



(12)发明专利申请

(10)申请公布号 CN 110232173 A

(43)申请公布日 2019.09.13

(21)申请号 201810184167.6

(22)申请日 2018.03.06

(71)申请人 福建天泉教育科技有限公司
地址 350212 福建省福州市长乐市湖南镇
大鹤村(福建天棣动漫科技有限公司)

(72)发明人 刘德建 王若龙 郭玉湖

(74)专利代理机构 福州市博深专利事务所(普
通合伙) 35214
代理人 林志峥

(51) Int. Cl.
G06F 17/24(2006.01)

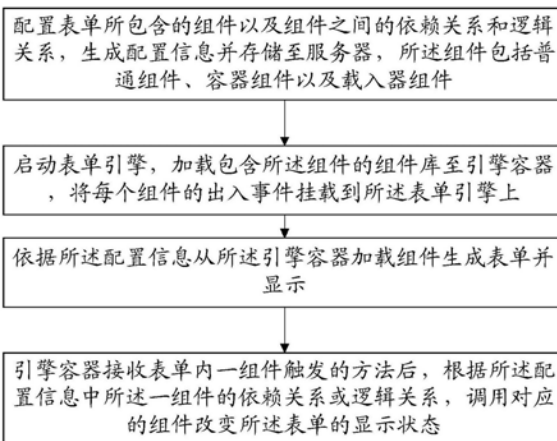
权利要求书2页 说明书10页 附图4页

(54)发明名称

基于可配置化表单引擎的表单配置方法及其系统

(57)摘要

本发明提供一种基于可配置化表单引擎的表单配置方法及其系统,方法包括配置表单所包含的组件以及组件之间的依赖关系和逻辑关系,组件包括普通组件、容器组件以及载入器组件;启动表单引擎,加载组件库至引擎容器,并将组件的出入事件挂在到引擎里面;依据配置信息生成表单;引擎容器接收表单内一组件触发的方法后,根据配置信息中所述一组件的依赖关系或逻辑关系,调用对应的组件改变表单的显示状态。本发明能够通过容器类组件、载入类组件以及普通组件的组合实现对表单页面布局和交互方式的自由定制;通过对组件间依赖关系和逻辑关系的配置实现对表单字段之间的关联关系进行自由定制,从而实现真正意义上的表单灵活地自定义布局。



1. 一种基于可配置化表单引擎的表单配置方法,其特征在于,包括:

配置表单所包含的组件以及组件之间的依赖关系和逻辑关系,生成配置信息并存储至服务器,所述组件包括普通组件、容器组件以及载入器组件;

启动表单引擎,加载包含所述组件的组件库至引擎容器,将每个组件的出入事件挂载到所述表单引擎上;

依据所述配置信息从所述引擎容器加载组件生成表单并显示;

引擎容器接收表单内一组件触发的方法后,根据所述配置信息中所述一组件的依赖关系或逻辑关系,调用对应的组件改变所述表单的显示状态。

2. 如权利要求1所述的基于可配置化表单引擎的表单配置方法,其特征在于,所述通过调用对应的组件改变所述表单的显示状态,具体为:

通过调用对应的组件的出入事件,改变所述表单的显示状态。

3. 如权利要求2所述的基于可配置化表单引擎的表单配置方法,其特征在于,通过logic事件订阅机制或dependencies事件回调机制调用所述对应的组件的出入事件。

4. 如权利要求1所述的基于可配置化表单引擎的表单配置方法,其特征在于,所述依据所述配置信息从所述引擎容器加载组件生成表单并显示,具体为:

载入并解析所述配置信息,从所述引擎容器中加载所述配置信息对应组件的实例;

通过集成器将所述组件的实例依据所述配置信息生成表单并显示。

5. 如权利要求4所述的基于可配置化表单引擎的表单配置方法,其特征在于,所述从所述引擎容器中加载所述配置信息对应组件的实例,具体为:

依据对应容器组件的配置信息,将所述容器组件所包含的子组件渲染成对应的页面布局或交互方式,所述子组件包括普通组件和/或载入器组件;

依据对应普通组件的配置信息,对普通组件进行渲染;

依据对应载入器组件的配置信息,将载入器组件对应的容器组件和普通组件渲染成对应的布局样式。

6. 如权利要求1所述的基于可配置化表单引擎的表单配置方法,其特征在于,所述依赖关系包括点击一组件或一组件的预设字段后显示另一组件,以及一组件取预设值时显示另一组件,以及一容器组件中各子组件的显示触发关系;

所述逻辑关系包括一组件对应的备选项的值将根据另一组件选中的值变化,以及组件对应载入器组件的显示触发方式。

7. 如权利要求1所述的基于可配置化表单引擎的表单配置方法,其特征在于,所述容器组件包括弹出框容器组件、逻辑容器组件、组容器组件、条件判断容器组件、列表容器组件、表格容器组件和选项卡容器组件;所述载入器组件包括底纹、阴影、特定颜色的星号。

8. 一种基于可配置化表单引擎的表单配置系统,其特征在于,包括模板编辑器、服务端、表单引擎,以及集成器;所述模板编辑器和表单引擎分别与所述服务端连接;所述集成器与所述表单引擎连接;

所述模板编辑器,用于配置表单所包含的组件以及组件之间的依赖关系和逻辑关系,生成配置信息并存储至服务器,所述组件包括普通组件、容器组件以及载入器组件;

所述表单引擎,用于在启动后,加载包含所述组件的组件库至引擎容器,将每个组件的出入事件挂载到所述表单引擎上;以及用于在其引擎容器接收表单内一组件触发的方法

后,根据所述配置信息中所述一组件的依赖关系或逻辑关系,调用对应的组件改变所述表单的显示状态;

所述集成器,用于依据所述配置信息生成表单并显示。

9.如权利要求8所述的基于可配置化表单引擎的表单配置系统,其特征在于,所述表单引擎,还用于将每个组件的出入事件挂载到所述表单引擎上;

