# WebViewSDK接口详解

目录

[WebViewSDK接口详解 1](#_Toc514688797)

[1标准方法 2](#_Toc514688798)

[1.1 ClientCertRequest 2](#_Toc514688799)

[1.2 ConsoleMessage 3](#_Toc514688800)

[1.3 CookieManager 4](#_Toc514688801)

[1.4 CookieSyncManager 8](#_Toc514688802)

[1.5 DownLoadInterface 11](#_Toc514688803)

[1.6 GeolocationPermissions 11](#_Toc514688804)

[1.6.1 GeolocationPermissions.Callback 13](#_Toc514688805)

[1.7 HttpAuthHandler 13](#_Toc514688806)

[1.8 JavascriptInterface 14](#_Toc514688807)

[1.9 JsPromptResult 14](#_Toc514688808)

[1.10 JsResult 15](#_Toc514688809)

[1.11 MimeTypeMap 15](#_Toc514688810)

[1.12 PermissionRequest 16](#_Toc514688811)

[1.13 SslErrorHandler 17](#_Toc514688812)

[1.14 URLUtil 18](#_Toc514688813)

[1.15 ValueCallback 21](#_Toc514688814)

[1.16 WebBackForwardList 22](#_Toc514688815)

[1.17 WebChromeClient 23](#_Toc514688816)

[1.17.1 WebChromeClient.CustomViewCallback 30](#_Toc514688817)

[1.18 WebHistoryItem 30](#_Toc514688818)

[1.19 WebIconDatabase 32](#_Toc514688819)

[1.19.1 WebIconDatabase.IconListener 33](#_Toc514688820)

[1.20 WebResourceRequest 34](#_Toc514688821)

[1.21 WebResourceResponse 35](#_Toc514688822)

[1.22 WebSettings 37](#_Toc514688823)

[1.23 WebStorage 59](#_Toc514688824)

[1.23.1 WebStorage.QuotaUpdater 60](#_Toc514688825)

[1.24 webview 61](#_Toc514688826)

[1.24.1 WebView.FindListener 98](#_Toc514688827)

[1.24.2 WebView.PictureListener 98](#_Toc514688828)

[1.25 WebViewClient 99](#_Toc514688829)

[1.26 WebViewDatabase 108](#_Toc514688830)

[2 扩展WebView方法 111](#_Toc514688831)

[2.1 WebViewEx 111](#_Toc514688832)

[2.2 WebSettingEx 115](#_Toc514688833)

[2.3 ExtensionClient 118](#_Toc514688834)

[2.4 WebVideoViewClient 121](#_Toc514688835)

# 1标准方法

## 1.1 ClientCertRequest

* public abstract class ClientCertRequest

该类所有方法应该在UI线程上调用。用户收到此类的一个实例作为[onReceivedClientCertRequest(WebView, ClientCertRequest)](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#onReceivedClientCertRequest(android.webkit.WebView,%20android.webkit.ClientCertRequest)) 的参数。该请求包含选择客户端证书的参数，例如请求证书的主机名和端口号、可接受的密钥类型和主体。用户应该调用其中一个类方法来指示如何处理客户端证书请求。WebView 使用[proceed(PrivateKey, X509Certificate[])](https://developer.android.google.cn/reference/android/webkit/ClientCertRequest.html#proceed(java.security.PrivateKey,%20java.security.cert.X509Certificate[]))和[cancel()](https://developer.android.google.cn/reference/android/webkit/ClientCertRequest.html#cancel())处理同一主机/端口对的客户端证书请求。用户可以使用[clearClientCertPreferences(Runnable)](https://developer.android.google.cn/reference/android/webkit/WebView.html#clearClientCertPreferences(java.lang.Runnable))清除缓存的数据 。

**Public constructor**

* ClientCertRequest(level 21)

ClientCertRequest()

**public**

* cancle(level 21)

void cancel( )

取消此请求。记住用户的选择并将其用于未来的请求。

* getHost(level 21)

[String](https://developer.android.google.cn/reference/java/lang/String.html) getHost( )

返回请求证书的服务器的主机名。

返回：

[String](https://developer.android.google.cn/reference/java/lang/String.html)

* getKeyTypes(level 21)

[String []](https://developer.android.google.cn/reference/java/lang/String.html) getKeyTypes( )

返回非对称密钥的可接受类型。

返回：

[String[]](https://developer.android.google.cn/reference/java/lang/String.html) 这个值可能是null。

* getPort(level 21)

int getPort( )

返回请求证书的服务器的端口号。

返回：

int 端口号

* getPrincipals(level 21)

[Principal []](https://developer.android.google.cn/reference/java/security/Principal.html) getPrincipals( )

返回匹配到私钥的证书颁发者。

返回：

[Principal[]](https://developer.android.google.cn/reference/java/security/Principal.html) 这个值可能是null。

* ignore(level 21)

void ignore( )

忽略现在的请求。不记住用户的选择。

* proceed(level 21)

void proceed([PrivateKey](https://developer.android.google.cn/reference/java/security/PrivateKey.html) privateKey，[X509Certificate[]](https://developer.android.com/reference/java/security/cert/X509Certificate.html) chain)

继续使用指定的私钥和客户端证书链。记住用户的选择并将其用于未来的请求。

参数：

privateKey 指定的私钥

chain X509Certificate

## 1.2 ConsoleMessage

* public class ConsoleMessage

公共类，表示来自WebCore的JavaScript控制台消息。这可能是通过调用其中一个console日志记录(例如console.log('...'))或页面上的JavaScript错误发出的。要接收这些消息的通知，请重写该[onConsoleMessage(ConsoleMessage)](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.html#onConsoleMessage(android.webkit.ConsoleMessage))方法。

**Public constructor**

* ConsoleMessage(level 8)

ConsoleMessage(String message， String sourceId， int lineNumber， [ConsoleMessage.MessageLevel](https://developer.android.google.cn/reference/android/webkit/ConsoleMessage.MessageLevel.html) msgLevel)

参数：

message String

sourceId String

lineNumber int

msgLevel ConsoleMessage.MessageLevel

**public**

* lineNumber(level 8)

int lineNumber( )

返回：

int

* message(level 8)

String message( )

返回：

[String](https://developer.android.google.cn/reference/java/lang/String.html)

* messageLevel(level 8)

[ConsoleMessage.MessageLevel](https://developer.android.google.cn/reference/android/webkit/ConsoleMessage.MessageLevel.html) messageLevel( )

返回：

[ConsoleMessage.MessageLevel](https://developer.android.google.cn/reference/android/webkit/ConsoleMessage.MessageLevel.html)

* SourceID(level 8)

[String](https://developer.android.google.cn/reference/java/lang/String.html) sourceId( )

