alpha Project页面去掉热力图的样式（P0）
3. C端Project Detail页+Tutorial板块优化（P0）
4. 后台管理系统，查看和编辑Tutorial（P0）
5. Tutorial自动分析系统（P1）
6. Task抓取系统（P1）
7. 可交互Tutorial自动生成系统（P1）
8. Quest类Tutorial
9. Community Tutorial
10. 项目入驻，自动同步task的完成状态到Community（P1）
11. 运营系统（P1）
12. 增加Tutorial数据源，dropstab+galxe（P2）

**功能明细**

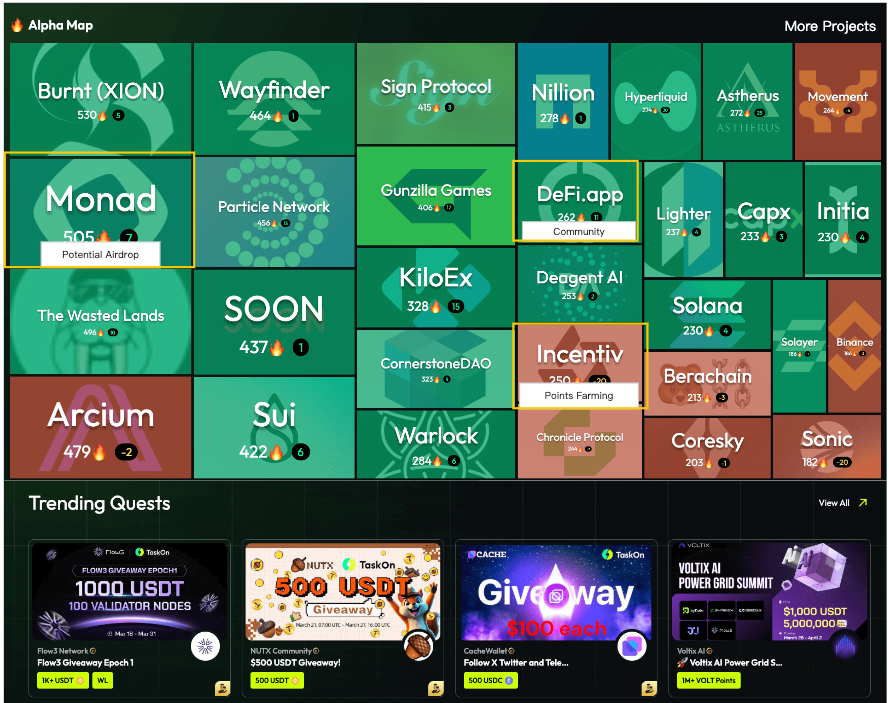
1. **首页增加热力图（Sam）**

**进入条件**

首页

**页面布局**

1. 顶部的位置，去掉Trending Quests上方的内容，替换成Alpha Map的这个热力图；
2. 如果热力图中的部分项目有标签
3. 有community的，标签展示为community
4. 没有community，但是有open状态的交互教程的，展示标签为Potential Airdrop（未发币）或者Points Farming（已发币）
5. 有标签的项目，增加高亮展示的样式
6. 右上角有more projects按钮



**交互逻辑**

1. 点击有交互项目，进入Project详情页，并定位在Tutorial tab；
2. 点击无交互项目，进入Project详情页，并定位在Overview tab；
3. 点击More Projects，进入Alpha Projects列表页；

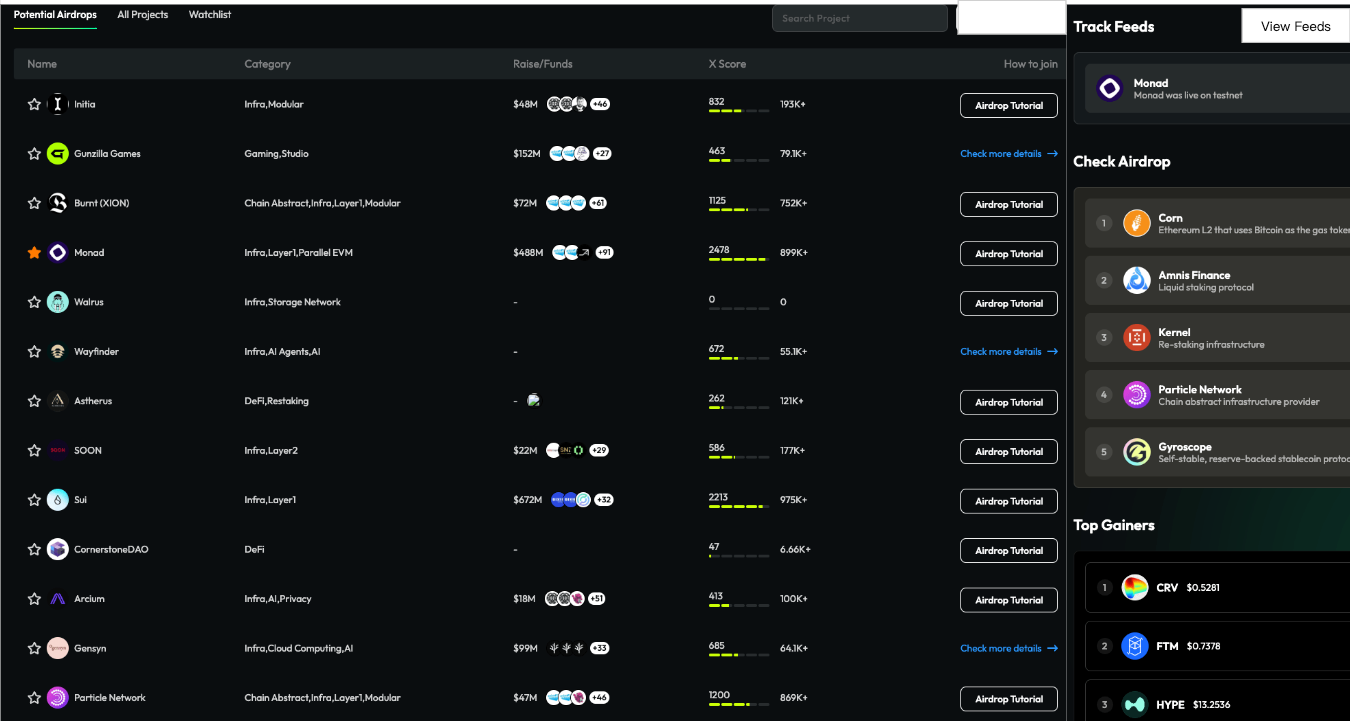
2. **Alpha Projects（Sam）**

**进入条件**

1. 路径保持为：/alpha/home，替换掉之前的页面
2. 去掉/alpha/projects

**页面布局**

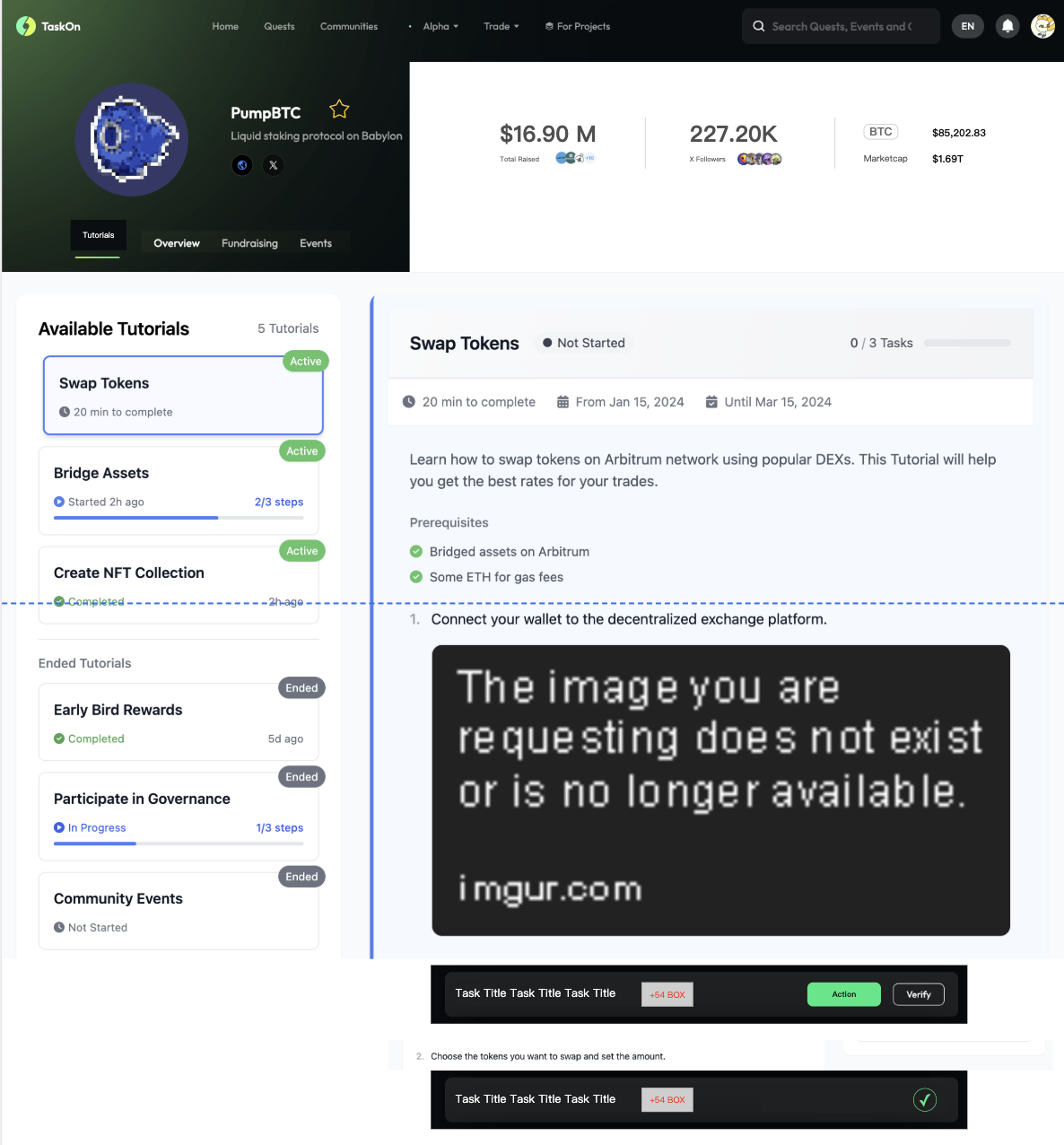
1. 左侧就是之前的/alpha/projects页面；
2. 右侧把之前/alpha/home页面的右侧三个板块方过来；
3. 把Track Feeds右边的more按钮加强，文案改为View Feeds；