所述表单引擎,具体通过调用对应的组件的出入事件,改变所述表单的显示状态。

10.如权利要求9所述的基于可配置化表单引擎的表单配置系统,其特征在于,所述表单引擎,具体通过logic事件订阅机制或dependencies事件回调机制调用所述对应的组件的出入事件。

基于可配置化表单引擎的表单配置方法及其系统

技术领域

[0001] 本发明涉及表单配置领域,具体说的是基于可配置化表单引擎的表单配置方法及其系统。

背景技术

[0002] 目前市面上现有的智能表单系统一般是通过xml来描述表单字段的顺序、字段的类型等信息,然后通过一个解析器来解析出对应的控件,最后再把解析出来的控件按照顺序排列在页面上面。这种做法表单的描述信息比较有限,而且会有几个问题:第一,只能顺序的把控件排列在页面上面,无法对表单进行复杂的页面布局。第二,只能呆板的把控件原原本本的摆在页面上面,无法定义控件之间的交互方式。第三,只能描述控件的数据类型是什么样的,无法定义字段之间的关联关系。Formtastic取自form(表单)和fantastic(奇思妙想的)的结合,寓意着这是一个灵活多变的表单引擎。

[0003] 随着公司业务的发展,大量的表单类型应用需要被开发。而不同类型的表单应用对页面布局、交互方式以及字段的关联关系都不尽相同。有些应用仅仅是因为几个字段的布局和交互方式不一样,就需要对整个应用进行重新开发,对原有的应用无法有效的复用。而且这样费时费力的开发完成后并无法产生沉淀,将来再有类似的应用在布局、交互和关联关系等方面但有差异就需要重新开发。因此如何抽象出表单类型应用的页面布局、交互方式以及字段间的关系成为了急需解决的问题。

发明内容

[0004] 本发明所要解决的技术问题是:提供一种基于可配置化表单引擎的表单配置方法及其系统,能够实现表单页面布局和交互方式的自由定制,以及表单内字段之间关联关系的自由定制。

[0005] 为了解决上述技术问题,本发明采用的技术方案为:

[0006] 一种基于可配置化表单引擎的表单配置方法,包括:

[0007] 配置表单所包含的组件以及组件之间的依赖关系和逻辑关系,生成配置信息并存储至服务器,所述组件包括普通组件、容器组件以及载入器组件;

[0008] 启动表单引擎,加载包含所述组件的组件库至引擎容器,将每个组件的出入事件挂载到所述表单引擎上;

[0009] 依据所述配置信息生成表单并显示;

[0010] 引擎容器接收表单内一组件触发的方法后,根据所述配置信息中所述一组件的依赖关系或逻辑关系,调用对应的组件改变所述表单的显示状态。

[0011] 本发明提供的另一个技术方案为:

[0012] 一种基于可配置化表单引擎的表单配置系统,包括模板编辑器、服务端、表单引擎,以及集成器;所述模板编辑器和表单引擎分别与所述服务端连接;所述集成器与所述表单引擎连接;

[0013] 所述模板编辑器,用于配置表单所包含的组件以及组件之间的依赖关系和逻辑关系,生成配置信息并存储至服务器,所述组件包括普通组件、容器组件以及载入器组件;

[0014] 所述表单引擎,用于在启动后,加载包含所述组件的组件库至引擎容器,将每个组件的出入事件挂载到所述表单引擎上;以及用于在其引擎容器接收表单内一组件触发的方法后,根据所述配置信息中所述一组件的依赖关系或逻辑关系,调用对应的组件改变所述表单的显示状态;

[0015] 所述集成器,用于依据所述配置信息生成表单并显示。

[0016] 本发明的有益效果在于:区别于传统的表单系统无法实现交互方式以及字段之间关联关系的灵活配置,只支持对页面布局简单定义(仅限于从上往下或者从左往右,并不能真正意义的实现表单页面的自定义布局)的不足。本发明只需基于可配置化表单引擎Formtastic进行简单的配置,将普通组件、容器组件以及载入器组件相互结合,实现对页面布局和交互方式的自由配置;通过配置表单内包含的各个组件之间的依赖关系和逻辑关系,实现字段之间的关联关系的个性化定制,从而实现真正意义上的表单自定义配置。

附图说明

[0017] 图1为本发明一种基于可配置化表单引擎的表单配置方法;

[0018] 图2为本发明实施例一中组件的事件机制中事件挂载的流程示意图;

[0019] 图3为本发明实施例一中表单配置信息的渲染流程示意图;

[0020] 图4为本发明实施例一中组件的事件机制中事件触发调用的流程示意图;

[0021] 图5为本发明实施例二中配置的表单效果示意图;

[0022] 图6为本发明实施例三的基于可配置化表单引擎的表单配置系统的结构组成连接示意图。

[0023] 标号说明:

[0024] 1、模板编辑器;2、服务端;3、表单引擎;4、集成器。

具体实施方式

[0025] 为详细说明本发明的技术内容、所实现目的及效果,以下结合实施方式并配合附图予以说明。

[0026] 本发明最关键的构思在于:基于可配置化表单引擎Formtastic将普通组件、容器组件以及载入器组件相互结合,实现对页面布局和交互方式的自由配置;通过配置表单内包含的各个组件之间的依赖关系和逻辑关系,实现字段之间的关联关系的个性化定制,从而实现真正意义上的表单自定义配置。

[0027] 本发明涉及的相关技术术语以及技术内容的解释:

[0028] 本发明涉及的可配置化表单引擎(Formtastic引擎),取自form(表单)和fantastic(奇思妙想的)的结合,寓意着这是一个灵活多变的表单引擎。在本发明中,Formtastic引擎也是通过解析配置信息来实现表单的可配置化,但是不同的是,配置里面包含了容器组件、载入器组件(loader组件)以及普通组件的相关信息。目前Formtastic引擎里面内置了许多容器类、loader类和普通类组件。本发明正是可以通过自定义三者之间的关联关系来配置出灵活多变的表单。