返回：

[String](https://developer.android.google.cn/reference/java/lang/String.html)

## 1.3 **CookieManager**

* public abstract class CookieManager

管理应用程序中[WebView](https://developer.android.com/reference/android/webkit/WebView.html)实例使用的Cookie 。Cookies根据RFC2109进行操作。

**Public constructor**

* CookieManager(level 22)

[CookieManager](https://developer.android.com/reference/android/webkit/CookieManager.html#CookieManager())( )

**Public**

* acceptCookie(level 1)

boolean acceptCookie( )

获取应用程序的[WebView](https://developer.android.com/reference/android/webkit/WebView.html)实例是否发送和接受cookie。

返回：

boolean 如果[WebView](https://developer.android.com/reference/android/webkit/WebView.html)实例发送并接受cookie，返回true。

* acceptThirdPartyCookies(level 21)

boolean acceptThirdPartyCookies([WebView](https://developer.android.com/reference/android/webkit/WebView.html) webview)

获取[WebView](https://developer.android.com/reference/android/webkit/WebView.html)是否允许设置第三方Cookie。

参数：

webview WebView：[WebView](https://developer.android.com/reference/android/webkit/WebView.html)实例

返回：

boolean 如果[WebView](https://developer.android.com/reference/android/webkit/WebView.html)允许接收第三方cookies，返回true

* allowFileSchemeCookies(level 12)

boolean allowFileSchemeCookies( )

获取应用程序的[WebView](https://developer.android.com/reference/android/webkit/WebView.html)实例是否发送和接收文件方案URL的Cookie。

返回：

boolean 如果[WebView](https://developer.android.com/reference/android/webkit/WebView.html)实例发送并接收文件方案URL的Cookie，返回true

* flush(level 21)

void flush( )

确保通过getCookie 获取的可访问的所有Cookie都写入永久存储。直到它完成并且可以执行I / O，其他操作才可以执行。

* getCookie(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getCookie([String](https://developer.android.com/reference/java/lang/String.html) url)

获取给定URL的cookie。

参数：

url String：请求Cookie的URL

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 将获取cookie作为字符串

* getInstance(level 1)

[CookieManager](https://developer.android.com/reference/android/webkit/CookieManager.html) getInstance( )

获取单例CookieManager实例。

返回：

[CookieManager](https://developer.android.com/reference/android/webkit/CookieManager.html) 单例CookieManager实例

* hasCookies(level 1)

boolean hasCookies( )

获取是否存储了Cookie。

返回：

boolean 如果有存储的cookie，返回 true

* removeAllCookie(level 1)

void removeAllCookie( )

**这种方法在API级别被废弃21**  
使用[**removeAllCookies(ValueCallback)**](https://developer.android.com/reference/android/webkit/CookieManager.html#removeAllCookies(android.webkit.ValueCallback%3Cjava.lang.Boolean%3E))来代替。

删除所有Cookie。

* removeAllCookies(level 21)

void removeAllCookies([ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) < [Boolean](https://developer.android.com/reference/java/lang/Boolean.html) > callback)

删除所有Cookie。这种方法是异步的。如果提供了[ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) ， 那么操作完成后将在当前线程的[Looper](https://developer.android.com/reference/android/os/Looper.html)上调用[onReceiveValue()](https://developer.android.com/reference/android/webkit/ValueCallback.html#onReceiveValue(T))。提供给回调的值指示是否删除了任何Cookie。如果您不需要知道操作何时完成或是否删除了任何Cookie，callback可以传递null，在这种情况下，可以安全地从没有Looper的线程调用该方法。

参数：

callback ValueCallback：在Cookie被移除时执行的回调，这个值可能是null。

* removeExpiredCookie(level 1)

void removeExpiredCookie( )

**此方法在API级别21中已弃用**  
WebView会自动处理过期的Cookie。

删除所有过期的Cookie。

* removeSessionCookie(level 1)

void removeSessionCookie( )

**这种方法在API级别被废弃21**  
使用[**removeSessionCookies(ValueCallback)**](https://developer.android.com/reference/android/webkit/CookieManager.html#removeSessionCookies(android.webkit.ValueCallback%3Cjava.lang.Boolean%3E))来代替。

删除所有会话Cookie，即没有过期日期的Cookie。

* removeSessionCookies(level 21)

void removeSessionCookies([ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) < [Boolean](https://developer.android.com/reference/java/lang/Boolean.html) > callback)

删除所有会话Cookie，即没有过期的Cookie。

这种方法是异步的。如果提供[ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) ， 那么操作完成后将在当前线程的[Looper](https://developer.android.com/reference/android/os/Looper.html)上调用[onReceiveValue()](https://developer.android.com/reference/android/webkit/ValueCallback.html#onReceiveValue(T))。提供给回调的值指示是否删除了任何Cookie。如果您不需要知道操作何时完成或是否删除了任何Cookie，callback可以传递null，在这种情况下，可以安全地从没有Looper的线程调用该方法。

参数：

callback ValueCallback：会话cookie已被删除时执行的回调，这个值可能是null。

* setAcceptCookie(level 1)

void setAcceptCookie(boolean accept)

设置应用程序的[WebView](https://developer.android.com/reference/android/webkit/WebView.html)实例是否应该发送和接受cookie。默认情况下，这被设置为true，此时WebView接受cookie。

当accept为true时， [setAcceptThirdPartyCookies](https://developer.android.com/reference/android/webkit/CookieManager.html#setAcceptThirdPartyCookies(android.webkit.WebView,%20boolean))和 [setAcceptFileSchemeCookies](https://developer.android.com/reference/android/webkit/CookieManager.html#setAcceptFileSchemeCookies(boolean)) 可以用来控制特定类型的cookie。

参数：

Accept boolean：[WebView](https://developer.android.com/reference/android/webkit/WebView.html)实例是否应该发送和接受cookie

* setAcceptFileSchemeCookies(level 12)

void setAcceptFileSchemeCookies(boolean accept)

设置应用程序的[WebView](https://developer.android.com/reference/android/webkit/WebView.html)实例是否应发送并接受文件系统URL的Cookie。文件系统URL使用cookie可能不安全，默认是关闭。除非您确定不会无意共享Cookie数据，否则请勿使用此功能。

参数：

accept boolean

* setAcceptThirdPartyCookies(level 21)

void setAcceptThirdPartyCookies([WebView](https://developer.android.com/reference/android/webkit/WebView.html) webview，boolean accept)

设置[WebView](https://developer.android.com/reference/android/webkit/WebView.html)是否允许设置第三方Cookie。并且可以在不同的WebView实例上有不同的设置。[KITKAT](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#KITKAT)或更低版本默认允许第三方Cookie。 [LOLLIPOP](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#LOLLIPOP)或更高版本默认禁止使用第三方Cookie。

