

混合视图的显示方法及装置

本申请要求 2016 年 03 月 24 日递交的申请号为 201610172774.1、发明名称为“混合视图的显示方法及装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5

技术领域

本发明涉及网页处理技术领域，尤其涉及一种混合视图的显示方法及装置。

背景技术

10 随着信息技术的快速发展，人们对移动终端设备上的 APP（Application，应用程序）要求越来越高，鉴于移动终端的局限性，移动终端上的 APP 由本地化应用(Native App)，到混合型应用(Hybrid APP)，再到基于 WEB 的应用 Web App，这一连串的变化都源于技术的更新和市场的需要。其中，Hybrid App 是指介于 web-app、native-app 这两者之间的 App，通过它可以同时展示某一个应用程序的网页视图和客户端视图，网页视图是运行
15 在浏览器上的 HTML5（超文本标记语言）页面，客户端视图是由 iOS/Android 开发语言编写的用于展示性的页面或控件，比如淘宝应用程序，通过 Hybrid App 可以混合展示淘宝应用程序网页视图和客户端视图。

目前，可以通过在移动终端 webView（客户端网页容器）当中添加子视图的方式实现混合型应用，即通过在 webView 中添加客户端视图实现应用中网页视图和客户端视图
20 的混合显示。如在 iOS 系统中通过向 webView.scrollView 中添加客户端视图；或者在 Android 系统中通过 webView.addView 方法可以直接向 webView 中添加客户端视图。然而，由于网页视图是依靠浏览器引擎进行渲染，而客户端视图是依靠操作系统的编译运行，两者环境不一样，因此在网页视图进行滚动变化时，造成现有混合视图显示不一致。

25 发明内容

鉴于上述问题，提出了本发明，以便提供一种克服上述问题或者至少部分地解决上述问题的混合视图的显示方法及装置。

为达到上述目的，本发明主要提供如下技术方案：

一方面，本发明实施例提供了一种混合视图的显示方法，该方法包括：

30 当接收到网页试图加载指令时，对所述网页视图进行加载，所述网页视图中包含有

用于添加客户端视图的占位元素；

从所述网页视图的加载结果中获取所述占位元素的位置信息；

在所述位置信息添加所述客户端视图；

显示所述添加有所述客户端视图的网页视图。

5 另一方面，本发明实施例还提供一种混合视图的显示装置，该装置包括：

加载单元，当接收到网页视图加载指令时，对所述网页视图进行加载，所述网页视图中包含有用于添加客户端视图的占位元素；

获取单元，从所述网页视图的加载结果中获取所述占位元素的位置信息；

添加单元，在所述位置信息添加所述客户端视图；

10 显示单元，显示所述添加有所述客户端视图的网页视图。

借由上述技术方案，本发明实施例提供的技术方案至少具有下列优点：

本发明实施例提供的一种混合视图的显示方法及装置，当接收到网页视图加载指令时，首先对网页视图进行加载，其中，所述网页视图中包含有用于添加客户端视图的占位元素，然后从所述网页视图的加载结果中获取所述占位元素的位置信息，并在所述位置信息添加所述客户端视图，最后显示所述添加有所述客户端视图的网页视图。与目前通过在 `webView` 当中直接添加子视图的方式实现混合视图的显示相比，本发明实施例在接收到网页视图加载指令时，首先从网页视图的加载结果中获取占位元素的位置信息，然后将客户端视图添加到占位元素位置上，以此实现网页视图和客户端视图的混合展示，由于本发明实施例中的占位元素可以通过浏览器引擎渲染，因此将客户端视图添加到加载后的占位元素对应的位置上，可以使客户端视图随着网页视图的滚动而滚动，从而通过本发明可以提高混合视图显示的准确度及一致性。

15 20

附图说明

通过阅读下文优选实施方式的详细描述，各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。附图仅用于示出优选实施方式的目的，而并不认为是对本发明的限制。而且在整个附图中，用相同的参考符号表示相同的部件。在附图中：

25

图 1 为本发明实施例提供的一种混合视图的显示方法流程图；

图 2 为本发明实施例提供的另一种混合视图的显示方法流程图；

图 3 为本发明实施例提供的一种混合视图的显示装置的组成框图；

30 图 4 为本发明实施例提供的另一种混合视图的显示装置的组成框图；

图 5 为本发明实施例提供的混合视图的界面示意图。

具体实施方式

下面将参照附图更详细地描述本公开的示例性实施例。虽然附图中显示了本公开的
5 示例性实施例，然而应当理解，可以以各种形式实现本公开而不应被这里阐述的实施例
所限制。相反，提供这些实施例是为了能够更透彻地理解本公开，并且能够将本公开的范围完整的传达给本领域的技术人员。

