



(12) 发明专利申请

(10) 申请公布号 CN 103164401 A

(43) 申请公布日 2013. 06. 19

(21) 申请号 201110405714. 7

(22) 申请日 2011. 12. 08

(71) 申请人 航天信息软件技术有限公司

地址 100195 北京市海淀区杏石口路甲 18
号

(72) 发明人 解军伟 瞿建军 李长山

(74) 专利代理机构 北京国昊天诚知识产权代理
有限公司 11315

代理人 张永林

(51) Int. Cl.

G06F 17/30 (2006. 01)

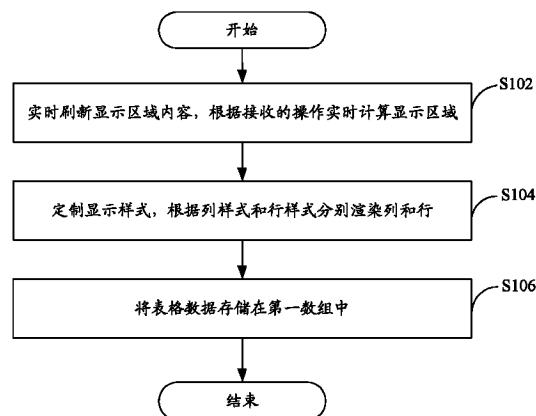
权利要求书1页 说明书4页 附图2页

(54) 发明名称

页面表格显示方法

(57) 摘要

本发明公开了一种页面表格显示方法，其包括：实时刷新显示区域内容，根据接收的操作实时计算显示区域；定制显示样式，根据列样式和行样式分别渲染列和行；将表格数据存储在第一数组中。通过本发明，使得表格展示数据变得方便，迅速，避免了大数据量加载时引起的页面死锁问题，克服了表格数据量大时 DOM 元素过多，加载和查找缓慢的缺点。



1. 一种页面表格显示方法,其特征在于,包括:
实时刷新显示区域内容,根据接收的操作实时计算显示区域;
定制显示样式,根据列样式和行样式分别渲染列和行;
将表格数据存储在第一数组中。
2. 根据权利要求 1 所述的方法,其特征在于,实时刷新显示区域内容包括:
根据表格数据和滚动条位置,动态计算显示区域中显示的列区间和行区间;
渲染显示区域内容。
3. 根据权利要求 2 所述的方法,其特征在于,动态计算显示区域中显示的列区间包括:
根据设置去除表头、表格合计条、表格分页条的高度。
4. 根据权利要求 2 所述的方法,其特征在于,渲染显示区域内容包括:
同步刷新表头数据;
同步刷新合计栏数据。
5. 根据权利要求 1 所述的方法,其特征在于,定制显示样式包括:
针对列,设置列样式和渲染器,通过列样式控制该列显示样式,渲染器精确控制到单元格的显示内容和样式;
针对行,设置行高度和行样式,通过隐藏在行数据里的变量控制该行的高度以及显示样式。
6. 根据权利要求 1 所述的方法,其特征在于,还包括:
将显示数据存储在第二数组中,从而实现数据过滤功能以及树表功能。
7. 根据权利要求 1 所述的方法,其特征在于,所述将表格数据存储在第一数组中,包括:
表格数据采用 json 方式来进行保存,并为每行制定唯一的键值。

页面表格显示方法

技术领域

[0001] 本发明涉及一种页面表格显示方法。

背景技术

[0002] 在实际应用中,表格的应用越来越广泛,尤其是信息系统,目前很多表格显示是把数据一次性全部加载,有少部分使用动态加载机制,但是加载方式是逐渐累加的方式,致使数据量大时,页面元素过多,内存占用很多,页面操作效率极低。另外对于目前表格对显示的控制很不方便,编写代码量很多,且不易维护。

发明内容

[0003] 本发明的主要目的在于提供一种页面表格显示方法,以解决现有技术存在的大数据量加载时引起的页面死锁、以及表格数据量大时文件对象模型 (Document Object Model, 简称 DOM) 元素过多,加载和查找缓慢的问题,其中:

[0004] 根据本发明实施例的页面表格显示方法包括:

[0005] 实时刷新显示区域内容,根据接收的操作实时计算显示区域;定制显示样式,根据列样式和行样式分别渲染列和行;将表格数据存储在第一数组中。

[0006] 其中,实时刷新显示区域内容包括:根据表格数据和滚动条位置,动态计算显示区域中显示的列区间和行区间;渲染显示区域内容。

[0007] 其中,动态计算显示区域中显示的列区间包括:根据设置去除表头、表格合计条、表格分页条的高度。

[0008] 其中,渲染显示区域内容包括:同步刷新表头数据;同步刷新合计栏数据。

[0009] 其中,定制显示样式包括:针对列,设置列样式和渲染器,通过列样式控制该列显示样式,渲染器精确控制到单元格的显示内容和样式;针对行,设置行高度和行样式,通过隐藏在行数据里的变量控制该行的高度以及显示样式。

[0010] 其中,还包括:将显示数据存储在第二数组中,从而实现数据过滤功能以及树表功能。

[0011] 其中,将表格数据存储在第一数组中,包括:表格数据采用 json 方式来进行保存,并为每行制定唯一的键值。

[0012] 根据本发明的技术方案,使得表格展示数据变得方便,迅速,避免了大数据量加载时引起的页面死锁问题,克服了表格数据量大时 DOM 元素过多,加载和查找缓慢的缺点。

附图说明

[0013] 此处所说明的附图用来提供对本发明的进一步理解,构成本申请的一部分,本发明的示意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。在附图中:

[0014] 图 1 是本发明实施例的页面表格显示方法的流程图;

[0015] 图 2 是本发明实施例的数据模型的示意图

[0016] 图 3 是本发明实施例的数据发生变化或滚动条发生变化时处理计算和显示的流程图；

[0017] 图 4 是本发明实施例的显示区域和占位区域的示意图。

具体实施方式

[0018] 本发明的主要思想在于，在表格数据发生变化时，动态计算显示区域；在滚动条发生变化时，动态计算应当显示的行区间和列区间，只绘制这个区间的页面元素，并利用绝对定位的方式，减少了依赖的页面元素；并且在数据显示时，采用列渲染的方式来定制某一列的显示样式和数据内容，行渲染则采用隐藏属性的方式来进行定制。

[0019] 为使本发明的目的、技术方案和优点更加清楚，以下结合附图及具体实施例，对本发明作进一步地详细说明。

[0020] 根据本发明的实施例，提供了一种页面表格显示方法。参考图 1，图 1 是本发明实施例的页面表格显示方法的流程图，如图 1 所示，该方法包括（步骤 S102-S106）：

[0021] 步骤 S102，实时刷新显示区域内容，根据接收的操作实时计算显示区域。

[0022] 其中，实时刷新显示区域内容包括：根据表格数据和滚动条位置，动态计算显示区域中应该显示的列区间和行区间，进而渲染显示区域内容；动态计算显示区域列区间是指：根据设置要去除表头、表格合计条、表格分页条等高度；而渲染显示区域内容是指：同步刷新表头数据，同步刷新合计栏数据。

[0023] 步骤 S104，定制显示样式，根据列样式和行样式分别渲染列和行。

[0024] 其中，定制显示样式具体包括：针对列，可以设置列样式和渲染器，通过列样式单独控制该列显示样式，渲染器则可以精确控制到单元格的显示内容和样式；针对行，可以设置行高和行样式，针对行的这些控制，通过隐藏在行数据里的变量进行控制，进而控制该行的高度以及显示样式；

[0025] 步骤 S106，将表格数据存储在第一数组中。

[0026] 表格的数据存储在内存数组（第一数组）中，而显示数据单独放在 view 数组（第二数组），这样可以实现数据过滤功能以及树表（treegrid）功能；

[0027] 通过上述实施例，本发明能够保证数据快速，方便的装载到表格中，加载到页面的 dom 元素做到了最少，节省了内存；由于数据显示和实际数据是分开的，使得行数据中可以保留很多隐藏数据和控制数据，极大的方便了编程。与现有技术相比，本发明解决了大数据量加载的性能问题，克服了页面 dom 元素过多造成的不方便，以及编程模式复杂的特点。

[0028] 参考图 2，本发明通过数据 - 显示 - 控制的模式对表格进行操作，数据保存在内存数组中，通过对显示区域的计算，来控制显示的行和列，只对可视区域内的行和列进行渲染。运用本发明的方法可以快速的显示一个表格数据，提高了效率；另外，为行、列以及单元格可以定制渲染样式和内容，方便了用户的定制显示。

