所谓的父子联系是指：

创建数据的时候，子数据是建立在父数据之上的。两个要有相关性。

一般的场景是：

* 新建时父数据与子数据在一个页面上创建。
* 新建子数据时，父数据已建立完成。但子数据是父数据中的某一项。（并不是关联关系，而是从属关系）

常见场景1：同页增删数据。



常见场景2：不同页增删数据。



解决方案：

注：如果出现类似的场景，我们会更偏向于建议使用**场景2**的方式来展现，但场景1也能支持，支持的方式为：

在card中设置显示某个页面。

整体的解决方案为：

1. 在提交页面中，可以设置父id中。这样，提交的所有数据都存一个父id。
2. 对于子数据，比如任务管理，需要创建子流程，设置子页面，然后在总页面中可以引用子页面。（这不应该属于逻辑，但因为我们之前的逻辑中一个流程生成一个数据库，所以暂时按
3. 这个方式来做）。
4. 查找时，可通过父id来查找。

其它变化

* 打开和导入页面时，可以导入所有页面。
* 所有生成的VUE中，都需要一个pageid, 在生成时，由commonaction写入
* Querybysk中新加一个参数，pageid，如果有这个参数时，则先取page中的flow和node信息，再做查询。这样，保证导入的列表，label等能取到对应的数据

对于url参数中的itemID，我们不必担心导入详情页时会产生itemid错误的情况，因为这本身就是业务逻辑的错误。

但我们需要考虑的是，对话框中显示页面时，因为url没有变化，产生的问题。

现在的对话框，在globlconfig存了数组，每个对话框是否显示都在里面。对话框自身可以存入pageid，每个页面也知道自己的pageid。。。。当从列表打开对话框的时候，把itemid存到globlconig

Current-z-index = 3000 // 每个对话框都会从这个基础之上加1.

[

{

Pageid: “true”

“Itemid”: item;

}

]

这样，详情页在打开的时候，先看看自己是否被显示在对话框中，如果是，从对话框对应的item中取id，而不是从URL中取。