



chrome.storage

描述:	使用 chrome.storage API 存储、获取用户数据，追踪用户数据的更改。
可用版本:	从 Chrome 20 开始支持。
权限:	"storage"
内容脚本:	完全支持。 了解更多

概述

这一 API 为应用的存储需要而特别优化，它提供了与 **localStorage API** 相同的能力，但是具有如下几个重要的区别：

- 您的应用的内容脚本可以直接访问用户数据，而不需要后台页面。
- 即使使用**分离式隐身行为**，用户的应用设置也会保留。
- 它是异步的，并且能够进行大量的读写操作，因此比阻塞和串行化的 **localStorage API** 更快。
- 用户数据可以存储为对象（**localStorage API** 以字符串方式存储数据）。
- 可以读取管理员为应用配置的企业策略（使用 **storage.managed** 和**架构**）。

清单文件

您必须在**应用的清单文件**中声明 "storage" 权限才能使用存储 API。例如：

```
{
  "name": "我的应用",
  ...
  "permissions": [
    "storage"
  ],
  ...
}
```

用法

如果要为您的应用储存用户数据，您可以使用 **storage.local**。

不应该储存机密的用户信息！存储区没有加密。

storage.managed 存储区是只读的。

存储空间与调用频率限制

chrome.storage 并不像一辆大卡车那样，而是像一系列的管道，如果您不理解这一点的话，这样的管道很容易被填满。如果当您存入消息时它们填满了，它将会变成细线，任何人向其中存入大量数据都有可能使操作产生延迟。

有关目前对存储 API 的限制以及超出限制的结果，请参见 [local](#) 的配额信息。

例子

以下例子检查用户在表单中保存的 CSS 代码，如果找到的话则存储下来。

```
function saveChanges() {  
  // 获取表单中保存的值。  
  var theValue = textarea.value;  
  // 确保包含代码。  
  if (!theValue) {  
    message("错误：没有指定值");  
    return;  
  }  
  // 使用百度浏览器应用程序的存储 API 保存它。  
  chrome.storage.local.set({'value': theValue}, function() {  
    // 通知保存完成。  
    message("设置已保存");  
  });  
}
```

如果您希望追踪数据对象的更改，您可以向 `onChanged` 事件添加监听器，每当存储有任何更改时将会产生该事件。如下是监听对已保存内容的更改的示例代码：

```
chrome.storage.onChanged.addListener(function(changes, namespace) {  
  for (key in changes) {  
    var storageChange = changes[key];  
    console.log("存储键“%s”（位于“%s”命名空间中）已更改。" +  
      "原来的值为“%s”，新的值为“%s”。",  
      key,  
      namespace,  
      storageChange.oldValue,  
      storageChange.newValue);  
  }  
});
```

摘要

类型
StorageChange
StorageArea
属性
local
managed
事件
onChanged

类型

StorageChange

属性		
any	(可选) oldValue	如果有旧值，则为项目的旧值。
any	(可选) newValue	如果有新值，则为项目的新值。

StorageArea

方法														
<p>get <code>StorageArea.get(string or array of string or object keys, function callback)</code></p> <p>从存储中获得一个或多个值。</p> <table><tr><th colspan="3">参数</th></tr><tr><td>string or array of string or object</td><td>(可选) keys</td><td>要获得的单个键、多个键的列表或者指定默认值的词典（参见对象的描述），空的列表或对象将会返回空的结果对象。要获得存储的所有内容，请传递 <code>null</code>。</td></tr><tr><td>function</td><td>callback</td><td>包含存储项目或者表示失败（这种情况下会设置 <code>runtime.lastError</code>）的回调函数。 <i>callback</i> 参数应该是一个如下形式的函数： <code>function(object items) {...};</code><table><tr><td>object</td><td>items</td><td>包含按键-值映射的项目对象。</td></tr></table></td></tr></table>			参数			string or array of string or object	(可选) keys	要获得的单个键、多个键的列表或者指定默认值的词典（参见对象的描述），空的列表或对象将会返回空的结果对象。要获得存储的所有内容，请传递 <code>null</code> 。	function	callback	包含存储项目或者表示失败（这种情况下会设置 <code>runtime.lastError</code> ）的回调函数。 <i>callback</i> 参数应该是一个如下形式的函数： <code>function(object items) {...};</code> <table><tr><td>object</td><td>items</td><td>包含按键-值映射的项目对象。</td></tr></table>	object	items	包含按键-值映射的项目对象。
参数														
string or array of string or object	(可选) keys	要获得的单个键、多个键的列表或者指定默认值的词典（参见对象的描述），空的列表或对象将会返回空的结果对象。要获得存储的所有内容，请传递 <code>null</code> 。												
function	callback	包含存储项目或者表示失败（这种情况下会设置 <code>runtime.lastError</code> ）的回调函数。 <i>callback</i> 参数应该是一个如下形式的函数： <code>function(object items) {...};</code> <table><tr><td>object</td><td>items</td><td>包含按键-值映射的项目对象。</td></tr></table>	object	items	包含按键-值映射的项目对象。									
object	items	包含按键-值映射的项目对象。												
<p>getBytesInUse <code>StorageArea.getBytesInUse(string or array of string keys, function callback)</code></p> <p>获得一个或多个项目正在使用的空间大小（以字节为单位）。</p> <table><tr><th colspan="3">参数</th></tr><tr><td>string or array of string</td><td>(可选) keys</td><td>要获得总计使用空间的单个键或多个键的列表，空的列表或对象将会返回 <code>0</code>。要获得所有存储占用的总空间，请传递 <code>null</code>。</td></tr><tr><td>function</td><td>callback</td><td>回调函数，将传递存储占用的空间大小，或者指示失败（这种情况下将会设置 <code>runtime.lastError</code>） <i>callback</i> 参数应该是一个如下形式的函数： <code>function(integer bytesInUse) {...};</code><table><tr><td>integer</td><td>bytesInUse</td><td>正在使用的存储空间大小，以字节为单位。</td></tr></table></td></tr></table>			参数			string or array of string	(可选) keys	要获得总计使用空间的单个键或多个键的列表，空的列表或对象将会返回 <code>0</code> 。要获得所有存储占用的总空间，请传递 <code>null</code> 。	function	callback	回调函数，将传递存储占用的空间大小，或者指示失败（这种情况下将会设置 <code>runtime.lastError</code> ） <i>callback</i> 参数应该是一个如下形式的函数： <code>function(integer bytesInUse) {...};</code> <table><tr><td>integer</td><td>bytesInUse</td><td>正在使用的存储空间大小，以字节为单位。</td></tr></table>	integer	bytesInUse	正在使用的存储空间大小，以字节为单位。
参数														
string or array of string	(可选) keys	要获得总计使用空间的单个键或多个键的列表，空的列表或对象将会返回 <code>0</code> 。要获得所有存储占用的总空间，请传递 <code>null</code> 。												
function	callback	回调函数，将传递存储占用的空间大小，或者指示失败（这种情况下将会设置 <code>runtime.lastError</code> ） <i>callback</i> 参数应该是一个如下形式的函数： <code>function(integer bytesInUse) {...};</code> <table><tr><td>integer</td><td>bytesInUse</td><td>正在使用的存储空间大小，以字节为单位。</td></tr></table>	integer	bytesInUse	正在使用的存储空间大小，以字节为单位。									
integer	bytesInUse	正在使用的存储空间大小，以字节为单位。												