[0029] 下面请参阅表1,将对本发明涉及的普通组件、容器组件以及载入器组件进行详细解释说明:

[0030]

	默认 Loader 类组件	功能性 Loader 类组件
普通类组件	按照自己定义的方式渲染	按照功能类载入器的定义好的方式渲染
容器类组件	按照容器定义好的方式,将子组件(子组件按照自己的方式进行渲染)进行渲染	当功能性 Loader 类组件遇到容器类组件可以有以下几种行为选择: 1、按照功能性载入器组件定义好的方式渲染; 2、按照容器组件定义好的方式渲染; 3、先按照功能性载入器组件定义好的方式渲染,然后再按照容器组件定义好的方式渲染; 4、先按照容器组件定义好的方式渲染,然后再按照功能性载入器组件渲染; 5、功能性载入器组件与容器组件互相结合渲染。

[0031] 表1

[0032] 其中,普通类组件:就是对传统html控件的封装并根据Formtastic协议进行开发的组件。它的作用就是封装出入事件并对组件进行必要的渲染。普通类组件会把从组件内部发往外部的数据根据协议进行封装,以此来告诉Formtastic引擎它触发的这个动作需要执行什么操作。同时也会把从外部发往内部的数据根据协议进行适配,把其适配成控件能够识别的信息,以此来提供给外部调用。

[0033] 容器类组件:容器类组件也是按照Formtastic协议进行开发的组件,它也会对消息的进出进行封装。但是不同的地方在于,容器类组件会定义子组件的组合方式或者交互方式,比如它可以事先定义好上中下结构,那么当该容器类组件与普通类组件结合后,普通类组件就会按照容器类组件事先布局好的样式进行排布,以此来达到灵活的页面布局的目的。又比如,它可以定义通过页面上的“+”和“-”按钮来对组件进行复制和删除,这样当该容器类组件与普通类组件相结合后,我们就可以方便的在页面上控制普通类组件的复制新增和删除。

[0034] 容器类组件一般可以定义组件的展现形式或者交互方式,比如田字格类的容器组件结合了普通组件,页面就会布局成田字格的形式,以此来达到页面布局的控制。又比如弹出框类的容器组件结合了普通类组件,就会形成了弹出框的交互效果,以此来达到了交互方式的控制。

[0035] Formtastic的容器与现有技术的智能表单引擎容器的概念不一样。现有的智能表单引擎容器只是用来呈现组件的一个地方,他们所谓的容器只不过是人为的方式给它起了个名字,它的作用仅仅是把基本的控件元素展示在页面上而已。而Formtastic的容器则表示容器本身也是一个组件,目前的容器组件有:弹出框容器组件、逻辑容器组件、组容器组件、条件判断容器组件、列表容器组件、表格容器组件、选项卡容器组件。事实上,Formtastic中的组容器组件就已经包含了现有的智能表单引擎容器的所有功能。这也是

Formtastic将表单渲染过程分成三种类型组件来进行组合的原因。因为这样可以实现许多不同功能的容器类型组件,然后再把这些容器组件和普通组件进行结合产生新的表单。

[0036] 比如选项卡容器组件,在这个容器组件里面我们可以动态的定义选项卡容器需要几个选项卡,每个选项卡的交互样式是怎样的,点击了选项卡之后触发的方法会做些什么事情。然后我们会给这个选项卡容器组件配上许多子组件(子组件可以是普通组件也可以是容器组件),这样子当选项卡容器组件在切换选项卡的时候就能有不同的子组件被显示出来。而现有的智能表单引擎容器只是把组件摆在上面,即便是他们真的配制出了一个选项卡组件也只能在这个组件内部自己切换,无法为选项卡组件的内容配置复杂的表单,也无法在点了选项卡组件后再做什么额外的事情。

[0037] 载入器组件(Loader类组件):Loader类组件的主要作用是对组件进行装饰,即定义组件的页面布局。一般不对数据的进出进行封装,当然如果需要也可以。它更多的是处理某个组件的布局样式,比如我可以定义一个验证Loader组件,其作用就是应用了该Loader的组件都会被强制加上红色“*”,需要进行验证,而验证的逻辑我们可以自由定制。它可以与容器类组件进行结合以此来对普通组件产生强大的效果。比如验证类loader组件结合了田字格容器组件对普通组件进行了控制,在田字格容器组件左上角的组件点击可以弹出框,右上角的组件可以自由复制新增和删除,左下角的组件为需要非空验证的组件,右下角的组件按照原来的方式进行展示。比如标签载入器组件结合了普通组件,就会给普通组件都加上一个选中的时候有阴影和底纹,非选中的时候就没有阴影和底纹,以此来达到了页面布局交互样式的控制。同时Loader类组件还可以与容器类组件相互结合来对普通组件进行控制,以此来达到更加复杂的页面布局以及交互效果。

[0038] 逻辑配置(Logic):Logic可以定义组件之间数据的相互依赖,比如当A组件取值true时B组件才能显示,这就可以配置B组件的订阅A组件的赋值事件,当B组件收到了A组件发出的事件后B组件才会显示出来。以此来定义组件之间的关联关系。

[0039] 依赖配置(Dependencies):Dependencies相对于Logic的事件订阅机制采用的是事件回调机制,当某个普通组件触发了某个方法后会被dependencies给捕获到,通过解析该方法并做一些必要的操作以此来实现对组件之间的控制。

[0040] 请结合上述内容,同时参照图1,本发明提供一种基于可配置化表单引擎的表单配置方法,包括:

[0041] 配置表单所包含的组件以及组件之间的依赖关系和逻辑关系,生成配置信息并存储至服务器,所述组件包括普通组件、容器组件以及载入器组件;