[0029] 其中，表格数据采用 json（JavaScript Object Notation）方式进行保存，并为每行制定唯一的键（key）值，方便对行数据的查找。

[0030] 另外，为了实现 treegrid 以及数据过滤功能，表格本身又内置了一个 view 数组（第二数组），用来保存需要显示的数据，而原有 data 提供数据来源，这样保证了显示和数据的隔离。

[0031] 表格初始化流程：

[0032] 1、对表格所在的 div 进行区域渲染，主要是表头，表体，合计条，分页条，表体锁定列这些区域的占位渲染；

[0033] 2、计算表格的整体宽高，分页条高度固定，合计条高度固定，表头高度取决于是否是多表头，以及多表头的行数，剩余高度是表体高度；

[0034] 3、定制一个隐藏焦点元素，并为其绑定事件，该元素的功能是获得页面焦点，等同于表格焦点功能，焦点的获得能使表格接收键盘事件，比如上下行选择功能；

[0035] 4、初始化表头，即列的定义；

[0036] 5、初始化数据；

[0037] 6、显示表格数据。

[0038] 其中，初始化表头的流程如下：

[0039] 1、对表头的定义进行遍历操作，计算表头的层次，并对表头对象加入隐藏属性，主要是表头所在的层次，在其父对象中的索引，跨子元素的个数；

[0040] 2、对表头的定义做索引，便于针对列的操作；

[0041] 3、处理表格锁定，对列进行拆分。

[0042] 其中，初始化数据的流程如下：

[0043] 1、初始化行数据，对于指定了唯一 key 列的行，直接使用该行对应的数据作为行索引，否则为其分配一个唯一的递加行号；

[0044] 2、初始化行的高度，若无指定，默认为内置行高；

[0045] 3、保存行的索引到索引对象；

[0046] 4、处理表格锁定时，数据的共享。

[0047] 参考图 3，显示表格数据的流程如下：

[0048] 1、计算表格显示内容数据，即 view 数组，若无过滤和 treegrid，view 即等同于表格自身的 data 数组；

[0049] 2、计算有合计行时，合计行的数据内容；

[0050] 3、计算表格表体占位区域大小，参考图 4，即所有列，所有行全部铺开时，这个区域的大小；

[0051] 4、根据上面的计算，设置占位区域大小，此时可视区域的滚动条状态就确定了；

[0052] 5、计算可视区域中，去除滚动条所占大小，剩余的空间大小；

[0053] 6、根据上面计算的空间大小，计算应该显示的行区间和列区间；

[0054] 7、计算可视区域的左侧距离和上方距离，即其在滚动区域内的绝对位置，第一次加载时，左侧距离和上方距离都是 0；

[0055] 8、渲染可视区域内容；

[0056] 9、渲染合计条区域内容。

[0057] 其中渲染可视区域内容的流程如下：

[0058] 1、使用 table 方式显示可视区域，先累加 html 字符串；

[0059] 2、遍历计算出来的行区间，加入 tr 标签；

[0060] 3、根据是否行选择标志，添加行选中样式；

[0061] 4、根据是否隔行换色属性，隔行添加颜色样式；

[0062] 5、根据行里是否有隐藏样式,添加自定义行样式 ;
[0063] 6、根据行里设置的行高,设置行的高度 ;
[0064] 7、遍历列来添加单元格的数据 ;
[0065] 8、把 html 填充入可视区域。

[0066] 其中,遍历列来添加单元格的数据的流程如下 :

[0067] 1、使用 td 来添加单元格 ;
[0068] 2、计算渲染的数据 ;
[0069] 3、添加定义的列样式 ;
[0070] 4、在 td 中定义一个 div 来装载单元格内容 ;
[0071] 5、设置单元格宽高。

[0072] 其中计算渲染的数据的流程如下

[0073] 1、先进行编码转换,替换 html 转义符 ;
[0074] 2、如果有列渲染器,则调用列渲染器进行数据处理和渲染,返回 html 字符串。

[0075] 当用户拖动横向或纵向滚动条时,根据拖动的位置进行处理可视区域的内容,滚动条滚动时的流程如下 :

