



(12)发明专利申请

(10)申请公布号 CN 105955714 A

(43)申请公布日 2016.09.21

(21)申请号 201610235863.6

(22)申请日 2016.04.15

(71)申请人 北京海尔广科数字技术有限公司
地址 100086 北京市海淀区知春路106号中关村皇冠假日写字楼15层
申请人 青岛海尔智能家电科技有限公司

(72)发明人 刘旭升

(74)专利代理机构 工业和信息化部电子专利中心 11010
代理人 焉明涛

(51)Int.Cl.
G06F 9/44(2006.01)

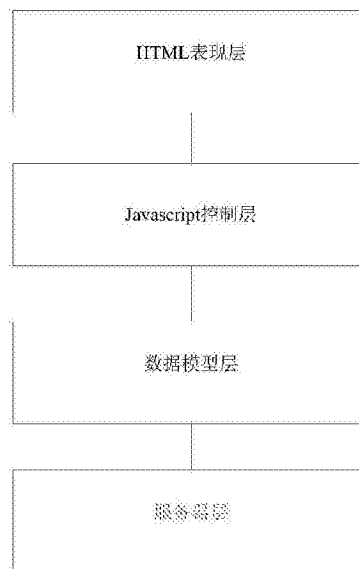
权利要求书1页 说明书5页 附图1页

(54)发明名称

混合开发前端MVC框架实现方法和系统

(57)摘要

本发明公开了一种混合开发前端MVC框架实现方法和系统,用以解决现有混合开发前端模式中程序代码管理逻辑混乱,不易维护和重构的问题。所述方法包括:表现层捕获用户界面中一页面模块的动作事件,并将所述动作事件通知控制层;所述控制层在接收到通知后,根据所述动作事件判断所述页面模块是否需要从服务层获取数据;在判定为是时,确定所述动作事件对应的数据模型,并将所述数据模型通知数据模型层;所述数据模型层调用所述数据模型,以向服务层请求所述数据;并在接收到所述服务层返回的所述数据后,将所述数据发送给所述控制层;所述控制层将所述数据填充到所述表现层,并刷新所述页面模块。



1. 一种混合开发的前端MVC框架实现方法,其特征在于,包括:
表现层捕获用户界面中一页面模块的动作事件,并将所述动作事件通知控制层;
所述控制层在接收到通知后,根据所述动作事件判断所述页面模块是否需要从服务层获取数据;在判定为是时,确定所述动作事件对应的数据模型,并将所述数据模型通知数据模型层;
所述数据模型层调用所述数据模型,以向服务层请求所述数据;以及接收所述服务层返回的所述数据,并将所述数据发送给所述控制层;
所述控制层将所述数据填充到所述表现层,并刷新所述页面模块。
2. 如权利要求1所述的方法,其特征在于,
所述表现层采用单页面应用模式。
3. 如权利要求1所述的方法,其特征在于,所述方法还包括:
将所述表现层的程序代码、所述控制层的程序代码和所述数据模型层的程序代码分离。
4. 如权利要求1-3中任意一项所述的方法,其特征在于,所述数据模型层向所述服务层请求所述数据时采用异步请求方式。
5. 如权利要求1-3中任意一项所述的方法,其特征在于,所述数据模型层在接收到所述服务层返回的所述数据后,将所述数据发送给所述控制层的步骤,还包括:
按照预定的数据格式封装所述数据,并将封装后的所述数据发送给所述控制层。
6. 一种混合开发的前端MVC框架实现系统,其特征在于,包括:
表现层,用于捕获用户界面中一个页面模块的动作事件,将所述动作事件通知控制层;以及用于接收所述控制层填充的数据,并在所述用户界面上展示刷新后的页面模块;
控制层,用于在接收到通知后,根据所述动作事件判断所述页面模块是否需要从服务层获取数据;在判定为是时,确定所述动作事件对应的数据模型,并将所述数据模型通知数据模型层;以及用于接收数据模型层发送的所述数据,将所述数据填充到所述表现层,并刷新所述页面模块;
数据模型层,用于调用所述数据模型,以向服务层请求所述数据;以及接收所述服务层返回的所述数据,并将所述数据发送给所述控制层。
7. 如权利要求6所述的系统,其特征在于,
所述表现层采用单页面应用模式。
8. 如权利要求6所述的系统,其特征在于,所述表现层的程序代码、所述控制层的程序代码和所述数据模型层的程序代码分离。
9. 如权利要求6-8中任意一项所述的系统,其特征在于,所述数据模型层向所述服务层请求所述数据时采用异步请求方式。
10. 如权利要求6-8中任意一项所述的系统,其特征在于,所述数据模型层在接收所述服务层返回的所述数据后,还用于按照预定的数据格式封装所述数据,并将封装后的所述数据发送给所述控制层。