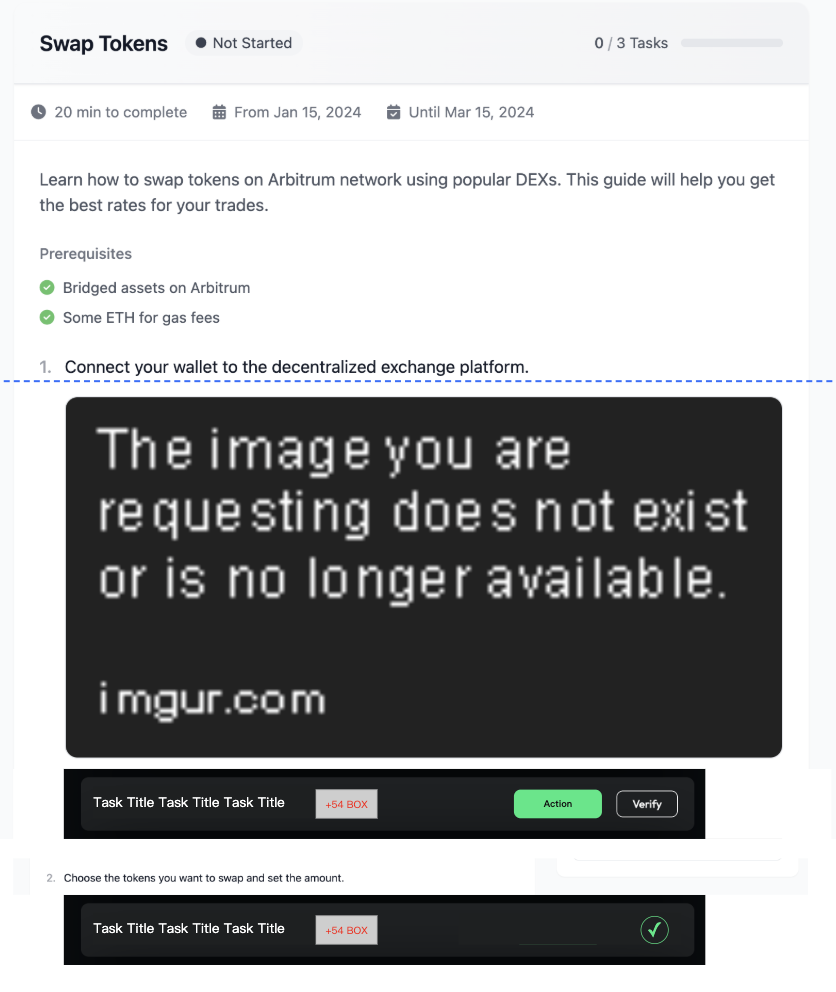
3. **C端Project Detail页+Tutorial板块优化（billy）**

**3.1 Tutorial 板块**

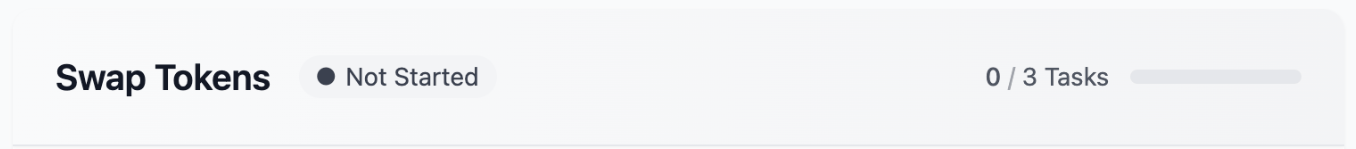
Tutorial Tab 位置调到左边第一位



**3.1.1 教程内容区**

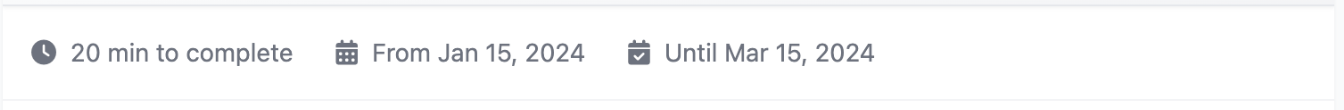


1. **粘性标题栏**



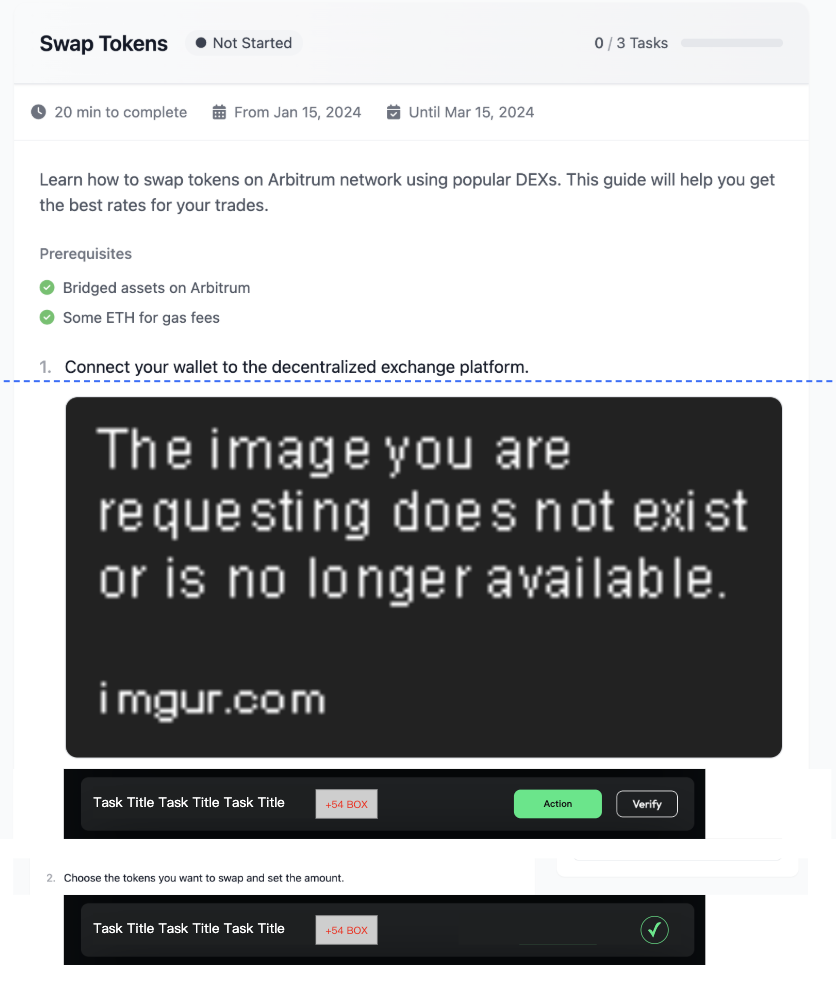
* **布局**：
* 显示教程标题和状态标签
* 进度指示器（完成任务数/总任务数）
* 滚动时保持可见
* **交互**：
* 随页面滚动始终保持在视图顶部
* 状态标签根据当前教程状态动态变化
* 进度指示器实时反映任务完成情况

