MongoDB术语和关系型数据库术语对照

- 表(table) -> 集合(collection)
- 行(row) -> 文档(document)

MongoDB的基本设计原则以及和关系型数据库的主要区别

(http://blog.mongodb.org/post/87200945828/6-rules-of-thumb-for-mongodb-schema-design-part-1)

1. 尽量采用文档嵌套的结构，除非有其他重要的原因。
2. 如果被嵌套的文档需要进行独立的操作，就是选择不嵌套的一个重要原因。子文档需要有存在于自己独立的集合里面。
3. 如果子文档有超过一，两百个实例，不要嵌套在主文档的数组里
4. 不要害怕在程序里做集合之间的连接(join)。如果检索设置的合适，性能不会比关系型数据库的表连接慢多少
5. 如果一个字段多数都是读操作，很少更新，那就比较适合非规范化。就是引入一定数据重复性
6. 数据库的设计完全由应用程序决定

Inkanban数据库的设计

基本的集合：

1. 表单模版（forms）

我们系统可以支持的所以表单模版都存在这里。用户从里面可以选择他们需要都表单。下图是一个捻股工单表模版的例子（截图，没有显示完整的结构）。所有表单基本都以树形结构表示，数据定义都来自Excel表格。其中也包含表头的定义(header)。

每个表单模版有一个selected字段表示是否这个公司选择了这个表单。由于MongoDB并没有固定好的“表”结构，所以我们也可以支持在每个节点加入selected字段，这样用户可以自己选择不只是那个表，还可以定义在一个表里面，那些字段可以选择。这样基本满足用户对一个表单的基本要求。

有些字段是可重复的，比如股捻向可能有n层，我们在第1层定义一个repeat: n的属性，表示这个字段是可重复的。

MongoDB在数据库的定义很灵活，其实全是由程序来决定。所以如果公司要求添加新的表单，在数据库里没有任何约束，关键是要程序里面可以支持。

需要完善的地方：

1. 目前我们还没有支持如何在表单里定义数据的传承型，比如捻股的很多字段来自其他表，我们需要设计每个字段能够引用其他表的数值。这也应该会影响到用户选表时的操作，因为这些传承数据隐含着表单之间的依赖性。
2. 可能还需要添加一些新的属性，比如valid，表示是否这个表可以选择；还有display，表示是否在非树状结构中显示表单时显示，比如在数据表显示一个表单时，显示哪几个字段，要不然太多字段，无法在数据表里显示。

```
{
    name: "捻股工单",
    selected: false,
    header: [
        {name: "form-no", text: "表单编号"},
        {name: "author", text: "编制"},
        {name: "created", text: "编制日期"},
        {name: "version", text: "版本号"},
        {name: "reviewer", text: "审核"},
        {name: "review-date", text: "审核日期"}
    ],
    body: [
        {
            text: "生产要求",
            type: "folder",
            order: 1,
            children: [
                {text: "制造号"},
                {text: "制造序列号"},
                {text: "机台号"},
                {text: "班别"},
                {text: "班长"},
                {text: "操作工"},
                {"text": "调产时间"...},
                {text: "标准产量", unit: "m/H"},
                {"text": "计划产量"...},
                {"text": "实际产量"...},
                {"text": "收线工字轮"...},
                {"text": "异常停机时间"...}
            ]
        },
        {
            text: "工艺要求",
            type: "folder",
            order: 2,
            children: [
                {text: "股结构"},
                {text: "强度", unit: "MPa"},
                {
                    text: "股捻向",
                    type: "folder",
                    children: [
                        {text: "第1层", repeat: "n"}
                    ]
                },
                {"text": "股径公差"...},
                {"text": "股捻距"...},
                {"text": "股内钢丝直径（由内到外）"...},
```

2. 表单实例(form_insts)

用户选好表单模版后，会生成新的具体表单实例，比如车间主任新创建一个捻股工单，这就是一个表单实例。表单实例的结构和表单是一致的，里面保存用户定义的每个字段的具体数据。也就是除了text之外，还有value。而且repeat: n也被替换成具体的层数，层数是用户创建这个工单实例是必输入的。

3. 用户(users)

保存创建好的用户。比如下图是一个用户的数据。里面有一些用户的基本信息，如姓名，邮件，所属角色(roles)，所属机构(orgs)，授权的表单(perms)，创建的表单实例(forms)。


```
{
    "_id" : ObjectId("55ef468bae73672d9c2d5575"),
    "username" : "1111",
    "first_name" : "三",
    "last_name" : "张",
    "email" : "zhang@insevo.cn",
    "roles" : [
        "车间主任"
    ],
    "orgs" : [
        "一车间"
    ],
    "perms" : [
        {
            "form_id" : "55fdb73b391dc4893b5b8345",
            "form_name" : "捻股工单",
            "read" : true,
            "write" : true
        }
    ],
    "forms" : [
        ObjectId("55fdde0cb713913b0647509f"),
        ObjectId("55fde483ee571810075ae142"),
        ObjectId("55fde49dee571810075ae143"),
        ObjectId("55fde600ee571810075ae144"),
        ObjectId("55fded6eee571810075ae145"),
        ObjectId("55fded86ee571810075ae146")
    ]
}
```

4. 系统数据(enums)

主要用来保存一些静态数据，比如所有用户角色，所有机构。以后还可以存一些其他静态数据。

5. 对话数据(sessions)

用户登录后的对话数据，cookie数据等。