

SRTP 实验数据表格模板

在本模板（1.0）中，我针对不同实验为代码做了一定程度的结构优化和逻辑优化，使之更适用于对不同实验进行高效的更改和适配。我们仍旧只对 component 目录下的 MyTable.jsx 做更改。如果对以下两类内容有了一定程度的了解，那么在更改表格的过程中遇到的阻力就会大大减小：

1. HTML 中 `<table />` 标签及其子标签（`<tr />`、`<td />`、`<th />` 等）的使用
2. `<input />` 标签的使用（获取输入的值）

我们本次以 **光栅常数的测量** 实验为例进行解释说明。

一、需要更改的参数



上图粗略展示了代码中部分需要更改的参数在 UI 对应的位置，在本模板中，大家需要改动的共有14个参数，代码中已用注释分割开来。在修改时 请务必确认以下的所有参数都满足要求。以下是简要说明。

| 序号 | 变量名/函数名 | 数据类型/ 函数返回值数据类型 | 说明 |
|----|--|---------------------------------|--|
| 1 | <code>tableTitle</code> | 字符串值 (<code>string</code>) | 表格标题 |
| 2 | <code>MyTableCell</code> 类中的 <code>render()</code> 函数 | jsx组件 | 数据单元格的排版 |
| 3 | <code>elementCount</code> | 数值 (<code>number</code>) | 数据单元格行数 本实验需要 4 个数据单元格行，分别是波长为参考光波长时的左游标读数、右游标读数和波长为带测光时的左游标读数、右游标读数。故设定 <code>elementCount = 4</code> |
| 4 | <code>allowGrowAndShrink</code> | 布尔值 (<code>boolean</code>) | 是否允许表格增删数据列 需要提到的是，若此项设为 <code>false</code> ，则增删表格列的按钮也会消失。借此提醒，一般情况下请不要擅自变动可编辑区域外的内容 |

| | | | |
|----|--|---------------------------------------|---|
| 5 | <code>dataHeader</code> | 数组 (object) | 初始的数据单元格的表头 |
| 6 | <code>changeTwoSides</code> | 数值 (number) | 数据单元格列数增长/收缩方式 设为 0 表示左右同时增长，如本实验中的数据需要由 <code>[-2, -1, 1, 2]</code> 变更为 <code>[-3, -2, -1, 1, 2, 3]</code> ，即为两侧增长，收缩同理（仅 <code>allowGrowAndShrink = true</code> 时本参数生效） |
| 8 | <code>buttonTitles</code> | 数组 (object) | 增删表格按钮的文字（仅 <code>allowGrowAndShrink = true</code> 时本参数生效） |
| 7 | <code>getNewRowEdge</code> | 可变对象 (number string boolean) | 增长后的新 数据单元格 表头的新元素 注!!! 该方法一般不必更改!!!。当遇到需两侧增长表格数据列的情形又需要自定义新表头元素时才对此值进行更改（仅 <code>allowGrowAndShrink = true</code> 时本参数生效） |
| 9 | <code>max</code> | 数值 (number) | 最大的数据单元格列数 若学生已通过按钮增加表格列数到此最大值，则再次点击按钮将不会增加表格数据列。如在本实验中，最多只允许学生记录5个明纹组，即表头为 <code>[-5, -4, -3, -2, -1, 1, 2, 3, 4, 5]</code> 共10列，故本实验中的 <code>max</code> 值设为 10（仅 <code>allowGrowAndShrink = true</code> 时本参数生效） |
| 10 | <code>min</code> | 数值 (number) | 最小数据单元格列数 若学生已通过按钮删减表格列数到此最大值，则再次点击按钮将不会增加表格数据列。（仅 <code>allowGrowAndShrink = true</code> 时本参数生效） |
| 11 | <code>tableBody()</code> | jsx组件 | 表格的排版 这是表格框内的表格排版。需要对 <code><table /></code> 标签及其子标签的使用有大概的了解。数据绑定见下方 表格排版调用数据对象 中对 <code>MyTable</code> 类的介绍 |
| 12 | <code>calculateCell(item)</code> | 可变对象 (number string boolean) | 计算每个单元格代表的值 |
| 13 | <code>judgeRule()</code> | 布尔值 (boolean) | 判断单元格中输入框中输入的值是否合法 若满足该条件则将输入框输入的值暂存至内存，否则不存入数据并在UI上进行视觉警示 |
| 14 | <code>experimentData(list, otherData)</code> | (指定要发送的实验数据) | 指定要发送的实验数据 |

二、表格排版调用数据对象

涉及到数据录入目前用到的只有输入框 `<input />`，故我将输入框和数据录入行为进行了函数的“封装”。经过改良，大家只需专注于如何调用这几个函数，而不必关心其细节。下面针对 `MyTableCell` 类（在 `MyTable` 中渲染单元格的类）和 `MyTable` 类分别介绍。

