

# 基于 JS 实现数据驱动页面的研究和应用

卫嘉敏 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,数据驱动成为解决前端脚本代码规模大、重用性低、扩展性差等问题的重要途径,本文给出了对数据驱动技术的相关研究及对基于 JavaScript 实现数据操作页面的应用实例。

**【关键词】**JavaScript; 数据驱动; Web 前端

## 1 “数据驱动”的研究意义

随着互联网技术的快速发展,从上个世纪 90 年代开始,经历了从 Web1.0、2.0 到 Web3.0 的过渡,互联网带给大众的体验由网络冲浪体验,变成交互体验,再到如今的个性化定制体验。前端作为大众与网络交互的桥梁,为顺应时代的变化及用户需求的日益增长,在互联网的革新中也不断的自我变革。在这场自我变革中越来越多的数据处理和业务逻辑处理开始偏向于前端<sup>[1]</sup>。业务处理逻辑及用户界面都变的更为复杂,这就在一定程度上造成了前端开发几方面的问题:

(1) 代码规模大并且与数据结构划分不清晰,数据与代码耦合,不便于修改及维护;

(2) 代码可重用性差,导致存在过多的不必要代码;

(3) 现有的数据的存储和提取方法导致页面响应速度缓慢,降低了用户体验<sup>[2]</sup>。当需要在这样的代码上进行版本的更新,便会是一件成本比较高的事情。

不得不提到的一句话是“编程人员常常从自己的代码中挣脱出来,回顾、分析实际情况,仔细考虑程序的数据,最终获得非常好的结果。实际上,数据的表现形式是编程的根本<sup>[3]</sup>。”代码的实现总是与数据紧密相关,数据的体现形式是代码的根本。由此当想要解决前端编程中所出现的这些问题,应该将更多的目光焦距在“数据”上。

鉴于这样的设计理念,编码的重点可以总结为两个方面:

(1) 数据与代码分离,使两者相互独立。(2) 数据控制过程。

“数据驱动”编程正是将编程的重心放于数据,在前端开发方面它相比于传统的前端开发方式,例如使用 jQuery 等前端库直接修改 DOM,大大的简化了代码量,特别是当交互复杂的时候,只关心数据的修改可以让代码变得非常清晰同时提高编码效率,下面就数据驱动技术进行相关探讨。

## 2 “数据驱动”相关研究

人人都在谈数据驱动,数据驱动系统、数据驱动的测试等论题在各类文章中被研究者们反复推敲。而数据驱动是什么?这一问题却鲜有人做出解释。在探讨这个问题之前我们要知道的是:(1) 数据驱动是一种设计思路。(2) 数据驱动的起点是数据,目的是通过数据的改变驱动相应改变。

数据驱动的概念可以归纳为:将数据进行组织形成信息,再对相关的信息加以整合和提炼,最终在提炼后的信息的基础上经过训练和拟合最终形成自动化的模型的过程。并且当数据进行改变或有新的数据输入的之时,系统会以智能的方式对操作进行抉择以达到数据驱动的目的,如图 1 所示。

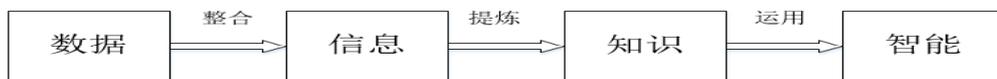


图 1 数据驱动概念模型

作为一种设计思路,数据驱动在各个领域都有相关的运用,本文主要探讨数据驱动在前端结合 JavaScript 的运用——“数据驱动页面”。

数据驱动编程是数据驱动页面的基础,同数据驱动一样它是一种编程思想,它的核心出发点在于:将编程的复杂度从程序设计转移至数据。因为数据的处理比复杂的逻辑设计简单的多。而这里的“数据”也不同于面向对象编程中的数据,数据驱动思想中的数据不仅是一组数据,它还定义了程序的控制流。