[0042] 启动表单引擎,加载包含所述组件的组件库至引擎容器,将每个组件的出入事件挂载到所述表单引擎上,使得引擎拥有支配控制组件的能力;

[0043] 依据所述配置信息从所述引擎容器加载组件生成表单并显示;

[0044] 引擎容器接收表单内一组件触发的方法后,根据所述配置信息中所述一组件的依赖关系或逻辑关系,调用对应的组件改变所述表单的显示状态。

[0045] 从上述描述可知,本发明具有以下有益效果:

[0046] 1、现有的智能表单引擎无法对表单的交互方式以及字段的关联关系进行自由定义,对表单页面布局也无法随意定制。本发明基于Formtastic引擎,可以方便的对表单的页面布局、交互方式以及字段之间的关联关系进行定义。

[0047] 2、现有的智能表单系统组件一旦有变化就要重新开发组件，而通过本发明的Formtastic引擎则是一次开发终身受用即便是有新的需求也只需要在原来的组件上进行扩展既可。

[0048] 3、现有的智能表单虽然方便但是配置起来依然需要不少时间，配置完还要各种调试，而本发明使用Formtastic引擎，配置后只需少许代码即可运行，相对来讲极大的减少了开发的时间。

[0049] 4、现有的智能表单虽然也可以在配置的时候就能看到效果，但是并不能方便的对配置信息进行修改，而使用了Formtastic引擎后，可视化操作界面可以方便的对页面进行配置，所见即所得。

[0050] 5、现有的智能表单系统二次开发较为复杂还要对接原来数据解析的格式，而Formtastic引擎通过许多机制提供给开发者便捷开发的入口。

[0051] 如图2和图3所示，进一步的，

[0052] 所述通过调用对应的组件改变所述表单的显示状态，具体为：

[0053] 通过调用对应的组件的出入事件，改变所述表单的显示状态。

[0054] 由上述描述可知，通过事先将每个组件的出入事件挂载在表单引擎上，能够实现对应事件触发时的快速调用。

[0055] 进一步的，可以通过Logic事件订阅机制、Dependencies事件回调机制、Courier事件拦截机制等许多方式提供给开发者便捷开发的入口，在此，具体通过logic事件订阅机制或dependencies事件回调机制所述对应的组件的出入事件。

[0056] 由上述描述可知，通过logic（逻辑配置）和dependencies（依赖配置）实现了对表单字段之间的关联关系进行自由定制。

[0057] 进一步的，所述依据所述配置信息从所述引擎容器加载组件生成表单并显示，具体为：

[0058] 载入并解析所述配置信息，从所述引擎容器中加载所述配置信息对应组件的实例；

[0059] 通过集成器将所述组件的实例依据所述配置信息生成表单并显示。

[0060] 由上述描述可知，能够通过表单引擎从其引擎容器中准确快速的找到所需的组件，并实例化组件，从而生成表单。

[0061] 进一步的，所述从所述引擎容器中加载所述配置信息对应组件的实例，具体为：

[0062] 依据对应容器组件的配置信息，将所述容器组件所包含的子组件渲染成对应的页面布局或交互方式，所述子组件包括普通组件和/或载入器组件；

[0063] 依据对应普通组件的配置信息，对普通组件进行渲染；

[0064] 依据对应载入器组件的配置信息，将载入器组件对应的容器组件和普通组件渲染成对应的布局样式。

[0065] 由上述描述可知，依据事先自定义的配置信息，对表单对应的组件进行渲染，获取对应的自由定制的表单页面布局和交互效果，以及表单字段之间的关联关系。

[0066] 进一步的，所述依赖关系包括点击一组件或一组件的预设字段后显示另一组件，以及一组件取预设值时显示另一组件，以及一容器组件中各子组件的显示触发关系；

[0067] 所述逻辑关系包括一组件对应的备选项的值将根据另一组件选中的值变化，以及

组件对应载入器组件的显示触发方式。

[0068] 由上述描述可知,能够对表单的各组件进行依赖关系和逻辑关系的自由定制,实现对表单的页面布局、交互效果,以及表单字段之间逻辑关系的自定义配置,最终实现真正意义上的表单自由配置。

[0069] 进一步的,所述容器组件包括弹出框容器组件、逻辑容器组件、组容器组件、条件判断容器组件、列表容器组件、表格容器组件和选项卡容器组件;所述载入器组件包括底纹、阴影、特定颜色的星号。

[0070] 由上述描述可知,提供多样式多功能的容器组件,并非简单的呈现组件,还提供与普通组件或载入器组件进行结合的功能,从而支持表单的灵活配置。

[0071] 本发明提供的另一个技术方案为:

[0072] 一种基于可配置化表单引擎的表单配置系统,包括模板编辑器、服务端、表单引擎,以及集成器;所述模板编辑器和表单引擎分别与所述服务端连接;所述集成器与所述表单引擎连接;

[0073] 所述模板编辑器,用于配置表单所包含的组件以及组件之间的依赖关系和逻辑关系,生成配置信息并存储至服务器,所述组件包括普通组件、容器组件以及载入器组件;

[0074] 所述表单引擎,用于在启动后,加载包含所述组件的组件库至引擎容器,将每个组件的出入事件挂载到所述表单引擎上;以及用于在其引擎容器接收表单内一组件触发的方法后,根据所述配置信息中所述一组件的依赖关系或逻辑关系,调用对应的组件改变所述表单的显示状态;

[0075] 所述集成器,用于依据所述配置信息生成表单并显示。

[0076] 从上述描述可知,本发明的有益效果在于:同时提供一种基于可配置化表单引擎的表单配置系统,能够根据容器组件、加载器组件、普通组件的相互配合实现表单的布局和交互效果的自由定制;通过依赖配置和逻辑配置实现了组件之间的关联关系自由定制。

[0077] 进一步的,所述表单引擎,还用于将每个组件的出入事件挂载到所述表单引擎上;