1. **信息栏**

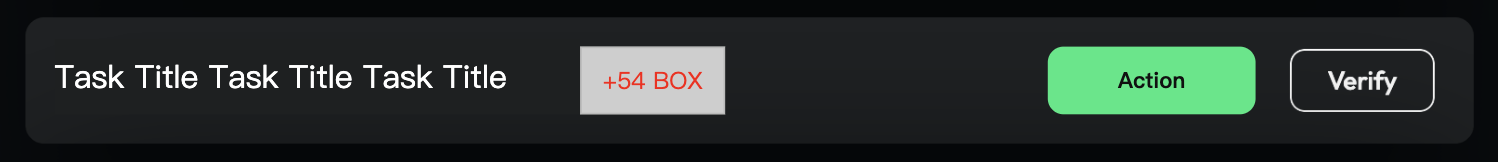


* **布局**：
* 估计完成时间
* 活动起止日期
* 非滚动固定位置
* **交互**：
* 在标题栏下方固定显示，不随内容滚动
* 时间信息根据教程配置自动展示

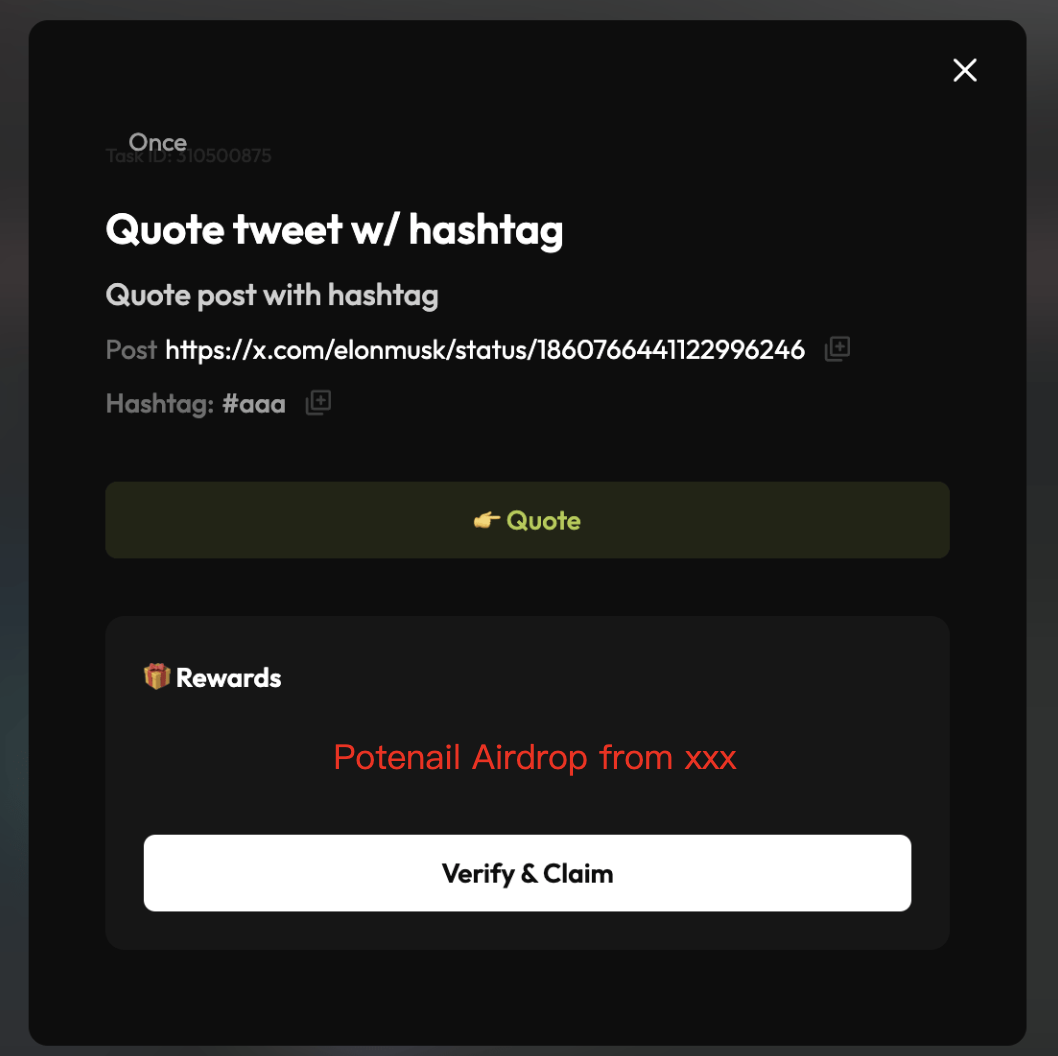
1. **教程内容区**



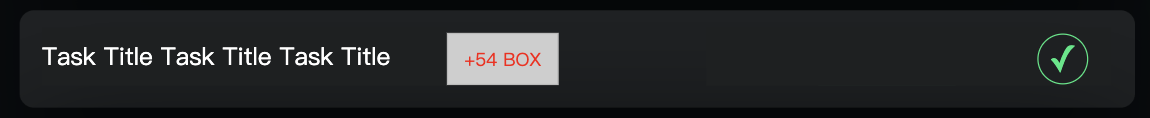
* **布局**：
* 教程描述
* 任务模块（若后端运营工具在教程内有配置任务)
* 任务卡片上要显示以下要素:

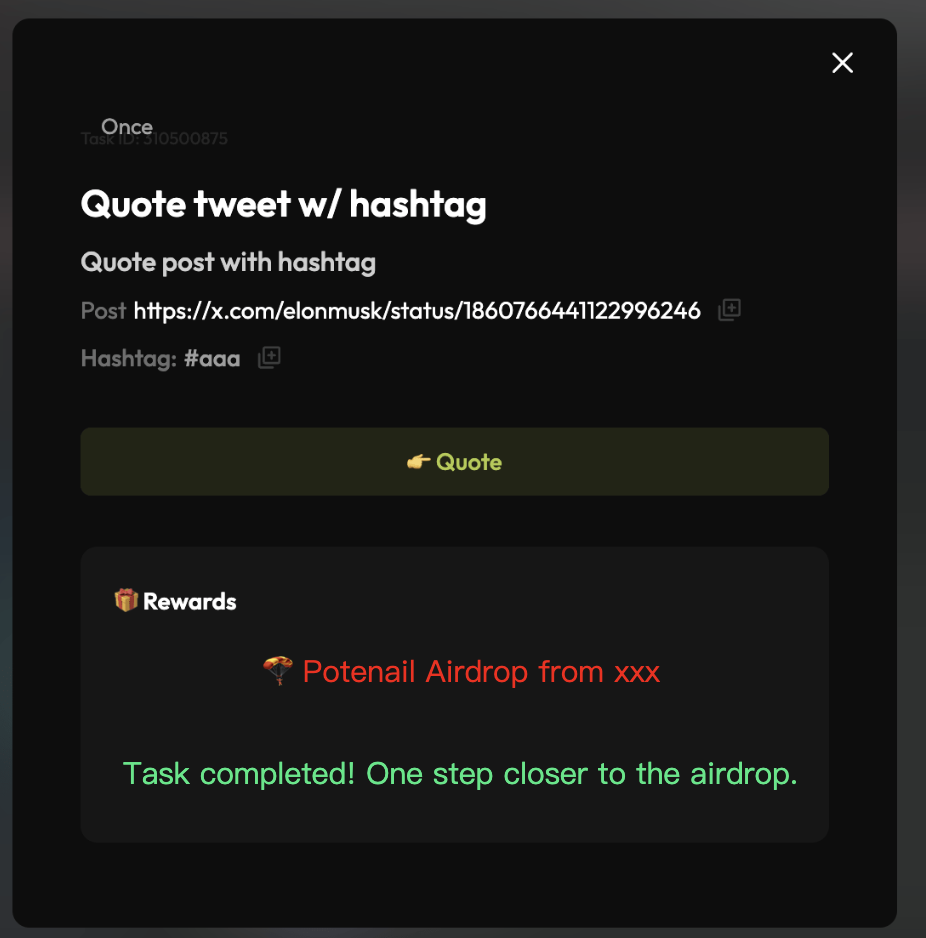


* 任务标题
* 任务奖励(本期没有, 后期会加上去, 需要预留位置)
* Action 按钮 (这个按钮有可能没有, 比如在 token balance 任务) 注: Action 按钮上的文案与现有community task 上的 **保持一致**
* Verify 按钮 (这个按钮有可能没有, 比如在浏览网页任务) 注: Verify 按钮上的文案不与 community task 保持一致, 统一使用 Verify 作为 button 文档
* 任务弹窗上要显示以下要素:

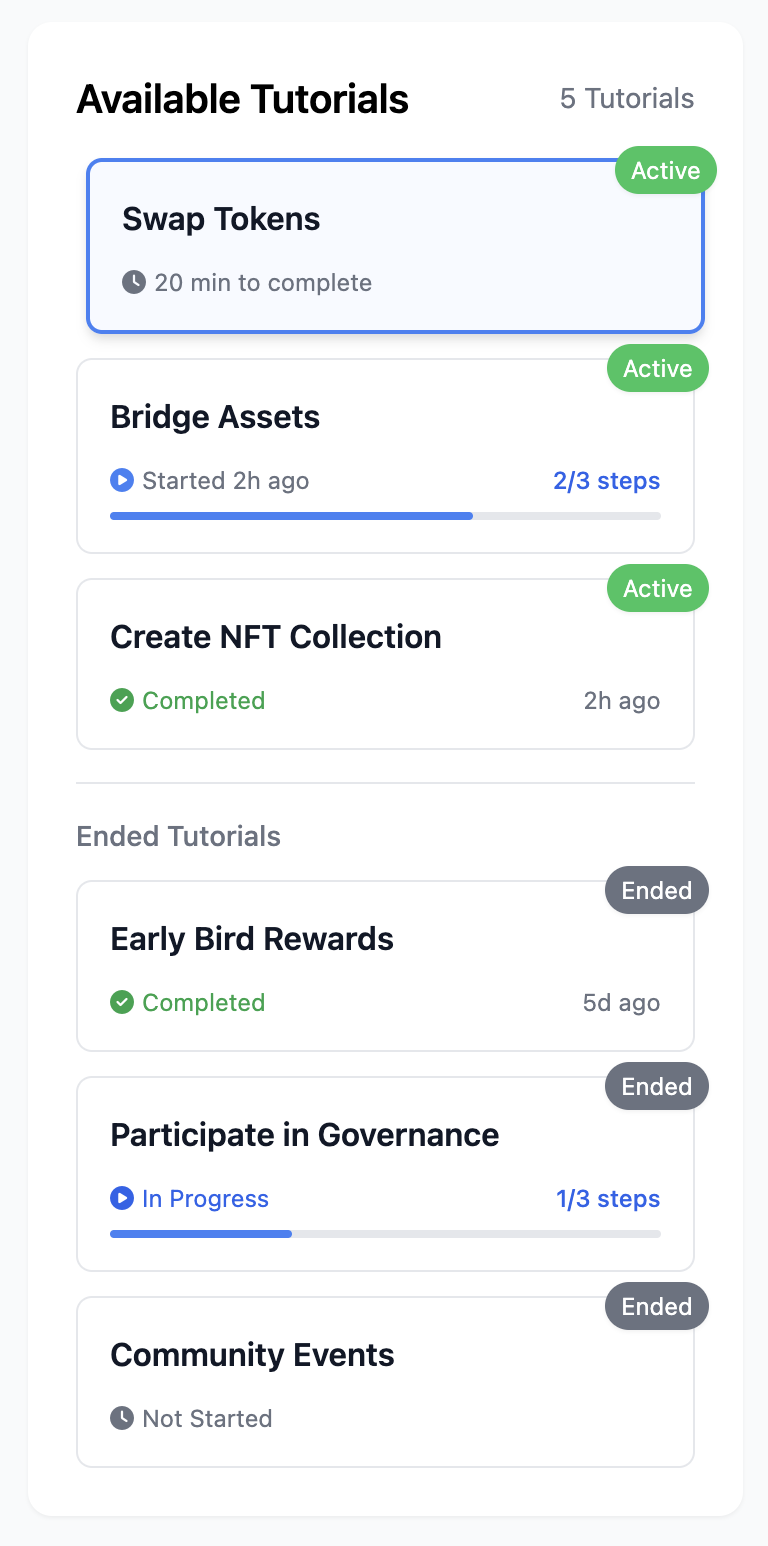


* 要将原有显示积分奖励的模块, 替换成 Potenail Airdrop from xxx ; xxx 是项目名称
* **交互**：
* 图片支持点击放大查看
* 任务卡片与 Community 的任务卡片交互逻辑保持一致
* 点击 Action 按钮, 跳转到对应的页面或打开任务弹窗
* 点击 Verify 按钮, 校验任务是否完成
* 任务完成后, 任务卡片和任务弹窗需要展示对应的完成样式





**3.1.2 教程卡片区**



**布局**：

* **教程标题**
* **教程状态**
* 根据教程的完成状态, 显示不同的**辅助信息**

**交互**

* **教程卡片点击**：点击后滚动至对应教程详情，同时高亮显示选中的卡片
* **进度滚动同步**：滚动查看教程时，右侧卡片列表自动同步高亮当前查看的教程

**3.1.2 状态系统**

状态定义表:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **状态类型** | **状态值** | **定义** | **触发条件** | **UI表现** |
| **教程活动状态** | 活跃 | 可参与并获得奖励 | 当前日期在活动范围内 | 正常色彩，功能完整 |
|  | 已结束 | 不可参与，仅供参考 | 超出结束日期或运营手动设置 | 灰色系，功能受限 |
| **用户完成状态** | 未开始 | 尚未开始任何步骤 | 首次查看教程 | 灰色进度条，时钟图标 |
|  | 进行中 | 部分完成任务 | 完成至少一个任务 | 蓝色进度条，播放图标 |
|  | 已完成 | 完成了教程内指定需要完成的操作 | 用户手动标记完成 | 勾选图标 |

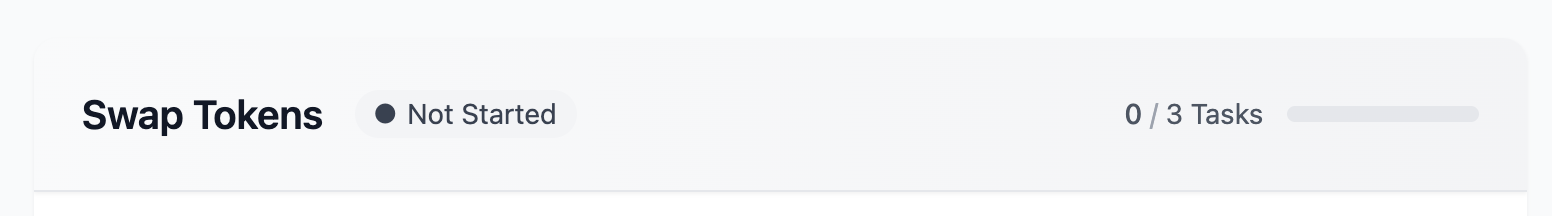
状态组合:

|  |  |
| --- | --- |
| **组合** | **主要特点** |
| 活跃 + 未开始 | 全功能可用，鼓励开始 |
| 活跃 + 进行中 | 突出进度，鼓励继续 |
| 活跃 + 已完成 | 展示完成信息，推荐新教程 |
| 已结束 + 未开始 | 显示已结束通知，内容参考 |
| 已结束 + 进行中 | 保留已完成进度，禁用未完成任务 |
| 已结束 + 已完成 | 显示历史完成信息 |

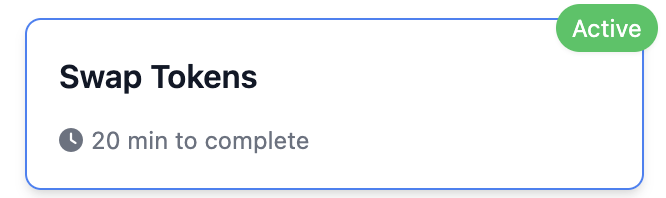
**3.1.3 活跃教程的UI与交互**

**活跃 + 未开始**

**教程详情页:**



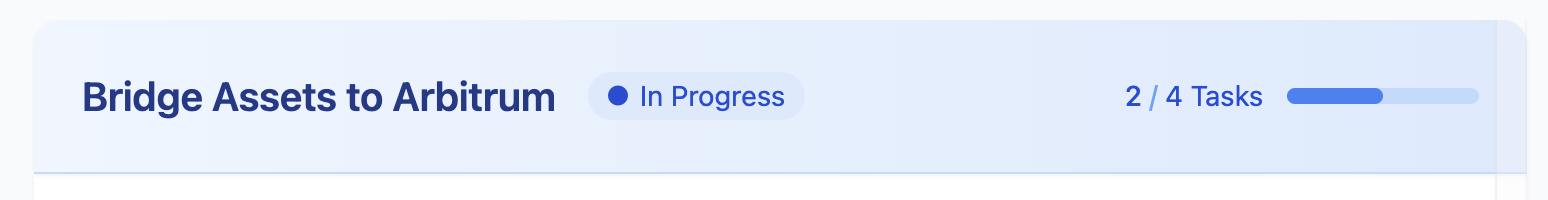
**教程卡片:**



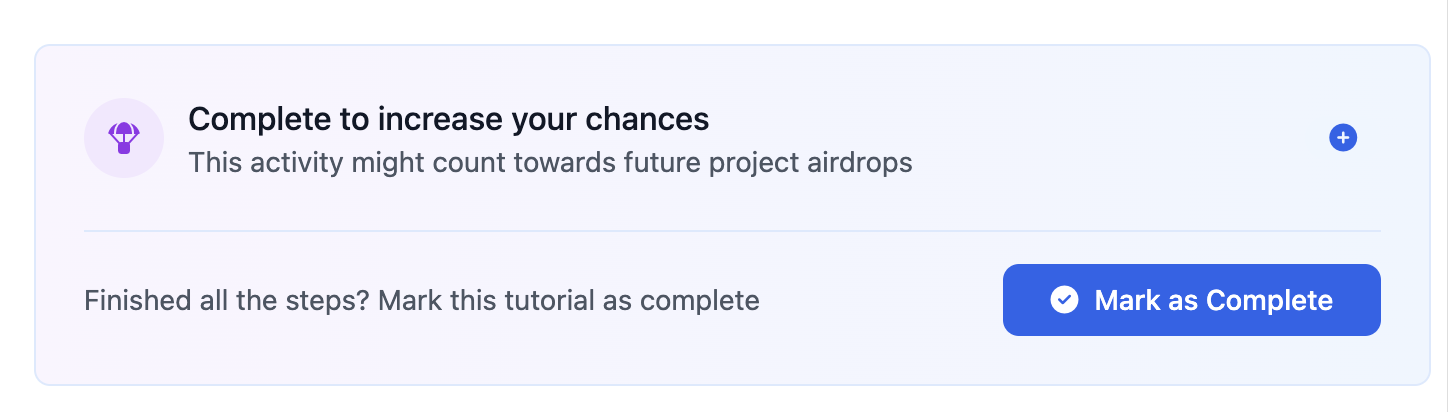
* **交互行为**
* 任务完成后自动更新状态为"进行中"