[0076] 1、计算滚动条的位置,跟记忆的上次滚动条位置做比较,判断是拖动的横向滚动条还是纵向滚动条 ;

[0077] 2、如果是横向滚动条,则计算列区间,同时更改可视区域的左侧距离 ;
[0078] 3、如果是纵向滚动条,则计算行区间,同时更改可视区域的上方距离 ;
[0079] 4、判断行区间和列区间是否发生了变化,如果没变化,直接返回 ;

[0080] 5、如果发生变化,则调用上述的刷新可视区域的流程,进行可视区域内容的重绘。

[0081] 综上所述,在表格数据发生变化时,动态计算显示区域;在滚动条发生变化时,动态计算应当显示的行区间和列区间,只绘制这个区间的页面元素,并利用绝对定位的方式,减少了依赖的页面元素;并且在数据显示时,采用列渲染的方式来定制某一列的显示样式和数据内容,行渲染则采用隐藏属性的方式来定制。通过本发明,使得表格展示数据变得方便,迅速,避免了大数据量加载时引起的页面死锁问题,克服了表格数据量大时 DOM 元素过多,加载和查找缓慢的缺点。

[0082] 以上所述仅为本发明的实施例而已,并不用于限制本发明,对于本领域的技术人员来说,本发明可以有各种更改和变化。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的权利要求范围之内。

