(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 106790276 A
(43)申请公布日 2017.05.31

(21)申请号 201710090347.3

(22)申请日 2017.02.20

(71)申请人 山东威尔数据股份有限公司
地址 264000 山东省烟台市莱山经济开发区明达西路12号

(72)发明人 孙晓悦

(74)专利代理机构 上海精晟知识产权代理有限公司 31253
代理人 孙福岭

(51)Int.Cl.
H04L 29/06(2006.01)
H04L 29/08(2006.01)

权利要求书1页 说明书4页 附图3页

(54)发明名称

一种基于浏览器协议的跨浏览器数据交互方法和装置

(57)摘要

本发明涉及一种基于浏览器协议的跨浏览器数据交互方法和装置，该方法包括：步骤1、所述用户终端浏览器根据浏览器协议利用JavaScript模块执行调用本地应用程序命令；步骤2、本地应用程序创建Webserver，进行本地模拟交互；步骤3、调用公共服务器的JavaScript脚本文件，动态添加JavaScript脚本文件，进行跨域数据发送；步骤4、数据发送完成后，在Webserver端控件设置回调函数，进一步载入回调指令进行数据回调；步骤5、回调完成后进行数据储存，删掉回调引用，对变量赋值，直接在回调中使用该变量实现数据返回。本发明实现了数据跨浏览器交互，提高了开发效率。

CN 106790276 A

1.一种基于浏览器协议的跨浏览器数据交互方法，其特征在于，不同用户终端安装有相同或不同的浏览器工具，所述方法包括：

步骤1、所述用户终端浏览器根据浏览器协议利用JavaScript模块执行调用本地应用程序命令；

步骤2、本地应用程序创建Webserver，通过JavaScript与Webserver进行本地模拟交互；

步骤3、调用公共服务器的JavaScript脚本文件，动态添加JavaScript脚本文件，进一步利用DOM中的JavaScript模块进行跨域数据发送；

步骤4、数据发送完成后，在Webserver端控件设置JavaScript回调函数，进一步载入JavaScript回调指令进行数据回调；

步骤5、回调完成后数据储存至JavaScript中，删掉回调引用，将JavaScript文件中的变量赋值，直接在回调中使用该变量实现数据返回。

2.根据权利要求1所述的一种基于浏览器协议的跨浏览器数据交互方法，其特征在于，所述步骤3进一步包括使用Get类请求找到头文件；使用Create语句创建一个JavaScript对象。

3.根据权利要求1所述的一种基于浏览器协议的跨浏览器数据交互方法，其特征在于，所述步骤3中单次发送数据的大小不超过2KB。

4.根据权利要求1所述的一种基于浏览器协议的跨浏览器数据交互方法，其特征在于，所述步骤4中进一步包括利用不可重入函数防止数据重入。

5.根据权利要求1所述的一种基于浏览器协议的跨浏览器数据交互方法，其特征在于，所述浏览器工具至少为Trident内核、Gecko内核、WebKit内核、Presto内核类型浏览器的一种。

6.一种基于浏览器协议的跨浏览器数据交互装置，其特征在于，包括：

应用调用模块，用于调用本地应用程序；

模拟交互模块，用于本地应用程序创建Webserver，并通过JavaScript进行本地模拟交互；

跨域数据发送模块，用于利用DOM中的JavaScript模块进行跨域数据发送；

回调模块、用于对返回数据进行回调。

7.根据权利要求6所述的一种基于浏览器协议的跨浏览器数据交互装置，其特征在于，进一步包括数据重入禁止模块，用于防止数据重入。

8.根据权利要求6所述的一种基于浏览器协议的跨浏览器数据交互装置，其特征在于，进一步包括数据取出模块，用于取出返回的数据变量。

9.根据权利要求6所述的一种基于浏览器协议的跨浏览器数据交互装置，其特征在于，所述跨域数据发送模块单次发送数据的大小不超过2KB。

10.根据权利要求6所述的一种基于浏览器协议的跨浏览器数据交互装置，其特征在于，该装置进一步至少包括Trident内核、Gecko内核、WebKit内核、Presto内核类型浏览器中的一种。

