

数据回填方法及装置

技术领域

本申请涉及互联网技术领域，尤其涉及数据回填方法及装置。

背景技术

5 用户在网页上执行账号注册、账号内容填写等操作时，需要在网页上的输入框中填写对应的数据内容，比如账号名称、账号密码、联系方式等信息。出于信息分类的考虑，各类信息的填写被分配至不同的网页中，比如用户在第一页面内填写了“姓名”和“年龄”后，通过点击第一页面上的“下一步”跳转至第二页面，同时将第一页面内填写的内容提交至服务器；而当用户在
10 第二页面进行操作时，希望对第一页面的“姓名”进行修改，则可以点击第二页面上的“上一步”返回第一页面。

当用户返回第一页面时，浏览器需要将用户先前填写的如“姓名”和“年龄”等内容，回填至第一页面中对应的输入框内，以便于用户的查看和修改。

然而在实际应用中，当用户填写的内容出现中文字符时，回填时往往会
15 出现“乱码”的情况，导致用户无法正常查看先前填写的内容，影响用户体验。

发明内容

有鉴于此，本申请提供一种数据回填方法及装置，可以顺利实现对页面数据的回填操作，且不会出现回填内容乱码的现象。

20 为实现上述目的，本申请提供技术方案如下：

根据本申请的第一方面，提出了一种数据回填方法，包括：

将终端的第一页面中的用户输入数据和所述第一页面中输入框的标识相关联地发送至服务器，以由所述服务器进行存储，且存储内容映射于所述

终端的 sessionID、所述第一页面的标识；

当由第二页面返回所述第一页面时，向所述服务器发送数据获取请求，所述数据获取请求中包含所述终端的 sessionID 和所述第一页面的标识；

接收所述服务器根据所述数据获取请求返回的所述第一页面中的用户
5 输入数据和相关联的所述第一页面中输入框的标识，并将所述第一页面中的用户输入数据回填至所述第一页面中对应的输入框内。

根据本申请的第二方面，提出了一种数据回填方法，包括：

关联存储终端发送的第一页面中的用户输入数据和所述第一页面中输入框的标识，且存储内容映射于所述终端的 sessionID、所述第一页面的标识；

10 接收所述终端在由第二页面返回所述第一页面时发送的数据获取请求；

根据所述数据获取请求中包含的所述终端的 sessionID 和所述第一页面的标识，向所述终端返回所述第一页面中的用户输入数据和相关联的所述第一页面中输入框的标识，以由所述终端将所述第一页面中的用户输入数据回填至所述第一页面中对应的输入框内。

15 根据本申请的第三方面，提出了一种数据回填装置，包括：

发送单元，将终端的第一页面中的用户输入数据和所述第一页面中输入框的标识相关联地发送至服务器，以由所述服务器进行存储，且存储内容映射于所述终端的 sessionID、所述第一页面的标识；

请求单元，当由第二页面返回所述第一页面时，向所述服务器发送数据
20 获取请求，所述数据获取请求中包含所述终端的 sessionID 和所述第一页面的标识；

回填单元，接收所述服务器根据所述数据获取请求返回的所述第一页面中的用户输入数据和相关联的所述第一页面中输入框的标识，并将所述第一页面中的用户输入数据回填至所述第一页面中对应的输入框内。

25 根据本申请的第四方面，提出了一种数据回填装置，包括：

数据存储单元，关联存储终端发送的第一页面中的用户输入数据和所述第一页面中输入框的标识，且存储内容映射于所述终端的 sessionID、所述第

一页面的标识;

请求接收单元,接收所述终端在由第二页面返回所述第一页面时发送的数据获取请求;

数据返回单元,根据所述数据获取请求中包含的所述终端的 sessionID 和
5 所述第一页面的标识,向所述终端返回所述第一页面中的用户输入数据和相
关联的所述第一页面中输入框的标识,以由所述终端将所述第一页面中的用
户输入数据回填至所述第一页面中对应的输入框内。