数据驱动编程将代码和数据结构进行划分,在改变程序逻辑的时候仅仅改变数据,而不改变代码。如

当需要处理成对的输入和输出时,我们可能会选择使用 switch 或者 if 判断语句来将输入进行判断再进行带入相应的处理函数中最后对应输出,这些处理函数都被封装在对象外部。而采用数据驱动的编程思想则会将封装在外部的函数代码放置于对象内作为私有函数,使用不同的数据时再调用对应的私有函数对相应内容进行改变。这样的编程思想相对于面向对象编程的优势在于:

(1) 方便修改数据,当修改数据时无需修改程序的流程;

(2) 提高了代码可读性,整个数据处理流程一目了然。同时将方法作为内部私有函数,也能使代码和程序

关联更紧密，并且这种紧密不是宏观上的耦合，而是使之具有更强的逻辑性：

(3) 减少了代码量使代码更加简洁。

数据驱动页面是以数据驱动在前端开发方面的运用，根据已有的研究结果可以将数据驱动页面的主要思想总

结为“在数据改变时页面重新渲染”，当后端数据改变的时候，通过 JavaScript 中相关代码的改变来驱动页面中 DOM 的改变，以达到页面重新渲染的目的。此时页面的渲染不再与 DOM 直接相关，而是将数据作为两者之间的桥梁，用数据来驱动页面的改变，如图 2 所示。

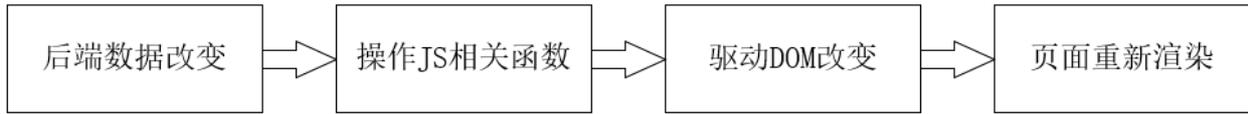


图 2 数据驱动流程

### 3 数据驱动页面的应用—电商网站购物车

在了解数据驱动思想及数据驱动页面后，以为电商网站购物车为例，进一步对数据驱动页面进行研究。这里购物车的主要设计目标是：

(1) 创建购物车用于保存商品信息，商品图片，商品单价，商品数量等商品信息。

(2) 在商品下方提示所选择的商品的总件数和总数量。

(3) 可以对商品数量进行修改，同时商品数量的修改关系到总价、总数量的改变。

(4) 可以对商品列进行选择、全选、取消全选、删除操作，同时各个操作也关联到相关数据的修改。

商品的各个信息以 JSON 数据格式保存，在数据结构设计中将所要描述的名称作为键，将描述内容作为值，例：“商品单价”：“10.00 ¥/500g”，创建商品表单的实现是以 forin 方法遍历键创建表单的 thead，forof 方法遍历值创建表单的 tbody 的方式构建整个购物车单，同时，当遍历到需要操作按钮的时候，为相应位置的 td 以 createChild 的方法插入相应的操作按钮来使购物车表单具有功能入口。

创建完成购物车表单之后便是对购物车中的各项功能实现进行探讨。可以考虑到实现这些功能的方法有多种，例如：将各种方法封装，在修改对应数据的时候调用对应的方法再调用封装的页面渲染函数对页面进行重新渲染。这样的方式看起来结构清晰但是数据的改变和界面的渲染没有联系在一起，他们之间的联系是通过代码的编写强行的融合在一起，这种联系不具有主动性，并且使用这种方式进行代码的编写在结构复杂的程序中不适用，因为代码的结构非常的松散不利于开发也不利于修改和维护。而结合以上研究不难发现所需功能中选择、全选、修改数量等无一不是数据驱动页面的表现，自然会联想到以数据驱动的方式来实现这些功能。

可能让人感到疑问的是，数量的改变关联总价的改变符合前文说到的数据驱动页面的核心“数据的改变关联页面的重新渲染”，但选择等功能为何体现了数据驱动页面。要明白的是数据驱动中的数据并不只是单纯的数据，数据是信息的一种表现形式，由此数据驱动中的数据在最底层应该是信息，数据驱动实质上是信息驱动。而按钮的点击是一种事件信息，复选框的选择，取消选