**活跃 + 进行中**

**教程详情页:**



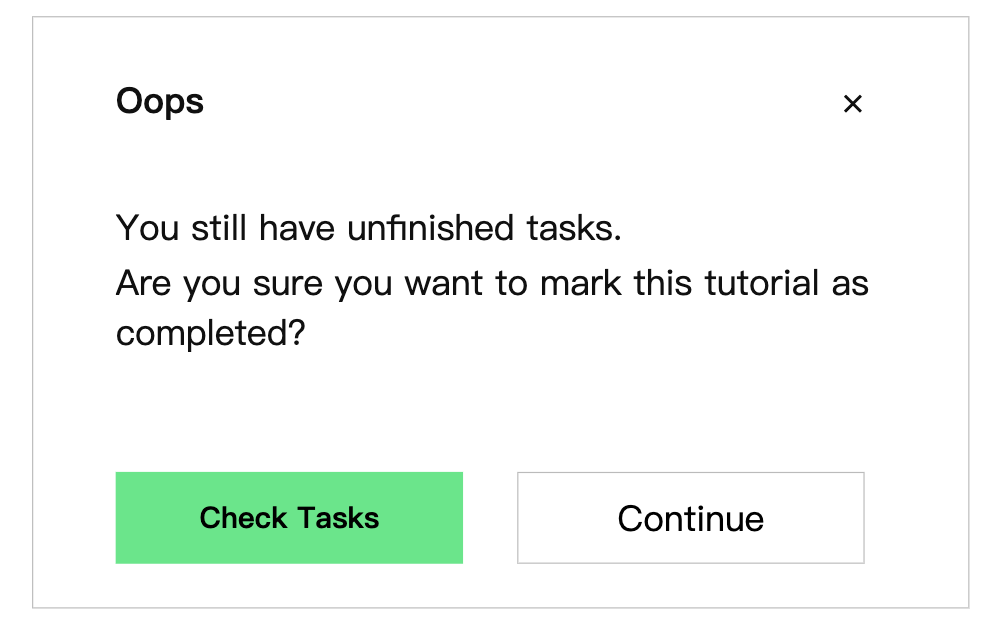
在教程详情页的底部提供**手动标记完成**的按钮:



**交互行为:**

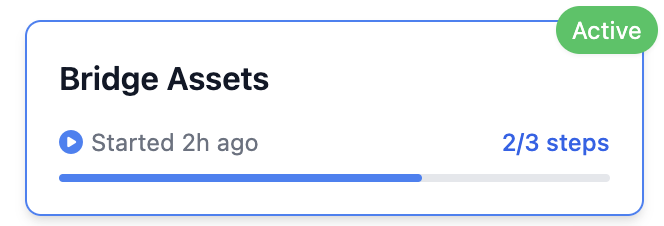
当用户未登录的情况下, 点击该按钮, 引导用户先登录, 登录完成后自动走以下判断逻辑:

1. 若教程内没有任务, 则点击后将教程状态标记为已完成
2. 若教程内有任务, 则点击后, 判断用户是否已完成所有任务:
3. 未完成, 弹窗跳出引导用户未完成所有任务, 是否仍需要标记为完成.
4. 点击 Check Task 后, 弹窗关闭, 页面定位至用户未完成的任务所在处(有多个的话, 定位到最靠前的一个)
5. 点击 Continue 后, 将教程状态标记为已完成
6. 点击右上角 ×, 弹窗关闭



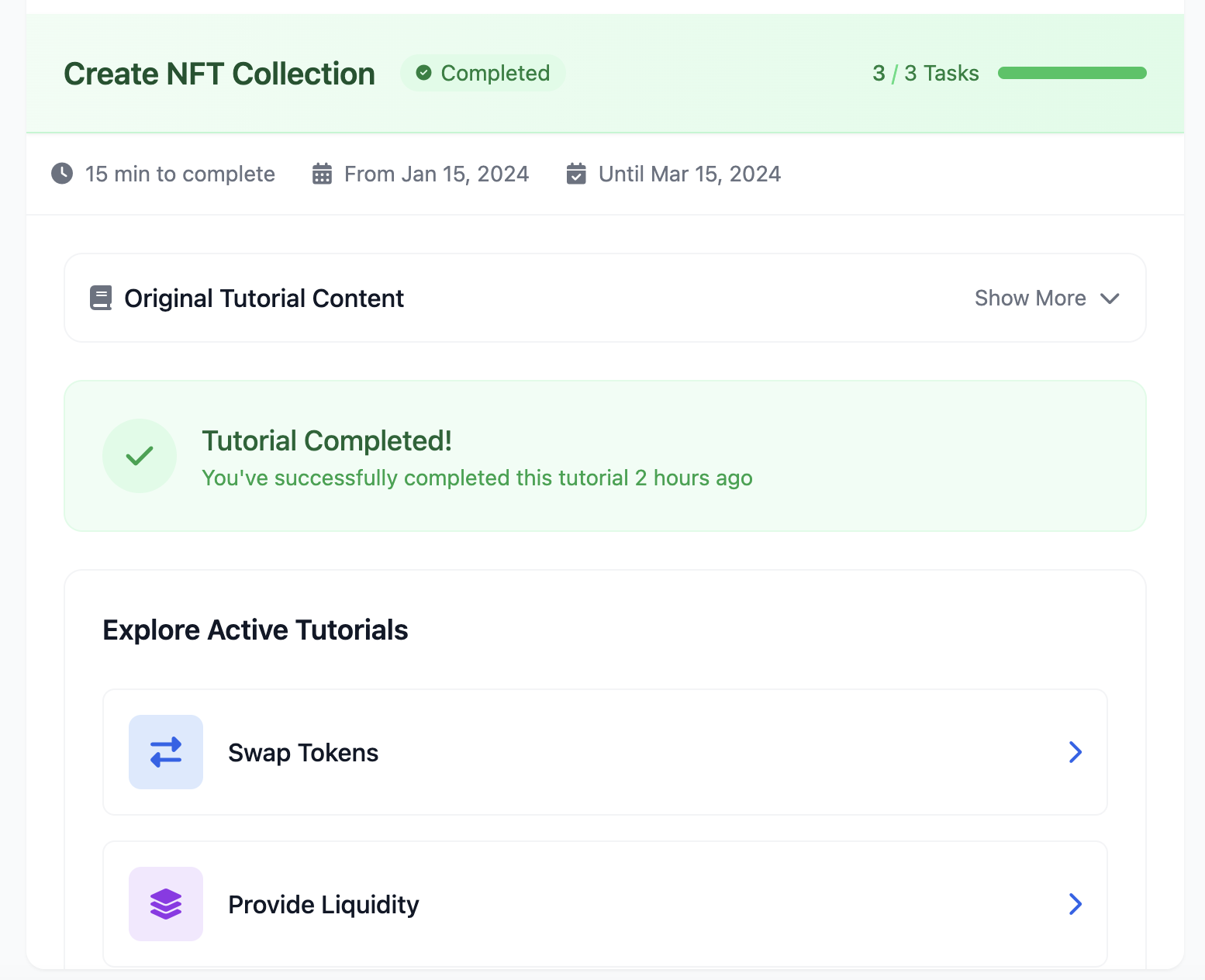
1. 已完成, 将教程状态标记为已完成

**教程卡片:**



**活跃 + 已完成**

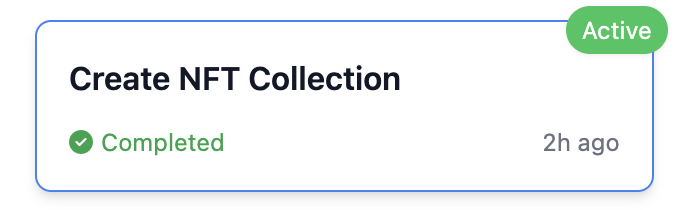
**教程详情页:**



**内容展示:**

* 收起教程
* 绿色完成标志及完成时间
* 下一步推荐区域突出显示, 推荐用户参与同一个项目下其他未完成且活跃的教程. 如果没有符合条件的教程, 则不需要显示该模块
* **交互行为**
* 点击教程展开/收起
* 通过推荐区导航至相关教程