参数：

webview WebView：[WebView](https://developer.android.com/reference/android/webkit/WebView.html)设置cookie的实例

accept boolean：该[WebView](https://developer.android.com/reference/android/webkit/WebView.html)实例是否应接受第三方Cookie

* setCookie方法(level 21)

void setCookie([String](https://developer.android.com/reference/java/lang/String.html) url， [String](https://developer.android.com/reference/java/lang/String.html) value， [ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) < [Boolean](https://developer.android.com/reference/java/lang/Boolean.html) > callback)

为给定的URL设置一个cookie。任何具有相同主机，路径和名称的现有cookie都将被替换为新的cookie。如果设置的cookie过期了，这个cookie将被忽略。

这种方法是异步的。如果提供了[ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html)， 操作完成后将在当前线程上的[Looper](https://developer.android.com/reference/android/os/Looper.html)调用[onReceiveValue()](https://developer.android.com/reference/android/webkit/ValueCallback.html#onReceiveValue(T))。提供给回调的值指示cookie是否设置成功。如果您不需要知道操作何时完成或是否设置了任何Cookie，callback可以传递null，在这种情况下，可以安全地从没有Looper的线程调用该方法。

参数：

url String：要为其设置Cookie的URL

value String：将Cookie作为字符串使用

callback ValueCallback：设置cookie时要执行的回调，这个值可能是null。

* setCookie方法(level 1)

void setCookie([String](https://developer.android.com/reference/java/lang/String.html) url， [String](https://developer.android.com/reference/java/lang/String.html) value)

为给定的URL设置一个cookie。任何具有相同主机，路径和名称的现有cookie都将被替换为新的cookie。如果设置的cookie过期，它将被忽略。

参数：

url String：要为其设置Cookie的URL

value String：将Cookie作为字符串使用

**protected**

* clone(level 1)

[Object](https://developer.android.com/reference/java/lang/Object.html) clone( )

创建并返回此对象的副本。“复制”的确切含义可能取决于对象的类别。一般意图是，对于任何对象x，表达式为：

x.clone( )！= x将是true，

并且表达式：

x.clone( ).getClass( )== x.getClass( )将是true，

但这些不是绝对的要求。虽然通常情况是：

x.clone( ).equals(x)

将是true，这不是绝对的要求。

按照惯例，返回的对象应该通过调用super.clone获得 。如果一个类和它的所有超类(除了 Object)遵守这个规定，那么 x.clone().getClass() == x.getClass()。

按照惯例，这个方法返回的对象应该独立于这个对象(被克隆)。为了实现这种独立性，可能有必要在返回它之前修改返回对象的一个​​或多个字段。通常，这意味着复制包含被克隆对象的内部“深层结构”的任何可变对象，并将这些对象的引用替换为这些副本的引用。如果一个类仅包含原始字段或对不可变对象的引用，那么通常情况下不需要修改返回的对象中的字段。

Object类 的clone方法执行特定的克隆操作。首先，如果这个对象的类没有实现接口Cloneable，则抛出CloneNotSupportedException。请注意，所有数组都被认为是实现了接口Cloneable并且clone数组类型的方法T[] 的返回类型是T[]，其中T是任何引用或原始类型。否则，此方法创建该对象的类的新实例，并使用该对象的相应字段的内容来初始化其所有字段，就像通过赋值一样; 这些字段的内容本身并不克隆。因此，此方法执行此对象的“浅拷贝”，而不是“深拷贝”操作。

这个类Object本身并没有实现这个接口 Cloneable，所以clone在一个类的对象上调用该方法Object将导致在运行时抛出一个异常。

返回：

[Object](https://developer.android.com/reference/java/lang/Object.html) 这个实例的一个克隆。

抛出：

CloneNotSupportedException

## 1.4 CookieSyncManager

* public final class CookieSyncManager

**此类在API级别21中已弃用**  
WebView现在会根据需要自动同步Cookie。您不再需要创建或使用CookieSyncManager。要手动强制同步，您可以使用CookieManager方法[**flush()**](https://developer.android.google.cn/reference/android/webkit/CookieManager.html#flush())，该方法是同步替换方法[**sync()**](https://developer.android.google.cn/reference/android/webkit/CookieSyncManager.html#sync())。

CookieSyncManager用于在RAM和永久存储之间同步浏览器Cookie。为了获得最佳性能，浏览器cookie被保存在RAM中。一个单独的线程在一个定时器的驱动下保存cookie。

要使用CookieSyncManager，主应用程序必须在应用程序启动时调用以下内容：

CookieSyncManager .createInstance (context)

要设置同步，应用程序必须在Activity.onResume( )调用

CookieSyncManager .getInstance( ).startSync ( )

在Activity.onPause( )中调用

 CookieSyncManager .getInstance( ).stopSync ( )

要获得即时同步而不是等待定时器触发，主机可以调用

CookieSyncManager .getInstance( ).sync ( )

同步时间间隔为5分钟，因此无论如何您都需要手动强制同步，例如[onPageFinished(WebView, String)](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#onPageFinished(android.webkit.WebView,%20java.lang.String))。请注意，即使sync( )也是异步发生的，所以不要在活动关闭的时候这样做。

**public**

* createInstance(level 1)

[CookieSyncManager](https://developer.android.google.cn/reference/android/webkit/CookieSyncManager.html) createInstance(Context context)

在上下文中创建一个单例CookieSyncManager实例

返回：

[CookieSyncManager](https://developer.android.google.cn/reference/android/webkit/CookieSyncManager.html) CookieSyncManager

* getInstance(level 1)

[CookieSyncManager](https://developer.android.google.cn/reference/android/webkit/CookieSyncManager.html) getInstance( )

获取[CookieSyncManager](https://developer.android.google.cn/reference/android/webkit/CookieSyncManager.html)的单例实例，如果之前没有调用[createInstance(Context)](https://developer.android.google.cn/reference/android/webkit/CookieSyncManager.html#createInstance(android.content.Context))，则会抛出IllegalStateException异常 。

返回：

[CookieSyncManager](https://developer.android.google.cn/reference/android/webkit/CookieSyncManager.html) CookieSyncManager

* resetSync(level 1)

void resetSync( )

**此方法在API级别27中已弃用。**  
由于WebView会自动同步Cookie，因此不再需要调用resetSync。

重置同步管理器的计时器。

* run(level 1)

void run( )

当对象使用Runnable实现接口来创建线程时，启动线程会导致run在该单独执行的线程中调用该对象的 方法。

* startSync(level 1)

void startSync( )

**此方法在API级别27中已弃用。**  
由于WebView会自动同步Cookie，因此不再需要调用startSync。

请求同步管理器开始同步。

* stopSync(level 1)

void stopSync( )

