

# 突破本地离线存储5M限制的JS库localforage简介

这篇文章发布于 2018年06月4日, 星期一, 01:25, 归类于 [JS实例](#)。阅读 12691 次, 今日 14 次 [15 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=7621>

本文可全文转载, 但需得到原作者书面许可, 同时保留原作者和出处, 摘要引流则随意。

## 一、localforage项目简介

`localforage` 项目地址: <https://github.com/localForage/localForage>

截止今日 (2018-06-03), 已经有12000+个星星, 万级别的项目。



`localforage` 用来本地存储数据的。

说到本地存储数据, 我们首先想到的是 `localStorage`, 应该很多小伙伴都用过, 使用很简单。然而, `localStorage` 却有下面一些缺点:

1. 存储容量限制, 大部分浏览器应该最多5M。我就遇到过`localStorage`存储字符串然后尺寸爆掉的情况。
2. 仅支持字符串, 如果是存对象还需要将使用`JSON.stringify`和`JSON.parse`方法互相转换, 有些啰嗦。
3. 读取都是同步的。大多数情况下, 还挺好使的。但, 如果存储数据比较大, 例如一张重要图片base64格式存储了, 再读可能会有可感知的延迟时间。

`localforage` 的作用就是用来规避上面 `localStorage` 的缺点, 同时保留 `localStorage` 的优点而设计的。从命名上就可以看出两者关系不浅。

`localStorage` 的优点是API非常简单, 使用很方便。于是, `localforage` 和 `localStorage` 一模一样。

至于 `localStorage` 的不足, `localforage` 和使用其他HTML5 API进行规避, 什么API呢? 是 `IndexedDB` 和 `WebSQL`。

也就是说, `localforage` 的逻辑是这样的: 优先使用 `IndexedDB` 存储数据, 如果浏览器不支持, 使用 `WebSQL`, 浏览器再不支持, 使用 `localStorage`。

然后API还是 `localStorage` 的API, 也就是数据增删改查通过 `get`, `set`, `remove`, `clear` 和 `length` 等API。

## 二、localforage使用案例一则

您可以狠狠地点击这里: [localforage存储blob格式的本地上传图片demo](#)

第一次进入这个demo页面, 只有一个上传图片按钮。

选择上传图片按钮, 则会以blob的形式直接把选择的图片在网页中显示处理, 例如:



此时，你再刷新页面，则显示的还是这张图片，因为使用 `localStorage` 把这张图片以blob数据形式存储在了本地。



相关HTML代码为：

```
<label for="fileImg">上传图片</label>
<input type="file" id="fileImg" accept="image/*" hidden>
<p id="result"></p>
```

然后JS代码如下：

```

<script src="./localforage.min.js"></script>
<script>
var eleResult = document.getElementById('result');
// 图片资源
var eleImg = document.createElement('img');
// 获取本地存储数据
localforage.getItem('zxxImg', function (err, value) {
    if (err == null && value) {
        eleImg.src = value;
        eleResult.appendChild(eleImg);
    }
});
// 选择的本地文件以Blob形式呈现
var reader = new FileReader();
reader.onload = function(event) {
    if (!eleImg.src) {
        eleResult.appendChild(eleImg);
    }
    var blob = URL.createObjectURL(new Blob([event.target.result]));
    eleImg.src = blob;
    // blob本地存储
    localforage.setItem('zxxImg', blob);
};
// 选择的文件对象
var file = null;
document.getElementById('fileImg').addEventListener('change', function (event) {
    file = event.target.files[0];
    // 选择的文件是图片
    if (file.type.indexOf("image") == 0) {
        reader.readAsArrayBuffer(file);
    }
});
</script>

```

可以看到，虽然localforage的API名称和localStorage一样，但是，在同步还是异步上却不同，localforage是异步执行的，用法示意如下。

```

localforage.getItem('key', function (err, value) {
    // 如果err不是null，则出错。否则value就是我们想要的值
});