由以上技术方案可见,本申请通过在服务器上对页面数据的关联存储,
使得页面数据由服务器进行统一的编解码操作,不会由于终端侧的浏览器类
10 型差异或者浏览器与服务器之间的不匹配,而导致出现字符乱码现象,确保
页面数据的顺利回填。

附图说明

图 1 示出了相关技术中的一种数据回填方法的流程图;

图 2 示出了本申请一示例性实施例的一种基于终端侧的数据回填方法的
15 流程图;

图 3 示出了本申请一示例性实施例的一种基于服务器侧的数据回填方法
的流程图;

图 4 示出了本申请一示例性实施例的一种数据回填方法的流程图;

图 5 示出了本申请一示例性实施例的一种终端的结构示意图;

20 图 6 示出了本申请一示例性实施例的一种基于终端侧的数据回填装置的
框图;

图 7 示出了本申请一示例性实施例的一种服务器的结构示意图;

图 8 示出了本申请一示例性实施例的一种基于服务器侧的数据回填装置
的框图。

具体实施方式

在本申请的技术方案中的“回填”应当理解为：终端根据用户操作返回某个页面时，将用户先前在该页面输入的数据，重新一一对应地填入该页面的输入框中，以便于用户的查看和编辑操作；由于数据并非首次填入，而是
5 从其他页面返回该页面时的再次填入，因而描述为“回填”操作。

请参考图 1，图 1 示出了相关技术中的一种数据回填方法的流程图，包括以下步骤：

步骤 102，获取终端浏览器的页面 1 中的用户输入数据。假定页面 1 为用户执行账号注册的网页，比如用户可以在页面 1 中的“姓名”和“年龄”
10 输入框中填入自己的姓名“张三”和年龄“10”。

步骤 104，用户发起向页面 2 的跳转，比如用户通过点击页面 1 中的“下一步”操作按键，触发该跳转操作。

步骤 106，终端浏览器通过 post 请求将页面 1 中的用户输入数据提交至服务器。

15 步骤 108，服务器接收页面 1 中的用户输入数据，并记录在服务器中。

步骤 110，服务器可以对记录的用户输入数据进行处理，比如数据校验等。

步骤 112，服务器将页面 1 中的用户输入数据回传至终端浏览器，由终端对其进行记录。

20 步骤 114，用户在页面 2 进行操作的时候，如果执行了向页面 1 的跳转操作，比如用户点击了页面 2 中的“上一步”按键，则触发了步骤 116 至步骤 122 中对页面 1 的数据回填操作。

步骤 116，终端浏览器向服务器发起 get 请求，该 get 请求中包含终端在步骤 112 中记录的页面 1 中的用户输入数据。

25 基于 get 请求的结构，其包括目标页面的 URL(Uniform Resource Locator，统一资源定位器)和需要传输的数据，且该数据直接跟在 URL 后面，比如：

“ http://www.123456789.com?name= 张 三 &age=10 ” ， 其 中
“http://www.123456789.com”为 URL 信息，“name=张三&age=10”为页面
1 中的用户输入数据，具体对应于输入框“name（姓名）”的用户输入数据
为“张三”、对应于输入框“age（年龄）”的用户输入数据为“10”。

5 实际上，用户可以直接从终端浏览器的地址栏中查看到 get 请求的内容，
比如上述的“http://www.123456789.com?name=张三&age=10”；然而，对于
服务器而言，并不能够直接接收到“name=张三”和“age=10”的数据，由
于“张三”为汉字，需要由终端浏览器采用自身编码方式对其执行编码操作，
比如当终端浏览器采用 UTF-8（万国码）方式编码时，将“张三”编译为
10 “&#x5F20&#x4E09”，则服务器接收到的 get 请求可以为
“http://www.123456789.com?name=&#x5F20&#x4E09&age=10”。