[0078] 所述表单引擎,具体通过调用对应的组件的出入事件,改变所述表单的显示状态。

[0079] 进一步的,所述表单引擎,具体通过logic事件订阅机制或dependencies事件回调机制调用所述对应的组件的出入事件。

[0080] 进一步的,所述表单引擎,还用于从服务器载入并解析所述配置信息,从所述引擎容器中加载所述配置信息对应组件的实例;

[0081] 所述集成器,具体通过集成器将所述组件的实例依据所述配置信息生成表单并显示。

[0082] 进一步的,所述集成器,具体依据对应容器组件的配置信息,将所述容器组件所包含的子组件渲染成对应的页面布局或交互方式,所述子组件包括普通组件和/或载入器组件;依据对应普通组件的配置信息,对普通组件进行渲染;依据对应载入器组件的配置信息,将载入器组件对应的容器组件和普通组件渲染成对应的布局样式。

[0083] 进一步的,所述依赖关系包括点击一组件或一组件的预设字段后显示另一组件,以及一组件取预设值时显示另一组件,以及一容器组件中各子组件的显示触发关系;

[0084] 所述逻辑关系包括一组件对应的备选项的值将根据另一组件选中的值变化,以及组件对应载入器组件的显示触发方式。

[0085] 进一步的,所述容器组件包括弹出框容器组件、逻辑容器组件、组容器组件、条件判断容器组件、列表容器组件、表格容器组件和选项卡容器组件;所述载入器组件包括底纹、阴影、特定颜色的星号。

[0086] 实施例一

[0087] 请参照图1至图4所示,本实施例提供一种基于可配置化表单引擎的表单配置方法,基于Formtastic表单引擎系统实现,所述表单引擎系统至少包括:表单引擎3、模板编辑器1、服务端2以及集成器4。

[0088] 如图1所示,所述方法包括:

[0089] S1:配置表单所包含的组件以及组件之间的依赖关系和逻辑关系,生成配置信息并存储至服务器,所述组件包括普通组件、容器组件以及载入器组件。

[0090] 其中,所述容器组件包括但不限于:弹出框容器组件、逻辑容器组件、组容器组件、条件判断容器组件、列表容器组件、表格容器组件和选项卡容器组件;所述载入器组件包括但不限于:底纹、阴影、特定颜色的星号。所述依赖关系包括但不限于:点击一组件或一组件的预设字段后显示另一组件,以及一组件取预设值时显示另一组件,以及一容器组件中各子组件的显示触发关系;所述逻辑关系包括但不限于:一组件对应的备选项的值将根据另一组件选中的值变化,以及组件对应载入器组件的显示触发方式。

[0091] 可选的,通过模板编辑器进行表单的配置并生成表单配置信息的数据。在本实施例中,重点是依赖关系的配置、逻辑关系的配置,然后生成配置信息,并将配置信息保存到服务端数据库中。组件之间依赖关系、逻辑关系的配置,例如将普通组件A通过依赖配置与弹出框容器组件B进行关联,依赖配置中说明,点击上述的组件A后,将触发组件B的显示自己的事件,实现点击弹出预选框功能;将普通组件C与普通组件D通过逻辑配置进行关联,逻辑配置中说明,组件D的备选项的值订阅了组件C的值,当组件C的值变化时组件D的备选项的值也将跟着变化,实现预选功能。

[0092] S2:启动表单引擎,将包含上述配置信息所需组件的组件库加载至引擎容器当中。

[0093] 具体的,如图2所示,表单引擎在启动的时候,将加载组件库(包括普通组件、容器组件以及载入器组件)的组件,然后将每个组件的出入事件都挂载到表单引擎上,使得引擎拥有支配控制组件的能力;同时,Formtastic表单引擎上挂载的组件保持着一种规范,对于组件触发的事件表单引擎都能接收到,对于组件需要接收的事件都有预留给引擎调用的地方。所以当这些组件(包括容器组件、加载器组件、普通组件)被托管到引擎容器中后,引擎就拥有了支配这些组件的能力。

[0094] S3:依据所述配置信息从所述引擎容器加载组件生成表单并显示。

[0095] 具体的,如图3所示,从服务端的数据库中载入配置信息,并对配置信息进行解析;然后依据从引擎容器中找到对应的组件的实例;再通过集成器将所述组件的实例依据所述配置信息集成在一起,生成相应的表单并显示。

[0096] 即,接入Formtastic表单引擎的应用从服务端取回表单的配置信息,传递给表单引擎。表单引擎解析配置信息,实例化组件,将组件托管在引擎容器中,根据配置信息生成表单。

[0097] 其中,组件实例化的过程即根据配置信息对组件进行渲染生成表单的过程,包括:

[0098] 依据对应一容器组件的配置信息,将该容器组件所包含的子组件渲染成对应的页

面布局或交互方式,子组件中可以包括普通组件或载入器组件,或同时包含二者;例如针对容器类组件,容器类组件的配置信息一般需要包含子组件,将根据容器的配置信息把子组件渲染成田字格布局或者弹窗等页面布局或者交互方式。

[0099] 依据对应一普通组件的配置信息,对该普通组件进行渲染。

[0100] 依据对应一载入器组件的配置信息,将该载入器组件对应的容器组件和普通组件渲染成对应的布局样式。载入器类组件会跟随在容器类组件或者普通类组件上面,来控制他们的布局样式。例如,在一组件上加上红色*标志。

[0101] 最后,通过集成器将上述组件集成在一起最终形成一个表单。

[0102] 此时显示的表单为初始化表单,还未接收用户在UI界面上进行信息输入、项目选定、项目展开等编辑操作。

[0103] S4:引擎容器接收表单内一组件触发的方法后,根据所述配置信息中所述一组件的依赖关系或逻辑关系,调用对应的组件改变所述表单的显示状态。

[0104] 用户在UI界面上对表单进行编辑,编辑过程中一组件触发的方法将被引擎容器接收到,引擎容器将根据这一组件触发的方法去改变与其关联的另一组件的状态。具体的,如图4所示,将通过logic事件订阅机制或dependencies事件回调机制调用所述另一组件在上述步骤S2中事先挂载在表单引擎上的出入事件来实现。相关的方法会改变组件的数据,组件接收到这个信息后会解析成控件能够识别的数据,最终实现定义的组件之间的关联关系。

