



chrome.webRequest

描述:	使用 <code>chrome.webRequest</code> API 监控与分析流量，还可以实时地拦截、阻止或修改请求。
可用版本:	从 Chrome 17 开始支持。
权限:	<code>"webRequest"</code> 主机权限

清单文件

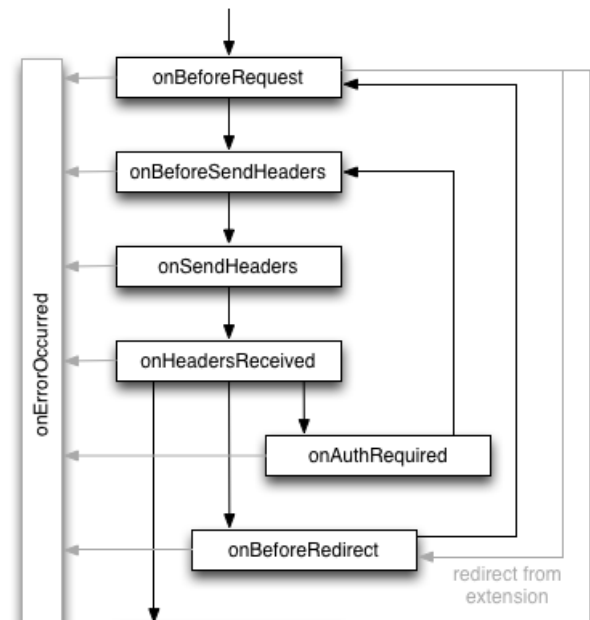
要使用网络请求 API，您必须在应用的清单文件中声明 `"webRequest"` 权限，以及您需要访问网络请求的所有主机的主机权限。如果您需要以阻塞方式使用网络请求 API，您还需要另外请求 `"webRequestBlocking"` 权限。例如：

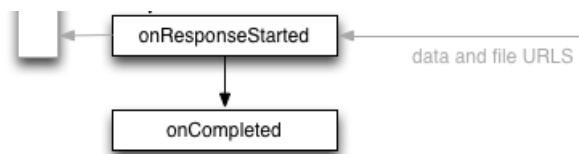
```
{
  "name": "我的应用",
  ...
  "permissions": [
    "webRequest",
    "*//*.google.com/"
  ],
  ...
}
```

请求的生命周期

网络请求 API 定义了一系列事件，在一次网络请求的生命周期内产生，您可以使用这些事件监控和分析流量。某些同步事件还允许您截获、阻止或者修改请求。

对于成功请求的事件生命周期如下图所示，接下来是事件的定义：





onBeforeRequest (可以为同步)

当请求即将发出时产生。这一事件在 TCP 连接建立前发送，可以用来取消或重定向请求。

onBeforeSendHeaders (可以为同步)

当请求即将发出并且初始标头已经准备好时产生。这一事件是为了使应用能够添加、修改和删除请求标

头 (*)。onBeforeSendHeaders 事件将传递给所有订阅者，所以不同的订阅者都可以尝试修改请求。有关具体如何处理的细节，请参见[实现细节](#)部分。这一事件可以用来取消请求。

onSendHeaders

当所有应用已经修改完请求标头并且展现最终 (*) 版本时产生。这一事件在标头发送至网络前触发，仅用于提供信息，并且以异步方式处理，不允许修改或取消请求。

onHeadersReceived (可以为同步)

每当接收到 HTTP(S) 响应标头时产生。由于重定向以及认证请求，对于每次请求这一事件可以多次产生。这一事件是为了使应用能够添加、修改和删除响应标头，例如传入的 Set-Cookie 标头。缓存指示是在该事件触发前处理的，所以修改 Cache-Control 之类的标头不会影响浏览器的缓存。它还允许您重定向请求。

onAuthRequired (可以为同步)

当请求需要用户认证时产生。这一事件可以同步处理，提供认证凭据。注意，应用提供的凭据可能无效，注意不要重复提供无效凭据，陷入无限循环。

onBeforeRedirect

当重定向即将执行时产生，重定向可以由 HTTP 响应代码或应用触发。这一事件仅用于提供信息，并以异步方式处理，不允许修改或取消请求。

onResponseStarted

当接收到响应正文的第一个字节时产生。对于 HTTP 请求，这意味着状态行和响应标头已经可用。这一事件仅用于提供信息，并以异步方式处理，不允许修改或取消请求。

onCompleted

当请求成功处理后产生。

onErrorOccurred

当请求不能成功处理时产生。

网络请求 API 保证对于每一个请求，onCompleted 或 onErrorOccurred 是最终产生的事件，除了如下例外：如果请求重定向至 data:// URL，onBeforeRedirect 将是最后报告的事件。