步骤 118，服务器在接收到来自终端浏览器的 get 请求后，按照自身的编
码方式对数据“name=&#x5F20&#x4E09&age=10”进行编解码操作。

当然，除了对数据进行编解码操作之外，服务器也会在业务层面上对接
15 收到的数据进行完整性校验、用户身份校验等处理。

步骤 120，服务器通过 get 请求的方式，将编解码后的数据返回终端浏览
器。

步骤 122，终端浏览器将接收到的数据回填至页面 1 中相应的输入框中。

然而，由于终端浏览器在步骤 112 中接收并记录了服务器传回的数据之
20 后，仍然需要通过步骤 116 至步骤 120 将记录的数据在服务器上“来回”传
输一次，使得数据在该传输过程中分别需要由终端浏览器和服务器进行编解
码操作，因而如果终端浏览器和服务器中的任一方无法识别对方采用的编解
码方式，就将导致其编码后的数据为乱码（与原始含义没有关联的字符串），
并最终作为回填数据而呈现于页面 1 中，使用户无法实现对数据的顺利查看，
25 极为影响用户体验。

可见，由于相关技术中需要将回填数据在终端浏览器和服务器上反复传
输，导致最终用于回填的数据需要经由终端浏览器和服务器分别进行编解码

操作，因而回填数据中包含中文时，极易因终端和服务器的编解码方式不匹配的原因，导致页面回填数据出现乱码现象。

需要说明的是：

在相关技术中，比如对于步骤 108 而言，服务器虽然会为终端浏览器（对应的 sessionID）分配相应的缓存空间，但该缓存空间内的存储数据的生命周期以“request（请求）”为单位，即服务器每当接收到一个请求时，就将该请求中包含的数据记录在该缓存空间中，并且删除先前的存储数据。因此，可以认为服务器仅对接收到的数据进行了“记录”，而非真正意义上的存储，因为终端浏览器无法对已发送的数据进行任意的查找和调取操作，而这也导致了终端浏览器和服务器之间必然需要执行图 1 所示的反复的数据传输。

但是，正由于相关技术的数据回填过程中，仅涉及到数据在终端浏览器和服务器之间传递，且服务器仅需对数据进行“记录”，使其整个方案虽然看似流程复杂，实际上属于“轻量级”的处理过程，因而广泛应用于终端浏览器的数据回填处理。

而为了避免相关技术中的数据回填乱码问题，本申请对上述技术方案进行了改进，通过避免数据的反复传输，能够顺利实现页面数据的回填操作。为对本申请进行进一步说明，提供下列实施例：

1、终端侧

请参考图 2，图 2 示出了本申请一示例性实施例的一种基于终端侧的数据回填方法的流程图，该方法应用于终端，可以包括下述步骤：

步骤 202，将终端的第一页面中的用户输入数据和所述第一页面中输入框的标识相关联地发送至服务器，以由所述服务器进行存储，且存储内容映射于所述终端的 sessionID、所述第一页面的标识。

在本实施例中，可以按照预设方式对第一页面中的用户输入数据和相关的输入框标识进行打包，并将打包数据发送至服务器。其中，打包数据可以采用 JSON（JavaScript Object Notation）格式，以便尽可能地减小传输数据量。

在本实施例中，当执行上述打包操作时，可以在用户每完成一个输入框的输入操作后，就将该输入框的标识和填入的数据进行打包，从而实现了该输入框标识和填入数据的关联；或者，也可以在用户触发向其他页面的跳转操作时，将该第一页面中的所有用户输入数据和输入框的标识相关联地执行打包操作。

在本实施例中，sessionID 为服务器为终端分配的唯一标识，具体由服务器在终端首次访问时进行分配，并由终端存储在本地 cookie 中，则终端每次访问该服务器时，均通过携带 cookie 中存储的该 sessionID 来表明自己的身份；而第一页面的标识可以为第一页面的 URL 或 URI（Uniform Resource Identifier，通用资源标识符）信息。因此，服务器可以通过 sessionID 和 URL/URI 信息的组合，准确分辨和识别出每个页面对应的存储内容。