**教程卡片:**



**3.1.4 已结束教程的UI与交互**

通用结束状态元素



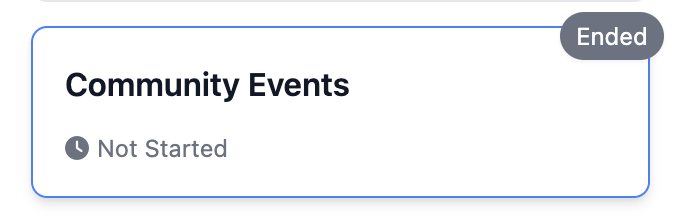
* 界面布局
* 教程标题右侧的状态标记为已结束
* 在教程内容板块中增加结束通知模块
* 置于内容顶部
* 显示教程结束时间信息

**已结束 + 未开始**

**教程内容:**

* 教程标题上隐藏任务进度
* 教程内容中隐藏所有任务模块
* 隐藏 Mark as complete 模块
* 在教程底部显示下一步推荐区域突出显示, 推荐用户参与同一个项目下其他未完成且活跃的教程. 如果没有符合条件的教程, 则不需要显示该模块

**教程卡片:**



**已结束 + 进行中**

**教程内容:**

* 禁用所有任务操作按钮

**教程卡片:**

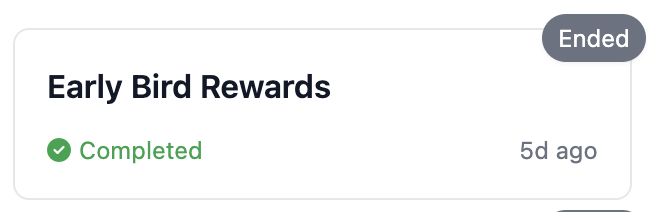


**已结束 + 已完成**

**教程内容:**

没变动

**教程卡片:**



* 完成标志使用灰绿色
* 显示完成时间

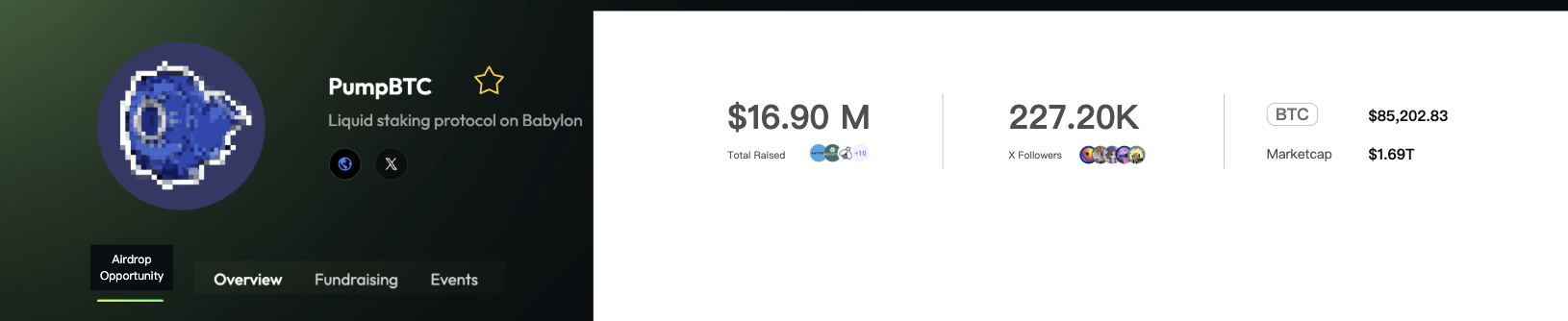
**3.1.5 数据统计**

需要以教程为维度, 记录以下指标:

|  |  |
| --- | --- |
| **指标** | **定义** |
| 教程浏览人数 | 查看过教程的用户数 |
| 教程任务参与人数 | 至少完成过1个任务的用户数 |
| 教程任务完成人数 | 教程下的任务全部都完成了的用户数 |
| 教程完成人数 | 用户标记了教程已完成的用户数 |

用户标记: user\_id

**3.2 Project Detail页头部优化**

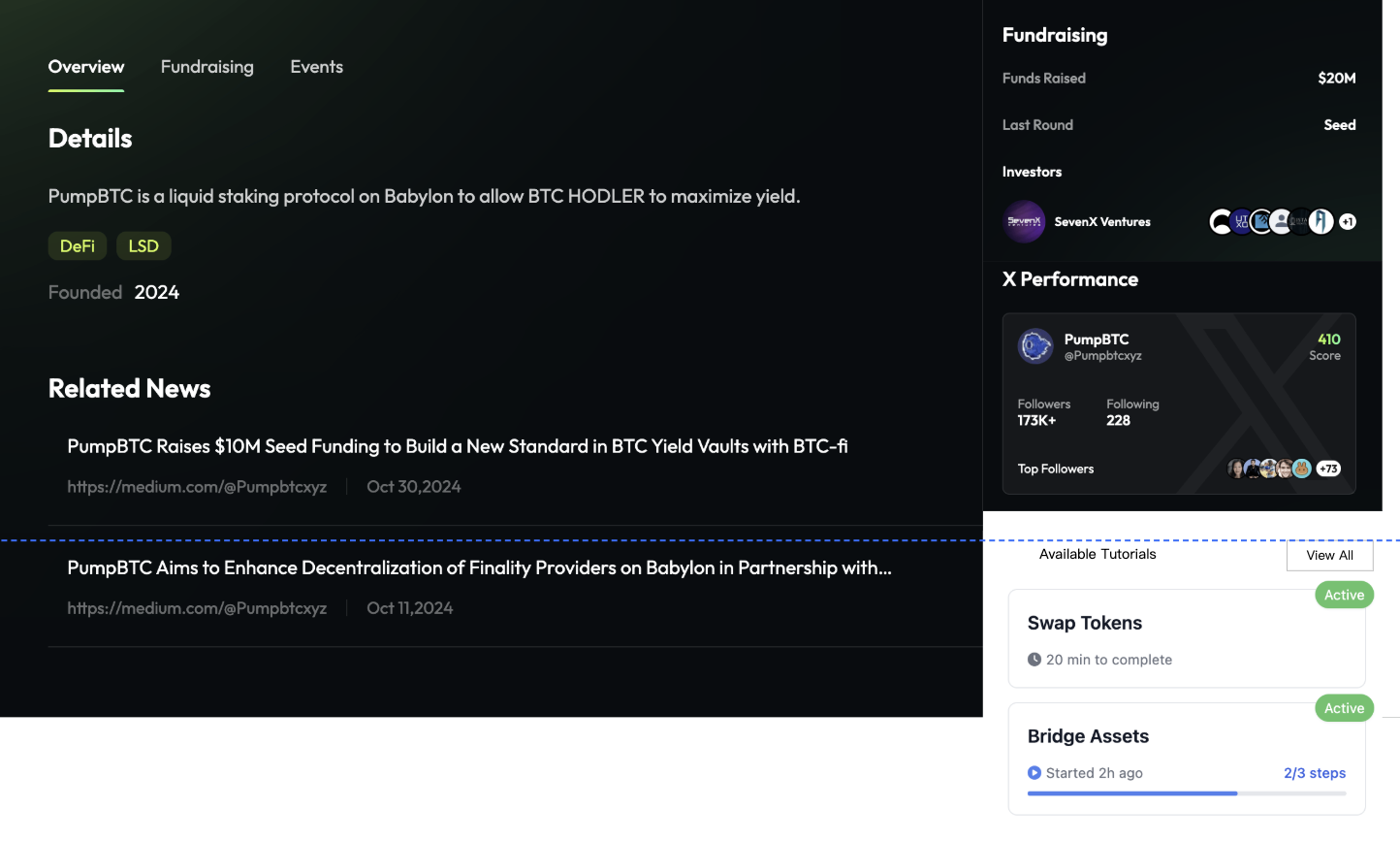


原来放在页面右侧代币, 融资, 推特模块去掉. 将模块的关键信息放在页面头部.