**此方法在API级别27中已被弃用。**  
调用stopSync不再有用，因为WebView会自动同步cookie。

请求同步管理器停止同步。删除队列中的任何SYNC\_MESSAGE以中断同步循环

* sync(level 1)

void sync( )

**这种方法在API级别被废弃27.**  
使用[**flush()**](https://developer.android.google.cn/reference/android/webkit/CookieManager.html#flush())代替。

强制同步管理器立即进行同步

**Protected**

* clone(level 1)

[Object](https://developer.android.google.cn/reference/java/lang/Object.html) clone( )

创建并返回此对象的副本。“clone”的确切含义可能取决于对象的类别。一般意图是，对于任何对象x，表达式为：

x.clone( )！= x将是true，

并且表达式：

x.clone( ).getClass( )== x.getClass( )将是true，

但这些不是绝对的要求。虽然通常情况是：

x.clone( ).equals(x)

将是true，这不是绝对的要求。

按照惯例，这个方法返回的对象应该独立于这个对象(被克隆)。为了实现这种独立性，可能有必要在返回它之前修改返回对象的一个​​或多个字段。通常，这意味着复制包含被克隆对象的内部“深层结构”的任何可变对象，并将这些对象的引用替换为这些副本的引用。如果一个类仅包含原始字段或对不可变对象的引用，那么通常情况下不需要修改返回的对象中的字段。

Object类 的clone方法执行特定的克隆操作。首先，如果这个对象的类没有实现接口Cloneable，则抛出CloneNotSupportedException。请注意，所有数组都被认为是实现了接口Cloneable并且clone数组类型的方法T[] 的返回类型是T[]，其中T是任何引用或原始类型。否则，此方法创建该对象的类的新实例，并使用该对象的相应字段的内容来初始化其所有字段，就像通过赋值一样; 这些字段的内容本身并不克隆。因此，此方法执行此对象的“浅拷贝”，而不是“深拷贝”操作。

这个类Object本身并没有实现这个接口 Cloneable，所以clone在一个类的对象上调用该方法Object将导致在运行时抛出一个异常。

返回：

[Object](https://developer.android.google.cn/reference/java/lang/Object.html) 这个实例的一个克隆

抛出

[CloneNotSupportedException](https://developer.android.google.cn/reference/java/lang/CloneNotSupportedException.html)

* onSyncInit(level 1)

void onSyncInit( )

* syncFromRamToFlash(level 1)

void syncFromRamToFlash( )

**这种方法在API级别被废弃21**  
使用[**flush()**](https://developer.android.google.cn/reference/android/webkit/CookieManager.html#flush())来代替。

## 1.5 DownLoadInterface

* public interface DownloadListener

**public**

* onDownloadStart(level 1)

abstract void onDownloadStart([String](https://developer.android.com/reference/java/lang/String.html) url， [String](https://developer.android.com/reference/java/lang/String.html) userAgent，

[String](https://developer.android.com/reference/java/lang/String.html) contentDisposition，

[String](https://developer.android.com/reference/java/lang/String.html) mimetype，long contentLength)

通知应用程序下载文件。该方法是一个虚方法。

参数：

url String：应该下载的内容的完整网址

userAgent String：用于下载的用户代理。

contentDisposition String：内容处置http头，如果存在。

mimetype String：服务器报告的内容的MIME类型

contentLength long：服务器报告的文件大小

## 1.6 GeolocationPermissions

* public class GeolocationPermissions

该类用于管理WebView的Geolocation JavaScript API的权限。地理位置权限由URI的主机，方案和端口组成。为了让网站内容使用Geolocation API，必须为该内容的来源授予权限。该类存储地理位置权限。来源的许可状态可以被允许或被拒绝。这个类使用字符串来表示一个来源。当来源尝试使用地理定位API时，但是当前没有为该来源设置许可状态，接下来会调用[WebChromeClient.onGeolocationPermissionsShowPrompt()](https://developer.android.com/reference/android/webkit/WebChromeClient.html#onGeolocationPermissionsShowPrompt(java.lang.String,%20android.webkit.GeolocationPermissions.Callback)) 为该来源设置权限状态。该类的方法可以随时用于修改和询问存储的地理位置权限。

**public**

* allow(level 7)

void allow([String](https://developer.android.com/reference/java/lang/String.html) origin)

允许指定的来源使用地理定位API。

参数：

origin String：为true时，允许使用

* clear(level 7)

void clear([String](https://developer.android.com/reference/java/lang/String.html) origin)

清除指定来源的地理位置权限状态。

参数：

origin String：清除指定地理位置权限

* clearAll(level 7)

void clearAll( )

清除所有来源的地理位置权限状态。

* getAllowed(level 7)

void getAllowed([String](https://developer.android.com/reference/java/lang/String.html) origin， [ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) < [Boolean](https://developer.android.com/reference/java/lang/Boolean.html) > callback)

获取指定来源的地理位置权限状态。

参数：

origin String：请求地理位置权限的来源

callback ValueCallback：一个[ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html)接收这个请求的结果。该对象的 [onReceiveValue()](https://developer.android.com/reference/android/webkit/ValueCallback.html#onReceiveValue(T)) 方法将被一个布尔值异步调用，该布尔值指示源可以使用Geolocation API。

* getInstance(level 7)

[GeolocationPermissions](https://developer.android.com/reference/android/webkit/GeolocationPermissions.html) getInstance()

获取此类的单例实例。在应用程序实例化[WebView](https://developer.android.com/reference/android/webkit/WebView.html)实例之前，不能调用此方法。

返回：

[GeolocationPermissions](https://developer.android.com/reference/android/webkit/GeolocationPermissions.html) [GeolocationPermissions](https://developer.android.com/reference/android/webkit/GeolocationPermissions.html)的单例实例

* getOrigins(level 7)

void getOrigins([ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) < [Set](https://developer.android.com/reference/java/util/Set.html) < [String](https://developer.android.com/reference/java/lang/String.html) >> callback)

获取存储地理位置权限的原始集合。

参数：

callback ValueCallback：一个[ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html)接收这个请求的结果。该对象的 [onReceiveValue()](https://developer.android.com/reference/android/webkit/ValueCallback.html#onReceiveValue(T)) 方法将与包含存储地理位置权限的来源的一组字符串异步调用。

1.6.1 GeolocationPermissions.Callback

* public static interface GeolocationPermissions.Callback

**public**

* Invoke(level 5)

abstract void invoke(origin，boolean retain)