步骤 204，当由第二页面返回所述第一页面时，向所述服务器发送数据获取请求，所述数据获取请求中包含所述终端的 sessionID 和所述第一页面的标识。

在本实施例中，第二页面与第一页面之间可以为“相邻”关系，即用户点击第一页面上的“下一步”按键时，将直接跳转至第二页面，以及用户点击第二页面上的“上一步”按键时，将直接跳转至第一页面；或者，第二页面与第一页面之间也可以不相邻，比如第一页面为“页面 1”，而第二页面为“页面 3”，则通过在第二页面上直接选择跳转目标为“页面 1”时，即可直接跳转至第一页面。

步骤 206，接收所述服务器根据所述数据获取请求返回的所述第一页面中的用户输入数据和相关联的所述第一页面中输入框的标识，并将所述第一页面中的用户输入数据回填至所述第一页面中对应的输入框内。

在本实施例中，终端执行回填操作的用户输入数据直接来自服务器，而无需终端浏览器从第一页面跳转至“下一步”时，由服务器将用户输入数据回传至终端浏览器进行记录（对应于图 1 所示的步骤 112），也无需使得用户输入数据在终端和服务器之间进行反复传输（对应于图 1 所示的步骤 116

至步骤 120)。实际上,本申请中的用户输入数据由服务器在步骤 202 执行存储后,在步骤 206 中直接从服务器传输至终端,而终端仅在需要执行回填操作时,向服务器发起数据获取请求(即步骤 204)即可,使得用户输入数据只需要在服务器上实现统一的编解码操作,而不需要考虑服务器与终端浏览器之间的匹配问题,因而不会由于服务器和终端浏览器之间不匹配而导致出现回填数据为乱码的情况。

在本实施例中,服务器将第一页面中的用户输入数据和相关联的输入框标识返回终端时可以采用流的形式,从而避免数据过多时,get 请求的 URL 过长而被截取,导致参数错误而无法正确回填数据。

10 2、服务器侧

请参考图 3,图 3 示出了本申请一示例性实施例的一种基于服务器侧的数据回填方法的流程图,该方法应用于服务器,可以包括以下步骤:

步骤 302,关联存储终端发送的第一页面中的用户输入数据和所述第一页面中输入框的标识,且存储内容映射于所述终端的 sessionID、所述第一页面的标识。

在本实施例中,服务器可以采用 Memcache 缓存框架对用户输入数据和输入框标识进行关联存储,以实现数据的高速存取。

步骤 304,接收所述终端在由第二页面返回所述第一页面时发送的数据获取请求。

在本实施例中,终端浏览器上的页面跳转后,服务器无需将用户输入数据等回传至终端(对应于图 1 所示的步骤 112),而只需要对终端发送的数据获取请求进行响应。

在本实施例中,数据获取请求可以采用 get 请求的形式,但由于服务器不会将用户输入数据和输入框标识等信息提前返回终端(对应于图 1 所示的步骤 112),因而 get 请求中无需包含用户输入数据和输入框标识等信息,而仅包含第一页面的 URL,并携带用于服务器执行终端识别的 sessionID。

步骤 306,根据所述数据获取请求中包含的所述终端的 sessionID 和所述

第一页面的标识，向所述终端返回所述第一页面中的用户输入数据和相关联的所述第一页面中输入框的标识，以由所述终端将所述第一页面中的用户输入数据回填至所述第一页面中对应的输入框内。

在本实施例中，服务器向终端发送的用户输入数据是步骤 302 中的存储内容，而非来自终端的 get 请求，因而该数据仅需服务器在存入和取出时执行字符编解码操作，无需终端浏览器对其进行编解码，因而不会由于终端浏览器和服务器之间不匹配，而导致出现字符乱码的情况。

