



chrome.permissions

描述:	使用 <code>chrome.permissions</code> API 在运行时而不是安装时请求 声明的可选权限 ，这样用户可以理解为什么需要这些权限，并且仅在必要时授予这些权限。
可用版本:	从 Chrome 16 开始支持。
了解更多:	声明权限

实现可选权限

第一步：确定哪些权限是必选的，哪些是可选的

应用既可以声明必选权限，也可以声明可选权限。大体上说：

- 应用的基本功能需要某些权限时使用必选权限。
- 应用的可选功能需要某些权限时使用可选权限。

必选权限的优势：

- **提示较少：**应用只需要提示用户一次，接受所有权限。
- **开发更容易：**必选权限确保存在。

可选权限的优势：

- **安全性更高：**应用以更少的权限运行，用户仅启用需要的权限。
- **可以向用户提供更多信息：**当用户启用相关功能时，应用可以解释为什么需要某种权限。
- **便于更新：**您的应用更新时，如果更新版本添加了可选而不是必选权限，百度浏览器不会为用户禁用它。

第二步：在清单文件中声明可选权限

和 `permissions` 字段使用相同的格式，使用 `optional_permissions` 键，在您的[应用的清单文件](#)中声明可选权限：

```
{
  "name": "我的应用",
  ...
  "optional_permissions": [ "tabs", "http://www.google.com/" ],
  ...
}
```

您可以将下列任意权限指定为可选[权限](#)：

- 主机权限
- clipboardRead
- clipboardWrite
- contextMenus
- cookies
- notifications
- pageCapture

- tabs
- webNavigation
- webRequest
- webRequestBlocking

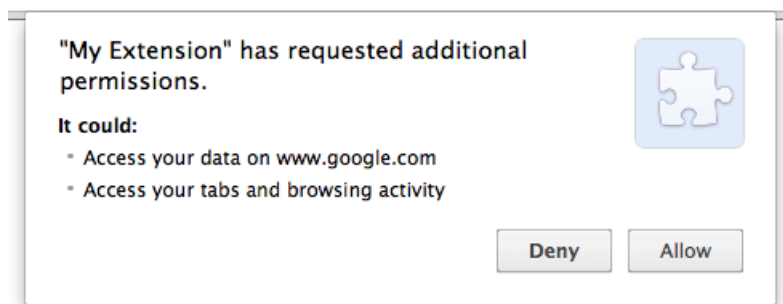
如果您希望请求只能在运行时发现的主机，您可以在 `optional_permission` 中包含 `"http://*"` 和/或 `"https://*"`，这样就允许您在 [Permissions.origins](#) 中指定任意来源，只要它具有匹配的协议。

第三步：请求可选权限

使用 `permissions.request()` 在用户操作下请求权限：

```
document.querySelector('#my-button').addEventListener('click', function(event) {  
    // 权限必须在用户操作下请求，例如按钮单击的事件处理函数。  
    chrome.permissions.request({  
        permissions: ['tabs'],  
        origins: ['http://www.google.com/']  
    }, function(granted) {  
        // 如果用户授予了这些权限，则回调函数的参数为 true。  
        if (granted) {  
            doSomething();  
        } else {  
            doSomethingElse();  
        }  
    });  
});
```

如果加入这些权限会导致与用户已经看见并接受的所不同的**权限警告**，百度浏览器会提示用户。例如，上面的代码可能会导致如下提示：



第四步：检查应用的当前权限

要检查您的应用是否拥有某个或某些权限，请使用 `permission.contains()`：

```
chrome.permissions.contains({  
    permissions: ['tabs'],  
    origins: ['http://www.google.com/']  
}, function(result) {  
    if (result) {  
        // 应用拥有这些权限。  
    } else {  
        // 应用没有这些权限。  
    }  
});
```

第五步：移除权限

您应该及时移除您不再需要的权限。在权限移除后，调用 `permissions.request()` 通常会再次添加权限而不再提示用户。

```
chrome.permissions.remove({
  permissions: ['tabs'],
  origins: ['http://www.google.com/']
}, function(removed) {
  if (removed) {
    // 权限已经移除。
  } else {
    // 权限没有移除（例如您试着移除必需的权限）
  }
});
```

