

MVVM 框架在 Web 前端的应用研究

孙连山,李云倩

(陕西科技大学 电气与信息工程学院, 陕西 西安 710021)

摘要:随着 Web 应用的日趋复杂,前端技术的不断推陈出新,MVVM 框架模式也得以在 Web 前端发挥作用。该文将对 MVVM 框架进行介绍,结合目前备受瞩目的前端 AngularJS 框架,阐述 MVVM 在 Web 前端的应用,并将其与前端传统开发思维对比,分析 MVVM 在 Web 前端开发中具有的优势。

关键词: MVVM; Web 前端框架; AngularJS

中图分类号: TP311 文献标识码: A 文章编号: 1009-3044(2016)06-0045-02

DOI: 10.14004/j.cnki.ckt.2016.0665

面向对象设计(OOD)的核心设计原则是使程序模块达到“高内聚,低耦合”,而 MVC、MVP、MVVM 等都是围绕这一核心原则相继产生的框架模式。随着 Web 应用的日趋复杂、前端规模不断扩大,Web 前端也开启了 MVC、MVVM 风潮,越来越工程化。

1 MVVM 框架模式思想

MVVM 框架模式是 Model-View-ViewModel 模式的简称,于 2005 年由微软的 WPF 和 Silverlight 的架构师 John Gossman 提出。MVVM 既吸收了 MVC 模式的精华,又利用了数据绑定(Data Binding)技术和 WPF 中命令 Command 技术对 MVP 模式进行了改进。其设计思想是“数据驱动界面”,与传统的“事件驱动界面”相比较,以数据为核心,使视图处于从属地位。数据绑定^[1]使表层视图会随底层数据地改变而改变,若用户修改了视图元素值,相当于透过视图元素直接修改了底层数据。Command 技术可接受 View 中用户的输入并做相应处理,它解耦了视图和视图模型。



图 1 MVVM 框架模式结构

MVVM 框架模式结构如图 1 所示,由模型(Model)、视图(View)、视图模型(ViewModel)三部分组成^[2]。MVVM 为 View 层单独定制了一个 Model,既 ViewModel。View 中的视图逻辑通过双向数据绑定和命令绑定到 ViewModel 的属性上,通过 ViewModel 得到 Model 中的数据,双向数据绑定实现了视图和数据模型的自动同步机制。

MVVM 中每个模块在系统中的职责不同。视图:用于界面呈现,它不直接与 Model 进行交互,而是通过 ViewModel 与 Model 通信,并可独立于 Model 的变化和修改,这也是 MVVM 低耦合思想的体现。视图模型:是对视图逻辑和 View 与 Model 模块之间状态控制的封装,是 View 和 Model 通信的桥梁,也是 MVVM 设计思想的核心部分。它包含绑定的数据集合,可根据用户输入通知修改 Model、响应 Model 中的数据变化事件,可发送

ViewModel 中的变化事件以通知 View 更新。一个 ViewModel 可以绑定到多个 View 上,即多个 View 可以复用同一个 ViewModel,这大大提高了代码的可重用性。模型:封装了与业务逻辑相关的数据,以及数据处理等事件。它不依赖于 View 和 ViewModel,可直接与数据库交互。

2 AngularJS 中 MVVM 框架模式的应用

2.1 AngularJS 简介

AngularJS 框架于 2009 年由 Google 首次发布。因其模块化思想,双向数据绑定,指令(Directives),测试驱动开发等特性,以及其 2.0 版本对移动设备的良好支持,目前已处于前端 JS 框架之领导地位。AngularJS 设计初衷是标准的 MVC 模式,但随着代码重构和 API 的重写,现在更是将 MVVM 框架模式表现得淋漓尽致。

2.2 AngularJS 中 MVVM 应用模式

在 AngularJS 中,MVVM 应用模式主要体现在以下几个部分,结构如图 2 所示。

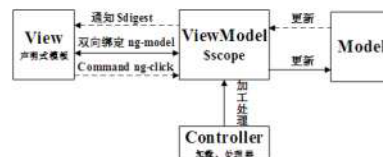


图 2 AngularJS 中 MVVM 应用模式

View:它专注于界面的显示和渲染。在 AngularJS 中,View 除了 HTML、CSS 这些视图代码,还包含 AngularJS 指令、表达式等的视图模板。View 不能直接与 Model 交互,视图对象需要通过 \$scope 这个 ViewModel 来引用。