混合开发前端MVC框架实现方法和系统

技术领域

[0001] 本发明涉及计算机领域,特别是涉及一种混合开发前端MVC框架实现方法和系统。

背景技术

[0002] 随着智能终端(手机、平板电脑等)的发展和普及,很多办公、娱乐及操作软件已从大型硬件电脑软件转变成终端APP(应用程序,Application的缩写),而终端APP的开发正逐步由单一iOS或Android开发模式向混合开发模式(iOS+HTML5+Javascript或Android+HTML5+Javascript)转变,HTML5+Javascript开发被称为混合开发模式中的前端开发。

[0003] 目前,由于前端开发兴起时间较短,大部分开发者对前端开发的理解停留在较陈旧模式中,使得前端开发代码管理逻辑混乱,不易维护和重构,增加了开发成本,从而造成了开发的应用程序刷新效率和数据存储性能较低,导致应用程序的用户体验不够好。

发明内容

[0004] 鉴于现有技术的缺陷,本发明目的在于提供一种混合开发前端MVC框架实现方法和系统,用以解决现有混合开发前端模式中程序代码管理逻辑混乱,不易维护和重构的问题。

[0005] 本发明目的主要是通过以下技术方案实现的:

[0006] 根据本发明的一个方面,本发明提供一种混合开发的前端MVC框架实现方法,包括:

[0007] 表现层捕获用户界面中一页面模块的动作事件,并将所述动作事件通知控制层;

[0008] 所述控制层在接收到通知后,根据所述动作事件判断所述页面模块是否需要从服务层获取数据;在判定为是时,确定所述动作事件对应的数据模型,并将所述数据模型通知数据模型层;

[0009] 所述数据模型层调用所述数据模型,以向服务层请求所述数据;以及接收所述服务层返回的所述数据,并将所述数据发送给所述控制层;

[0010] 所述控制层将所述数据填充到所述表现层,并刷新所述页面模块。

[0011] 进一步,所述表现层采用单页面应用模式。

[0012] 进一步,所述方法还包括:

[0013] 将所述表现层的程序代码、所述控制层的程序代码和所述数据模型层的程序代码分离。

[0014] 进一步,所述数据模型层向所述服务层请求所述数据时采用异步请求方式。

[0015] 进一步,所述数据模型层在接收到所述服务层返回的所述数据后,将所述数据发送给所述控制层的步骤,还包括:

[0016] 按照预定的数据格式封装所述数据,并将封装后的所述数据发送给所述控制层。

[0017] 根据本发明的另一个方面,本发明还提供一种混合开发的前端MVC框架实现系统,包括:

[0018] 表现层,用于捕获用户界面中一个页面模块的动作事件,将所述动作事件通知控制层;以及用于接收所述控制层填充的数据,并在所述用户界面上展示刷新后的页面模块;

[0019] 控制层,用于在接收到通知后,根据所述动作事件判断所述页面模块是否需要从服务层获取数据;在判定为是时,确定所述动作事件对应的数据模型,并将所述数据模型通知数据模型层;以及用于接收数据模型层发送的所述数据,将所述数据填充到所述表现层,并刷新所述页面模块;

[0020] 数据模型层,用于调用所述数据模型,以向服务层请求所述数据;以及接收所述服务层返回的所述数据,并将所述数据发送给所述控制层。

[0021] 进一步,所述表现层采用单页面应用模式。

[0022] 进一步,所述表现层的程序代码、所述控制层的程序代码和所述数据模型层的程序代码分离。

[0023] 进一步,所述数据模型层向所述服务层请求所述数据时采用异步请求方式。

[0024] 进一步,所述数据模型层在接收所述服务层返回的所述数据后,还用于按照预定的数据格式封装所述数据,并将封装后的所述数据发送给所述控制层。

[0025] 本发明有益效果如下:

[0026] 本发明的方法和系统通过使控制层采用数据注入的方式将数据传给表现层,从而实现表现层的程序代码和控制层的程序代码的分离;同时通过控制层调用数据模型层的相应数据模型,并使该数据模型向服务层请求数据,从而实现数据模型层的程序代码和控制层的程序代码的分离;从而使混合开发的表现层,控制层及数据层彼此分立,模块维护独立互不影响,使程序代码管理逻辑清晰,易维护和重构;并且通过页面局部刷新,降低应用数据请求所占内存,提高运行性能,为软件及APP开发带来更好的用户体验。

附图说明

[0027] 图1是本发明实施例中混合开发前端MVC框架实现方法的主流程图;

[0028] 图2是本发明实施例中混合开发前端MVC框架实现系统的框架示意图。

具体实施方式

[0029] 为了解决现有混合开发前端模式中程序代码管理逻辑混乱,不易维护和重构的问题,本发明提供了一种混合开发前端MVC(Model(对象实例)View(视图)Controller(控制器))框架实现方法和系统,以下结合附图以及实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅用以解释本发明,并不限定本发明。

[0030] 实施例一

[0031] 如图1所示,本发明实施例提供一种混合开发的前端MVC框架实现方法,包括:

[0032] 步骤A、捕捉用户操作,也就是说,表现层捕获用户界面中一页面模块的动作事件,并将所述动作事件通知控制层;

[0033] 步骤B、调用数据处理,也就是说,所述控制层在接收到通知后,根据所述动作事件判断所述页面模块是否需要从服务层获取数据;在判定为是时,确定所述动作事件对应的数据模型,并将所述数据模型通知数据模型层;

[0034] 步骤C、请求数据,也就是说,所述数据模型层调用所述数据模型,以向服务层请求

所述数据;并在接收到所述服务层返回的所述数据后,将所述数据发送给所述控制层;

[0035] 步骤D、服务层返回数据;

[0036] 步骤E、返回数据到控制器,也就是说,所述数据模型层接收所述服务层返回的所述数据,将所述数据发送给所述控制层;

[0037] 步骤F、局部刷新页面填充数据,也就是说,所述控制层将所述数据填充到所述表现层,并刷新所述页面模块。

[0038] 具体说,表现层用来提供用户界面,用户界面包括若干页面模块;也就是说表现层向用户提供展示页面;页面展示的内容包括浏览器或APP中用户能看到的所有内容,包括文字、图片、动画和加载提示,以及用户自己输入的文字及表情等。

[0039] 数据是指从服务器拿到的所有动态变化的内容,包括动态更新显示的新闻、广告、用户信息、订单信息、用户习惯和设备当前状态等。

[0040] 控制层用来判断表现层的用户操作,并调用数据模型层请求数据;用户操作泛指用户在页面中做出的所有动作事件,包括:输入动作、点击动作和滑动动作等,例如注册时输入信息,登陆时输入用户名密码,点击浏览下一页,在 APP中滑动页面加载更多内容,点击某商品进入商品详细介绍页面等。

[0041] 数据模型层用于保存多个功能模块(即数据模型),每个功能模块对应与表现层中的一个动作,用于实现表现层中相应动作所要实现的功能。

[0042] 现有前端开发模式中,是在HTML中引入或嵌入相关的Javascript代码,这种开发模式使得代码逻辑不够清晰,代码维护繁琐;并且这种开发模式在表现层需要数据是,需要刷新整个页面,从而造成了服务端处理需要处理整个页面数据,从而页面加载时间较长;并且页面之间逻辑耦合较高,不易分离。

[0043] 本发明实施例通过在表现层的用户界面需要载入数据时,控制层向数据模型层发送数据请求,以使所述数据模型层返回所述数据,所述控制层接收到所述数据后,刷新所述用户界面上所述数据所属的页面模块,并将所述数据填充到所述表现层,从而使控制层采用数据注入的方式将数据传给表现层,从而实现表现层的程序代码和控制层的程序代码的分离,使程序代码管理逻辑清晰,易维护和重构;并且通过页面局部刷新,降低应用数据请求所占内存,提高运行性能,为软件及APP开发带来更好的用户体验。

[0044] 在本发明的一个优选的实施方式中,还对上述方法进行了优化,具体来说,所述方法还包括:

[0045] 所述表现层采用单页面应用模式,只提供一个页面作为用户界面。

[0046] 单页面应用模式具体包括:一个应用(包括门户网站及APP)只有一个页面入口(用户界面);所有的请求及页面变化通过控制层控制实现,不需要发送url改变请求,节省流量提高加载速度。