(*) 注意网络请求 API 向应用展现的是网络栈的一种抽象。单个 URL 请求在内部可以分割为几个 HTTP 请求（例如从一个大文件获取单独的字节范围）或者可以不与网络通信就由网络栈处理。由于这一原因，这一 API 不会提供最终发送至网络的 HTTP 标头。例如，所有与缓存相关的标头对应用都是不可见的。

如下是当前不提供给 onBeforeSendHeaders 事件的标头列表，这一列表不保证是完整的或者不会变化：

- Authorization
- Cache-Control
- Connection
- Content-Length
- Host
- If-Modified-Since
- If-None-Match
- If-Range
- Partial-Data
- Pragma
- Proxy-Authorization
- Proxy-Connection
- Transfer-Encoding

只有应用具有相应的[主机权限](#)时，网络请求 API 才会暴露相关的请求。此外，只能访问下列协议的请

请求：[http://](#)、[https://](#)、[ftp://](#)、[file://](#) 或 [chrome-extension://](#)。此外，某些请求的 URL 即使使用了以上某种协议也会被隐藏。例如，[chrome-extension://other_extension_id](#)，其中 [other_extension_id](#)不是处理该请求的应用标识符；还有 [https://www.google.com/chrome](#) 及其他（该列表并不完整）。此外来自您的应用的同步 XMLHttpRequest 将对阻塞的事件处理函数隐藏，以免产生死锁。注意，对于一些支持的协议，由于对应协议本身的性质，可以产生的事件会受到限制。例如，对于“file:”协议，只会产生 [onBeforeRequest](#)、[onResponseStarted](#)、[onCompleted](#) 和 [onErrorOccurred](#) 事件。

概念

如以下几节所述，网络请求 API 的事件使用请求标识符，当您注册事件监听器时您还可以指定过滤器以及可选的额外信息。

请求标识符

每一个请求由请求标识符标识，这一标识符在浏览器会话以及应用的上下文中保证唯一，在请求的生命周期内保持不变，可以用来匹配同一请求的其他事件。注意在 HTTP 重定向或 HTTP 认证时，多个 HTTP 请求将映射至同一个网络请求。

注册事件监听器

要为网络请求注册事件处理函数，您使用通常 [addListener\(\)](#) 函数的一种变形。除了指定回调函数，您还必须指定过滤器参数，另外您还可以指定一个可选的额外信息参数。

网络请求API的 [addListener\(\)](#) 三个参数定义如下：

```
var callback = function(details) {...};
var filter = {...};
var opt_extrInfoSpec = [...];
```

如下是监听 [onBeforeRequest](#) 事件的一个例子：

```
chrome.webRequest.onBeforeRequest.addListener(
  callback, filter, opt_extrInfoSpec);
```

每一个 [addListener\(\)](#) 调用必须传递回调函数，作为第一个参数。将向这一回调函数传递包含当前 URL 请求详情的词典，词典中的信息取决于具体事件类型以及 [opt_extrInfoSpec](#) 的内容。

如果可选的 [opt_extrInfoSpec](#) 数组包含 'blocking' 字符串（仅允许用于特定事件），回调函数将以同步方式处理。这意味着请求将阻塞，直到回调函数返回。在这一种情况下，回调函数可以返回 [webRequest.BlockingResponse](#) 对象，确定这一请求进一步的生命周期。取决于当前上下文，这一响应允许取消或重定向某个请求（[OnBeforeRequest](#)），取消请求或修改标头（[onBeforeSendHeaders](#)），或者提供认证凭据（[onAuthRequired](#)）。

[webRequest.RequestFilter](#) 类型的 [filter](#) 参数允许通过不同的方式限制为哪些请求产生事件：

URLs

[URL 匹配表达式](#)，例如 [*://www.google.com/foo*bar](#)。

类型

请求类型，例如 ["main_frame"](#)（为顶层框架加载的文档）、["sub_frame"](#)（为内嵌框架加载的文档）和 ["image"](#)（网站上的图片）。请参见 [webRequest.RequestFilter](#)。