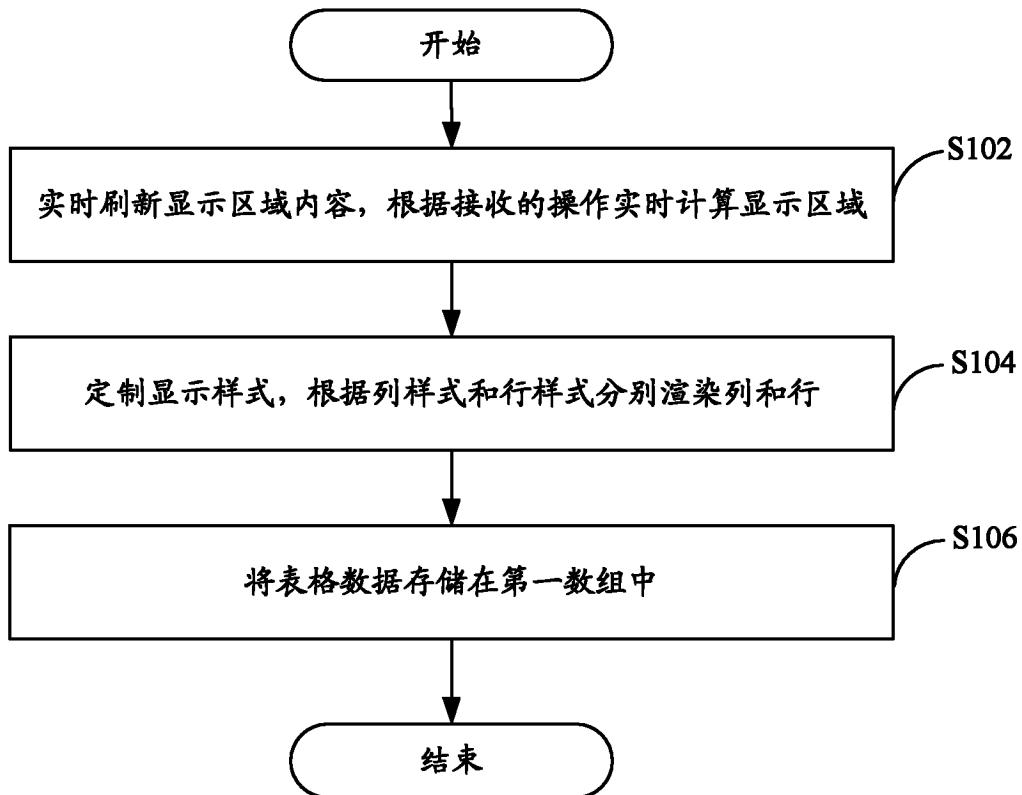


图 1

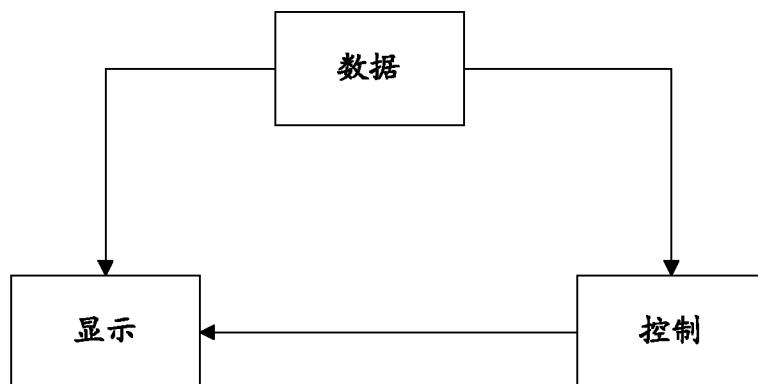


图 2

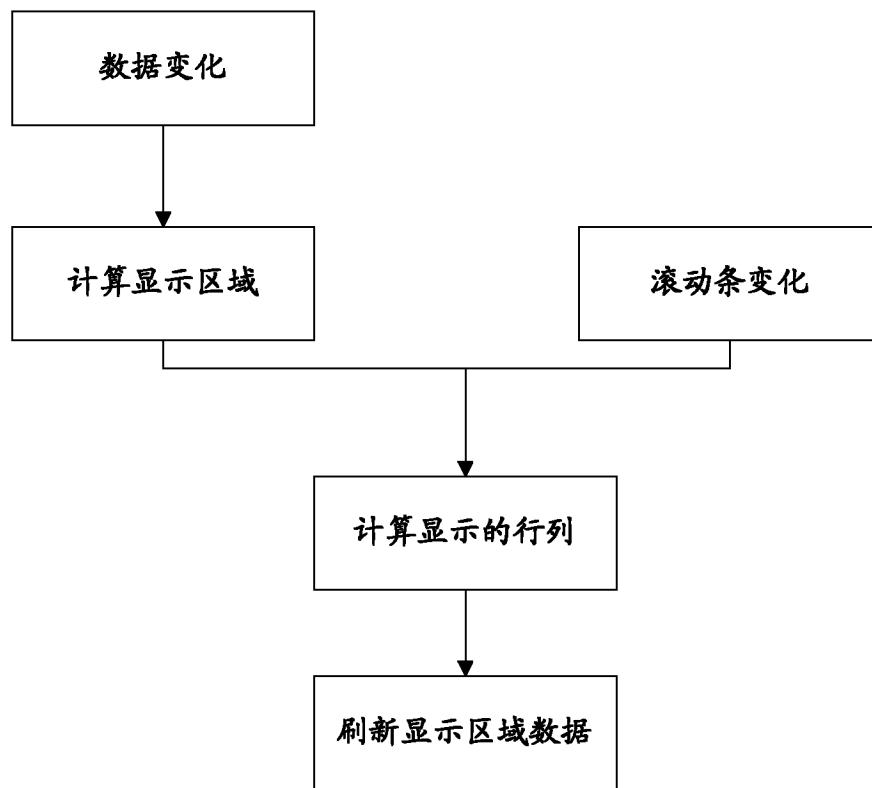


图 3

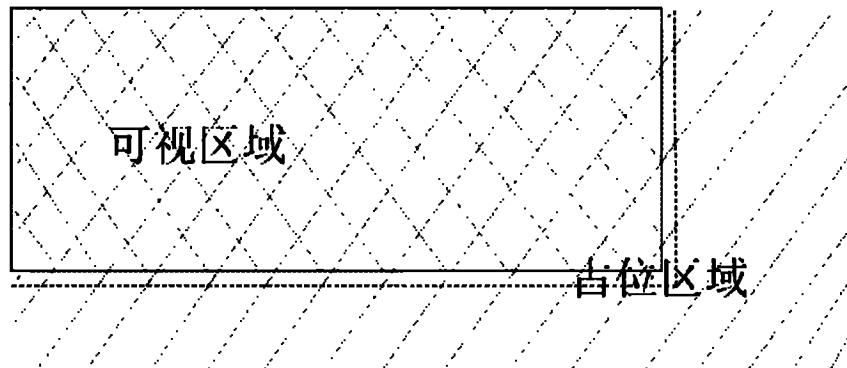


图 4