[0047] 在本发明的一个优选的实施方式中,还对上述方法进行了优化,具体来说,所述表现层捕获用户界面中一页面模块的动作事件的步骤之前,还包括:

[0048] 将所述表现层的程序代码、所述控制层的程序代码和所述数据模型层的程序代码分离。

[0049] 也就是说,采用模块化开发方式,模块之间可继承。模块化开发方式使开发者像添加模块一样添加新的开发功能,所有功能互不影响,独立管理。具体说,需要新增功能时,

只需新增一个模块进行开发,此模块会作为单一独立模块,与其他功能模块互相之间不会影响,且可选择继承其他模块功能。例如,HTML页面和Javascript代码都可作为单一片段,按需加载,使用完后实例自动销毁,合理管理内存空间。

[0050] 在本发明的一个优选的实施方式中,还对上述方法进行了优化,具体来说,

[0051] 数据模型向所述服务层发送数据请求的方式为异步请求。

[0052] 在本发明的一个优选的实施方式中,还对上述方法进行了优化,具体来说,所述数据模型层接收所述数据后,按照预定的数据格式封装所述数据,并将封装后的所述数据发送给所述控制层。

[0053] 也就是说,在本优选的实施方式中,控制层主要用来分析处理用户界面需要载入的数据,根据需要载入的数据,调用数据模型层中的相应数据模型(model),使数据模型从服务器中获取该数据。比如:用户点击注册,控制层需调用注册的model并发送用户信息到服务器进行注册;又如:用户点击下一页,控制层负责页面正确跳转并且展示页面跳转的动画(比如手机APP中新页面是从右向左滑动出现)。

[0054] 页面中所有的动作或者请求,都有控制层先进行分析,然后处理。

[0055] 数据模型层主要负责发送请求,并且封装请求回来的数据。控制层负责处理页面所有的动作和逻辑,并决定是否需要调用数据模型层来发送请求,以及调用哪个数据模型来发送请求。并在接收到服务层返回的数据后,在S102中每次只需重绘用户需要看到变化的部分,从而减轻服务端的处理负担,提高网页响应速度,且将数据交互从服务端转移到浏览器端,做到数据按用户所需加载的目的。

[0056] 如图1所示,以HTML5+Javascript为例,详细描述本发明实施例方法,本发明实施例方法包括:

[0057] A:HTML表现层用来展示页面及数据,在用户触发某种事件后捕获事件并通知控制层;例如:用户在海尔商城首页看到所有的商品列表,点击某一个商品,会触发一个“点击”事件,控制层会监控到。

[0058] B:Javascript控制层用来分析表现层捕获的事件类型及相对应的处理机制,调用该事件对应的数据模型请求数据;例如:控制层分析当前的“点击”事件是什么类型,比如这个事件不是注册,不是登陆,不是页面跳转,而是点击商品需要进入商品详情页面。

[0059] 控制层控制页面跳转到商品详情页,并且调用取商品详情的数据model发送该商品数据的请求。

[0060] C:数据模型层收到数据请求后,发送Ajax请求后台服务器;

[0061] D:服务器返回数据;

[0062] E:数据模型层按照规定的数据格式封装取回的数据并传回控制层;

[0063] F:控制层接收数据,并局部刷新对应的页面模块,将数据填充到HTML表现层中。

[0064] 也就是说,按照此流程,混合开发的表现层,控制层及数据层彼此分立,模块维护独立互不影响,无论从编码或者维护方面考察,都更加简便,且更符合面向对象逻辑。

[0065] 实施例二

[0066] 如图2所示,本发明实施例提供一种混合开发的前端MVC框架实现系统,包括:

[0067] 表现层,用于捕获用户界面中一个页面模块的动作事件,将所述动作事件通知控制层;以及用于接收所述控制层填充的数据,并在所述用户界面上展示刷新后的页面模块;

[0068] 控制层,用于在接收到通知后,根据所述动作事件判断所述页面模块是否需要从服务层获取数据;在判定为是时,确定所述动作事件对应的数据模型,并将所述数据模型通知数据模型层;以及用于接收数据模型层发送的所述数据,将所述数据填充到所述表现层,并刷新所述页面模块;