标签页标识符

某个标签页的标识符。

窗口标识符

某个窗口的标识符。

取决于事件类型，您可以在 [opt_extrInfoSpec](#) 中指定字符串，获取有关请求的附加信息。这样是为了仅在明确请求时才提供请求数据的有关详情。

实现细节

在开发使用网络请求API的应用时，理解如下几个实现细节是很重要的：

冲突的解决

在网络请求 API 的当前实现中，如果至少一个应用要求取消请求，则请求将被取消。如果一个应用取消了一个请求，所有应用都会收到 `onErrorOccurred` 事件的通知。一次只有一个应用允许重定向请求或者修改标头。如果多于一个应用尝试修改请求，最近安装的应用具有优先级，而忽略所有其他应用。如果应用修改或重定向的要求被忽略，也不会收到通知。

缓存

百度浏览器使用两种缓存：磁盘缓存和十分快速的内存缓存。内存缓存的生命周期与渲染器进程（大致与每个标签页对应）的生命周期相关，通过内存缓存响应的请求对网络请求API不可见。如果请求处理函数更改了它的行为（例如根据哪些请求被阻止而作出的行为），简单的页面刷新不一定能够体现这一更改的行为。要确保行为更改生效，请调用 `handlerBehaviorChanged()` 来清洗内存缓存。然而不要经常调用，清洗缓存是一项十分昂贵的操作。您不需要在注册或取消注册事件处理函数后调用 `handlerBehaviorChanged()`。

时间戳

网络请求事件的 `timeStamp` 属性仅保证内部的一致性，将两个事件的时间相比较会得到它们之间正确的时间差，但是与应用内的当前时间（例如通过 `(new Date()).getTime()`）比较则可能会导致不可预料的结果。

错误处理

如果您使用无效的的参数注册事件，会引发 JavaScript 错误，事件处理程序也不会注册。如果事件处理的过程中产生错误，或者事件处理程序返回无效的 `BlockingResponse` 对象，应用的控制台中会记录错误消息，同时忽略对应请求的处理程序。

示例

如下例子演示如何阻止所有发送至 `www.evil.com` 的请求：

```
chrome.webRequest.onBeforeRequest.addListener(
  function(details) {
    return {cancel: details.url.indexOf("//www.evil.com") != -1};
  },
  {urls: ["<all_urls>"]},
  ["blocking"]);
```

由于这一函数使用了阻塞事件处理函数，您将需要在清单文件中声明 "webRequest" 以及 "webRequestBlocking" 权限。

以下例子通过另一种更高效的方式达到相同目的，因为不发送至 `www.evil.com` 的请求不必传递给应用：

```
chrome.webRequest.onBeforeRequest.addListener(
  function(details) { return {cancel: true}; },
  {urls: ["*://www.evil.com/*"]},
  ["blocking"]);
```

以下例子演示如何将所有请求中的 User-Agent 标头删除：

```
chrome.webRequest.onBeforeSendHeaders.addListener(
  function(details) {
```

```
return {details: {
  for (var i = 0; i < details.requestHeaders.length; ++i) {
    if (details.requestHeaders[i].name === 'User-Agent') {
      details.requestHeaders.splice(i, 1);
      break;
    }
  }
  return {requestHeaders: details.requestHeaders};
},
{urls: ["<all_urls>"]},
["blocking", "requestHeaders"]];
```

摘要

类型
RequestFilter
HttpHeaders
BlockingResponse
UploadData
属性
MAX_HANDLER_BEHAVIOR_CHANGED_CALLS_PER_10_MINUTES
方法
handlerBehaviorChanged – chrome.webRequest.handlerBehaviorChanged(function callback)
事件
onBeforeRequest
onBeforeSendHeaders
onSendHeaders
onHeadersReceived
onAuthRequired
onResponseStarted
onBeforeRedirect
onCompleted
onErrorOccurred

类型

RequestFilter

描述应用于网络请求事件的过滤器对象。

属性		
array of string	urls	URL 或 URL 匹配表达式列表，不匹配任何 URL 的请求会被过滤出

		去。
array of enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlhttprequest", or "other"	(可选) types	请求类型的列表，不匹配任何一种类型的请求会被过滤出去。
integer	(可选) tabId	
integer	(可选) windowId	

HttpHeaders

array of object
包含 HTTP 标头的数组，每一项标头都通过词典表示，包含 **name** 属性，以及 **value** 或 **binaryValue** 中的某一属性。