由上述实施例可知，本申请通过将用户输入数据和输入框标识关联存储于服务器，且服务器对接收到的数据执行了真正意义上的“存储”，而非相关技术中的“记录”，即服务器上的存储数据的生命周期以“session(会话)”为单位（相当于终端浏览器当前被分配的 sessionID 的有效期），使得终端在实现页面数据的回填操作时，无需将用户输入数据和输入框标识在终端和服务器之间反复传输，而直接由服务器将其存储内容发送至终端即可，即由服务器对用户输入数据等存储内容进行统一的编解码操作，无需经由终端浏览器的编解码操作，也不会由于服务器和终端浏览器之间不匹配而导致出现数据乱码的情况，从而顺利实现用户输入数据的回填操作。

下面针对终端和服务器之间的信息交互过程，结合图 4 进行详细说明。其中，图 4 示出了本申请一示例性实施例的一种数据回填方法的流程图，如图 4 所示，该方法可以包括以下步骤：

步骤 402，获取终端浏览器的页面 1 中的用户输入数据。假定页面 1 为用户执行账号注册的网页，比如用户可以在页面 1 中的“姓名”和“年龄”输入框中填入自己的姓名“张三”和年龄“10”。

步骤 404，用户发起向页面 2 的跳转，比如用户通过点击页面 1 中的“下一步”操作按键，触发该跳转操作。

步骤 406，终端浏览器通过 post 请求将页面 1 中的用户输入数据提交至服务器。

步骤 408，服务器接收页面 1 中的用户输入数据，并将用户输入数据与

输入框标识进行关联存储。

在本实施例中，服务器可以将终端的 sessionID 和页面 1 的 URL 或 URI 作为 key (关键词)，将用户输入数据和相关联的输入框标识存储至缓存中，比如 Memcache 缓存；或者，服务器也可以将接收到的数据存储至其他形式的存储空间，比如数据库中，本申请并不对此进行限制。

步骤 410，服务器可以对记录的用户输入数据进行处理，比如数据校验等。

步骤 412，用户在页面 2 进行操作的时候，如果执行了向页面 1 的跳转操作，比如用户点击了页面 2 中的“上一步”按键，则触发了步骤 414 至步骤 420 中对页面 1 的数据回填操作。

步骤 414，终端浏览器向服务器发起 get 请求。

在本实施例中，由于服务器并未提前将页面 1 的用户输入数据传回终端，且本申请也不需要将用户输入数据在终端浏览器和服务器之间进行反复传输，因而 get 请求中仅需要包含页面 1 的 URL，而无需将用户输入数据和输入框标识等添加在该 URL 后方而发送至服务器，区别于图 1 所示的步骤 116。因此，用户在终端浏览器中的地址栏中，看到的信息区别于上述的“http://www.123456789.com?name= 张三 &age=10”，而仅为“http://www.123456789.com”。

步骤 416，根据 get 请求中包含的页面 1 的 URL 或者 URI，以及终端发送该 get 请求时一并传输的 sessionID，从缓存中查找对应于“sessionID+URL”或“sessionID+URI”组合的存储内容，即对应于页面 1 的用户输入数据和对应的输入框标识。

步骤 418，将页面 1 的用户输入数据等查找到的存储内容传回终端。

步骤 420，终端浏览器根据服务器传回的数据，将其中的用户输入数据回填至页面 1 中对应的输入框内。

在上述实施例中，通过服务器对终端提交的数据进行关联存储，以及服务器根据终端发起的 get 请求，将对应的数据传回终端来实现数据回填，使

得终端接收到的用户输入数据仅在步骤 406（服务器对接收到的数据进行编码操作，以供存储于缓存）和步骤 418（服务器对查找到的数据进行解码操作后，以供回传至终端）中，由服务器一方执行了统一的字符编解码操作，而无需终端浏览器进行编解码操作，则无需终端浏览器和服务器的编解码操作匹配，且不会导致回填数据的乱码情况。