为提供的来源设置地理位置权限状态，此方法为虚方法。

参数：

origin String：设置权限的来源

allow boolean：是否允许来源使用地理定位API

retain boolean：是否应该在WebView当前显示的页面的生命周期之外保留权限

## 1.7 HttpAuthHandler

* public class HttpAuthHandler

代表HTTP身份验证的请求。这个类的实例由WebView创建并传递给它 [onReceivedHttpAuthRequest(WebView, HttpAuthHandler, String, String)](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#onReceivedHttpAuthRequest(android.webkit.WebView,%20android.webkit.HttpAuthHandler,%20java.lang.String,%20java.lang.String))。应用程序必须调用要么[proceed(String, String)](https://developer.android.google.cn/reference/android/webkit/HttpAuthHandler.html#proceed(java.lang.String,%20java.lang.String))或[cancel()](https://developer.android.google.cn/reference/android/webkit/HttpAuthHandler.html#cancel())设置的WebView请求响应。

**public**

* Cancle(level 1)

void cancel( )

WebView取消认证请求。

* proceed(level 1)

proceed(String username， String password)

指示WebView继续使用给定的凭据进行身份验证。使用此方法的凭证可以从WebView的商店中使用[getHttpAuthUsernamePassword(String, String)](https://developer.android.google.cn/reference/android/webkit/WebView.html#getHttpAuthUsernamePassword(java.lang.String,%20java.lang.String))。

参数：

username String

password String

* useHttpAuthUsernamePassword(level 1)

boolean useHttpAuthUsernamePassword( )

获取为当前主机([onReceivedHttpAuthRequest(WebView, HttpAuthHandler, String, String)](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#onReceivedHttpAuthRequest(android.webkit.WebView,%20android.webkit.HttpAuthHandler,%20java.lang.String,%20java.lang.String))即被调用的主机)存储的凭证是否适合使用。如果证书先前已被服务器拒绝当前请求，则证书不适用。

返回：

boolean 证书是否适合使用

**相关函数：**

* [getHttpAuthUsernamePassword(String, String)](https://developer.android.google.cn/reference/android/webkit/WebView.html#getHttpAuthUsernamePassword(java.lang.String,%20java.lang.String))

## 1.8 JavascriptInterface

* public abstract @interface JavascriptInterface

允许将方法暴露给JavaScript的注释。从API级别 [JELLY\_BEAN\_MR1](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN_MR1)和以上版本开始使用，只有明确标有此注释的方法才可用于JavaScript代码。查看 [addJavascriptInterface(Object, String)](https://developer.android.com/reference/android/webkit/WebView.html#addJavascriptInterface(java.lang.Object,%20java.lang.String))获取更多关于它的信息。

## 1.9 JsPromptResult

* public class JsPromptResult

用于处理JavaScript提示请求的公共类。WebChromeClient将以[onJsPrompt(WebView, String, String, String, JsPromptResult)](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.html#onJsPrompt(android.webkit.WebView,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20android.webkit.JsPromptResult))JsPromptResult实例作为参数接收调用。该参数用于将此用户对话框提示的结果返回给WebView实例。客户端可以调用cancel( )来取消对话框，或者使用用户的输入confirm( )以确认对话框。

**public**

* confirm(level 1)

void confirm([String](https://developer.android.google.cn/reference/java/lang/String.html) result)

处理来自用户的响应。

参数:

result String

## 1.10 JsResult

* public class JsResult

这个类的一个实例在各种[WebChromeClient](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.html)通知中作为参数传递。该对象用作底层JavaScript源请求的句柄，并为客户端提供了一种办法来指示是否应该继续此操作。

**public**

* Cancle(level 1)

void cancel( )

处理用户取消对话框。

* confirm(level 1)

void confirm( )

处理来自用户的确认响应。

## 1.11 MimeTypeMap

* public class MimeTypeMap

MIME类型到文件扩展名的双向映射。另见[guessContentTypeFromName(String)](https://developer.android.google.cn/reference/java/net/URLConnection.html#guessContentTypeFromName(java.lang.String)) 和[guessContentTypeFromStream(InputStream)](https://developer.android.google.cn/reference/java/net/URLConnection.html#guessContentTypeFromStream(java.io.InputStream))。这个类和URLConnection共享相同的MIME类型的数据库。

**public**

* getExtensionFromMimeType(level 1)

[String](https://developer.android.google.cn/reference/java/lang/String.html) getExtensionFromMimeType([String](https://developer.android.google.cn/reference/java/lang/String.html) mimeType)

返回给定MIME类型的注册扩展。请注意，某些MIME类型映射到多个扩展名。此调用将返回给定MIME类型的最常用扩展名。

参数：

mimeType String：MIME类型(即文本/纯文本)

返回：

[String](https://developer.android.google.cn/reference/java/lang/String.html) 给定的MIME类型的扩展名， 可以为null。

* getFileExtensionFromUrl(level 1)

[String](https://developer.android.google.cn/reference/java/lang/String.html) getFileExtensionFromUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

如果没有扩展名，则返回文件扩展名或空字符串。此方法是一种获取url扩展的便捷方法。

返回：

[String](https://developer.android.google.cn/reference/java/lang/String.html) 给定网址的文件扩展名。

* getMimeTypeFromExtension(level 1)

[String](https://developer.android.google.cn/reference/java/lang/String.html) getMimeTypeFromExtension([String](https://developer.android.google.cn/reference/java/lang/String.html) extension)

返回给定扩展的MIME类型。

参数：

extension String：不带前导'。'的文件扩展名

* getSingleton(level 1)

[MimeTypeMap](https://developer.android.google.cn/reference/android/webkit/MimeTypeMap.html) getSingleton( )

获取MimeTypeMap的单例实例。

返回：

[MimeTypeMap](https://developer.android.google.cn/reference/android/webkit/MimeTypeMap.html) MIME类型映射的单例实例。

* hasExtension(level 1)

boolean hasExtension(String extension)

如果给定的扩展有注册MIME类型，返回true。

参数：

extension String：不带前导'.'的文件扩展名

返回:

boolean

* hasMimeType(level 1)

boolean hasMimeType([String](https://developer.android.google.cn/reference/java/lang/String.html) mimeType)

返回true如果给定的MIME类型在地图上的条目。

参数：

mimeType String：MIME类型(即文本/纯文本)