为使本发明技术方案的优点更加清楚，下面结合附图和实施例对本发明作详细说明。

本发明实施例提供了一种混合视图的显示方法，如图 1 所示，所述方法包括：

10 101、当接收到网页视图加载指令时，对所述网页视图进行加载。

其中，所述网页视图中包含有用于添加客户端视图的占位元素。在本发明实施例中，
网页视图加载指令可通过打开应用程序中的一个具体页面触发，如选择支付宝中花呗的
还款按钮，招财宝中的预约购买按钮等，本发明实施例不做具体限定。

需要说明的是，所述网页视图中包含的占位元素是可以改变对象大小或宽高的元素。
15 在本发明实施例中，所述占位元素具体可以为块级元素，或表现形式为块级元素的行级
元素，本发明实施例不做具体限定。例如，通过块级元素<div id=" native-view" ></div>
表示占位元素，或是通过表现形式为块级元素的行级元素<span id=" native-view"
style=" display:block" > 表示占位元素，由于上述行级元素中设置的
style=" display:block" 代表的是让对象成为块级元素，因此上述行级元素实际上的表现
20 形式为块级元素，从而该行级元素是改变对应的大小或宽度的。

102、从所述网页视图的加载结果中获取所述占位元素的位置信息。

对于本发明实施例，由于是对网页视图进行的加载，而网页视图就是 HTML 页面，
因此在本发明实施例中是通过浏览器渲染引擎对待显示网页视图进行加载的。需要说明
的是，本发明中对于不同的终端设备操作系统是通过其系统内部方式监听网页视图加载，
25 如在 iOS （苹果移动设备操作系统）当中，是通过接口 webViewDidFinishLoad (iOS)获
知网页视图加载是否完成；在 Android（安卓操作系统）当中，是通过接口 onPageFinished
(Android)获知网页视图加载是否完成。

在本发明实施例中，在待显示网页视图的加载结果中通过 webview（客户端网页视
图容器）执行 javascript（脚本语言）获取占位元素的位置信息，根据占位元素的标示方
30 式不同，这里获取占位元素的方式有差异。例如，获取占位元素的位置信息的脚本代码

为:

```
(function(){  
  var ele = document.getElementById('native-view');  
  var xPoint = ele.getBoundingClientRect().left;  
5  var yPoint = ele.getBoundingClientRect().top + document.body.scrollTop;  
  return JSON.stringify({x: xPoint, y: yPoint});  
})();
```

其中, 上述代码中'native-view'表示的是占位元素的标识, xPoint 表示占位元素相对浏览器视窗的左边位置距离信息, yPoint 用于获得页面中占位元素相对于浏览器视窗上
10 边的位置, 因此通过坐标轴中的 x 和 y 就可以确定网页视图的加载结果中占位元素的位置。

103、在所述位置信息添加所述客户端视图。

其中, 所述客户端视图是由相应手机操作系统语言编写并编译的原生视图控件, 例如 iOS 的 UISwitch (开关控件), Android 的 TextView (文本显示) 或者用户编写的
15 自定义视图控件。

104、显示所述添加有所述客户端视图的网页视图。

对于本发明实施例, 当接收到网页视图加载指令时, 首先从网页视图的加载结果中获取占位元素的位置信息, 然后将客户端视图对应添加到占位元素的位置上, 以此实现网页视图和客户端视图的混合展示, 由于本发明实施例中的占位元素可以通过浏览器引擎渲染, 因此将客户端视图添加到加载后的占位元素对应的位置上, 可以使客户端视图
20 随着网页视图的滚动而滚动, 从而通过在网页视图中添加占位元素可实现流布局的混合视图展现, 进而通过本发明可以提高混合视图显示的准确度及一致性。

本发明实施例提供的一种混合视图的显示方法, 当接收到网页视图加载指令时, 首先对所述网页视图进行加载, 其中, 所述网页视图中包含有用于添加客户端视图的占位
25 元素, 然后从所述网页视图的加载结果中获取所述占位元素的位置信息, 并在所述位置信息添加所述客户端视图, 最后显示所述添加有所述客户端视图的网页视图。与目前通过在 webView 当中直接添加子视图的方式实现混合视图的显示相比, 本发明实施例在接收到网页视图加载指令时, 首先从网页视图的加载结果中获取占位元素的位置信息, 然后将客户端视图对应添加到占位元素的位置上, 以此实现网页视图和客户端视图的混合
30 展示, 由于本发明实施例中的占位元素可以通过浏览器引擎渲染, 因此将客户端视图添