步骤 422，当用户完成对页面 1 的查看或编辑操作后，若触发向其他页面的跳转操作，比如跳转至页面 3。

步骤 424，类似于步骤 406，由终端浏览器通过 post 请求将最新用户输入数据和输入框标识相关联地发送至服务器。

10 步骤 426，服务器根据终端的 sessionID 和页面 1 的标识，查找已存储的用户输入数据和相关联的输入框标识，并与当前接收到的数据进行匹配，从而确定同一输入框标识对应的已存储用户输入数据和最新用户输入数据是否存在差异。

步骤 428，针对存在数据差异的情况，对服务器的存储内容进行更新。

15 在上述实施例中，由服务器对页面 1 的用户输入数据进行差异判断和数据更新。除了由服务器实现数据差异判断，也可以由终端对用户输入数据的变更情况进行检测，比如将最新用户输入数据与步骤 418 中接收到的用户输入数据进行比较，若存在数据差异，则将相应的最新用户输入数据和输入框标识相关联地发送至服务器，以由服务器进行数据更新处理。

20 由上述实施例可知，本申请通过在服务器上执行真正意义上的“存储”操作，即服务器将接收到的所有数据均与 key 相关联地存储，使得终端浏览器可以随时根据 key 来调取对应的存储数据，从而减少了终端浏览器与服务器之间的数据传输次数，以解决数据传输过程导致的乱码问题。

图 5 示出了根据本申请的一示例性实施例的终端的示意结构图。请参考图 5，在硬件层面，该终端包括处理器、内部总线、网络接口、内存以及非易失性存储器，当然还可能包括其他业务所需要的硬件。处理器从非易失性存储器中读取对应的计算机程序到内存中然后运行，在逻辑层面上形成数据

回填装置。当然，除了软件实现方式之外，本申请并不排除其他实现方式，比如逻辑器件抑或软硬件结合的方式等等，也就是说以下处理流程的执行主体并不限定于各个逻辑单元，也可以是硬件或逻辑器件。

请参考图 6，在软件实施方式中，该数据回填装置可以包括发送单元、

5 请求单元和回填单元。其中：

发送单元，将终端的第一页面中的用户输入数据和所述第一页面中输入框的标识相关联地发送至服务器，以由所述服务器进行存储，且存储内容映射于所述终端的 sessionID、所述第一页面的标识；

请求单元，当由第二页面返回所述第一页面时，向所述服务器发送数据
10 获取请求，所述数据获取请求中包含所述终端的 sessionID 和所述第一页面的标识；

回填单元，接收所述服务器根据所述数据获取请求返回的所述第一页面中的用户输入数据和相关联的所述第一页面中输入框的标识，并将所述第一页面中的用户输入数据回填至所述第一页面中对应的输入框内。

15 可选的，所述发送单元具体用于：

将所述第一页面中的用户输入数据和所述第一页面中输入框的标识相关联地生成为预设格式的打包数据；

将所述打包数据发送至所述服务器。

可选的，所述预设格式为 JSON 格式。

20 可选的，还包括：

第一更新单元，当由所述第一页面切换至其他页面时，将所述第一页面中的最新用户输入数据和所述第一页面中输入框的标识相关联地发送至所述服务器，以由所述服务器对映射于所述终端的 sessionID 和所述第一页面的标识的存储内容进行更新。

25 可选的，还包括：

比较单元，当由所述第一页面切换至其他页面时，将所述第一页面中的最新用户输入数据与所述服务器返回的用户输入数据进行比较；

第二更新单元，若比较结果为不完全相同，则将存在差异的用户输入数据和对应的输入框的标识相关联地发送至所述服务器，以由所述服务器更新所述存储内容。

图 7 示出了根据本申请的一示例性实施例的服务器的示意结构图。请参考图 7，在硬件层面，该服务器包括处理器、内部总线、网络接口、内存以及非易失性存储器，当然还可能包括其他业务所需要的硬件。处理器从非易失性存储器中读取对应的计算机程序到内存中然后运行，在逻辑层面上形成数据回填装置。当然，除了软件实现方式之外，本申请并不排除其他实现方式，比如逻辑器件抑或软硬件结合的方式等等，也就是说以下处理流程的执行主体并不限于各个逻辑单元，也可以是硬件或逻辑器件。