[0069] 数据模型层,用于调用所述数据模型,以向服务层请求所述数据;并在接收到所述服务层返回的所述数据后,将所述数据发送给所述控制层。

[0070] 其中,所述表现层采用单页面应用模式,只提供一个入口页面作为用户界面。

[0071] 所述表现层的程序代码、所述控制层的程序代码和所述数据模型层的程序代码分离。

[0072] 所述数据模型向所述服务层发送数据请求的方式为异步请求。

[0073] 所述数据模型层在接收所述服务层返回的所述数据后,还用于按照预定的数据格式封装所述数据,并将封装后的所述数据发送给所述控制层。

[0074] 本发明实施例的系统可使开发者更容易管理展示页面与逻辑控制代码,模式由面向脚本语言开发转为面向对象开发。需要新增功能时,只需新增一个模块进行开发,此模块会作为单一独立模块,与其他功能模块互相之间不会影响,且可选择继承其他模块功能。HTML页面和Javascript代码都可作为单一片段,按需加载,使用完后实例自动销毁,合理管理内存空间。

[0075] 尽管为示例目的,以上已经公开了本发明的优选实施例,但本领域的技术人员将意识到各种改进、增加和取代也是可能的,因此,本发明的范围应当不限于上述实施例。

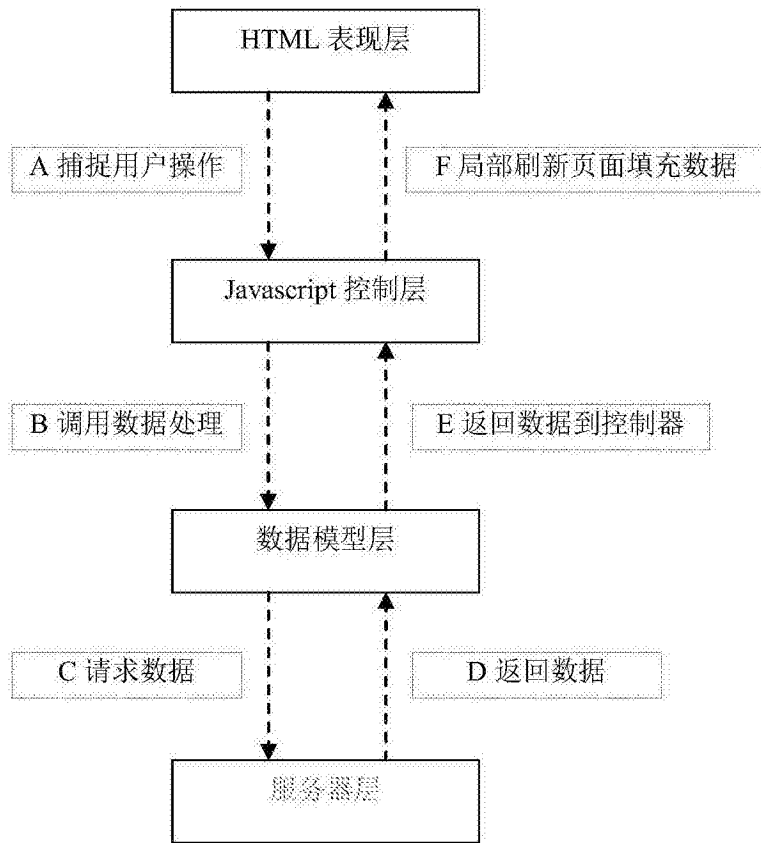


图1

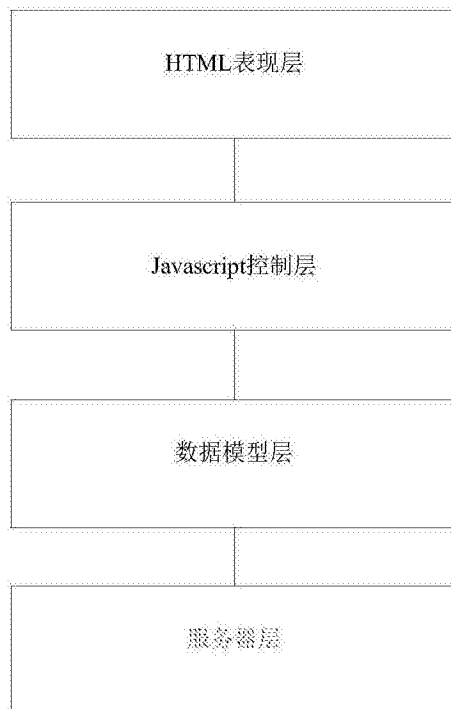


图2