一种基于浏览器协议的跨浏览器数据交互方法和装置

技术领域

[0001] 本发明涉及一种基于浏览器协议的跨浏览器数据交互方法和装置，属于互联网技术领域。

背景技术

[0002] URL Protocol协议又称浏览器协议，是浏览器通用的一种协议，通常被用于一些外部应用的调用，比如：迅雷下载、调用阿里旺旺或QQ进行客户服务等。由于浏览器协议只能是单向一次性交互，只能一次性调用应用程序，且无返回值，调用后页面无法与外部应用进行交互，所以一直以来只是做一些简单的应用。

[0003] 控件是对数据和方法的封装，拥有自己的属性和方法，属性是控件数据的简单访问者，方法则是控件的一些简单而可见的功能。由于web应用的多样性，很多应用要与系统进行交互，例如：发卡器要跟USB通讯、安全支付需要扫描系统环境等。这部分功能浏览器上是没有权限去完成的，必须使用控件的方式。目前浏览器是百家争鸣的时期，每种浏览器使用各自的开发语言进行控件开发。由于浏览器开发及语言方案不同，开发者不得不学习各种浏览器的开发语言，而且要分别调试，大大降低了开发效率。此外，不同浏览器运行环境不同，即使内核相同也需要分开进行打包安装，甚至部分浏览器要求进入插件商店进行审核，大大降低了发布效率。

发明内容

[0004] 本发明目的在于解决浏览器使用不同开发语言进行控件开发，同一控件在不同浏览器上不能同时使用问题，提供一种基于浏览器协议的跨浏览器数据交互方法和装置。

[0005] 现阶段，浏览器工具主要包括Trident内核、Gecko内核、WebKit内核、Presto内核四种类型浏览器，不同用户终端安装有相同或不同的浏览器工具，本发明解决上述技术问题的技术方案如下：一种基于浏览器协议的跨浏览器数据交互方法，所述方法包括：

[0006] 步骤1、所述用户终端浏览器根据浏览器协议利用JavaScript模块执行调用本地应用程序命令；

[0007] 步骤2、本地应用程序创建Webserver，通过JavaScript与Webserver进行本地模拟交互；

[0008] 步骤3、调用公共服务器的JavaScript脚本文件，动态添加JavaScript脚本文件，进一步利用DOM中的JavaScript模块进行跨域数据发送，单次发送数据的大小不超过2KB；进一步使用Get类请求找到头文件；使用Create语句创建一个JavaScript对象；

[0009] 步骤4、数据发送完成后，在Webserver端控件设置JavaScript回调函数，进一步载入JavaScript回调指令进行数据回调；进一步包括利用不可重入函数防止数据重入；