请参考图 8，在软件实施方式中，该数据回填装置可以包括数据存储单元、请求接收单元和数据返回单元。其中：

数据存储单元，关联存储终端发送的第一页面中的用户输入数据和所述第一页面中输入框的标识，且存储内容映射于所述终端的 sessionID、所述第一页面的标识；

请求接收单元，接收所述终端在由第二页面返回所述第一页面时发送的数据获取请求；

数据返回单元，根据所述数据获取请求中包含的所述终端的 sessionID 和所述第一页面的标识，向所述终端返回所述第一页面中的用户输入数据和相关的所述第一页面中输入框的标识，以由所述终端将所述第一页面中的用户输入数据回填至所述第一页面中对应的输入框内。

可选的，还包括：

数据接收单元，接收所述终端在由所述第一页面切换至其他页面时发送的所述第一页面中的最新用户输入数据和相关的所述第一页面中输入框的标识；

数据比较单元，将所述第一页面中的最新用户输入数据与已存储的所述第一页面中的用户输入数据进行比较；

内容更新单元，若比较结果为不完全相同，则对映射于所述终端的 sessionID 和所述第一页面的标识的存储内容进行更新。

因此，本申请通过在服务器上对页面数据的关联存储，使得页面数据由服务器进行统一的编解码操作，不会由于终端侧的浏览器类型差异或者浏览器与服务器之间的不匹配，而导致出现字符乱码现象，确保页面数据的顺利回填。

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。

内存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)。内存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。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。

还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

以上所述仅为本申请的较佳实施例而已，并不用以限制本申请，凡在本申请的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本申请保护的范围之内。

权 利 要 求 书

1. 一种数据回填方法，其特征在于，包括：

将终端的第一页面中的用户输入数据和所述第一页面中输入框的标识相关联地发送至服务器，以由所述服务器进行存储，且存储内容映射于所述
5 终端的 sessionID、所述第一页面的标识；

当由第二页面返回所述第一页面时，向所述服务器发送数据获取请求，所述数据获取请求中包含所述终端的 sessionID 和所述第一页面的标识；

接收所述服务器根据所述数据获取请求返回的所述第一页面中的用户输入数据和相关联的所述第一页面中输入框的标识，并将所述第一页面中的
10 用户输入数据回填至所述第一页面中对应的输入框内。

2. 根据权利要求 1 所述的方法，其特征在于，所述将终端的第一页面中的用户输入数据和所述第一页面中输入框的标识相关联地发送至服务器，包括：

将所述第一页面中的用户输入数据和所述第一页面中输入框的标识相关联地生成为预设格式的打包数据；
15

将所述打包数据发送至所述服务器。

3. 根据权利要求 2 所述的方法，其特征在于，所述预设格式为 JSON 格式。

4. 根据权利要求 1 所述的方法，其特征在于，还包括：

20 当由所述第一页面切换至其他页面时，将所述第一页面中的最新用户输入数据和所述第一页面中输入框的标识相关联地发送至所述服务器，以由所述服务器对映射于所述终端的 sessionID 和所述第一页面的标识的存储内容进行更新。

5. 根据权利要求 1 所述的方法，其特征在于，还包括：

25 当由所述第一页面切换至其他页面时，将所述第一页面中的最新用户输入数据与所述服务器返回的用户输入数据进行比较；

若比较结果为不完全相同，则将存在差异的用户输入数据和对应的输入框的标识相关联地发送至所述服务器，以由所述服务器更新所述存储内容。

6. 一种数据回填方法，其特征在于，包括：

关联存储终端发送的第一页面中的用户输入数据和所述第一页面中输入框的标识，且存储内容映射于所述终端的 sessionID、所述第一页面的标识；

接收所述终端在由第二页面返回所述第一页面时发送的数据获取请求；

根据所述数据获取请求中包含的所述终端的 sessionID 和所述第一页面的标识，向所述终端返回所述第一页面中的用户输入数据和相关联的所述第一页面中输入框的标识，以由所述终端将所述第一页面中的用户输入数据回填至所述第一页面中对应的输入框内。