加到加载后的占位元素对应的位置上，可以使客户端视图随着网页视图的滚动而滚动，从而通过本发明可以提高混合视图显示的准确度及一致性。

本发明实施例提供了另一种混合视图的显示方法，如图 2 所示，所述方法包括：

201、在所述网页视图中添加所述占位元素。

5 其中，每个占位元素设置有对应的标识信息，所述标识信息用于唯一标识占位元素。每个占位元素对应的标识信息具体可以通过 class 或者 id 设置。例如，在以块级元素<div class=" native-view1 " ></div>表示的占位元素中，是通过 class 设置占位元素的标识信息，即该占位元素的标识信息为 native-view1；在以表现形式为块级元素的行级元素表示的占位元素中，是通过 id 设置
10 占位元素的标识信息，即该占位元素的标识信息为 native-view2。

需要说明的是，网页视图中包含的占位元素是可以改变对象大小或宽高的元素。在本发明实施例中，所述占位元素具体可以为块级元素，或表现形式为块级元素的行级元素，本发明实施例不做具体限定。对于本发明实施例，将占位元素设置成块级元素或是表现形式为块级元素的行级元素，可以在后续步骤当中设置占位元素的宽高，即通过占
15 位元素可以调整客户端视图的大小，从而通过本发明实施例可以提高客户端视图显示的精确度及一致性。

202、设置各个占位元素分别对应的客户端视图。

其中，所述客户端视图是由相应手机操作系统语言编写并编译的原生视图控件，例如 iOS 的 UISwitch (开关控件)， Android 的 TextView （文本显示）或者用户编写的
20 自定义视图控件，本发明实施例不做具体限定。需要说明的是，由于页面视图当中可能包括多个客户端视图，而每个占位元素可以添加一个客户端视图，因此在网页视图当中添加多个占位元素之后，还需要设置各个占位元素分别对应的客户端视图。在本发明实施例中，通过设置各个占位元素分别对应的客户端视图，可以在后续步骤中将多个客户端视图相应的添加到占位元素对应的位置信息上。

25 203、当接收到网页视图加载指令时，对所述网页视图进行加载。

其中，所述网页视图中包含有用于添加客户端视图的占位元素，在本发明实施例中，网页视图加载指令可通过打开应用程序中的一个具体页面触发，如选择支付宝中花呗的还款按钮，招财宝中的预约购买按钮等，本发明实施例不做具体限定。

204、判断是否能够从所述网页视图的加载结果中获取到所述占位元素的位置信息。

30 需要说明的是，本发明中对于不同的终端设备操作系统是通过其系统内部方式监听

网页视图的加载，如在 iOS（苹果移动设备操作系统）当中，是通过接口 `webViewDidFinishLoad` (iOS) 获知网页视图是否加载完成；在 Android（安卓操作系统）当中，是通过接口 `onPageFinished` (Android) 获知网页视图是否加载完成。其中，所述占位元素的位置信息为页面元素在可视区的位置，及占位元素的左，上分别相对浏览器视窗的位置。

205a、若是，则获取所述占位元素的位置信息。

在本发明实施例中，若够从所述待显示网页视图的加载结果中获取到所述占位元素的位置信息，则获取所述占位元素的位置信息。

206a、在所述各个位置信息上，分别添加与所述占位元素对应的客户端视图。

10 对于本发明实施例，所述占位元素的表现形式为块级元素，在步骤 206 a 之后，所述方法还包括：将所述占位元素的状态信息配置成所述客户端视图对应的状态信息。其中，所述占位元素的状态信息具体可以为状态元素的宽高信息、分辨率信息等，本发明实施例不做具体限定。需要说明的是，占位元素的表现形式为块级元素具体可以为：块级元素或者表现形式为块级元素的行级元素，即一切可以改变占位元素的状态信息的元素即可，本发明实施例不做具体限定。

在本发明实施例中，将所述占位元素的状态信息配置成所述客户端视图对应的状态信息包括：将所述占位元素的宽高信息设置成所述客户端视图对应的宽高信息。例如，可通过以下代码设置转为元素的宽高信息：

```
document.getElementById( 'native-view' ).style.height = '400px' ;  
20 document.getElementById( 'native-view' ).style.width = '100px' ;
```