**布局:**

1. 融资模块
2. 融资总额
3. 融资机构
4. 推特模块
5. 推特粉丝
6. Top Followers
7. 代币模块
8. Token Symbol
9. 代币价格
10. 代币市值

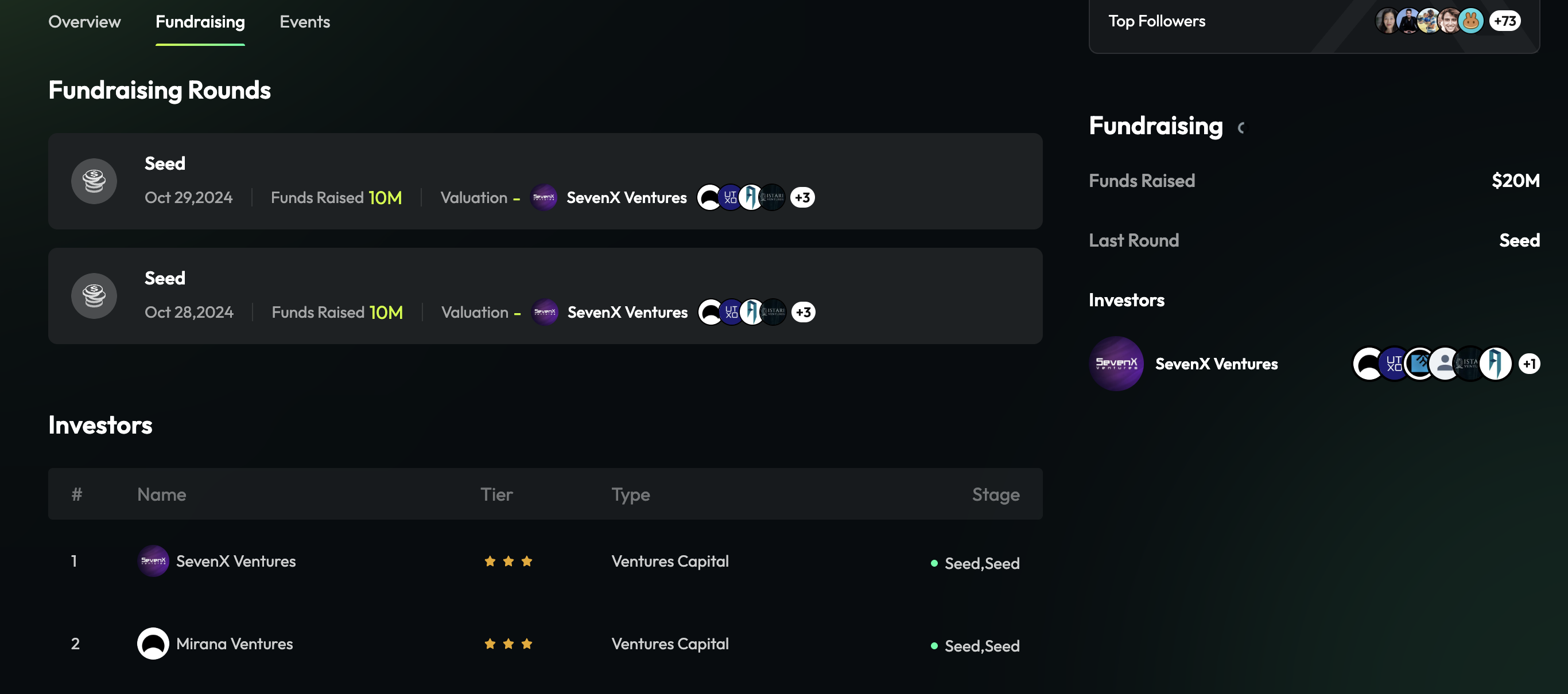
**3.3 Overview 优化**



**右侧布局:**

1. 原来的 Fund Raising 板块(样式需要重新设计, 强调融资金额)
2. X performance 板块(样式需要重新设计, 强调粉丝数)
3. 新增 Available Guides 板块, 放活跃且用户未完成的教程, 点击教程卡片后跳转至教程 Tab, 并定位至该教程; 点击 View all 跳转到教程 Tab

**3.4 Fundraising 优化**



**右侧布局:**

1. 放原来的 Fund Raising 板块(样式需要重新设计, 强调融资金额)

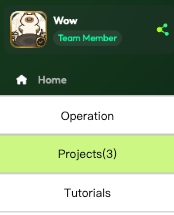
**3.5 去掉 Quest Tab**

如题

4. **后台管理系统，查看和编辑Tutorial（Sam）**

**准备工作**

1. 运营创建一个新的community，并且该community能在C端隐藏，不可见
2. 后台编辑Tutorial的板块，只有该community能看到和使用
3. 左侧菜单栏如下
4. Operation下面有Projects和Tutorials两个菜单
5. Projects后面的数字=在热力图上，并且todo>0的项目数（默认选项）

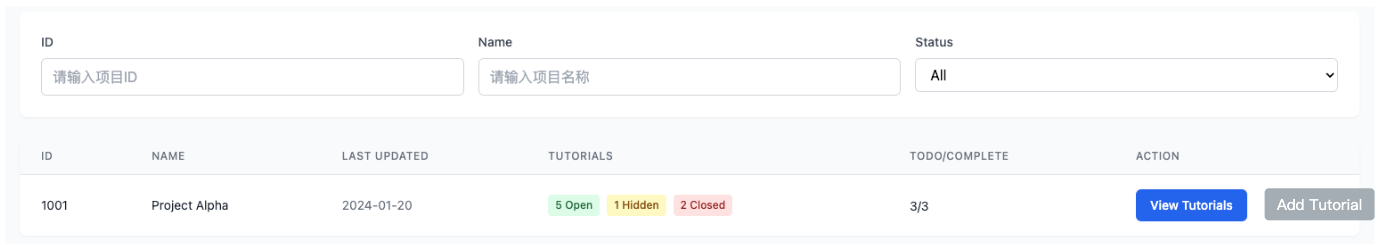


4.1 **Project列表页**

**进入条件**

左侧菜单栏，点击Projects

**页面布局**



**Project字段表**

1. ID
2. Name
3. Last Updated
4. Tutorials：Open/Hiden/Closed，比如5/1/2，代表5篇Open、1篇Hiden、2篇Closed
5. Hiden：所有Tutorial中，标题中带有（Galxe/Layer3/Layer 3/Zealy/Intract/Guild），设置为Hiden，C端不展示
6. Todo/Complete：3/3，代表3篇todo，3篇complete，todo的定义是Open+Hiden去掉标记为Complete剩下的
7. Action：操作按钮，按钮一个是View Tutorials，一个是Add Tutorial

**筛选项**

1. ID：精确搜索
2. Name：模糊搜索
3. Status：下拉选择
4. All：全部Project
5. Have Tutorial：筛选出有Tutorial的项目
6. Todo：筛选出todo>0的项目
7. Hot：筛选出在热力图上，并且todo>0的项目（默认选项）

**交互逻辑**

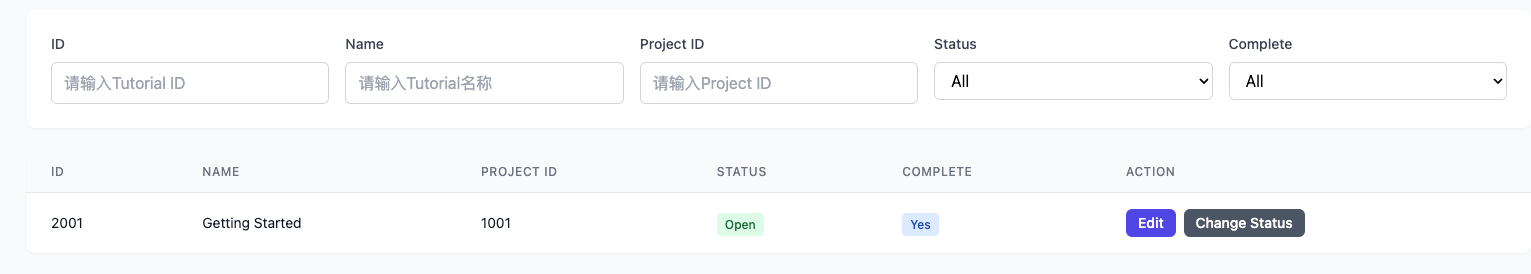
1. 点击View Tutorials，跳转到Tutorial列表页，并设置筛选出该Project
2. 点击Add Tutorial，打开Tutorial详情页，所有字段初始为空，标题和正文区域为必填，点击Save时创建一篇新的Tutorial
3. 搜索和筛选

4.2 **Tutorial列表页**

**进入条件**

1. 左侧栏点击Tutorials
2. 从Project列表，点击View Tutorial，带着Project ID作为搜索参数，来到该页面

**页面布局**



**Tutorial字段表**

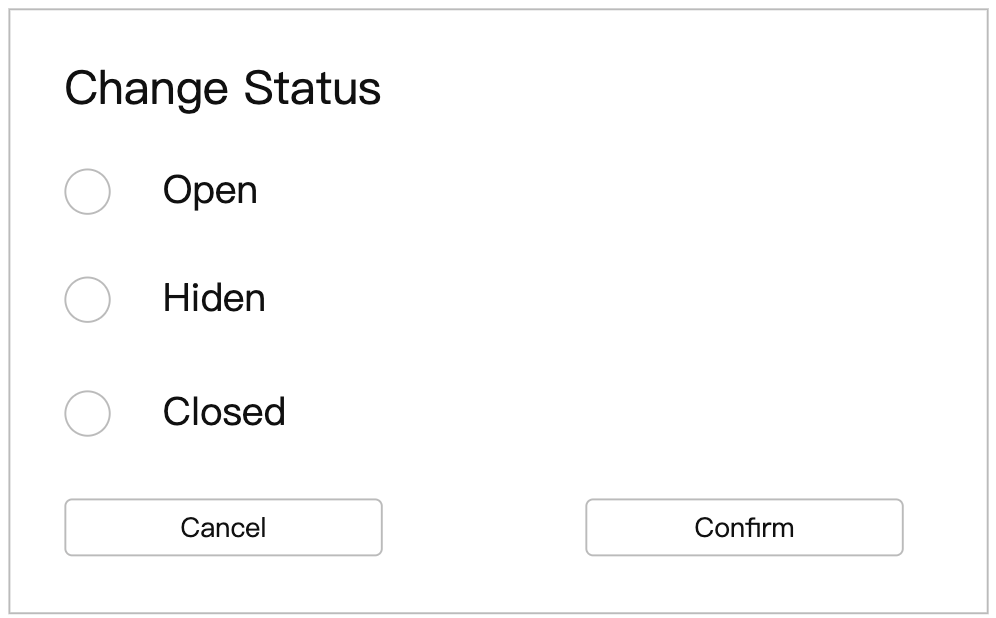
1. ID
2. Name
3. 所属Project ID
4. Last Updated
5. Status：Open/Hiden/Closed
6. Complete：Yes/No
7. Action：操作按钮，一个是Edit，一个是Change Status，一个是Top