[0010] 步骤5、回调完成后数据储存至JavaScript中，删掉回调引用，将JavaScript文件中的变量赋值，直接在回调中使用该变量实现数据返回。

[0011] 一种基于浏览器协议的跨浏览器数据交互装置，包括：

[0012] 应用调用模块，用于调用本地应用程序；

[0013] 模拟交互模块，用于本地应用程序创建Webserver，并通过JavaScript进行本地模拟交互；

[0014] 跨域数据发送模块，用于利用DOM中的JavaScript模块进行跨域数据发送；

[0015] 回调模块、用于对返回数据进行回调。

[0016] 进一步，包括数据重入禁止模块，用于防止数据重入。

[0017] 进一步，包括数据取出模块，用于取出返回的数据变量。

[0018] 进一步，跨域数据发送模块单次发送数据的大小不超过2KB。

[0019] 进一步，该装置至少包括Trident内核、Gecko内核、WebKit内核、Presto内核类型浏览器中的一种。

[0020] 本发明的有益效果是：无需针对不同浏览器开发不同控件，能够解决控件跨浏览器使用问题，实现数据跨浏览器交互，提高了开发效率。

附图说明

[0021] 图1为基于浏览器协议的跨浏览器数据交互方法示意图；

[0022] 图2为跨域数据传输部分代码举例；

[0023] 图3为返回数据回调部分代码举例；

[0024] 图4为基于浏览器协议的跨浏览器数据交互装置结构示意图；

[0025] 图5为常见浏览器HTTP Get请求URL的最大长度列表。

具体实施方式

[0026] 以下结合附图对本发明的原理和特征进行描述，所举实例只用于解释本发明，并非用于限定本发明的范围。

[0027] 实施例1：

[0028] 图1为本发明第一实施例中基于浏览器协议的跨浏览器数据交互方法示意图，一种可以实现跨浏览器使用的控件解决方案。

[0029] 具体而言，一种基于浏览器协议的跨浏览器数据交互方法，包括：

[0030] 步骤1、所述用户终端浏览器根据浏览器协议利用JavaScript模块执行调用本地应用程序命令；

[0031] 步骤2、本地应用程序创建Webserver，通过JavaScript与Webserver进行本地模拟交互；

[0032] 步骤3、调用公共服务器的JavaScript脚本文件，动态添加JavaScript脚本文件，进一步利用DOM中的JavaScript模块进行跨域数据发送，单次发送数据的大小不超过2KB；进一步使用Get类请求找到头文件；使用Create语句创建一个JavaScript对象；

[0033] 步骤4、数据发送完成后，在Webserver端控件设置JavaScript回调函数，进一步载入JavaScript回调指令进行数据回调；进一步包括利用不可重入函数防止数据重入；

[0034] 步骤5、回调完成后数据储存至JavaScript中，删掉回调引用，将JavaScript文件中的变量赋值，直接在回调中使用该变量实现数据返回。

[0035] 本发明采用上述方案，可以实现JavaScript与Webserver进行本地模拟交互，通过

DOM中的JavaScript模块，能够实现数据的本地和跨域传输，进一步借助回调函数进行数据回调，不可重入函数防止数据重入。

[0036] 实施例2：

[0037] 如图2所示，其中跨域数据传输涉及的代码字段举例如下：

[0038] “//找到文档头

[0039] var head＝document.getElementsByTagName('head')[0]；

[0040] //生成一个JS对象

[0041] var jscript＝document.createElement('script')；

[0042] jscript.type＝'text/javascript'；

[0043] //发送数据

[0044] jscript.src＝"http://127.0.0.1:"+jlink_port+"/"+"发送的数据"；

[0045] //插入元素(兼容IE6必须insertBefore)

[0046] head.insertBefore(jscript,head.firstChild)；”

[0047] 实施例3：

[0048] 如图3所示，其中数据回调与防止重入代码举例如下：

[0049]

```
“//载入成功时执行
 jscript.onload = jscript.onreadystatechange = function()
 {
    if((!this.readyState||this.readyState ===
"loaded"||this.readyState === "complete"))
    {
       //防止重入
       jscript.onload = jscript.onreadystatechange = null;
    }
 }”
```

[0050] 实施例:4：

[0051] 如图4所示，与上述方法对应，本发明还提供一种基于浏览器协议的跨浏览器数据交互装置1，包括：

[0052] 应用调用模块101，用于调用本地应用程序；

[0053] 模拟交互模块102，用于本地应用程序创建Webserver，并通过JavaScript进行本地模拟交互；

[0054] 跨域数据发送模块103，用于利用DOM中的JavaScript模块进行跨域数据发送；

[0055] 回调模块105、用于对返回数据进行回调。

[0056] 数据重入禁止模块104，用于防止数据重入。

[0057] 数据取出模块106，用于取出返回的数据变量。

[0058] 该装置进一步至少包括Trident内核、Gecko内核、WebKit内核、Presto内核类型浏览器中的一种。

[0059] 实施例5：

[0060] 如图5所示，常见浏览器HTTP Get请求URL的最大长度，其中：

[0061] IE6.0:URL最大长度2083个字符，超过最大长度后无法提交。

[0062] IE7.0:URL最大长度2083个字符，超过最大长度后仍然能提交，但是只能传过去2083个字符。

[0063] Firefox 3.0.3:URL最大长度7764个字符，超过最大长度后无法提交。

[0064] Opera 9.52:URL最大长度7648个字符，超过最大长度后无法提交。

[0065] Google Chrome 2.0.168:URL最大长度7713个字符，超过最大长度后无法提交。

[0066] 由于使用的Get类请求，提交的数据受URL长度限制，当递交数据控制在2K以内，具体而言数据需控制在市面所用浏览器URL最大长度值最小的那一个范围内，从而满足控件数据跨浏览器交互的数据量。