其中，400 为客户端视图的实际高度，100 为客户端视图的实际宽度。对于本发明实施例，将占位元素的宽高信息设置成所述客户端视图对应的宽高信息，可以使添加的客户端视图在待显示网页视图当中全部展现出来，从而通过将所述占位元素的宽高信息设置成所述客户端视图对应的宽高信息，可以提高客户端视图的显示精确度，进而提高了混合视图的显示精确度及一致性。

25 对于本发明实施例，在所述位置信息添加所述客户端视图包括：从所述网页视图的加载结果中获取各个占位元素分别对应的位置信息；在所述各个位置信息上，分别添加与所述占位元素对应的客户端视图。在本发明实施例当中，若网页视图当中存在多个占位元素时，则在网页视图加载完成后，会从加载结果中获取各个占位元素分别对应的位置信息，然后在各个占位元素的位置信息上添加与占位元素对应的客户端视图，以此实

现在网页视图当中添加多个客户端视图。

207a、显示所述添加有所述客户端视图的网页视图。

205b、若否，则输出网页视图加载错误提示信息。

其中，步骤 205b 是与步骤 205a 的并列步骤，若不能够从所述网页视图的加载结果
5 中获取到所述占位元素的位置信息，则输出网页视图加载提示信息，以提示用于从新加载所述网页视图。需要说明的是，在终端设备出现无网络时或是网页视图加载过程中出现错误等状况时，将到会导致不能够从所述网页视图的加载结果中获取到所述占位元素的位置信息。

对于本发明实施例，可以应用的场景如图 5 所示，但不仅限于于此，包括：当接收
10 到网页视图加载指令时，首先对网页视图进行加载，即对图 5 当中网页视图虚线框内的网页元素进行加载，然后判断是否能够从所述网页视图的加载结果中获取到所述占位元素的位置信息，即判断是否能够从网页视图的加载结果中获取到占位元素为“联系我们”的位置信息，若能够从所述网页视图的加载结果中获取到“联系我们”占位元素的位置信息，则获取“联系我们”占位元素的位置信息，并在 webView 中获取的位置信息上
15 添加所述客户端视图，即在图 5 中“联系我们”虚线框对应位置的 webView 中添加客户端视图，以此实现网页视图和客户端视图的混合显示；若不能够从所述网页视图的加载结果中获取到所述占位元素的位置信息，则输出网页视图加载错误提示信息，以提示用于从新加载所述网页视图。在本发明实施例中，由于本发明实施例中的占位元素可以通过浏览器引擎渲染，因此将客户端视图添加到加载后的占位元素对应的位置上，可以
20 使客户端视图随着网页视图的滚动而滚动，即图 5 中“联系我们”虚线框内的客户端视图可以随着网页视图虚线框内的网页视图滚动而滚动，因此通过在网页视图中添加占位元素可实现流布局的混合视图展现，从而通过本发明可以提高混合视图显示的准确度及一致性。

本发明实施例提供的另一种混合视图的显示方法，当接收到网页视图加载指令时，
25 首先对所述网页视图进行加载，其中，所述网页视图中包含有用于添加客户端视图的占位元素，然后从所述网页视图的加载结果中获取所述占位元素的位置信息，并在所述位置信息添加所述客户端视图，最后显示所述添加有所述客户端视图的网页视图。与目前通过在 webView 当中直接添加子视图的方式实现混合视图的显示相比，本发明实施例在接收到网页视图加载指令时，首先从网页视图的加载结果中获取占位元素的位置信息，
30 然后将客户端视图对应添加到占位元素的位置上，以此实现网页视图和客户端视图的混

合展示，由于本发明实施例中的占位元素可以通过浏览器引擎渲染，因此将客户端视图添加到加载后的占位元素对应的位置上，可以使客户端视图随着网页视图的滚动而滚动，从而通过本发明可以提高混合视图显示的准确度及一致性。

进一步地，本发明实施例提供一种混合视图的显示装置，如图 3 所示，所述装置包
5 括：加载单元 31，获取单元 32，添加单元 33，显示单元 34。

加载单元 31，当接收到网页视图加载指令时，对所述网页视图进行加载，所述网页视图中包含有用于添加客户端视图的占位元素；

获取单元 32，从所述网页视图的加载结果中获取所述占位元素的位置信息；

添加单元 33，在所述位置信息添加所述客户端视图；

10 显示单元 34，显示所述添加有所述客户端视图的网页视图。

需要说明的是，本发明实施例提供一种混合视图的显示装置所涉及各功能单元的其他相应描述，可以参考图 1 所示方法的对应描述，在此不再赘述，但应当明确，本实施例中的装置能够对应实现前述方法实施例中的全部内容。