ViewModel:它负责 View 和 Model 的交互、协作。ViewModel 给 View 提供显示数据,并提供了 View 中 Command 事件操作 Model 的方式。在 AngularJS 中,\$scope 对象充当了这个 ViewModel 的角色。\$scope 被称为控制器的功能模块包装,它提供一些 API 来监控 View 状态,可以把数据模型和函数暴露给视图(UI 模板),它包括数据引用关系、控制器定义行为、视图处理页面布局以及相应的处理跳转等内容。

Model:它是与应用程序的业务逻辑相关的数据的封装载体,是业务领域的对象。AngularJS通过数据模型Model驱动,以JavaScript对象的属性的形式呈现。通过数据绑定技术,视图会根据数据模型的变化自动更新,因而模型也是Web应用程序开发和设计中的焦点。

AngularJS中,Model与ViewModel通过\$scope对象互动。\$scope对象监听Model的变化,通过View来发送和渲染,由HTML来展示代码。Model并不关心会被如何显示或操作,所以Model中也不会包含任何界面显示相关的逻辑。在web页面中,大部分Model都是来自Ajax的服务端返回数据或者是全局的配置对象,而AngularJS中内置的服务\$http非常强大,可直接替代Jquery的Ajax函数,它封装和处理这些与Model相关的业务逻辑。

Controller:AngularJS提供了无状态的Controller,它并不是MVVM模式的核心元素。Controller可设置模型的初始状态,它组合一个或者多个服务(service)来获取业务领域Model并将其放在ViewModel对象上,使得应用界面在启动加载的时候达到一种可用的状态。另外,Controller可监控模型其余部分的变化并采取相应的动作。

AngularJS内嵌了jQuery,它内部实现的一个jQuery子集,包含了常用的jQuery DOM操作方法,事件绑定等,所以我们只需使用JS控制ViewModel,不用关注数据如何呈现到页面,由框架更新Model和View。对于用户交互Command事件(如ng-Click、ng-Change、ng-If等)则会转发到转到\$scope的某个行为逻辑上,通过ViewModel来实现对Model的改变,对于Model的任何改变也会随之反应在ViewModel之上,再通过\$scope的“脏检查机制”(\$digest)来更新到View,从而实现View和Model的分离。

2.3 AngularJS中MVVM应用模式与Web前端传统开发思维对比

jQuery是以事件驱动为中心、面向网页编程的传统前端开发思维的代表。它专注于View层的变化和用户的操作,在对网页元素的定位和操作上需花费较多精力,对于数据处理却是弱项。随着界面和逻辑的日趋复杂,再使用JS或者jQuery去控制DOM会越来越不易,尤其对具有复杂交互的项目,JS逻辑会变得臃肿,交互逻辑分散。而jQuery template技术,虽可实现模

块之间解耦,但无法实现数据和视图展现的解耦。AngularJS是新一代前端开发思维的体现,以Model为中心、面向数据编程。它减少对网页元素的定位和操作,避免了Jquery中DOM操作对网页结构的破坏。AngularJS把模型和视图绑定在一起,实现联动,改变模型,DOM就可以随之进行改变,甚至绑定DOM的事件也可以直接跟着进行改变,让View和Model的进一步分离和解耦,减少了前端开发工作量,提高了开发效率。

3 MVVM框架在Web前端开发过程中的优势

MVVM适合编写大型Web应用前端JS框架,其优势如下。在团队层面,MVVM改变了软件开发方式。由于View与ViewModel之间的松耦合关系,使得开发团队与设计团队分工明确,设计团队只需产出用户友好的界面,而开发团队则专注于业务逻辑和数据,提高了开发效率。在架构层面,模块间松耦合关系使得模块间相互依赖性降低,项目架构更稳定,扩展性得到提高,后续如需增加新模块,能做到最小的改动。在代码层面,通过合理的规划分层ViewModel,可提高代码重用性,使整个逻辑结构更为简洁。另外,MVVM的引入能更有效地组织应用结构,使项目模块变得清晰化、条理化,增强了代码可读性,降低了前端测试难度。

4 结束语

MVVM框架模式归根结底还是MVC精心优化后的结果,它可兼容当下使用的MVC模式。或许有些人认为,MVVM是以更复杂的方式存储DOM和数据绑定关系,比较耗内存、耗性能,但是当Web应用程序的功能达到一个量级且代码开始需要以更高效的方式组合时,使用优秀的开发框架反而会提高网站的性能。

参考文献:

- [1] 刘立. MVVM模式分析与应用[J]. 微型电脑应用,2012(12): 57-60.
- [2] 陈涛. MVVM设计模式及其应用研究[J]. 计算机与数字工程, 2014(10):1982-1985.
- [3] 李龙澍,华晓飞. Silverlight下的MVVM模式的应用[J]. 计算机技术与发展,2013(12):203-207.