BlockingResponse

用于在 **extraInfoSpec** 参数中指定 "blocking" 的事件处理函数的返回值，允许事件处理函数修改网络请求。

属性								
boolean	(可选) cancel	如果为 true ，则取消请求。在 onBeforeRequest 事件中使用，用来阻止请求的发送。						
string	(可选) redirectUrl	仅用于 onBeforeRequest 和 onHeadersReceived 事件的返回值。如果设置了该属性，就会阻止原始请求的发送/完成，并重定向至指定的 URL。允许重定向至非 HTTP 协议的 URL，例如 data: 。重定向操作产生的重定向通常使用原来的请求方法，以下情况例外：如果在 onHeadersReceived 阶段产生了重定向，重定向请求将使用 GET 方法发出。						
HttpHeaders	(可选) requestHeaders	仅用于 onBeforeSendHeaders 事件的返回值。如果设置了这一属性，请求将改用这些标头发出。						
HttpHeaders	(可选) responseHeaders	仅用于 onHeadersReceived 事件的返回值。如果设置了这一属性，则假定服务器返回了这些响应标头。为了限制冲突数目（对于每一个请求，只有一个应用可以修改 responseHeaders ），只有当您确实需要修改标头时才应该返回 responseHeaders 。						
object	(可选) authCredentials	仅用于 onAuthRequired 事件的返回值。如果设置了这一属性，发出的请求将使用提供的凭据。 <table><tr><td>string</td><td>username</td><td></td></tr><tr><td>string</td><td>password</td><td></td></tr></table>	string	username		string	password	
string	username							
string	password							

UploadData

从 **Chrome 23** 开始支持。

包含 URL 请求中上传的数据。

属性		
any	(可选) bytes	包含数据的一份拷贝的 ArrayBuffer 。
string	(可选) file	包含文件路径与名称的字符串。

属性

20	<code>chrome.webRequest.MAX_HANDLER_BEHAVIOR_CHANGED_CALLS_PER_10_MINUTES</code>	从 Chrome 23 开始支持。 10 分钟内 <code>handlerBehaviorChanged</code> 能够被调用的次数。 <code>handlerBehaviorChar</code> 是一个昂贵的函数，不应该常调用。
----	--	--

方法

handlerBehaviorChanged

`chrome.webRequest.handlerBehaviorChanged`(function callback)

当网络请求处理函数的行为发生更改时，为了避免缓存导致的不正确处理，需要调用这一函数。这一函数调用比较昂贵，不要经常调用。

参数		
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() {...};</code>

事件

onBeforeRequest

当请求即将发生时产生。

addListener

`chrome.webRequest.onBeforeRequest.addListener`(function callback,)

参数		

function

callback

callback 参数应该是一个如下形式的函数：

function(object details) {...};

object	details	string	requestId	请求标识符。请求标识符在浏览器中唯一，所以，它们可以用来联系同一事件。	
		string	url		
		string	method	标准 HTTP 方法。	
		integer	frameId	0 表示请求发生在主框架中，正数表示的子框架标识符。如果加载了（子）文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标签页中保证唯一。	
		integer	parentFrameId	包含发送请求框架的框架（即父框架的标识符，如果不存在父框架则为 -1。	
		object	(可选) requestBody	从 Chrome 23 开始支持。	
				包含 HTTP 请求正文数据，仅在 extendedInfo 包含 "requestBody" 时才会提供。	
string	(可选) error			获取数据错误	
object	(可选) formData	如果为 POST 且正文为 UTF-8 键/值对或者 multipart/form-data 形式，返回一个字典，字典的一个键值对对应一个 FormData 对象。字典的键为 "name"，值为 "value"。字典的值为：[{ 'name': 'value', 'value': 'value' }]			
array of UploadData	(可选) raw	如果为 POST 正文，返回一个 FormData 对象的 raw 数据。			

						而不是外层框架的标识符。框架标识符在标签页中保证唯一。
				integer	parentFrameId	包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。
				integer	tabId	产生请求的标签页标识符。如果请求与标签页无关则为 -1。
				enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlhttprequest", or "other"	type	请求的资源将如何使用。
				double	timeStamp	该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。
				HttpHeaders	（可选）requestHeaders	将要与这一请求一同发送的 HTTP 请求标头。

监听器返回值

([BlockingResponse](#))

如果“extraInfoSpec”参数指定了 "blocking"，事件监听器应该返回该类型的对象。

onSendHeaders

当请求即将发送至服务器的前一刻产生（前面 onBeforeSendHeaders 事件处理函数作出的修改可以在这一事件产生时体现）。

addListener

chrome.webRequest.onSendHeaders.addListener([function](#) callback,)

参数						
function	callback	<p><i>callback</i> 参数应该是一个如下形式的函数：</p> <p>function(object details) {...};</p>				
		object	details			
				string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。
				string	url	
				string	method	标准 HTTP 方法。
				integer	frameId	0 表示请求发生在主框架