7. 根据权利要求 6 所述的方法，其特征在于，还包括：

接收所述终端在由所述第一页面切换至其他页面时发送的所述第一页面中的最新用户输入数据和相关联的所述第一页面中输入框的标识；

将所述第一页面中的最新用户输入数据与已存储的所述第一页面中的用户输入数据进行比较；

若比较结果为不完全相同，则对映射于所述终端的 sessionID 和所述第一页面的标识的存储内容进行更新。

8. 一种数据回填装置，其特征在于，包括：

发送单元，将终端的第一页面中的用户输入数据和所述第一页面中输入框的标识相关联地发送至服务器，以由所述服务器进行存储，且存储内容映射于所述终端的 sessionID、所述第一页面的标识；

请求单元，当由第二页面返回所述第一页面时，向所述服务器发送数据获取请求，所述数据获取请求中包含所述终端的 sessionID 和所述第一页面的标识；

回填单元，接收所述服务器根据所述数据获取请求返回的所述第一页面中的用户输入数据和相关联的所述第一页面中输入框的标识，并将所述第一页面中的用户输入数据回填至所述第一页面中对应的输入框内。

9. 根据权利要求 8 所述的装置, 其特征在于, 所述发送单元具体用于:
将所述第一页面中的用户输入数据和所述第一页面中输入框的标识相关联地生成预设格式的打包数据;

将所述打包数据发送至所述服务器。

5 10. 根据权利要求 9 所述的装置, 其特征在于, 所述预设格式为 JSON 格式。

11. 根据权利要求 8 所述的装置, 其特征在于, 还包括:

第一更新单元, 当由所述第一页面切换至其他页面时, 将所述第一页面中的最新用户输入数据和所述第一页面中输入框的标识相关联地发送至所述
10 服务器, 以由所述服务器对映射于所述终端的 sessionID 和所述第一页面的标识的存储内容进行更新。

12. 根据权利要求 8 所述的装置, 其特征在于, 还包括:

比较单元, 当由所述第一页面切换至其他页面时, 将所述第一页面中的最新用户输入数据与所述服务器返回的用户输入数据进行比较;

15 第二更新单元, 若比较结果为不完全相同, 则将存在差异的用户输入数据和对应的输入框的标识相关联地发送至所述服务器, 以由所述服务器更新所述存储内容。

13. 一种数据回填装置, 其特征在于, 包括:

数据存储单元, 关联存储终端发送的第一页面中的用户输入数据和所述
20 第一页面中输入框的标识, 且存储内容映射于所述终端的 sessionID、所述第一页面的标识;

请求接收单元, 接收所述终端在由第二页面返回所述第一页面时发送的数据获取请求;

数据返回单元, 根据所述数据获取请求中包含的所述终端的 sessionID 和
25 所述第一页面的标识, 向所述终端返回所述第一页面中的用户输入数据和关联的所述第一页面中输入框的标识, 以由所述终端将所述第一页面中的用户输入数据回填至所述第一页面中对应的输入框内。

14. 根据权利要求 13 所述的装置，其特征在于，还包括：

数据接收单元，接收所述终端在由所述第一页面切换至其他页面时发送的所述第一页面中的最新用户输入数据和相关联的所述第一页面中输入框的标识；

- 5 数据比较单元，将所述第一页面中的最新用户输入数据与已存储的所述第一页面中的用户输入数据进行比较；

内容更新单元，若比较结果为不完全相同，则对映射于所述终端的 sessionID 和所述第一页面的标识的存储内容进行更新。