1. `MyTableCell` 类

该类渲染的是 数据单元格 中的控件。本例中需要“xx度xx分”的格式，故有下方代码片段

```
/**
 * 计算每个单元格代表的值
 * @param {object} item 单元格中输入框的值的数组
 */
calculateCell = (item) =>
    item.value[0] + item.value[1] / 60

/**
 * 单元格中的输入框输入的值是否合法的判定规则
 * @param {number} num 单元格中输入框的序号
 * @param {object} value 单元格中相应序号的输入框的值
 */
judgeRule = (num, value) =>
    (num == 0) && (value >= 0 && value < 360) || (num == 1) && (value >= 0 && value < 60)

//指定要发送的实验数据
```

其中使用 `this.getTableDataInput(num)` 方法将输入框放入单元格，其中 `num` 参数为单元格中 **输入框的序号**。

例如在本实验中，`num` 为 0 的输入框（即 `this.getTableDataInput(0)`）表示“度”，而 `this.getTableDataInput(1)` 则表示“分”。相应地：

- 在 `calculateCell(item)` 中用 `item.value` 表示单元格输入框的数组，数组下标即为 `num`。通过图中的算式计算得到一个单元格代表的角度值。
- 在 `judgeRule(num, value)` 中，`num` 为 **输入框的序号**，`value` 为 **该序号的输入框的值**。本实验的 `judgeRule(num, value)` 实现的是，同学在单元格中第一个输入框中的值只能在 $0 \leq value < 360$ 范围内；第一个输入框中的值只能在 $0 \leq value < 60$ 范围内。

2. MyTable 类

我们在排版时涉及到的输入框已封装到以下两处，这两处只用在 `tableBody()` 下。下面分别介绍其用法。

- ```
o this.state.tableData[num].map(item => this.getTableCell(item))
```

这一行渲染的是数据单元格的行，这一行看似既长又复杂，实则仅需更改 `num` 一个参数，至于其他处则不必更改。

- o `this.getOtherDataInput(num, hint)`

我们注意到在数据单元格区域外也有一些其他需要同学输入的数据。我们用该方法生成对应的输入框。当这些输入框不在表格中时，就把它们写在“其他输入框”区域

**SRTP 实验数据表格模板**

光栅测量实验数据记录表格

增加列按钮

| 次数         | -2                  | -1         | 1          | 2          |
|------------|---------------------|------------|------------|------------|
| 参考光波长 (nm) | 左游标读数<br>0 度<br>0 分 | 0 度<br>0 分 | 0 度<br>0 分 | 0 度<br>0 分 |
| 参考光波长 (nm) | 右游标读数<br>0 度<br>0 分 | 0 度<br>0 分 | 0 度<br>0 分 | 0 度<br>0 分 |
| 待测光        | 左游标读数<br>0 度<br>0 分 | 0 度<br>0 分 | 0 度<br>0 分 | 0 度<br>0 分 |
|            | 右游标读数<br>0 度<br>0 分 | 0 度<br>0 分 | 0 度<br>0 分 | 0 度<br>0 分 |

上传 填入模拟数据

其他输入框的区域

```

<table>
 <tr>
 <th>次数</th>
 <th>-2</th>
 <th>-1</th>
 <th>1</th>
 <th>2</th>
 </tr>
 <tr>
 <td>参考光波长 (nm)</td>
 <td>左游标读数</td>
 <td>0 度</td>
 <td>0 度</td>
 <td>0 度</td>
 <td>0 度</td>
 </tr>
 <tr>
 <td>参考光波长 (nm)</td>
 <td>右游标读数</td>
 <td>0 度</td>
 <td>0 度</td>
 <td>0 度</td>
 <td>0 度</td>
 </tr>
 <tr>
 <td>待测光</td>
 <td>左游标读数</td>
 <td>0 度</td>
 <td>0 度</td>
 <td>0 度</td>
 <td>0 度</td>
 </tr>
 <tr>
 <td></td>
 <td>右游标读数</td>
 <td>0 度</td>
 <td>0 度</td>
 <td>0 度</td>
 <td>0 度</td>
 </tr>
</table>

```

当然，以上两种方法中的 `num` 都必须与 `experimentData` (见上方表格最后一项) 的相应数值对应，大家可查看示例代码了解其对应规则。其中 `list[num].map(this.calculateCell)` 只需更改 `num` 的值，其他写法不必更改。

## 三、多个表格

---

有的实验需要多于一个表格，这时的做法是，直接复制 `MyTable` 类并重命名为 `MyTable2`，在根目录 `index.js` 中按如图方式更改 import 语句和标签。分别编辑两个类，互不干扰。

## 四、验证与交付

---

在确保排版没有太大问题的前提下，点击上传按钮，按 `F12` 打开控制台 (console) 可看到打印出的要上传的数据。确保完全符合戴文清服务器端的规定后可将 `MyTable.jsx` 文件发给我。虽尽量简化 react 表格的更改流程，不过还是辛苦大家！