

一种基于模块化编程思想的数据驱动表格

张正宇 李 林

四川大学锦城学院计算机与软件学院 四川 成都 611731

【摘要】 随着互联网技术的发展,出现了许多的在线工具,利用这些工具可以有效的提高工作效率。本文描述了一种可自定义编辑特性的表格,表现层通过用户自行导入的数据和自行配置的属性,将可编辑表格变成用户直接管理的应用。业务层利用模块化编程的思想,将用户提供的数据和修改的配置属性传递到各个不同的功能处理模块,最后将每个功能模块处理的结果返回,统一构建为真正意义上的数据驱动表格。用户只需要关注输入的数据和传入的参数,功能部分都由每个模块实现。

【关键词】 数据驱动; 模块化; Web 前端

引言

随着前后端分离开发思想的广泛应用,前端更加注重 MVVM 模型中的视图层。面对用户开发、提高用户体验已经成为了开发过程中的基本要求^[1]。对于用户来说,软件交互的使用流畅度和界面美观度都是至关重要的。那么作为开发人员,利用模块化思想将业务处理分成多个模块,由此来降低耦合性,实现真正意义上的低耦合,高内聚,简化代码^[2]。其次,将用户自定义输入的数据,通过业务模块的处理,输出用户想要得到的效果。基于模块化的思想,实现代码的一次编写,多次利用,提高了软件运行效率。

本文研究的项目使用原生 JavaScript 语言开发,针对需要进行表格开发的开发者提供可编辑表格的生成器。在这个项目中,用户可以直接将 JSON 数据和部分配置信息作为参数传入口函数,进而就可以生成用户所需要的可编辑表格。

1 传统方法概述

相较于传统的表格来说,开发者需要进行的操作是编写 HTML 代码,在 HTML 标签中加入数据,需要生成多少行或者多少列表格,就需要添加多少对 HTML 标签。如果按照传统方法来生成表格的话有以下两个缺点:首先,用户输入的数据如果随时随地都要变化,那么开发者随时要在 HTML 代码上进行更改,其次,针对一些有特殊的数据列,无法进行相应的管控。综上所述,传统的表格生成方法并没有在表格上做太多的扩展,生成的数据是固定,并且不能配置。就算是借助现有的第三方 UI 库,也需要对 HTML 代码进行编辑,如果在一些特殊情况下,需要对一些数据进行编辑权限上的操作的话,固定的表格和不具备配置性的表格就很显得很无力。然而使用本文提到的可编辑表格恰好解决了这个缺点,表格的属性可以由用户来设置,更加灵活,在已有数据的基础上,对生成的表格进行更加完美的编辑设置操作。

2 模块设计

(1) 入口函数设计

第一步,可以先定义一个页面的加载事件,在加载事件的函数体中。可以调用 initTable() 入口函数。

第二步,在入口函数中,传入标题头、表格内容、不可编辑列、样式字符串和分数规则。在函数体中,分别调用 setHeader、initdata、getNode、removeIt 和 setStyle 函数。分别对应渲染表格标题头、渲染表格行、绑定表格点击事件、绑定表格行的删除事件、设置表格整体样式。在上述过程中的每个函数都对应了每个模块,负责了不同的功能。参数设计如下:

(2) 渲染标题头模块

通过拿到入口函数传入的数据 header,执行该模块代码,实现动态字符串拼接,再将标题头按照传入的数据渲染出来。

(3) 渲染表内数据模块并注册可编辑点击事件

通过拿到参数传入的表内容 dataList,将数据映射到对应的表格中。再通过对 DOM 进行操作,将表头和表内容挂载在 DOM 上,实现动态数据表格的生成。再对每个可以编辑的 td 绑定点击事件,这样才可以动态插入 input 框,在生成的 input 框上再绑定失焦事件,目的是为了判断新输入数据是否满足设置的分数要求。

(4) 设置不可编辑列模块

通过对用户传入的参数来执行下一步操作。用户传入一个数组,该模块通过用户传入的数组来对不同的列进行设置不可编辑属性。

(5) 设置每列分数范围模块

这里的主要作用是针对用户的需求,如果用户想要设置一个最大值的话,默认是从 0 开始,到设置的最大值。如果用户想要设置一个区间的话,可以传入多个数组,每个数组的元素按照从左到右为最小值和最大值。

(6) 设置整体表格样式模块

用户可以选择传入选择性的表格样式字符串,如果传的为空值,则默认为初始样式,反之为用户选择的对应样式。

(7) 生成可编辑表格设置删除事件

用户可以对每一行进行删除操作,删除某一行后数据会对应重新渲染。

3 功能实现

(1) 渲染标题头模块

在 setHeader 函数中,把获得的标题头数据通过

for 循环, 将数组里面的元素取出来, 然后做一个字符串的动态拼接, 然后再通过对 DOM 的操作, 获取网页上的 thead, 然后把拼接好的字符串放到 thead 的 innerhtml 里面。