[0105] 实施例二

[0106] 请参照图5,本实施例对应实施例一,提供一具体运用场景,详细说明使用Formtastic表单引擎系统如何自由定义表单的页面布局,组件的交互方式,字段的关联关系等。

[0107] 在本具体运用场景中:

[0108] 组件A:是一个普通组件,它不需要有特殊的底纹阴影样式;用来显示干系人的列表信息。

[0109] 组件B:是一个弹出框容器组件,它不需要有特殊的底纹阴影样式,它用来将某一条干系人的信息显示出详细信息出来。

[0110] 组件C:是一个普通组件,它有一个底纹阴影的样式,它用来选择干系人分类。

[0111] 组件D:是一个普通组件,它有一个底纹阴影的样式,它用来选择干系人名称。

[0112] 如图5所示,期望配置获取以下表单效果:

[0113] 点击组件A能有一个弹出框来弹出选中这条记录的详细信息。

[0114] 点击了组件A的某个字段,在弹出框可以定位聚焦到相应的组件。

[0115] 在弹出框选择干系人分类也就是组件C;每选中一次组件C,组件D的备选项的值就会自动变化;变化的值都是与当下选中的组件C的值有关的数据。

[0116] 在弹出框里面,希望每选中一个字段,对应的字段就能有底纹阴影的样式,并且当选中了下一个,上一个的阴影就没掉了恢复成原来的样子。

[0117] 现有的智能表单引擎只能简单的把组件放到页面上而已,上述这么复杂的布局以及交互,在现有的表单引擎都是无法实现的。

[0118] 在本具体运用场景中,可以基于Formtastic表单引擎通过以下步骤实现:

[0119] 步骤一:在模板编辑器上配置表单。

[0120] 依据上述需求,将配置普通组件A和容器组件B;然后在组件A的配置信息里面做了依赖配置。依赖配置里面说明了,当点击了组件A后会触发的方法名称、相关的数据。在容器组件B里面配置了普通组件C、普通组件D;在组件D做了逻辑配置,逻辑配置里面说明了组件D的备选项的值要自动根据组件C选中的值而变化。给组件C和组件D配上标签载入器组件,这个标签载入器组件可以在该组件选中的时候给组件加上阴影,在组件没被选中的时候隐藏阴影恢复正常。

[0121] 步骤二:将上述配置信息保存到服务端的数据库中。

[0122] 步骤三:在引用表单引擎的地方启动表单引擎,表单引擎把组件库里面的包含组件A、组件B、组件C、组件D等组件全部载入到引擎容器里面,并把各个组件的出入方法都挂在到引擎上面,以供引擎支配这些组件。

[0123] 步骤四:从服务端获取配置信息;根据配置信息当前的表单将加载组件A、组件B、组件C、组件D。

[0124] 步骤五:集成器将根据配置信息生成表单。

[0125] 组件A被直接显示的放在页面上面。容器组件B被隐藏在页面上面,组件C和组件D被放到了容器组件B的里面一起隐藏起来。无标签加载器组件把组件A加载成无标签的形式。所谓的无标签形式就是没有底纹、阴影单纯的把组件显示出来。无标签加载器组件把组件B加载成无标签形式。标签加载器组件把组件C和组件D加载成一旦选中就有底纹、阴影,一旦选中离开底纹、阴影就消失的形式。至此集成器把容器组件、加载器组件、普通组件都集成好了,并且生成表单显示到了页面上面。

[0126] 步骤六:接下来是组件的交互效果。

[0127] 点击了组件A的某个位置触发了A的方法,该方法发送出去后被表单引擎所捕获,表单引擎获得组件A触发的方法后,根据表单配置信息里面配置的“依赖配置”,知道它想要把组件B打开,并且定位到组件B里面的某个位置,所以表单引擎找到组件B里面合适的方法并触发了该方法,使得组件B被显示出来,同时也找到合适的方法定位到了组件B的某个位置。接下来选中组件C,组件C的标签加载器组件产生了作用使得组件C拥有了底纹、阴影。接着选中组件C的具体值,选中组件C的值这个事件发送出来后被表单引擎所捕获到,表单引擎根据组件C和组件D的“逻辑配置”知道组件D的备选项的值订阅了组件C的值,当组件C的值变化的时候组件D的备选项的值也要跟着变化,因此表单引擎去触发了组件D的某个方法使得组件D的备选项的值自动变化了。

[0128] 自此流程结束。总结而言,Formtastic表单引擎系统起到的有益效果:根据容器组件、加载器组件、普通组件的相互配合实现了表单的布局和交互效果。通过依赖配置和逻辑配置实现了组件之间的关联关系。

[0129] 实施例三

[0130] 本实施例对应实施例一和实施例二,提供一种基于可配置化表单引擎的表单配置系统,如图6所示,包括模板编辑器1、服务端2、表单引擎3,以及集成器4;所述模板编辑器1和表单引擎3分别与所述服务端2连接;所述集成器4与所述表单引擎3连接;

[0131] 所述模板编辑器,用于配置表单所包含的组件以及组件之间的依赖关系和逻辑关系,生成配置信息并存储至服务器,所述组件包括普通组件、容器组件以及载入器组件;其

中,所述依赖关系包括点击一组件或一组件的预设字段后显示另一组件,以及一组件取预设值时显示另一组件,以及一容器组件中各子组件的显示触发关系;所述逻辑关系包括一组件对应的备选项的值将根据另一组件选中的值变化,以及组件对应载入器组件的显示触发方式。所述容器组件包括弹出框容器组件、逻辑容器组件、组容器组件、条件判断容器组件、列表容器组件、表格容器组件和选项卡容器组件;所述载入器组件包括底纹、阴影、特定颜色的星号。