**筛选项**

1. ID：精确搜索
2. Name：模糊搜索
3. Project ID：精确搜索
4. Status：下拉选择
5. All：所有Tutorial
6. Open
7. Hiden
8. Closed
9. Complete：下拉选择
10. All
11. Complete
12. Not Complete

**交互逻辑**

1. 点击Edit按钮，进入Tutorial详情页
2. 点击Change Status，弹窗修改状态
3. 状态只能单选
4. 选择状态后点击Confirm，修改生效

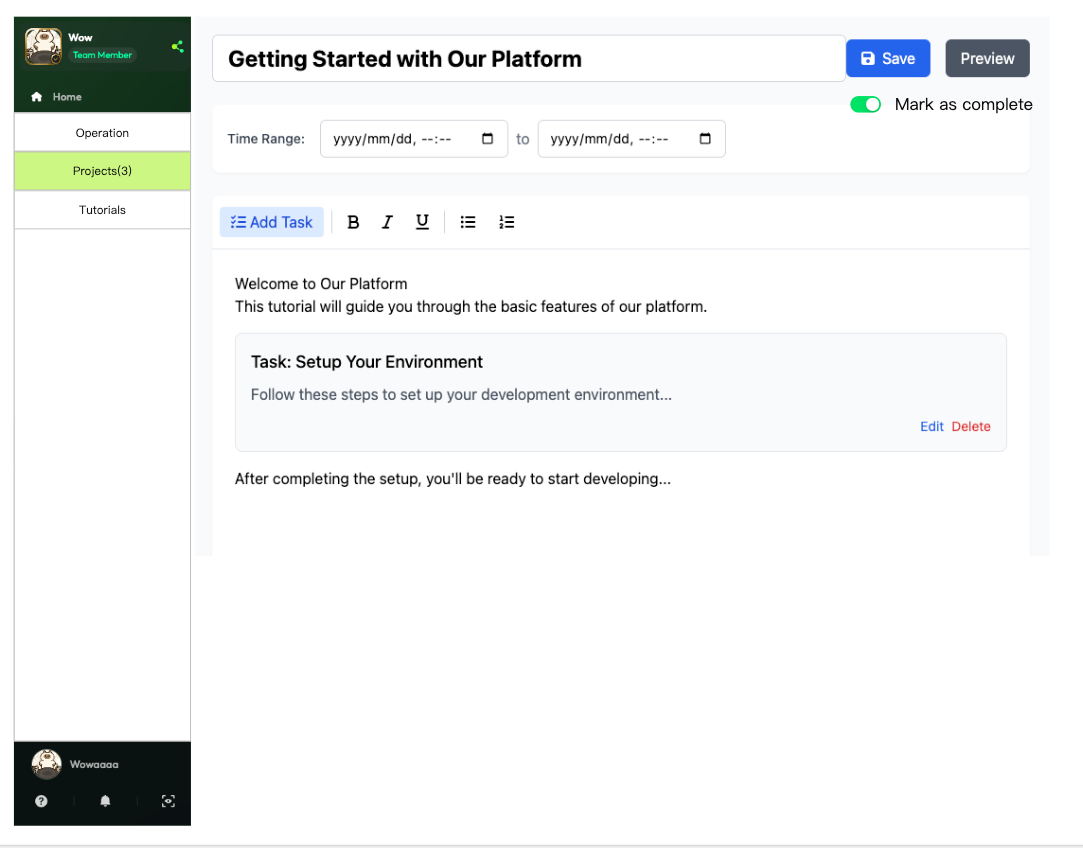


1. 点击Top，把该文档放在该Project Tutorial的第一个位置
2. 搜索和筛选功能

4.3 **Tutorial详情页**

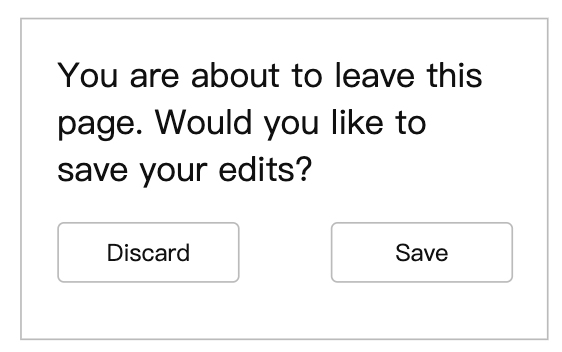
**页面布局**

1. Tutorial Name
2. 时间范围（Optional）
3. 正文区域：富文本编辑器，能直接渲染出来页面样式和task卡片
4. Add Task按钮

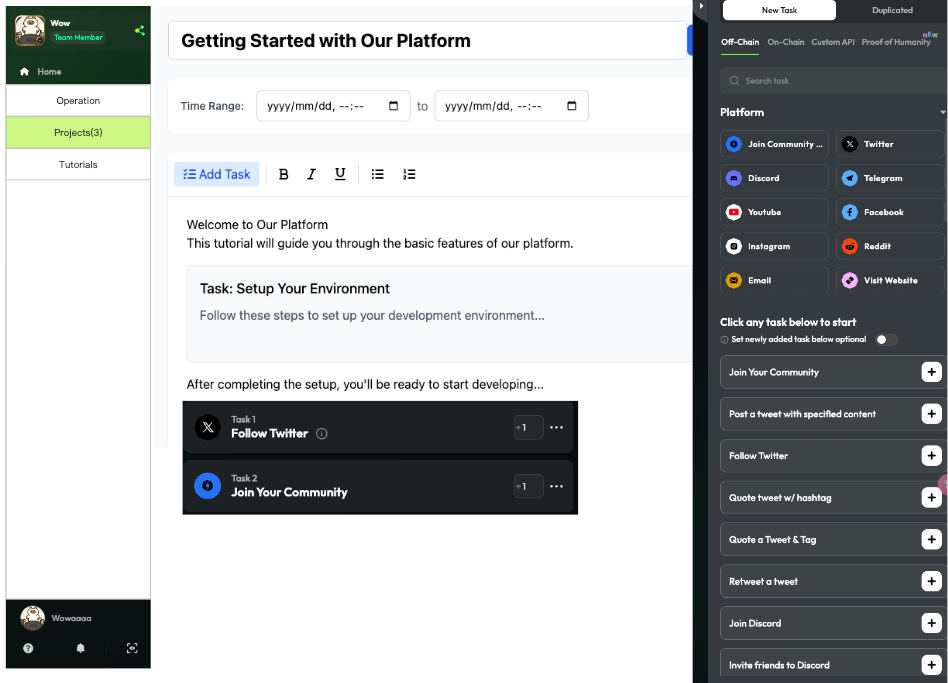


**交互逻辑**

1. 可以直接编辑标题、时间和正文区域，但是点击Save才保存修改；
2. 如果没有Save，要切换到其他页面，则弹窗提醒用户是否要Save；



1. 点击Preview，打开一个新页面预览C端效果；
2. 不保存技术上是否可以预览？成本高不高？
3. 点击Add Task，右边侧滑出来添加task的组件
4. 右侧点击添加某个task，直接插入在正文区域中光标所在的位置，并且立即收起右侧的添加task组件
5. 其他添加task流程，与当前quest中添加task保持一致
6. 这里添加的task，需要和community中的task一样，有独立的链接和task id
7. 任何task，都可以随时编辑，不做任何限制
8. 加积分的功能暂时禁用掉，后续设计好积分和运营体系再开放
9. 如果添加的task，与任意教程中的某个task的动作和参数完全一致，则不用新建task，而是用相同的task id，用户只要做完了task，在其他教程中看到同一个task，状态是已完成



1. 点击Save后，保存修改并生效，C端可以立即看到新文档（这里要处理一下缓存机制，要主动清掉C端该Tutorial的缓存）
2. 打开Mark as complete，文档Complete状态变更为Yes，关闭Mark as complete，文档Complete状态变更为No

5. **Tutorial分析+Task抓取+Tutorial自动生成**

5.1 **Tutorial分析**

5.2 **Task抓取**

5.3 **Tutorial自动生成**

6. **项目入驻流程**

7. **运营体系**

[热点+运营需求](https://ontology.larksuite.com/wiki/EYSTwFQX3iTqdkkCxDzuFVkvskt)

8. **增加Tutorial数据源**

8.1 **Dropstab**

8.2 **Galxe**