```

### 三、localforage和indexDB的区别

**indexDB** 为本地数据库存储，其功能非常强大，再复杂的结构存储都不在话下。**localStorage** 只是使用了其功能中的一部分，很多功能受限，例如，**localStorage** 一次只能存一个字段。

我之前有专门写过文章介绍HTML5 **indexedDB**，文章名为：[“HTML5 indexedDB前端本地存储数据库实例教程”](#)。

**indexDB** 几乎空间无限，性能也不错，各种数据结构都支持，为何总感觉在业内不温不火呢？

我觉得很重要的原因之一就是上手成本，包括2方面：

1. 前端需要了解数据库的一些基本概念，例如表，游标，事务，锁等。而业界普遍的前端都是与页面打交道的，数据库操作属于后端的后端了，离的有些远，于是，很多前端都不了解，需要从零开始的数据库学习。

2. indexedDB的API又多又长又纷杂，学习成本高，容易记不住，网上好的资源少。

`localforage` 的出现可谓曲线救国，通常我们的数据存储并不需要特别复杂，只要不是完完全全的离线开发，`localforage` 足矣。既不浪费 `indexedDB` 的好，又避开了 `indexedDB` 高上手成本的坑。从这个角度看，`indexedDB` 应该要谢谢 `localforage`。

当然，如果存储的数据是负责的多行多列表结构，我建议还是老老实实花点功夫学习学习 `indexedDB` 的使用。

## 四、结束语

`indexedDB` IE10+浏览器支持。因此，如果想要使用 `localforage` 存储任意格式数据，需要注意下浏览器的兼容性问题，例如，本地图片存储Blob数据，IE9肯定是不支持的。这些浏览器怕是只能存字符串了。

一个东西是否有生命力，看的不是其是否强大，而是是否足够简单。

就酱紫，感谢阅读！

//zxx: 今天骑士、勇士第二场，骑士凶多吉少，我觉得第四节很可能会有垃圾时间。第一场着实可惜了。

《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

(本篇完) // 想要打赏? 点击[这里](#)。有话要说? 点击[这里](#)。



« JS实现照片图片变成黑白线条线稿

《CSS世界》女主角诚寻靠谱一起奋斗之人»

### 猜你喜欢

- 翻译：清除各个浏览器中的数据研究
- HTML5 localStorage本地存储实际应用举例
- HTML5 indexedDB前端本地存储数据库实例教程
- zSlide-基于CSS3/HTML5演示文档jQuery插件
- 借助Service Worker和cacheStorage缓存及离线开发
- 翻译-你必须知道的28个HTML5特征、窍门和技术
- 遐想：如果没有IE6和IE7浏览器...
- 理解DOMString、Document、FormData、Blob、File、ArrayBuffer数据类型
- XMLHttpRequest实现HTTP协议下文件上传断点续传
- JS前端创建html或json文件并浏览器导出下载
- JS检测PNG图片是否有透明背景、抠图等相关处理

分享到： 1

标签： Blob, indexedDB, localforage, localStorage, 本地存储, 离线存储

## 发表评论（目前15条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

### 1. yunfeng说道：

2018年12月21日 15:22



保存图片的Demo，不应该保存Blob的URL地址，这个地址生命周期和localStorage是不一致的。当Blob释放掉的时候，Blob URL就读不到数据了。正确做法应该是缓存ArrayBuffer的二进制数据，然后取出cache的时候，再URL.createObjectURL

[回复](#)

张鑫旭说道：

2018年12月22日 18:27

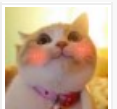
学习了。

[回复](#)



### 2. LinQi说道：

2018年07月2日 11:36



突然发现网页右上角张老师的简介那边的《CSS世界》的传送门是https的，然后访问不了...不知道是不是证书过期了.. <http://www.cssworld.cn/>，这个是可以的

[回复](#)

张鑫旭说道：

2018年07月4日 22:59

感谢反馈！

[回复](#)



### 3. 寂寞看书说道：

2018年06月14日 11:59



看了一下caniuse，indexDB在移动端的兼容性挺好的，感觉是能用了

[回复](#)

### 4. 莫问道说道：

2018年06月12日 11:34



那么极端情况只能使用localStorage并且满了，还是一样吧

[回复](#)

5. BaoMax说道:

2018年06月9日 10:43

只要清空缓存加载图片就会加载不出来, 是否考虑存base64格式的图片, 不存储blob, 如果存blob的话localstorage应该也不会有问题吧, 因为blob只是个url, blob:http://www.zhangxinxu.com/8d8ee0e5-a1c0-4942-a321-c084178fb414 这样

[回复](#)



6. 和尚爱敲钟说道:

2018年06月5日 14:48

为什么刷新后图片加载不了, 报错Not allowed to load local resource: blob:null/73379553-ad9a-4e92-bdea-5e58b62c1be0

[回复](#)



张鑫旭说道:

2018年06月5日 14:59

是我的demo页面, 还是自己测试的页面? 是http协议访问的吗?