[0132] 所述服务端,用来保存表单配置信息以及相应的返回表单配置信息;

[0133] 所述表单引擎,表单引擎用来实例化组件、托管组件、控制组件之间的互相调用。具体用于在启动后,加载包含所述组件的组件库至引擎容器;以及用于在其引擎容器接收表单内一组件触发的方法后,根据所述配置信息中所述一组件的依赖关系或逻辑关系,调用对应的组件改变所述表单的显示状态。其中,除了这些方便的配置Formtastic还提供了方便开发的接口,通过courier事件拦截机制对组件的相关行为进行拦截,开发者可以在拦截中进行必要的操作,然后再返回出去,以此来实现用户自定义功能的实现。

[0134] 所述集成器,用于依据所述配置信息生成表单并显示。

[0135] 其中,所述表单引擎,还用于将每个组件的出入事件挂载到所述表单引擎上;具体通过logic事件订阅机制或dependencies事件回调机制调用所述对应的组件的出入事件,改变所述表单的显示状态。

[0136] 所述表单引擎,还用于从服务器载入并解析所述配置信息,从所述引擎容器中加载所述配置信息对应组件的实例。

[0137] 所述集成器,具体用于将所述组件的实例依据所述配置信息生成表单并显示。进一步具体依据对应容器组件的配置信息,将所述容器组件所包含的子组件渲染成对应的页面布局或交互方式,所述子组件包括普通组件和/或载入器组件;依据对应普通组件的配置信息,对普通组件进行渲染;依据对应载入器组件的配置信息,将载入器组件对应的容器组件和普通组件渲染成对应的布局样式。

[0138] 综上所述,本发明提供一种基于可配置化表单引擎的表单配置方法及系统,能够通过容器组件、载入器组件以及普通组件的组合实现对表单页面布局和交互方式的自由定制;通过对组件间依赖关系和逻辑关系的配置实现了对表单字段之间的关联关系进行自由定制。本发明能够实现真正意义上的表单灵活地自定义布局,丰富表单的变现形式和功能。

[0139] 以上所述仅为本发明的实施例,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等同变换,或直接或间接运用在相关的技术领域,均同理包括在本发明的专利保护范围内。