摘要

类型
Permissions
方法
<code>getAll</code> – <code>chrome.permissions.getAll(function callback)</code>
<code>contains</code> – <code>chrome.permissions.contains(Permissions permissions, function callback)</code>
<code>request</code> – <code>chrome.permissions.request(Permissions permissions, function callback)</code>
<code>remove</code> – <code>chrome.permissions.remove(Permissions permissions, function callback)</code>
事件
<code>onAdded</code>
<code>onRemoved</code>

类型

Permissions

属性		
array of string	(可选) permissions	权限名称列表（不包括主机权限）。此处列出的所有权限都必须在清单文件的 <code>optional_permissions</code> 列表中出现。
array of string	(可选) origins	主机权限列表。此处列出的所有权限都必须为清单文件 <code>optional_permissions</code> 列表中出现的 主机子集。例如，如果 <code>http://*.example.com/</code> 或 <code>http://*</code> 包含在 <code>optional_permissions</code> 中，您可以请求 <code>http://help.example.com/</code> 来源。路径部分忽略。

方法

getAll

chrome.permissions.getAll(function callback)

获得应用当前拥有的所有权限。

参数					
function	callback	<div><div>callback 参数应该是一个如下形式的函数：</div><div><div>function(Permissions permissions) {...};</div><table><tr><td>Permissions</td><td>permissions</td><td>应用的当前权限。</td></tr></table></div></div>	Permissions	permissions	应用的当前权限。
Permissions	permissions	应用的当前权限。			

contains

chrome.permissions.contains(Permissions permissions, function callback)

检查应用是否拥有指定权限。

参数					
Permissions	permissions				
function	callback	<div><div>callback 参数应该是一个如下形式的函数：</div><div><div>function(boolean result) {...};</div><table><tr><td>boolean</td><td>result</td><td>如果应用拥有指定权限则为 true。</td></tr></table></div></div>	boolean	result	如果应用拥有指定权限则为 true。
boolean	result	如果应用拥有指定权限则为 true。			

request

chrome.permissions.request(Permissions permissions, function callback)

请求访问指定的权限，这些权限必须已在清单文件中的 optional_permissions 字段中定义。如果请求权限过程中发生任何问题，将会设置 runtime.lastError。

参数					
Permissions	permissions				
function	(可选) callback	<div><div>如果您指定了 callback 参数，它应该是一个如下形式的函数：</div><div><div>function(boolean granted) {...};</div><table><tr><td>boolean</td><td>granted</td><td>如果用户授予了指定权限则为 true。</td></tr></table></div></div>	boolean	granted	如果用户授予了指定权限则为 true。
boolean	granted	如果用户授予了指定权限则为 true。			

remove

chrome.permissions.remove(Permissions permissions, function callback)

移除指定权限的访问。如果移除权限过程中发生任何问题，将会设置 `runtime.lastError`。

参数					
Permissions	permissions				
function	(可选) callback	如果您指定了 <i>callback</i> 参数，它应该是一个如下形式的函数： <code>function(boolean removed) {...};</code> <table><tr><td>boolean</td><td>removed</td><td>如果权限已移除则为 <code>true</code>。</td></tr></table>	boolean	removed	如果权限已移除则为 <code>true</code> 。
boolean	removed	如果权限已移除则为 <code>true</code> 。			

事件

onAdded

应用获得新权限时产生。

addListener

`chrome.permissions.onAdded.addListener(function callback)`

参数					
function	callback	<i>callback</i> 参数应该是一个如下形式的函数： <code>function(Permissions permissions) {...};</code> <table><tr><td>Permissions</td><td>permissions</td><td>新获得的权限。</td></tr></table>	Permissions	permissions	新获得的权限。
Permissions	permissions	新获得的权限。			

onRemoved

对权限的访问从应用移除时产生。

addListener

`chrome.permissions.onRemoved.addListener(function callback)`

参数					
function	callback	<i>callback</i> 参数应该是一个如下形式的函数： <code>function(Permissions permissions) {...};</code> <table><tr><td>Permissions</td><td>permissions</td><td>已移除的权限。</td></tr></table>	Permissions	permissions	已移除的权限。
Permissions	permissions	已移除的权限。			