[illegible]

onHeadersReceived

当接收到 HTTP 响应标头时产生。

addListener

```
chrome.webRequest.onHeadersReceived.addListener(function callback, )
```

参数

function	callback	<p><i>callback</i> 参数应该是一个如下形式的函数：</p> <p>function(object details) {...};</p> <table><tr><td rowspan="2">object</td><td rowspan="2">details</td><td colspan="2">从 Chrome 31 开始支持。</td></tr><tr><td><table><tr><td>string</td><td>requestId</td><td>请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。</td></tr><tr><td>string</td><td>url</td><td></td></tr></table></td></tr></table>			object	details	从 Chrome 31 开始支持。		<table><tr><td>string</td><td>requestId</td><td>请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。</td></tr><tr><td>string</td><td>url</td><td></td></tr></table>	string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。	string	url	
object	details	从 Chrome 31 开始支持。													
		<table><tr><td>string</td><td>requestId</td><td>请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。</td></tr><tr><td>string</td><td>url</td><td></td></tr></table>	string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。	string	url								
string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。													
string	url														

参数

function	callback	<div><div>callback 参数应该是一个如下形式的函数：</div><div>function(object details, function callback) {...};</div></div> <table><tr><td>object</td><td>details</td><td><table><tr><td>string</td><td>requestId</td><td>请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。</td></tr><tr><td>string</td><td>url</td><td></td></tr><tr><td>string</td><td>method</td><td>标准 HTTP 方法。</td></tr><tr><td>integer</td><td>frameId</td><td>0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。</td></tr><tr><td>integer</td><td>parentFrameId</td><td>包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。</td></tr><tr><td>integer</td><td>tabId</td><td>产生请求的标签页标识符。如果请求与标签页无关则为 -1。</td></tr><tr><td>enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlHttpRequest", or "other"</td><td>type</td><td>请求的资源将如何使用。</td></tr><tr><td>double</td><td>timeStamp</td><td>该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。</td></tr><tr><td>string</td><td>scheme</td><td>认证方式，例如基本 ("basic") 或摘要式 ("digest")。</td></tr><tr><td>string</td><td>（可选）realm</td><td>服务器提供的认证域（如果有的话）。</td></tr><tr><td>object</td><td>challenger</td><td>请求认证的服务器。<table><tr><td>string</td><td>host</td><td></td></tr></table></td></tr></table></td></tr></table>	object	details	<table><tr><td>string</td><td>requestId</td><td>请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。</td></tr><tr><td>string</td><td>url</td><td></td></tr><tr><td>string</td><td>method</td><td>标准 HTTP 方法。</td></tr><tr><td>integer</td><td>frameId</td><td>0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。</td></tr><tr><td>integer</td><td>parentFrameId</td><td>包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。</td></tr><tr><td>integer</td><td>tabId</td><td>产生请求的标签页标识符。如果请求与标签页无关则为 -1。</td></tr><tr><td>enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlHttpRequest", or "other"</td><td>type</td><td>请求的资源将如何使用。</td></tr><tr><td>double</td><td>timeStamp</td><td>该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。</td></tr><tr><td>string</td><td>scheme</td><td>认证方式，例如基本 ("basic") 或摘要式 ("digest")。</td></tr><tr><td>string</td><td>（可选）realm</td><td>服务器提供的认证域（如果有的话）。</td></tr><tr><td>object</td><td>challenger</td><td>请求认证的服务器。<table><tr><td>string</td><td>host</td><td></td></tr></table></td></tr></table>	string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。	string	url		string	method	标准 HTTP 方法。	integer	frameId	0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。	integer	parentFrameId	包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。	integer	tabId	产生请求的标签页标识符。如果请求与标签页无关则为 -1。	enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlHttpRequest", or "other"	type	请求的资源将如何使用。	double	timeStamp	该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。	string	scheme	认证方式，例如基本 ("basic") 或摘要式 ("digest")。	string	（可选）realm	服务器提供的认证域（如果有的话）。	object	challenger	请求认证的服务器。 <table><tr><td>string</td><td>host</td><td></td></tr></table>	string	host	
object	details	<table><tr><td>string</td><td>requestId</td><td>请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。</td></tr><tr><td>string</td><td>url</td><td></td></tr><tr><td>string</td><td>method</td><td>标准 HTTP 方法。</td></tr><tr><td>integer</td><td>frameId</td><td>0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。</td></tr><tr><td>integer</td><td>parentFrameId</td><td>包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。</td></tr><tr><td>integer</td><td>tabId</td><td>产生请求的标签页标识符。如果请求与标签页无关则为 -1。</td></tr><tr><td>enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlHttpRequest", or "other"</td><td>type</td><td>请求的资源将如何使用。</td></tr><tr><td>double</td><td>timeStamp</td><td>该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。</td></tr><tr><td>string</td><td>scheme</td><td>认证方式，例如基本 ("basic") 或摘要式 ("digest")。</td></tr><tr><td>string</td><td>（可选）realm</td><td>服务器提供的认证域（如果有的话）。</td></tr><tr><td>object</td><td>challenger</td><td>请求认证的服务器。<table><tr><td>string</td><td>host</td><td></td></tr></table></td></tr></table>	string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。	string	url		string	method	标准 HTTP 方法。	integer	frameId	0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。	integer	parentFrameId	包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。	integer	tabId	产生请求的标签页标识符。如果请求与标签页无关则为 -1。	enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlHttpRequest", or "other"	type	请求的资源将如何使用。	double	timeStamp	该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。	string	scheme	认证方式，例如基本 ("basic") 或摘要式 ("digest")。	string	（可选）realm	服务器提供的认证域（如果有的话）。	object	challenger	请求认证的服务器。 <table><tr><td>string</td><td>host</td><td></td></tr></table>	string	host				
string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。																																							
string	url																																								
string	method	标准 HTTP 方法。																																							
integer	frameId	0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。																																							
integer	parentFrameId	包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。																																							
integer	tabId	产生请求的标签页标识符。如果请求与标签页无关则为 -1。																																							
enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlHttpRequest", or "other"	type	请求的资源将如何使用。																																							
double	timeStamp	该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。																																							
string	scheme	认证方式，例如基本 ("basic") 或摘要式 ("digest")。																																							
string	（可选）realm	服务器提供的认证域（如果有的话）。																																							
object	challenger	请求认证的服务器。 <table><tr><td>string</td><td>host</td><td></td></tr></table>	string	host																																					
string	host																																								