set

StorageArea.set(object items, function callback)

设置多个项目。

参数		
object	items	包含要更新的键/值对的对象，存储中的其他键/值对不会受到影响。 像数值之类的原生值会以预期的方式序列化，除了 Array （按照预期的方式序列化）、 Date 和 RegExp （以字符串表示形式序列化）以外， typeof 为 "object" 和 "function" 的值通常序列化为 {} 。
function	（可选）callback	在成功或失败（这种情况下会设置 runtime.lastError ）时调用。 如果您指定了 callback 参数，它应该是一个如下形式的函数： function() {...};

remove

StorageArea.remove(string or array of string keys, function callback)

从存储中移除一个或多个项目。

参数		
string or array of string	keys	单个键或多个键的列表，表示要移除的内容。
function	（可选）callback	在成功或失败（这种情况下会设置 runtime.lastError ）时调用。 如果您指定了 callback 参数，它应该是一个如下形式的函数： function() {...};

clear

StorageArea.clear(function callback)

从存储中删除所有值。

参数		
function	（可选）callback	在成功或失败（这种情况下会设置 runtime.lastError ）时调用。 如果您指定了 callback 参数，它应该是一个如下形式的函数： function() {...};

属性

StorageArea	chrome.storage.local	位于 local （本机）存储区下的项目仅对每一台计算机有效。 <table><tr><th colspan="3">属性</th></tr><tr><td>5,242,880</td><td>QUOTA_BYTES</td><td>可以在本地存储区储存的数据量大小（以字节为单位），计算方式为每一个值 JSON 字符串化的结果加上每一个键的长度。如果应用拥有 unlimitedStorage 权限则会忽略这一值。</td></tr></table>	属性			5,242,880	QUOTA_BYTES	可以在本地存储区储存的数据量大小（以字节为单位），计算方式为每一个值 JSON 字符串化的结果加上每一个键的长度。如果应用拥有 unlimitedStorage 权限则会忽略这一值。
属性								
5,242,880	QUOTA_BYTES	可以在本地存储区储存的数据量大小（以字节为单位），计算方式为每一个值 JSON 字符串化的结果加上每一个键的长度。如果应用拥有 unlimitedStorage 权限则会忽略这一值。						

StorageArea	chrome.storage.managed	从 Chrome 33 开始支持。 managed 存储区中的项目由域管理员设置，对应用来说只读，尝试修改这一命名空间会导致错误。

事件

onChanged

当一个或多个项目更改时产生。

addListener

chrome.storage.onChanged.addListener(function callback)

参数								
function	callback	<p>callback 参数应该是一个如下形式的函数：</p> <pre>function(object changes, string areaName) {...};</pre> <table><tr><td>object</td><td>changes</td><td>一个对象，将更改的每一个键映射到该项目对应的 StorageChange 对象。</td></tr><tr><td>string</td><td>areaName</td><td>从 Chrome 22 开始支持。 这一更改对应的存储区（"local"（本机）或 "managed"）。 </td></tr></table>	object	changes	一个对象，将更改的每一个键映射到该项目对应的 StorageChange 对象。	string	areaName	从 Chrome 22 开始支持。 这一更改对应的存储区（"local"（本机）或 "managed"）。
object	changes	一个对象，将更改的每一个键映射到该项目对应的 StorageChange 对象。						
string	areaName	从 Chrome 22 开始支持。 这一更改对应的存储区（"local"（本机）或 "managed"）。						