本发明实施例提供一种混合视图的显示装置，当接收到网页视图加载指令时，首
15 先对所述网页视图进行加载，其中，所述网页视图中包含有用于添加客户端视图的占位元素，然后从所述网页视图的加载结果中获取所述占位元素的位置信息，并在所述位置信息添加所述客户端视图，最后显示所述添加有所述客户端视图的网页视图。与目前通过在 webView 当中直接添加子视图的方式实现混合视图的显示相比，本发明实施例在接收到网页视图加载指令时，首先从网页视图的加载结果中获取占位元素的位置信息，然
20 后将客户端视图对应添加到占位元素的位置上，以此实现网页视图和客户端视图的混合展示，由于本发明实施例中的占位元素可以通过浏览器引擎渲染，因此将客户端视图添加到加载后的占位元素对应的位置上，可以使客户端视图随着网页视图的滚动而滚动，从而通过本发明可以提高混合视图显示的准确度及一致性。

进一步地，本发明实施例提供另一种混合视图的显示装置，如图 4 所示，所述装置
25 包括：加载单元 41，获取单元 42，添加单元 43，显示单元 44。

加载单元 41，当接收到网页视图加载指令时，对所述网页视图进行加载，所述网页视图中包含有用于添加客户端视图的占位元素；

获取单元 42，从所述网页视图的加载结果中获取所述占位元素的位置信息；

添加单元 43，在所述位置信息添加所述客户端视图；

30 显示单元 44，显示所述添加有所述客户端视图的网页视图。

对于本发明实施例，所述装置还包括：设置单元 45；

所述添加单元 43，在所述网页视图添加所述占位元素，每个占位元素设置有对应的标识信息；

所述设置单元 45，设置各个占位元素分别对应的客户端视图。

5 对于本发明实施例，所述占位元素的表现形式为块级元素，所述装置还包括：

配置单元 46，将所述占位元素的状态信息配置成所述客户端视图对应的状态信息。

所述配置单元 46，将所述占位元素的宽高信息设置成所述客户端视图对应的宽高信息。

对于本发明实施例，所述添加单元 43 包括：

10 获取模块 431，从所述网页视图的加载结果中获取各个占位元素分别对应的位置信息；

添加模块 432，在所述各个位置信息上，分别添加与所述占位元素对应的客户端视图。

对于本发明实施例，所述装置还包括：

15 输出单元 47，若未从所述网页视图的加载结果中获取到所述占位元素的位置信息，则输出网页视图加载错误提示信息。

需要说明的是，本发明实施例提供一种混合视图的显示装置所涉及各功能单元的其他相应描述，可以参考图 2 所示方法的对应描述，在此不再赘述，但应当明确，本实施例中的装置能够对应实现前述方法实施例中的全部内容。

20 本发明实施例提供的另一种混合视图的显示装置，当接收到网页视图加载指令时，首先对所述网页视图进行加载，其中，所述网页视图中包含有用于添加客户端视图的占位元素，然后从所述网页视图的加载结果中获取所述占位元素的位置信息，并在所述位置信息添加所述客户端视图，最后显示所述添加有所述客户端视图的网页视图。与目前通过在 `webView` 当中直接添加子视图的方式实现混合视图的显示相比，本发明实施例在
25 接收到网页视图加载指令时，首先从网页视图的加载结果中获取占位元素的位置信息，然后将客户端视图对应添加到占位元素的位置上，以此实现网页视图和客户端视图的混合展示，由于本发明实施例中的占位元素可以通过浏览器引擎渲染，因此将客户端视图添加到加载后的占位元素对应的位置上，可以使客户端视图随着网页视图的滚动而滚动，从而通过本发明可以提高混合视图显示的准确度及一致性。

30 所述混合视图的显示装置包括处理器和存储器，上述加载单元、获取单元、添加单

元、显示单元、设置单元、配置单元和输出单元等均作为程序单元存储在存储器中，由处理器执行存储在存储器中的上述程序单元来实现相应的功能。

处理器中包含内核，由内核去存储器中调取相应的程序单元。内核可以设置一个或以上，通过调整内核参数来提高混合视图显示的准确率。

- 5 存储器可能包括计算机可读介质中的非永久性存储器，随机存取存储器(RAM)和/或非易失性内存等形式，如只读存储器(ROM)或闪存(flash RAM)，存储器包括至少一个存储芯片。