								integer	port		
						boolean	isProxy	如果为 Proxy-Authenticate（代理服务器认证）则为 true，否则如果为 WWW-Authenticate 则为 false。			
						HttpHeaders	（可选） responseHeaders	与响应一起接收到的 HTTP 响应标头。			
						string	statusLine	响应的 HTTP 状态行，如果是 HTTP/0.9 响应（即没有状态行的响应）则为 'HTTP/0.9 200 OK' 字符串。			
					function	（可选） callback	如果“extraInfoSpec”参数指定了 "asyncBlocking"，事件监听器应该调用该函数，提供处理事件的结果。 如果您指定了 callback 参数，它应该是一个如下形式的函数： function(BlockingResponse response) {...};				
							BlockingResponse	response	事件处理的结果：可以设置 authCredentials 属性，提供认证凭据；也可以将 cancel 属性设置为 true，取消请求并显示错误页面；或者不设置任何属性，不采取任何行动。		

监听器返回值

(**BlockingResponse**)

如果“extraInfoSpec”参数指定了 "blocking"，事件监听器应该返回该类型的对象。

onResponseStarted

当响应正文的第一个字节接收到时产生。对于 HTTP 请求，这意味着状态行及请求标头已经可用。

addListener

chrome.webRequest.onResponseStarted.addListener(function callback,)

参数									
function	callback	<i>callback</i> 参数应该是一个如下形式的函数： function(object details) { ... };							
		object	details						
				string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。			
				string	url				

当服务器产生的延迟即将进行时产生。

addListener

chrome.webRequest.onBeforeRedirect.addListener(function callback,)

