

浅谈移动项目的本地存储

黄华林

(广东女子职业技术学院 广东 广州 511450)

摘要:介绍了Html5本地存储的相关概念,以及Html5本地存储的一些特性和操作本地存储的API,通过本地存储的典型应用场景来示范本地存储的使用,说明了移动项目中使用本地存储的优势,给移动应用开发工程师提供了指导和借鉴意义。

关键词:本地存储; Localstorage; Html5; 移动应用

中图分类号:TP393 文献标识码:A 文章编号:1009-3044(2015)19-0049-03

DOI:10.14004/j.cnki.ckt.2015.1588

据报道,2014年底全球智能手机应用订阅总量或将超越全球总人口,移动应用似乎已经成为人们生活密不可分的一部分。排除那些还没普及智能手机的国家与地区,简单计算,几乎平均每位用户的智能手机上都安装着1个以上的应用程序:社交、餐饮、交通、租住、旅游、购物。众所周知,随着html5的普及,越来越多的移动应用朝着传统网页的运行模式发展,这些应用时不时进行联网,或加载用户数据,或开始引用云端存储用户进度等。但不少业界认为,随着移动应用存储技术的发展,本地实时存储将成为最主流的数据保存方式。

1 HTML5本地存储Localstorage

在Web的发展史上,在很长时间内Cookies是唯一可以使用的在用户本地存储少量数据的方法。但Cookies有一些非常明显的缺陷,限制了它的应用:

1) Cookie会被附加在每个HTTP请求中,所以无形中增加了流量,对移动应用非常不利。

2) 由于在HTTP请求中的Cookie是明文传递的,所以安全性成问题。

3) Cookie的大小限制在4 KB左右,对于复杂的存储需求来说是不够用的。

移动应用中,数据都由Server端提供,但移动应用一般都会支持offline使用模式,因此需要移动客户端本地存储数据。HTML5的本地存储能够很好地解决问题。简单来说,HTML5的本地存储就是一种让网页可以把键值对存储在浏览器客户端的方法。像Cookie一样,这些数据不会因为用户打开新网站,刷新页面,乃至关闭浏览器而消失。而与Cookie不同的是,这些数据不会每次随着HTTP请求被发送到服务器端,这是HTML5规范的一部分,Html5使用JavaScript来存储和访问数据。这一接口会被浏览器原生支持,不用依赖任何第三方插件。

1.1 HTML5本地存储的一些特性

1) 大小限制在5M,各个浏览器不一致。

2) 在隐私模式下不可读取。

3) 本质是在读写文件,数据多的话会比较卡。

4) 不能被爬虫爬取,不要用本地存储完全取代URL传参。

1.2 存储对象分类

HTML5的本地存储提供了一种全新的不同于cookie的方式,让Web页面在浏览器中以键值对的形式在本地存储数据。本地存储对象分为两种,sessionStorage和localStorage。

sessionStorage:生命周期为当前窗口或标签页,一旦窗口或标签页被永久关闭了,那么所有通过sessionStorage存储的数据也就被清空了。

localStorage:将数据保存在客户端硬件设备上,生命周期是永久,这意味着除非用户显示在浏览器提供的UI上清除localStorage信息,否则这些信息将永远存在。

不同浏览器无法共享localStorage或sessionStorage中的信息。相同浏览器的不同页面间可以共享相同的localStorage(页面属于相同域名和端口),但是不同页面或标签页间无法共享sessionStorage的信息。

localStorage和sessionStorage一样都是用来存储客户端临时信息的对象,两者区别就是一个作为临时保存,一个长期保存。

1.3 本地存储的操作示范

localStorage和sessionStorage的用法基本上相同,这里以localStorage为例作代码示范。

1) 检查浏览器是否支持本地存储

```
if(window.localStorage){
    alert('浏览器支持localStorage');
}
else{
    alert('浏览器不支持localStorage');
}
```

收稿日期:2015-05-29

基金项目:本文系广东女子职业技术学院科学、教学研究基金资助项目(项目编号为:B201308)

作者简介:黄华林(1981-),男,广东电白县人,讲师,硕士,主要研究方向为计算机技术应用、软件测试。

2)本地存储数据的操作

存储数据的方法就是直接给 window.localStorage 添加一个属性,例如:window.localStorage.attr_a 或者 window.localStorage["attr_a"]。属性的读取、写入、删除操作方法很简单,是以键值对的方式存在的,常见操作如下:

```
localStorage.attr_a = 8;           //设置 attr_a 为"8"
localStorage["attr_a"] = "hello"; //设置 attr_a 为"hello"
localStorage.setItem("attr_b","world"); //设置 attr_b 为"world"

var str1 = localStorage["attr_a"]; //获取 attr_a 的值
var str2 = localStorage.attr_a;    //获取 attr_a 的值
var str3 = localStorage.getItem("attr_b"); //获取 attr_b 的值
localStorage.removeItem("attr_c"); //清除 attr_c 的值
```

3)本地存储的监听

HTML5 的本地存储提供了一个 storage 事件,可以监听对键值对的变化,使用方法如下:

```
if(window.addEventListener)
{
    window.addEventListener("storage",handle_storage,false);
}
else
{
    if(window.attachEvent)
    {
        window.attachEvent("onstorage",handle_storage);
    }
}

function handle_storage(e)
{
    if(!e){e=window.event;}
    showStorage();
}
```