[回复](#)



小宝说道:

2018年06月7日 09:15

在本地的可以, 用他的demo哪里是http访问的会有问题, 自己写一个试试就好了啊

[回复](#)



小宝说道:

2018年06月7日 09:43

补充, 在关闭网页的时候createObjectURL 会自动释放, 再次打开网页也会的不到你要的图片 (虽然你的地址还在)

[回复](#)



7. u3u说道:

2018年06月4日 20:18

> 开发者模式看了下, 刷新居然图片不显示, 原来是 disable cache 开启了, 关了后正常显示了。这个能优化吗?

我还发现上传后打开开发者工具切换到移动模式后刷新图片也不显示

[回复](#)



长江长江我是黄河说道:

2018年06月5日 11:10

我发现移动状态下上传刷新没事。若是从pc状态上传切换到移动模式再刷新就会是你说的酱紫。从移动模式上传切到pc刷新也是。我觉得实际应用情况下没问题, 手机就是手机, 没有切换功能。具体能不能优化, 期待中...

[回复](#)



8. 码农说道:

2018年06月4日 15:35

开发者模式看了下, 刷新居然图片不显示, 原来是disable cache开启了, 关了后正常显示了。这个能优化吗?

[回复](#)



9. dddd说道:

2018年06月4日 09:48

jiu jiang zi

[回复](#)



## 最新文章

- » [常见的CSS图形绘制合集](#)
- » [粉丝群第1期CSS小测点评与答疑](#)
- » [分享三个纯CSS实现26个英文字母的案例](#)
- » [小tips: 纯CSS实现打字动画效果](#)
- » [CSS/CSS3 box-decoration-break属性简介](#)
- » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的?](#)
- » [周知: CSS -webkit-伪元素选择器不再导致整行无效](#)

## 今日热门

- » [常见的CSS图形绘制合集](#) <sup>(179)</sup>
- » [粉丝群第1期CSS小测点评与答疑](#) <sup>(113)</sup>
- » [未来必热: SVG Sprite技术介绍](#) <sup>(111)</sup>
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) <sup>(85)</sup>
- » [让所有浏览器支持HTML5 video视频标签](#) <sup>(83)</sup>
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) <sup>(80)</sup>
- » [CSS3下的147个颜色名称及对应颜色值](#) <sup>(78)</sup>
- » [小tips: 纯CSS实现打字动画效果](#) <sup>(73)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(70)</sup>
- » [分享三个纯CSS实现26个英文字母的案例](#) <sup>(70)</sup>

## 今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) <sup>(76)</sup>
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) <sup>(64)</sup>
- » [看, for..in和for..of在那里吵架!](#) <sup>(60)</sup>
- » [是时候好好安利下LuLu UI框架了!](#) <sup>(47)</sup>
- » [原来浏览器原生支持JS Base64编码解码](#) <sup>(35)</sup>
- » [妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现](#) <sup>(33)</sup>
- » [炫酷H5中序列图片视频化播放的高性能实现](#) <sup>(31)</sup>
- » [CSS scroll-behavior和JS scrollToView让页面滚动平滑](#) <sup>(30)</sup>
- » [windows系统下批量删除OS X系统.DS\\_Store文件](#) <sup>(26)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(26)</sup>

## 猜你喜欢

- [翻译: 清除各个浏览器中的数据研究](#)
- [HTML5 localStorage本地存储实际应用举例](#)
- [HTML5 indexedDB前端本地存储数据库实例教程](#)
- [zSlide-基于CSS3/HTML5演示文档jQuery插件](#)
- [借助Service Worker和cacheStorage缓存及离线开发](#)
- [翻译-你必须知道的28个HTML5特征、窍门和技术](#)
- [遐想: 如果没有IE6和IE7浏览器...](#)
- [理解DOMString、Document、FormData、Blob、File、ArrayBuffer数据类型](#)
- [XMLHttpRequest实现HTTP协议下文件上传断点续传](#)
- [JS前端创建html或json文件并浏览器导出下载](#)
- [JS检测PNG图片是否有透明背景、抠图等相关处理](#)