参数

function	callback	<div><div>callback 参数应该是一个如下形式的函数：</div><div>function(object details) {...};</div><table><tr><th>object</th><th>details</th><td></td><td></td></tr><tr><td>string</td><td>requestId</td><td colspan="2">请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。</td></tr><tr><td>string</td><td>url</td><td colspan="2"></td></tr><tr><td>string</td><td>method</td><td colspan="2">标准 HTTP 方法。</td></tr><tr><td>integer</td><td>frameId</td><td colspan="2">0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。</td></tr><tr><td>integer</td><td>parentFrameId</td><td colspan="2">包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。</td></tr><tr><td>integer</td><td>tabId</td><td colspan="2">产生请求的标签页标识符。如果请求与标签页无关则为 -1。</td></tr><tr><td>enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlHttpRequest", or "other"</td><td>type</td><td colspan="2">请求的资源将如何使用。</td></tr><tr><td>double</td><td>timeStamp</td><td colspan="2">该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。</td></tr><tr><td>string</td><td>（可选）ip</td><td colspan="2">服务器 IP 地址，注意这可能是一个字面的 IPv6 地址。</td></tr><tr><td>boolean</td><td>fromCache</td><td colspan="2">表示响应是否从磁盘缓存中获取。</td></tr></table></div>	object	details			string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。		string	url			string	method	标准 HTTP 方法。		integer	frameId	0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。		integer	parentFrameId	包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。		integer	tabId	产生请求的标签页标识符。如果请求与标签页无关则为 -1。		enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlHttpRequest", or "other"	type	请求的资源将如何使用。		double	timeStamp	该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。		string	（可选）ip	服务器 IP 地址，注意这可能是一个字面的 IPv6 地址。		boolean	fromCache	表示响应是否从磁盘缓存中获取。	
object	details																																													
string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。																																												
string	url																																													
string	method	标准 HTTP 方法。																																												
integer	frameId	0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。																																												
integer	parentFrameId	包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。																																												
integer	tabId	产生请求的标签页标识符。如果请求与标签页无关则为 -1。																																												
enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlHttpRequest", or "other"	type	请求的资源将如何使用。																																												
double	timeStamp	该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。																																												
string	（可选）ip	服务器 IP 地址，注意这可能是一个字面的 IPv6 地址。																																												
boolean	fromCache	表示响应是否从磁盘缓存中获取。																																												

					<table><tr><td>integer</td><td>statusCode</td><td>服务器返回的标准 HTTP 状态代码。</td></tr><tr><td>string</td><td>redirectUrl</td><td>新的 URL。</td></tr><tr><td>HttpHeaders</td><td>(可选) responseHeaders</td><td>与响应一起接收到的 HTTP 响应标头。</td></tr><tr><td>string</td><td>statusLine</td><td>响应的 HTTP 状态行，如果是 HTTP/0.9 响应（即没有状态行的响应）则为 'HTTP/0.9 200 OK' 字符串。</td></tr></table>	integer	statusCode	服务器返回的标准 HTTP 状态代码。	string	redirectUrl	新的 URL。	HttpHeaders	(可选) responseHeaders	与响应一起接收到的 HTTP 响应标头。	string	statusLine	响应的 HTTP 状态行，如果是 HTTP/0.9 响应（即没有状态行的响应）则为 'HTTP/0.9 200 OK' 字符串。
integer	statusCode	服务器返回的标准 HTTP 状态代码。															
string	redirectUrl	新的 URL。															
HttpHeaders	(可选) responseHeaders	与响应一起接收到的 HTTP 响应标头。															
string	statusLine	响应的 HTTP 状态行，如果是 HTTP/0.9 响应（即没有状态行的响应）则为 'HTTP/0.9 200 OK' 字符串。															

onCompleted

当请求完成时产生。

addListener

chrome.webRequest.onCompleted.addListener(function callback,)

参数																								
function	callback	<p><i>callback</i> 参数应该是一个如下形式的函数：</p> <p>function(object details) {...};</p> <table><tr><th>object</th><th>details</th><td></td></tr><tr><td>string</td><td>requestId</td><td>请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。</td></tr><tr><td>string</td><td>url</td><td></td></tr><tr><td>string</td><td>method</td><td>标准 HTTP 方法。</td></tr><tr><td>integer</td><td>frameId</td><td>0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。</td></tr><tr><td>integer</td><td>parentFrameId</td><td>包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。</td></tr><tr><td>integer</td><td>tabId</td><td>产生请求的标签页标识符。如果请求与标签页无关则为 -1。</td></tr></table>		object	details		string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。	string	url		string	method	标准 HTTP 方法。	integer	frameId	0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（ type 为 "main_frame" 或 "sub_frame"）， frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。	integer	parentFrameId	包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。	integer	tabId	产生请求的标签页标识符。如果请求与标签页无关则为 -1。
object	details																							
string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。																						
string	url																							
string	method	标准 HTTP 方法。																						
integer	frameId	0 表示请求发生在主框架中，正数表示发出请求的子框架标识符。如果加载了（子）框架的文档（ type 为 "main_frame" 或 "sub_frame"）， frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。																						
integer	parentFrameId	包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。																						
integer	tabId	产生请求的标签页标识符。如果请求与标签页无关则为 -1。																						