择也是一种状态信息，并且这些信息的改变无一例外的使页面重新渲染，符合数据改变页面重新渲染的数据驱动页面思想，由此解答了上面提出的问题。修改数据功能实现步骤：

(1) 渲染页面的时候为标题为商品数量的每一列插入一个 type=“number”的 input 标签用于进行数据的修改；

(2) 定义函数分别用于修改总数量、修改总金额、修改每件商品金额；

(3) 为 input 标签绑定 onchange 事件，当修改数据时触发事件，调用预定义好的函数对相应数据进行重新计算并更改；

(4) 根据函数的调用自动对页面重新渲染。

通过以上步骤可以看出“数据改变页面重新渲染”这一数据驱动页面的核心思想，当数据修改时调用函数再渲染页面也体现了程序设计的逻辑性。选择功能实现步骤：

(1) 选择、全选、取消全选功能也是当渲染页面时在遍历到索引为“选择”或“全选”时在对应位置插入一个 type=“checkbox”的 input 标签，作为接口；

(2) 定义函数用于修改总数量、修改总金额、判断全选 / 取消全选、设置全选 / 取消全新；

(3) 在 input 标签上绑定 onchange 事件，当 checkbox 的 checked 属性改变时调用函数修改选中商品的总价及总数量，并且修改当前 tr 背景颜色事件；

(4) 页面自动重新渲染。

选择功能与数据改变功能相似，当选中或者取消选中时由于选中状态的变化会驱动页面中相关信息的变化，而与数据改变功能不同的是当选择事件触发时对应的 tr 的样式会变化，而这一点也正是数据驱动导致页面样式变化的体现，如图 3 所示。

| 全选 <input type="checkbox"/>         | 商品图片  | 商品描述  | 商品单价          | 商品数量 | 操作                   | 总价          |
|-------------------------------------|---|-------|---------------|------|----------------------|-------------|
| <input type="checkbox"/>            |  | 新鲜的山药 | 10.00 ¥ /500g | 1    | <a href="#">删除商品</a> | 10 ¥        |
| <input checked="" type="checkbox"/> |  | 新鲜的草莓 | 15.00 ¥ /500g | 3    | <a href="#">删除商品</a> | 45 ¥        |
| 已选商品 3 件, 合计 (不含运费):                |   |       |               |      |                      | <b>45 ¥</b> |

图 3 购物车模块页面视图

综上, 可以得出数据驱动页面主要可以分为以下两个部分:

- (1) 数据驱动使数据的改变同步页面数据的改变;
- (2) 数据驱动使数据的改变同步页面样式的改变;

#### 4 结语

本文基于互联网场景下对数据驱动页面模式进行了探讨, 对其原理进行了阐述, 并针对该模式的相关应用场景进行了研究。研究发现, 以数据驱动为主要思想的 Web 前端编程模式解决了实际开发中代码重用率低、规模大、扩展性差等长期影响 Web 前端开发的难题。

综上所述, 数据驱动的开发模式对于当前以数据为中心的 Web 应用场景具有极强的适用性。数据驱动作为一种思想, 它的运用不仅限于代码设计, 目前对数据驱动的研究还具有一定的局限性, 但随着未来各项技术的

不断发展, 以数据驱动作为主要思想的研究领域定将长足发展。

#### 【参考文献】

- [1] 校丽丽, 杨雷, 吴珏. 一种基于数据驱动模型的 VirtualDOM 树的构建方法 [J]. 西南科技大学学报, 2018, 33(01): 68-74.
- [2] 刘先莉. 基于数据驱动模型的 Web 前端框架研究与实现 [D]. 西南科技大学, 2019.
- [3] 布鲁克斯 (Frederick P. Brooks, Jr.), 汪颖译. 《人月神话》[M]. 北京: 清华大学出版社, 2002. 11