# Vue界面开发规划

## 总体规划

根据《Vue框架需求》，整个工作量分为业务和控件两部分。其中业务是最重要的，相对而言，控件的需求不紧急，只要有text控件模拟即可。

因此开发首先以业务开发为准。业务中List，Insert，Update，Detail是主体核心，应当优先开发。而List界面是入口，所以必须第一个开发。

## 开发顺序

### 列表页面

#### 两个指令

* v-tk-dialog-url
* v-tk-float-url

#### 列表页面架构

* 展示查询区块，以及通过Ajax从后端获取数据，并绑定在列表中显示
* 解决操作栏的显示，以及应用相应的操作权限
* 支持分页，排序操作
* 支持查询
* 支持多Tab

### 编辑页面（一）

通过指令v-tk-dialog-url或者v-tk-float-url进入编辑界面

先不考虑控件，所有字段都是text

* 展示编辑页面，通过Ajax从后端获取数据，绑定显示
* 编辑后，向后端提交
* 后端如果返回校验错误，在text上以合适的行为显示
* 后端处理成功，返回指令，根据指令进行恰当的行为（指令需要弄成插件方式，方便以后扩展）
* 前端控件布局，处理好一行一个控件，两个控件，三个控件，四个控件的布局方式

### 详情页面（一）

通过指令v-tk-dialog-url或者v-tk-float-url进入编辑界面

* 展示详情页面，通过Ajax从后端获取数据，绑定显示
* 能绑定详细页面的按钮操作
* 前端控件布局，处理好一行一个控件，两个控件，三个控件，四个控件的布局方式

### 编辑页面（二）

* 考虑多记录的问题，多记录有两种布局方式，一种字段少，可以用Table布局。如果字段多，还是按照NormalTable布局
* 多表的布局方式，无论单记录还是多记录表，考虑平铺，Tab，以及混合的方式

### 详情页面（二）

* 考虑多记录的问题。当前多记录的方式是留个架子，等到点击Tab时，发现没有数据，通过Ajax从后端获取。现在类似，可以考虑生成一个架子，等到点击Tab时，发现没有数据，通过Ajax从后端获取Json，然后同架子绑定
* 多记录的时候，需要考虑支持行记录的操作（如修改、删除，最好类似于列表界面），同时考虑全局的操作（如新增）
* 如果可能，考虑多记录不是按需获取，而是直接从Detail的Json中获取的可能性
* 多表的布局方式，无论单记录还是多记录表，考虑平铺，Tab，以及混合的方式

### 树页面

* 树型展示，最好能用现成的树型进行封装，留有点击扩展，支持节点上下移动，支持拖拽操作
* 树型展示，需要通过Ajax从后端获取数据，展示数据（返回的数据可能是一层的，也可能是多层的）。要求可以在展示的时候，自动跳到选中节点的位置。
* 树型数据维护，数据维护可以采用前面的对话框或者侧推方式来维护，也可以考虑直接嵌入右边区域页面，这个看实际开发的容易程度，但是有一点，不许跳转到新的页面在进行维护了。此外，维护完毕，左右可能都需要刷新。新建后取消，这种特例需要考虑。