

HTML5: Offline Web Application

1. [浏览器支持](#)
2. [API 参考](#)
3. [DEMO](#)
4. [注意事项](#)
5. [兼容解决方案](#)
6. [参考文献](#)

一、浏览器支持

Platform	MAC				WIN			
Browser	OPERA	FIREFOX	SAFARI	CHROME	OPERA	FIREFOX		SAFARI
Version	10.63	3.6	5	7	10.63	3.6	4.03	5
offline	√	√	√	√	√	√	√	√

二、API 参考

1. **API SPEC:** <http://www.w3.org/TR/offline-webapps/>

2. **HTML**

1. 在 `<html />` 中添加 `manifest` 属性：

```
<html manifest="http://www.example.com/example.manifest">
...
</html>
```

2. 创建 `manifest` 文件：`foo.manifest`

```

CACHE MANIFEST
# '#'号是注释，变更注释，也会重新缓存，因此我们常用它来做版本标记
# version 0.3 2010.10.23

# 在 .manifest 文件中的路径，默认会被缓存
# 路径可以是相对的，也可以是绝对的
img/a.png

# CACHE 标记下面的文件都会被缓存
CACHE:
img/b.img

# NETWORK 标记下面的文件都会保持在线
NETWORK:
*

# FALLBACK 是当文件有网络请求，但又处于离线状态的一种替代方案
FALLBACK:
/ajax /ajax_offline

```

3. Scripting API

(1) 浏览器检测

```
var hasApplicationCache = !! (window.applicationCache);
```

(2) 在线还是离线？

navigator 属性	描述
navigator.onLine	<pre> if (navigator.onLine) { // 当前在线 } else { // 当前离线 } </pre>

```

window.online = function(){
    // 离线
}

```

事件	描述
online	在线状态
offline	离线事件

(3) MIME Type

text/cache-manifest

(4) Cache States

状态码	描述
0	UNCACHED
1	IDLE
2	CHECKING
3	DOWNLOADING
4	UPDATEREADY
5	OBSOLETE

状态事件	事件对应的状态
onchecking	CHECKING
ondownloading	DOWNLOADING
onupdateready	UPDATEREADY
onobsolete	OBSOLETE
oncached	IDLE
onerror	发生错误
onnoupdate	没有变更
onprogress	正在缓存中

(5) 更新缓存

方法	描述
window.applicationCache.update()	请求浏览器更新离线数据的缓存

三、DEMO

[HTML5 application Cache](#)

四、注意事项

- 因为 `cache manifest` 的主要应用对象是离线应用，因此其会缓存引用 `manifest` 的页面。如果在一般页面中使用，一定要非常明确自己的使用方法和地方是否正确，不注意的话，容易导致更新不会被显示出来。
- 在 Firefox 的 `about:cache` 可以查看缓存的设备

五、兼容解决方案

1. 离线缓存 Apache Cache 等，没有前端的缓存解决方案。
2. 页面引用文件缓存，使用 `iframe`：[HTML5 缓存: cache manifest](#)

六、参考文献

- BOOK: 《Pro HTML5 Programming》
- [HTML5 SPEC: offline web application](#)
- [A Beginner's Guide to Using the Application Cache](#)
- [HTML5 & CSS3 Support, Web Design Tools & Support](#)