返回：

boolean 如果在map中有一个mimeType条目，返回true 。

## 1.12 PermissionRequest

* public abstract class PermissionRequest

该类定义了一个权限请求，并在Web内容请求访问受保护资源时使用。通过[onPermissionRequest(PermissionRequest)](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.html#onPermissionRequest(android.webkit.PermissionRequest))和 [onPermissionRequestCanceled(PermissionRequest)](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.html#onPermissionRequestCanceled(android.webkit.PermissionRequest))传递相关事件。无论是[grant()](https://developer.android.google.cn/reference/android/webkit/PermissionRequest.html#grant(java.lang.String[]))或者[deny()](https://developer.android.google.cn/reference/android/webkit/PermissionRequest.html#deny())必须在UI线程中调用。即使在较旧的Android版本上运行时，将来版本的WebView中可能会请求未在此处定义的受保护资源的新名称。为避免无意中授予对新权限的请求，您应该传递您想授予的特定权限[grant()](https://developer.android.google.cn/reference/android/webkit/PermissionRequest.html#grant(java.lang.String[]))，并避免编写如下示例所示的代码：

permissionRequest.grant(permissionRequest.getResources( ))//这是错误的！

有关新受保护资源的信息，请参阅WebView的发行说明。

**Public constructor**

* PermissionRequest(level 21)

PermissionRequest( )

**public**

* deny(level 21)

void deny( )

调用此方法来拒绝请求。

* getOrigin(level 21)

[Uri](https://developer.android.google.cn/reference/android/net/Uri.html) getOrigin( )

调用此方法以获取试图访问受限资源的网页的来源。

返回：

Uri 试图访问受限资源的网页内容的来源。

* getResources(level 21)

[String []](https://developer.android.google.cn/reference/java/lang/String.html) getResources( )

调用此方法获取网页尝试访问的资源。

返回：

[String[]](https://developer.android.google.cn/reference/java/lang/String.html) Web内容想要访问的资源数组。

* grant(level 21)

void grant([String []](https://developer.android.google.cn/reference/java/lang/String.html) resources)

调用此方法为origin授予访问给定资源的权限。授予的权限仅对此WebView有效。

参数：

resources String：准予按来源访问的资源，要授予请求，返回的请求资源[getResources()](https://developer.android.google.cn/reference/android/webkit/PermissionRequest.html#getResources()) 必须是等于或授权资源的子集。此参数旨在避免授予意外权限，特别是在Web内容请求新资源时。

## 1.13 SslErrorHandler

* public class SslErrorHandler

处理SSL错误的请求。这个类的实例由WebView创建并传递给 [onReceivedSslError(WebView, SslErrorHandler, SslError)](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#onReceivedSslError(android.webkit.WebView,%20android.webkit.SslErrorHandler,%20android.net.http.SslError))。应用程序必须调用[proceed()](https://developer.android.google.cn/reference/android/webkit/SslErrorHandler.html#proceed())或[cancel()](https://developer.android.google.cn/reference/android/webkit/SslErrorHandler.html#cancel())响应WebView请求。

**public**

* cancle(level 1)

void cancel( )

取消此请求和有错误的WebView的挂起的请求。

* proceed(level 1)

void proceed( )

继续使用SSL证书。

## 1.14 URLUtil

* public final class URLUtil

**Public constructor**

* URLUtil(level 1)

URLUtil( )

**public**

* composeSearchUrl(level 1)

[String](https://developer.android.google.cn/reference/java/lang/String.html) composeSearchUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) inQuery，

[String](https://developer.android.google.cn/reference/java/lang/String.html) template， [String](https://developer.android.google.cn/reference/java/lang/String.html) queryPlaceHolder)

参数：

inQuery String

template String

queryPlaceHolder String

返回：

[String](https://developer.android.google.cn/reference/java/lang/String.html)

* decode(level 1)

byte []decode(byte [] url)

参数：

url byte

返回：

byte[]

抛出：

[IllegalArgumentException](https://developer.android.google.cn/reference/java/lang/IllegalArgumentException.html)

* guessFileName(level 1)

String guessFileName(String url，

String contentDisposition， String mimeType)

使用URL和contentDisposition来猜测下载可能具有的规范文件名。如果没有定义文件扩展名，则根据MIME类型添加

参数：

url String：网址到内容

contentDisposition String：Content-Disposition HTTP头或 null

mimeType String：MIME类型的内容或 null

返回：

[String](https://developer.android.google.cn/reference/java/lang/String.html) 建议文件名

* guessUrl(level 1)

String guessUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) inUrl)

清除(如果可能)用户输入的网址

参数：

inUrl String

返回：

[String](https://developer.android.google.cn/reference/java/lang/String.html)

* isAboutUrl(level 1)

boolean isAboutUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

参数：

url String

boolean 如果url是about：url，返回true。

* isAssetUrl(level 1)

boolean isAssetUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

参数：

url String

返回：

boolean  如果url是一个资产文件，返回true。

* isContentUrl(level 1)

boolean isContentUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

确定url是不是内容url。

参数：

url String

返回：

boolean  如果该网址是内容：url ，返回true。

* isCookielessProxyUrl(level 1)

boolean isCookielessProxyUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

**此方法已在API级别3中弃用。**  
无Cookie代理不再受支持。

参数：

url String

返回：

boolean 如果url是一个代理url，允许来自文件url的无Cookie网络请求 。

* isDataUrl(level 1)

boolean isDataUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

参数：

url String

返回：

boolean 如果url是一个data：url ，返回true 。

* isFileUrl(level 1)

boolean isFileUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

参数：

url String

返回：

boolean 如果url是一个本地文件，返回true 。

* isHttpUrl(level 1)

boolean isHttpUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

参数：

url String

返回：

boolean 如果url是一个http：url，返回true 。

* isHttpsUrl(level 1)

boolean isHttpsUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

参数：

url String

返回：

boolean 如果url是一个https：url，返回true 。

* isJavaScriptUrl(level 1)

boolean isJavaScriptUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

参数：

url String

返回：

boolean 如果url是一个javascript：url，返回true 。

* isNetworkUrl(level 1)

boolean isNetworkUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

参数：

url String

返回：

boolean 如果url是一个网络网址，返回true 。

* isValidUrl(level 1)

boolean isValidUrl([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

参数：

url String

返回：

boolean 如果该网址是有效的，返回true 。

* stripAnchor(level 1)

[String](https://developer.android.google.cn/reference/java/lang/String.html) stripAnchor([String](https://developer.android.google.cn/reference/java/lang/String.html) url)

剥离锚点的网址。

参数：

url String

返回：

[String](https://developer.android.google.cn/reference/java/lang/String.html)

## 1.15 ValueCallback

* public interface ValueCallback

用于异步提供值的回调接口。

* onReceiveValue(level 7)

abstract void onReceiveValue(T value)

当值可用时调用。

参数：

value T： 值

## 1.16 WebBackForwardList

* public abstract class WebBackForwardList

该类包含WebView的后退/前进列表。WebView.copyBackForwardList( )将返回这个类的副本，用于检查列表中的条目。

**Public constructor**

* WebBackForwardList(level 22)

WebBackForwardList( )

**public**

* getCurrentIndex(level 1)

int getCurrentIndex( )