(2) 渲染表内数据模块并注册可编辑点击事件

在 initdata 函数中, 先获取 tbody 元素, 然后定义一个变量用来存放拼接的表内容字符串。由于表格分成了行和列, 所以在拼接的时候需要两层循环, 第一层 for 循环用来循环生成 tr, 然后第二层 for 循环用来拼接第一层 tr 里面的 td。这里默认的是从第三列开始就是可编辑表格的元素了, 所以从 2 的时候开始, 拼接的元素都带有 name 属性, 属性值为 grade, 并且元素的类名为 json 数据中传入的 key 值, 等拼接到 tr 数组长度减 2 的时候, 就默认拼接一列非编辑列, 最后再拼接上删除按钮。

再将拼接好的字符串赋值到获取到的 tbody 元素中渲染。非交互的表格就生成了。生成之后, 给每一个有 name 属性值为 grade 的 td 元素绑定点击事件, 在点击的时候, 获得当前点击的 td 单位, 同时传 this 参数, this 参数表示当前点击的单元格 td。ScoreArray 表示分数规则, 用来设置当前列的分数规则。

最重要的部分是点击之后要通过判断设置的属性和分数规则来实现 input 框插入操作, 并且在 input 框失焦之后将输入的新数据放回点击的 td。至于插入 input 元素操作。要先判断当前 table 中是否已经有新插入的 input 元素, 避免和其他 td 同时插入 input 出错。在每次点击 td 元素的时候, 都要获取当前页面 input 的数量, 再执行下面的操作。如果页面没有插入 input, 再判断 editable 属性。如果当前 td 的 editable 属性为 false, 那么说明是不可编辑列, 那么就不能插入。如果不存在这个属性的话, 那么说明可以进行插入操作。

插入 input 框之前, 保留 td 的值, 接下来用户对 input 框进行输入数据, 输入完成失去焦点的时候, 对输入的值进行判断, 调用 checkScore 函数, 通过判断输入的分数是否满足分数规则来返回布尔值, 如果为 true, 说明满足当前 td 元素是满足当前所在列的分数条件的, 那么就将 input 框的值赋给 td, 并且计算总分, 渲染到页面, 否则弹出错误提示框。

(3) 设置不可编辑列模块

在设置不可编辑列的时候, 要从 DOM 元素挂载结束后开始。通过上述模块的挂载后, 就可以获取到全部可以编辑的列, 获得每一列的类名, 对传入的不可编辑列数组使用 for 循环, 不可编辑列中的 number 类型元素, 对应了类名数组中元素的下标, 然后通过下标获取该列类名, 并对该类名赋值 editable 属性, 属性值为 false。则为不可编辑列。

(4) 设置每列分数范围模块

这里是 checkScore 函数, 是用来判断 input 框输入的分数是否满足条件。判断用户传入的是单个 number 还是 array, 如果是单个 number 类型, 就意味着用户只想单独设置某一列不可编辑, 那么就通过传入的参数的类型, 执行下一步的 for 循环判断操作, 最后, 函数返回一个布尔值, 用于表示是否通过数据有效性验证。

(5) 设置整体表格样式模块

先对用户传入的字符串进行一个判断。Style || "table-normal" 这段代码等同于 style == "" ? "table-normal":style。表示, 如果用户设置的样式为空的话, 那么就会设置默认样式, “:” 前为判断为真的值, “:” 后为判断为假的值。判断之后再对 DOM 元素操作, 获得标签名为 table 的元素, 获得之后设置该元素的类名为用户选择的样式类名, 如图 1 和图 2 所示。

姓名	语文	数学	英语	总分	操作
小王	100	80	91	283	删除
小曾	88	87	92	279	删除

图 1 table-green 样式

姓名	语文	数学	英语	总分	操作
小王	100	80	91	283	删除
小曾	88	87	92	279	删除

图 2 默认样式

(6) 生成可编辑表格设置删除事件

在已经生成的表格上, 获取删除按钮的类名, 使用 for 循环绑定每一行删除按钮的点击事件。在绑定的点击事件中, 使用 JavaScript 中的 splice 方法, 对 dataList 中的元素进行删除操作, 之后再把删除元素后的数组再次渲染上页面。

4 结论

本文介绍了如何利用模块化、数据驱动的思想来开发一个表格应用, 降低了代码维护上的复杂度, 并且, 数据驱动的思想在本文当中只是起到了小部分作用, 通常我们的 web 页面是由很多组件组成的, 比如导航栏、侧边栏等等, 这些都可以提取出来当作组件的形式。然后每个组件都是由不同的模块来做处理, 这样更加减少了代码的耦合性。如果在以后的前端开发中使用到数据驱动的思想, 那么开发起来就更加容易维护, 不用再去考虑数据有多少, 也不用对 HTML 代码进行操作, 可以根据数据的多少来决定你 web 页面的内容。利用好数据驱动生成的 web 页面, 会更加的精炼, 架构更加牢固。

【参考文献】

- [1] 肖慧明. 网站前端模块化开发策略研究 [J]. 数字技术与应用, 2018, 36(3): 144.
- [2] 颜玲霞. 关于网站前端模块化开发策略研究 [J]. 数码世界, 2020(3): 57.
- [3] 李为民. 软件开发中的数据驱动设计思想 [J]. 电脑编程技巧与维护, 2012(24): 14-15.
- [4] 王忠群. 面向对象技术和基于数据驱动的软件重用 [J]. 计算机应用, 1999(08): 13-15.