每种设备的保养内容不一样，需要填写的也不一样。

每家医院都不一样。如四院关于对监护仪二级保养参数如《多参数监护仪二级保养标准.xlsx》

这就需要有一种灵活的定义方式去定义要保养的内容。

保养的内容定义和结果都使用json字符串来保存。数据库中保存着每种类型的设备在某一个医院的保养内容。在保养界面，前端需要根据后端返回的该设备类型的保养内容生成保养表单。

用户填写完以后，把整个填写结果存储到数据库字段中去。下次查询结果的时候，需要根据结果加载页面。不单单生成内容，也要生成结果。**注意，新建保养的时候，由于结果是空的，所以界面生成根据设备类型的保养内容模板来生成。如果，已经保养过了，则根据保养内容模板来生成界面，然后根据数据库中存储的结果，来一一对应赋值显示在界面上。这么设计的目的是，设备类型的保养内容也许会修改，修改以后，之前来的保养结果，要能正常显示。如果添加了检测项，则新加的为空，老的还应该能够正常加载。**

# 保养内容的定义

保养内容可以分为2级，保养项目（subject）和保养记录（item），参考《多参数监护仪二级保养标准.xlsx》，只有两级，不再过滤灵活。

|  |  |
| --- | --- |
| Key | 描述 |
| subjects | 一组保养项目，数组 |
| subject\_name | 保养项目的名称 |
| items | 保养项目下面的一组保养记录 |
| item\_name | 保养记录的名称 |
| control\_type | 控件类型，和html 控件名字一样 |
| item\_key | 定义每一个item的key，全局唯一，用来存储和查看结果的时候赋值结果 |
| item\_value | 保养记录的结果 |

前端通过json，去构造界面

**示例如下：**

{

"subjects": [

{

"subject\_name": "外观",

"items": [

{

"item\_name": "1.监护仪外壳是否干净、无污迹",

"control\_type": "checkbox",

"item\_key": "item\_1"

},

{

"item\_name": "2.外壳、按键、旋钮、接口和附件是否无机械性损坏",

"control\_type": "checkbox",

"item\_key": "item\_2"

}

]

},

{

"subject\_name": "电气安全",

"items": [

{

"item\_name": "1.电源线是否无磨损、绝缘性良好",

"control\_type": "checkbox",

"item\_key": "item\_3"

},

{

"item\_name": "2.测试电池性能，并对电池进行充放电维护",

"control\_type": "checkbox",

"item\_key": "item\_4"

}

]

},

{

"subject\_name": "性能测试",

"items": [

{

"item\_name": "1.监护仪时钟是否正确",

"control\_type": "checkbox",

"item\_key": "item\_5"

},

{

"item\_name": "2.按键功能检测，保证所有按键功能正常",

"control\_type": "checkbox",

"item\_key": "item\_6"

},

{

"item\_name": "3.心电、血氧、血压等参数检测是否正常",

"control\_type": "checkbox",

"item\_key": "item\_7"

},

{

"item\_name": "4.电池是否处于良好状态并工作正常（电池可支持20分钟以上）",

"control\_type": "checkbox",

"item\_key": "item\_8"

},

{

"item\_name": "5.开机运行过程中是否有异常声音，保证平稳运行",

"control\_type": "checkbox",

"item\_key": "item\_9"

}

]

},

{

"subject\_name": "报警测试",

"items": [

{

"item\_name": "各监测指标（心电、血氧、血压、呼吸等导联声光报警功能是否正常",

"control\_type": "checkbox",

"item\_key": "item\_10"

}

]

}

]

}

# 保养结果的存储

结果就是上一节内容的结果，就是在每个item下面加一个item\_value。使用内容中的item\_key和item\_value存储在数据库中。

示例如下：

{

"result": [

{

"item\_key": "item\_1",

"item\_value": true

},

{

"item\_key": "item\_2",

"item\_value": false

}

]

}

# 数据库设计修改

保养设备模板：tb\_pm\_device\_template表，根据医院和设备类型id获取模板，也就是保养内容。

在tb\_pm\_case表中添加了pm\_result字段，用于存储结果。这个字段是：保养结果，是一个json字符串，模板来源于tb\_pm\_device\_template表。

# 推荐接口

1. 根据医院和设备类型id获取保养模板。
2. 修改存储保养记录的接口，把json字符串存储到pm\_result字段中。

# 测试数据

select \* from tb\_device where device\_id = 100002894;

select \* from tb\_pm\_device\_template；