利用这些简单的本地存储操作的 API,可以完成大量任务,比如说个性化或跟踪用户行为,这些对移动应用开发人员来说是重要的设计任务。但是本地存储还有一个更为重要的应用场景:高速缓存。利用本地存储,移动客户端可以很容易地从服务器高速缓存数据,让用户无需等待缓慢的服务器回调,并且最小化了对服务器上数据的需求量。

2 移动项目中本地存储的应用

用户使用移动应用最多的操作之一是购买某项服务或产品,移动应用中虚拟购物车的数据存储是本地存储的典型应用场景。

2.1 虚拟购物车的需求分析

虚拟购物车的本质就是一个意向订单,含有用户信息和该用户的意向商品的信息。典型的虚拟购物车基本需求如下:

- 1)虚拟购物车记录用户意向商品的相关信息,如商品编号、名称、单价、数量等。主要功能是添加、删除、修改意向商品信息。
- 2)添加商品到购物车时,若商品已存在于购物车中,则该商品数量相应增加,否则直接记入。
- 3)删除购物车中的商品。
- 4)修改购物车中商品的数量,可增可减,当数量减少为0时,即可从购物车中删除该商品。

2.2 基于本地存储的购物车实现

由于商品对象信息最终以字符串信息进行本地存储,需要引入 JSON 库,来处理商品对象与商品对象字符串之间的转换。在如下的 JavaScript 代码中,myLS 对象操作本地储存,用本地存储记录购物车信息("myCart" 键)。product 是商品类,shop_cart 是购物车类,shop_cart 内部实现了商品信息的添加、删除、修改。基于本地存储的购物车的实现代码如下:

```
myLS = {
    setParam : function(name,value){
        localStorage.setItem(name,value)
    },
    getParam : function(name){
        return localStorage.getItem(name)
    }
};

product = { p_id:0, p_mingcheng:"", p_shuliang:0, p_danjia:0.00};

shop_cart = {
    //添加商品到购物车
    add_p:function(product){
        var myCart = myLS.getParam("myCart");
        if(myCart == null || myCart == "")
        {
            //购物车为空
            var jsonstr = {"p_list":[{"p_id":product.p_id,"p_mingcheng":product.p_mingcheng,"p_shuliang":product.p_shuliang,"p_danjia":product.p_danjia}]};
            //保存购物车信息
            myLS.setParam("myCart",""+JSON.stringify(jsonstr));
        }
        else
        {
            var jsonstr = JSON.parse(myCart.substr(1,myCart.length));
            var p_list = jsonstr.p_list;
            var result=false;
            //查找购物车中是否已有该商品
            for(var i in p_list){
                if(p_list[i].p_id == product.p_id){
                    p_list[i].p_shuliang = parseInt(p_list[i].p_shuliang)+parseInt(product.p_shuliang);
                    result = true;
                }
            }
            if(!result){
                //购物车中未有该商品
                p_list.push({"p_id":product.p_id,"p_mingcheng":product.p_mingcheng,"p_shuliang":product.p_shuliang,"p_danjia":product.p_danjia});
            }
            myLS.setParam("myCart",""+JSON.stringify(jsonstr));
        }
    }
}
```

```

    },
    //修改商品数量
    edit_p_shuliang:function(p_id,p_shuliang){
        var myCart = myLS.getParam("myCart");
        var jsonstr = JSON.parse(myCart.substr(1,myCart.
length));
        var p_list = jsonstr.p_list;
        for(var i in p_list)
        {
            if(p_list[i].p_id == p_id)
            {
                p_list[i].p_shuliang = parseInt(p_shuliang);
                myLS.setParam("myCart","" + JSON.stringify(json-
str));
                return;
            }
        }
    },
    //删除购物车中商品 p_id
    delete_p:function(p_id){
        var myCart = myLS.getParam("myCart");
        var jsonstr = JSON.parse(myCart.substr(1,myCart.
length));
        var p_list = jsonstr.p_list;
        var list = [];
        for(var i in p_list)
        {
            if(p_list[i].p_id != p_id)
            {
                list.push(p_list[i]);
            }
        }
    }
}

```

```

    }
    jsonstr.p_list = list;
    myLS.setParam("myCart","" + JSON.stringify(jsonstr));
},
//获取购物车中的所有商品
get_p_list:function(){
    var myCart = myLS.getParam("myCart");
    var jsonstr = JSON.parse(myCart.substr(1,myCart.
length));
    var p_list = jsonstr.p_list;
    return p_list;
}
};

```

3 结束语

Html5的本地存储给移动 Web App 开发人员提供了巨大的客户端存储空间,摆脱了 Cookie 模式下的限制,开启了数据的本地高速缓存,振奋人心。Html5的本地存储将大大改善移动应用的性能,并能实现移动应用的离线使用,极大的提高用户体验,这正是移动应用开发人员所努力追求的。

参考文献:

- [1] 刘天寅. HTML5 与未来的 WEB 应用平台[J]. 阴山学刊: 自然科学版, 2010(2).
- [2] 严伟中. 关于 HTML5 的核心技术研究与应用[J]. 网络安全技术与应用, 2014(3).
- [3] 许中博, 董彬. 利用 HTML5 的本地存储实现图书馆网站的个性化[J]. 科技资讯, 2013(11).
- [4] 冯永亮. HTML5 本地数据存储技术研究[J]. 西安文理学院学报: 自然科学版, 2014(3).
- [5] 唐俊开. HTML5 移动 Web 开发指南[M]. 北京: 电子工业出版社, 2012.