获取当前历史项目的索引。这个索引可以用来直接索引到数组列表中。

返回：

int 当前索引从0 ... n或-1(如果列表为空)

* getCurrentItem(level 1)

[WebHistoryItem](https://developer.android.google.cn/reference/android/webkit/WebHistoryItem.html) getCurrentItem( )

返回当前历史项目。如果列表为空，则返回null。

返回：

[WebHistoryItem](https://developer.android.google.cn/reference/android/webkit/WebHistoryItem.html) 当前历史项目。这个值可能为null。

* getItemAtIndex(level 1)

[WebHistoryItem](https://developer.android.google.cn/reference/android/webkit/WebHistoryItem.html) getItemAtIndex(int index)

获取给定索引的历史记录。索引范围是从0 ... n，其中0是第一项，n是最后一项。

参数：

index int：要检索的索引。

返回：

[WebHistoryItem](https://developer.android.google.cn/reference/android/webkit/WebHistoryItem.html)

* getSize(level 1)

int getSize( )

获取后退/前进列表的总大小。

返回：

int 列表的总大小

**protected**

* clone(level 1)

[WebBackForwardList](https://developer.android.google.cn/reference/android/webkit/WebBackForwardList.html) clone( )

通过WebView的客户端克隆要在UI线程中使用的整个对象。这创建了一个不应该被任何webkit包类修改的副本。

返回：

[WebBackForwardList](https://developer.android.google.cn/reference/android/webkit/WebBackForwardList.html) 这个实例的一个克隆。

## 1.17 WebChromeClient

* public class WebChromeClient

**Public constructor**

* WebChromeClient(level 1)

WebChromeClient( )

**public**

* getDefaultVideoPoster(level 7)

Bitmap getDefaultVideoPoster( )

不播放时，视频元素由“poster”图片表示。要使用的图片可以通过HTML中的视频标签的海报属性指定。如果该属性不存在，则将使用默认poster。该方法允许ChromeClient提供该默认图片。

返回：

[Bitmap](https://developer.android.google.cn/reference/android/graphics/Bitmap.html) 位图用作默认poster的图片，如果没有这样的图像可用，返回null。

* getVideoLoadingProgressView(level 7)

[View](https://developer.android.google.cn/reference/android/view/View.html) getVideoLoadingProgressView( )

获得全屏视频缓冲时显示的视图。应用程序可以重写此方法以提供包含微调器或类似物的视图。

返回:

[View](https://developer.android.google.cn/reference/android/view/View.html) 查看在视频加载时显示的视图。这个值可能是null。

* getVisitedHistory(level 7)

void getVisitedHistory([ValueCallback](https://developer.android.google.cn/reference/android/webkit/ValueCallback.html) < [String []](https://developer.android.google.cn/reference/java/lang/String.html) > callback)

获取所有访问过的历史项目的列表，用于链接着色

参数：

callback ValueCallback

* onCloseWindow(level 1)

void onCloseWindow([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) webview)

通知应用程序关闭给定的WebView并在必要时将其从视图系统中删除。调用了这个方法后，WebCore停止在这个窗口中的任何加载，并删除了JavaScript中的任何交叉脚本功能。

参数：

Window WebView：需要关闭的WebView。

* onConsoleMessage(level 8)

boolean onConsoleMessage([ConsoleMessage](https://developer.android.google.cn/reference/android/webkit/ConsoleMessage.html) consoleMessage )

向应用程序报告JavaScript控制台消息。ChromeClient应该重写它以按照认为合适的方式处理日志消息。

参数：

consoleMessage ConsoleMessage：包含控制台消息详细信息的对象。

返回：

boolean 如果消息由客户端处理，返回true 。

* onConsoleMessage(level 7)

void onConsoleMessage([String](https://developer.android.google.cn/reference/java/lang/String.html) message， int lineNumber， String sourceID)

**这种方法在API级别被废弃8.**  
使用[**onConsoleMessage(ConsoleMessage)**](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.html#onConsoleMessage(android.webkit.ConsoleMessage)) 来代替。

向应用程序报告JavaScript错误消息。ChromeClient应该重写它以按照合适的方式处理日志消息。

参数;

message String：要报告的错误消息。

lineNumber int：错误的行号。

sourceID String：导致错误的源文件的名称。

* onCreateWindow(level 1)

boolean onCreateWindow([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， boolean isDialog，

boolean isUserGesture， View resultMsg)

请求应用程序创建一个新窗口。如果应用程序选择遵守此请求，则应从此方法返回true，创建一个新的WebView以承载该窗口，将其插入到View系统中，并使用新的WebView作为参数将提供的resultMsg消息发送到其目标。如果应用程序选择不遵守请求，则应从此方法返回false。这个方法的默认实现什么也不做，因此返回false。

参数:

view WebView：发起新窗口请求的WebView。

isDialog boolean：如果true，新窗口应该是对话框，而不是全尺寸窗口。

isUserGesture boolean：true如果请求是由用户手势启动的，例如用户点击链接。

resultMsg Message：创建新WebView时发送的消息。resultMsg.obj是一个 对象。这应该用于通过调用来传输新的WebView 。[WebView.WebViewTransport](https://developer.android.google.cn/reference/android/webkit/WebView.WebViewTransport.html)[WebView.WebViewTransport.setWebView(WebView)](https://developer.android.google.cn/reference/android/webkit/WebView.WebViewTransport.html#setWebView(android.webkit.WebView))

返回：

boolean true如果应用程序将创建一个新窗口，则应返回此方法，在这种情况下，应将resultMsg发送到其目标。否则，这个方法应该返回false。false从这个方法返回 ，但也发送resultMsg将导致未定义的行为。

* onExceededDatabaseQuota(level 5)

void onExceededDatabaseQuota([String](https://developer.android.google.cn/reference/java/lang/String.html) url， [String](https://developer.android.google.cn/reference/java/lang/String.html) databaseIdentifier，

long quota,long estimatedDatabaseSize, long totalQuota,

[WebStorage.QuotaUpdater](https://developer.android.com/reference/android/webkit/WebStorage.QuotaUpdater.html) quotaUpdater)

**此方法在API级别19中已弃用。**  
此方法不再被调用; WebView现在使用HTML5 / JavaScript配额管理API。

通知客户端特定源的Web SQL数据库超出配额并请求新的配额。客户端必须通过调用[updateQuota(long)](https://developer.android.google.cn/reference/android/webkit/WebStorage.QuotaUpdater.html#updateQuota(long)) 提供的[WebStorage.QuotaUpdater](https://developer.android.google.cn/reference/android/webkit/WebStorage.QuotaUpdater.html)实例的方法进行响应 。可以为新配额设置的最小值是当前配额。默认以当前配额实现响应，因此配额不会增加。