			<table><tr><td>enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlhttprequest", or "other"</td><td>type</td><td>请求的资源将如何使用。</td></tr><tr><td>double</td><td>timeStamp</td><td>该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。</td></tr><tr><td>string</td><td>(可选) ip</td><td>服务器 IP 地址，注意这可能是一个字面的 IPv6 地址。</td></tr><tr><td>boolean</td><td>fromCache</td><td>表示响应是否从磁盘缓存中获取。</td></tr><tr><td>integer</td><td>statusCode</td><td>服务器返回的标准 HTTP 状态代码。</td></tr><tr><td>HttpHeaders</td><td>(可选) responseHeaders</td><td>与响应一起接收到的 HTTP 响应标头。</td></tr><tr><td>string</td><td>statusLine</td><td>响应的 HTTP 状态行，如果是 HTTP/0.9 响应（即没有状态行的响应）则为 'HTTP/0.9 200 OK' 字符串。</td></tr></table>	enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlhttprequest", or "other"	type	请求的资源将如何使用。	double	timeStamp	该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。	string	(可选) ip	服务器 IP 地址，注意这可能是一个字面的 IPv6 地址。	boolean	fromCache	表示响应是否从磁盘缓存中获取。	integer	statusCode	服务器返回的标准 HTTP 状态代码。	HttpHeaders	(可选) responseHeaders	与响应一起接收到的 HTTP 响应标头。	string	statusLine	响应的 HTTP 状态行，如果是 HTTP/0.9 响应（即没有状态行的响应）则为 'HTTP/0.9 200 OK' 字符串。
enum of "main_frame", "sub_frame", "stylesheet", "script", "image", "object", "xmlhttprequest", or "other"	type	请求的资源将如何使用。																						
double	timeStamp	该信号触发的时间，以 1970 年 1 月 1 日午夜开始所经过的毫秒数表示。																						
string	(可选) ip	服务器 IP 地址，注意这可能是一个字面的 IPv6 地址。																						
boolean	fromCache	表示响应是否从磁盘缓存中获取。																						
integer	statusCode	服务器返回的标准 HTTP 状态代码。																						
HttpHeaders	(可选) responseHeaders	与响应一起接收到的 HTTP 响应标头。																						
string	statusLine	响应的 HTTP 状态行，如果是 HTTP/0.9 响应（即没有状态行的响应）则为 'HTTP/0.9 200 OK' 字符串。																						

onErrorOccurred

当错误发生时产生。

addListener

chrome.webRequest.onErrorOccurred.addListener(function callback,)

参数																			
function	callback	<p><i>callback</i> 参数应该是一个如下形式的函数：</p> <p>function(object details) {...};</p> <table><tr><th>object</th><th>details</th><td></td></tr><tr><td>string</td><td>requestId</td><td>请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。</td></tr><tr><td>string</td><td>url</td><td></td></tr><tr><td>string</td><td>method</td><td>标准 HTTP 方法。</td></tr><tr><td>integer</td><td>frameId</td><td>0 表示请求发生在主框架中，正数表示发出请求</td></tr></table>			object	details		string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。	string	url		string	method	标准 HTTP 方法。	integer	frameId	0 表示请求发生在主框架中，正数表示发出请求
object	details																		
string	requestId	请求标识符。请求标识符在浏览器会话中保证唯一，所以，它们可以用来联系同一请求的不同事件。																	
string	url																		
string	method	标准 HTTP 方法。																	
integer	frameId	0 表示请求发生在主框架中，正数表示发出请求																	

						<p>的子框架标识符。如果加载了（子）框架的文档（type 为 "main_frame" 或 "sub_frame"），frameId 指定该框架的标识符，而不是外层框架的标识符。框架标识符在标签页中保证唯一。</p>
				integer	parentFrameId	包含发送请求框架的框架（即父框架）标识符，如果不存在父框架则为 -1。
				integer	tabId	产生请求的标签页标识符。如果请求与标签页无关则为 -1。
				enum of "main_frame" , "sub_frame" , "stylesheet" , "script" , "image" , "object" , "xmlhttprequest" , or "other"	type	请求的资源将如何使用。
				double	timeStamp	该信号触发的时间，以 1970 年 1 月 1 日 午夜开始所经过的毫秒数表示。
				string	（可选） ip	服务器 IP 地址，注意这可能是一个字面的 IPv6 地址。
				boolean	fromCache	表示响应是否从磁盘缓存中获取。
				string	error	错误描述。这一字符串 <i>不保证</i> 在不同的版本间保持向后兼容，您决不能分析它的内容并由此作出处理。