[0067] 以上所述仅为本发明的较佳实施例，并不用以限制本发明，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保护范围之内。


用户终端浏览器根据浏览器协议利用JavaScript模块执行调用本地应用程序命令
本地应用程序创建Webserver，通过JavaScript与Webserver进行本地模拟交互
调用公共服务器的JavaScript脚本文件，动态添加JavaScript脚本文件，进一步利用DOM中的JavaScript模块进行跨域数据发送
数据发送完成后，在Webserver端控件设置JavaScript回调函数，进一步载入JavaScript回调指令进行数据回调
回调完成后数据储存至JavaScript中，删掉回调引用，将JavaScript文件中的变量赋值，直接在回调中使用该变量实现数据返回

图1

```
//找到文档头
var head = document.getElementsByTagName('head')[0];
//生成一个JS对象
var jscript = document.createElement('script');
jscript.type = 'text/javascript';
//发送数据
jscript.src = "http://127.0.0.1:" + jlink_port + "/" + "发送的数据";
//插入元素(兼容IE6必须insertBefore)
head.insertBefore(jscript,head.firstChild);
```

图2

```
//载入成功时执行
jscript.onload = jscript.onreadystatechange = function()
{
    if((!this.readyState||this.readyState === "loaded"||this.readyState ===
"complete"))
    {
        //防止重入
        jscript.onload = jscript.onreadystatechange = null;
    }
}
```

图3

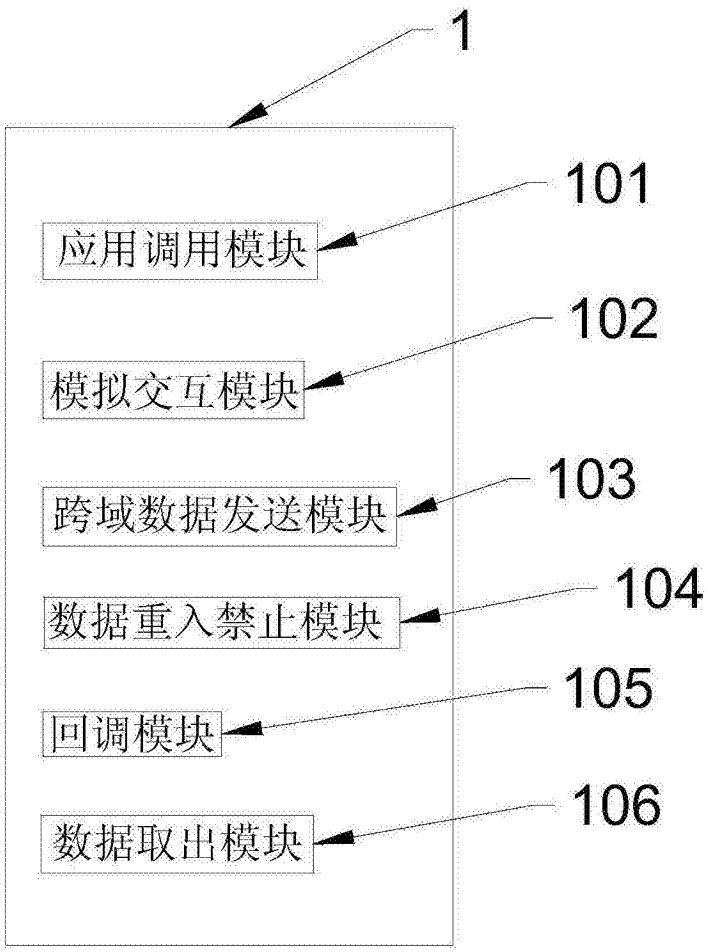

图4

浏览器名称	URL 最大长度
IE6.0	2083 字符
IE7.0	2083 字符
Firefox 3.0.3	7764 字符
Opera 9.52	7648 字符
Google Chrome 2.0.168	7713 字符

图5