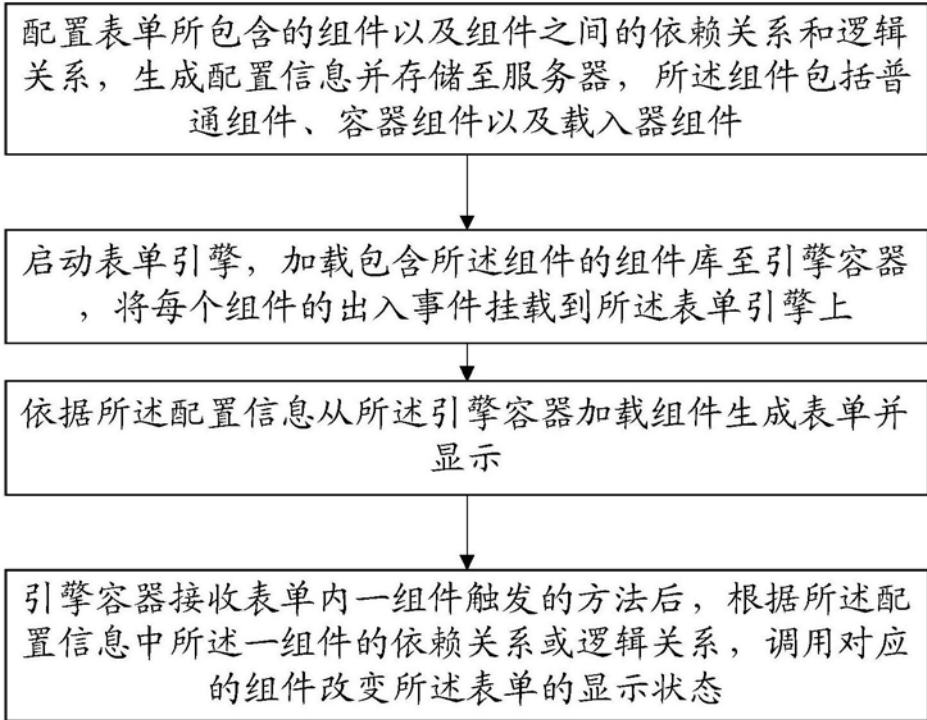


图1

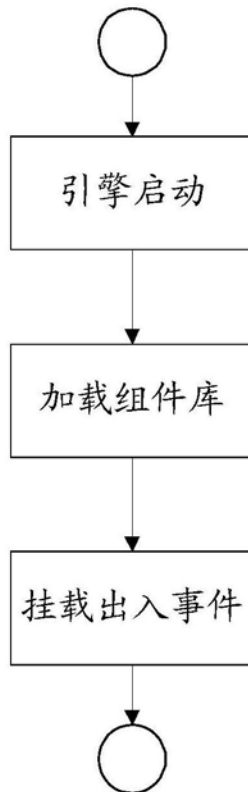


图2

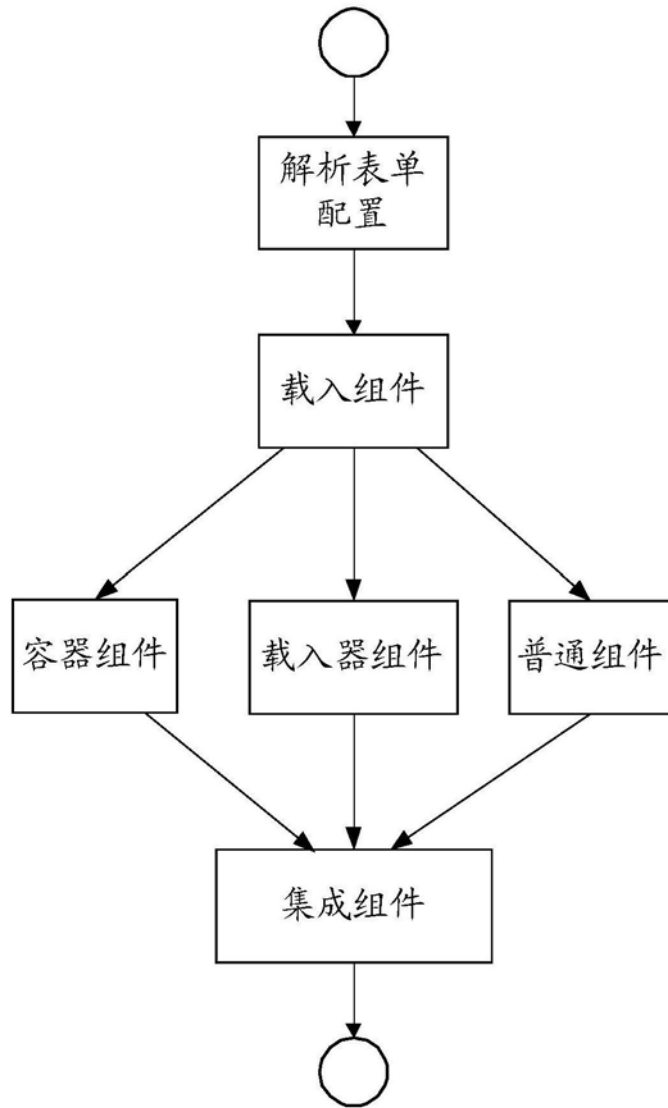


图3

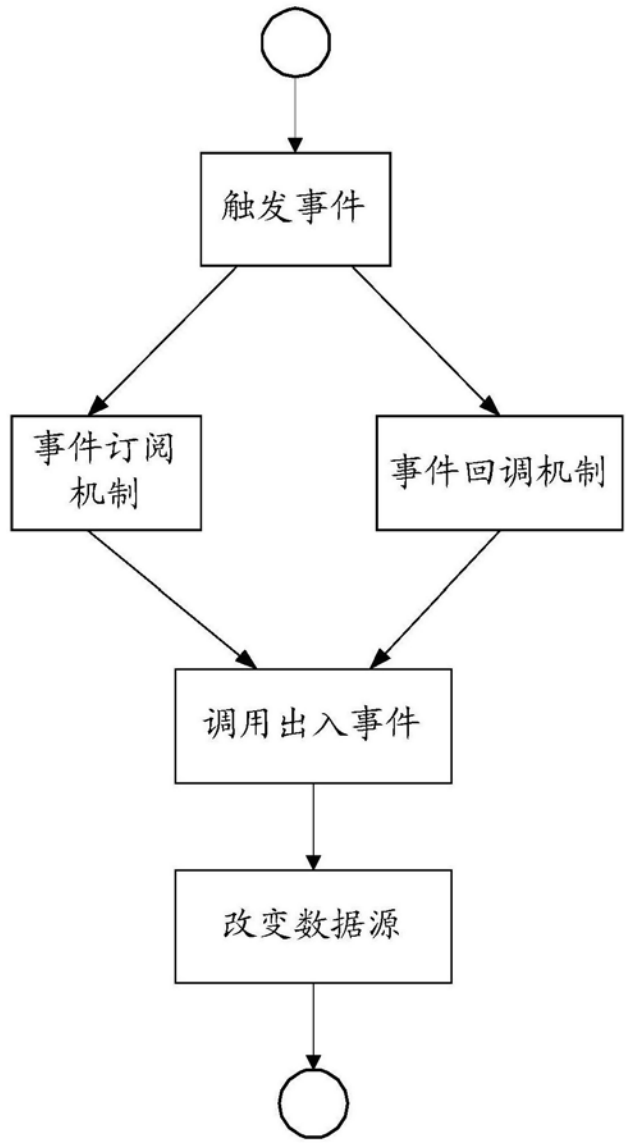


图4



图5

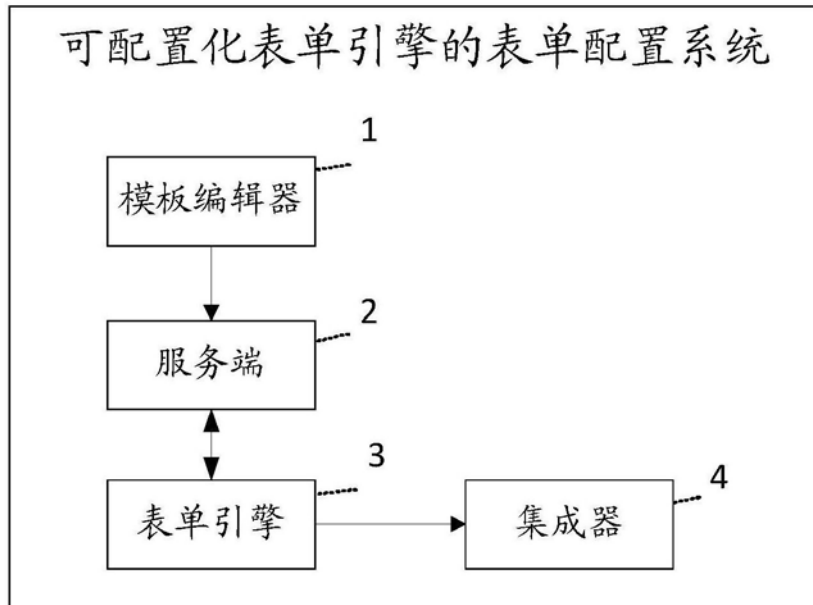


图6