本申请还提供了一种计算机程序产品，当在数据处理设备上执行时，适于执行初始化有如下方法步骤的程序代码：当接收到网页视图显示位置切换指令时，对待显示网页
10 视图进行加载，所述待显示网页视图中包含有用于添加客户端视图的占位元素；从所述待显示网页视图的加载结果中获取所述占位元素的位置信息；在所述位置信息添加所述客户端视图；显示所述添加有所述客户端视图的网页视图。

本领域内的技术人员应明白，本申请的实施例可提供为方法、系统、或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的
15 实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器、CD-ROM、光学存储器等）上实施的计算机程序产品的形式。

本申请是参照根据本申请实施例混合视图的显示方法及装置、和计算机程序产品的流程图和 / 或方框图来描述的。应理解可由计算机程序指令实现流程图和 / 或方框图中的
20 的每一流程和 / 或方框、以及流程图和 / 或方框图中的流程和 / 或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的装置。

25 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能。

30 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或

其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和 / 或方框图一个方框或多个方框中指定的功能的步骤。

在一个典型的配置中，计算设备包括一个或多个处理器 (CPU)、输入/输出接口、网络接口和内存。

- 5 存储器可能包括计算机可读介质中的非永久性存储器，随机存取存储器(RAM)和/或非易失性内存等形式，如只读存储器(ROM)或闪存(flash RAM)。存储器是计算机可读介质的示例。

计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。

- 10 计算机的存储介质的例子包括，但不限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器 (CD-ROM)、数字多功能光盘 (DVD) 或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计
15 算设备访问的信息。按照本文中的界定，计算机可读介质不包括暂存电脑可读媒体 (transitory media)，如调制的数据信号和载波。

以上仅为本申请的实施例而已，并不用于限制本申请。对于本领域技术人员来说，本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等，均应包含在本申请的权利要求范围之内。

权利要求书

1、一种混合视图的显示方法，其特征在于，包括：

当接收到网页试图加载指令时，对所述网页视图进行加载，所述网页视图中包含有用于添加客户端视图的占位元素；

5 从所述网页视图的加载结果中获取所述占位元素的位置信息；

在所述位置信息添加所述客户端视图；

显示所述添加有所述客户端视图的网页视图。

2、根据权利要求 1 所述的方法，其特征在于，对所述网页视图进行加载之前，所述方法还包括：

10 在所述网页视图中添加所述占位元素，每个占位元素设置有对应的标识信息；

设置各个占位元素分别对应的客户端视图。

3、根据权利要求 2 所述的方法，其特征在于，所述占位元素的表现形式为块级元素，在所述位置信息添加所述客户端视图之后，所述方法还包括：

将所述占位元素的状态信息配置成所述客户端视图对应的状态信息。

15 4、根据权利要求 3 所述的方法，其特征在于，将所述占位元素的状态信息配置成所述客户端视图对应的状态信息包括：

将所述占位元素的宽高信息设置成所述客户端视图对应的宽高信息。

5、根据权利要求 2 所述的方法，其特征在于，在所述位置信息添加所述客户端视图包括：

20 从所述显示网页视图的加载结果中获取各个占位元素分别对应的位置信息；

在各个所述位置信息上，分别添加与所述占位元素对应的客户端视图。

6、根据权利要求 1 所述的方法，其特征在于，所述方法还包括：

若未从所述网页视图的加载结果中获取到所述占位元素的位置信息，则输出网页视图加载错误提示信息。

25 7、一种混合视图的显示装置，其特征在于，包括：

加载单元，当接收到网页视图加载指令时，对所述网页视图进行加载，所述网页视图中包含有用于添加客户端视图的占位元素；

获取单元，从所述网页视图的加载结果中获取所述占位元素的位置信息；

添加单元，在所述位置信息添加所述客户端视图；

30 显示单元，显示所述添加有所述客户端视图的网页视图。

8、根据权利要求 7 所述的装置，其特征在于，所述装置还包括：设置单元；

所述添加单元，在所述网页视图添加所述占位元素，每个占位元素设置有对应的标识信息；

所述设置单元，设置各个占位元素分别对应的客户端视图。

5 9、根据权利要求 8 所述的装置，其特征在于，所述占位元素的表现形式为块级元素，所述装置还包括：

配置单元，将所述占位元素的状态信息配置成所述客户端视图对应的状态信息。

10、根据权利要求 9 所述的装置，其特征在于，

所述配置单元，将所述占位元素的宽高信息设置成所述客户端视图对应的宽高信息。

10 息。