参数：

url String：触发​​通知的网页的网址

databaseIdentifier String：超出配额的数据库的标识符。

quota long：原始的配额，以字节为单位

estimatedDatabaseSize long：违规数据库的估计大小，以字节为单位

totalQuota long：所有来源的总配额，以字节为单位

quotaUpdater WebStorage.QuotaUpdater：必须使用 [WebStorage.QuotaUpdater](https://developer.android.com/reference/android/webkit/WebStorage.QuotaUpdater.html) 实例通知WebView获取新配额。

* onGeolocationPermissionsHidePrompt(level 5)

void onGeolocationPermissionsHidePrompt( )

通知应用程序，前一次调用的地理位置权限请求 [onGeolocationPermissionsShowPrompt()](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.html#onGeolocationPermissionsShowPrompt(java.lang.String,%20android.webkit.GeolocationPermissions.Callback)) 已被取消。因此任何相关的用户界面应该隐藏。

* onGeolocationPermissionsShowPrompt(level 5)

void onGeolocationPermissionsShowPrompt([String](https://developer.android.google.cn/reference/java/lang/String.html) origin，

[GeolocationPermissions.Callback](https://developer.android.google.cn/reference/android/webkit/GeolocationPermissions.Callback.html) callback)

通知宿主应用程序指定来源的Web内容正在尝试使用Geolocation API，但目前没有为该来源设置权限状态。应用程序应该使用所需的权限状态调用指定的回调。详情请参阅 [GeolocationPermissions](https://developer.android.google.cn/reference/android/webkit/GeolocationPermissions.html)。

请注意，对于面向Android N及更高版本的SDK(API级别> [M](https://developer.android.google.cn/reference/android/os/Build.VERSION_CODES.html#M))的应用程序，仅针对源自https等安全来源的请求调用此方法。在不安全的来源地理位置请求会被自动拒绝。

参数：

origin String：试图使用Geolocation API的网页内容的来源。

callback GeolocationPermissions.Callback：用于设置原始权限状态的回调。

* onHideCustomView(level 7)

void onHideCustomView( )

通知应用程序当前页面已退出全屏模式。应用程序必须隐藏自定义视图，即。[onShowCustomView(View, WebChromeClient.CustomViewCallback)](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.html#onShowCustomView(android.view.View,%20android.webkit.WebChromeClient.CustomViewCallback))当内容进入全屏时传递给View 。另请参阅“全屏支持”文档[WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html)。

* onJsAlert(level 1)

boolean onJsAlert([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view，[String](https://developer.android.google.cn/reference/java/lang/String.html) url， [String](https://developer.android.google.cn/reference/java/lang/String.html) message， [JsResult](https://developer.android.google.cn/reference/android/webkit/JsResult.html) result)

通知客户端显示一个JavaScript警告对话框。如果客户端返回true，WebView将假定客户端将处理该对话框。如果客户返回false，程序将继续执行。

参数：

view WebView：启动回调的WebView。

url String：请求对话的页面的URL。

message String：要在窗口中显示的消息。

result JsResult：一个JsResult来确认用户输入。

返回：

boolean boolean客户端是否处理警报对话框。

* onJsBeforeUnload(level 1)

boolean onJsBeforeUnload([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [String](https://developer.android.google.cn/reference/java/lang/String.html) url，

[String](https://developer.android.google.cn/reference/java/lang/String.html) message， [JsResult](https://developer.android.google.cn/reference/android/webkit/JsResult.html) result)

通知客户端显示一个对话框，以确认远离当前页面。这是onbeforeunload javascript事件的结果。如果客户端返回true，WebView将假定客户端将处理确认对话框并调用相应的JsResult方法。如果客户端返回false，默认值true将返回到JavaScript以接受远离当前页面的导航。默认行为是返回false。将JsResult设置为true离开当前页面，false将取消导航。

参数：

view WebView：启动回调的WebView。

url String：请求对话的页面的URL。

message String：要在窗口中显示的消息。

result JsResult：一个JsResult，用于发送用户对javascript的响应。

返回：

boolean boolean客户端是否处理确认对话框。

* onJsConfirm(level 1)

boolean onJsConfirm([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [String](https://developer.android.google.cn/reference/java/lang/String.html) url， [String](https://developer.android.google.cn/reference/java/lang/String.html) message， [JsResult](https://developer.android.google.cn/reference/android/webkit/JsResult.html) result)

通知客户端向用户显示一个确认对话框。如果客户端返回true，WebView将假定客户端将处理确认对话框并调用相应的JsResult方法。如果客户端返回false，则默认值false将返回到javascript。默认行为是返回false。

参数：

view WebView：启动回调的WebView。

url String：请求对话的页面的URL。

message String：要在窗口中显示的消息。

result JsResult：一个JsResult，用于发送用户对javascript的响应。

返回：

boolean boolean客户端是否处理确认对话框。

* onJsPrompt(level 1)

boolean onJsPrompt([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [String](https://developer.android.google.cn/reference/java/lang/String.html) url， [String](https://developer.android.google.cn/reference/java/lang/String.html) message，

[String](https://developer.android.google.cn/reference/java/lang/String.html) defaultValue， [JsPromptResult](https://developer.android.google.cn/reference/android/webkit/JsPromptResult.html) result)

告诉客户端向用户显示一个提示对话框。如果客户端返回true，WebView将假定客户端将处理提示对话框并调用相应的JsPromptResult方法。如果客户端返回false，则默认值false将返回到javascript。默认行为是返回false。

参数：

view WebView：启动回调的WebView。

url String：请求对话的页面的URL。

message String：要在窗口中显示的消息。

defaultValue String：提示对话框中显示的默认值。

result JsPromptResult：用于将用户的响应发送给JavaScript的JsPromptResult。

返回：

boolean boolean客户端是否处理提示对话框。

* onJsTimeout(level 7)

boolean onJsTimeout( )

**此方法在API级别17中已弃用。**  
此方法不再受支持，也不会被调用。

通知客户端JavaScript执行已经超时。客户可以决定是否中断执行。如果客户端返回true，则JavaScript将被中断。如果客户返回false，执行将继续。请注意，在继续执行的情况下，超时计数器将被重置，如果脚本没有在下一个检查点完成，则回调将继续发生。

返回：

boolean boolean是否应该中断JavaScript执行。

* onPermissionRequest(level 21)

void onPermissionRequest([PermissionRequest](https://developer.android.google.cn/reference/android/webkit/PermissionRequest.html) requst)

通知应用程序Web内容正在请求访问指定资源的权限，并且当前未授予或拒绝该权限。应用程序必须调用[grant(String[])](https://developer.android.google.cn/reference/android/webkit/PermissionRequest.html#grant(java.lang.String[])) 或[deny()](https://developer.android.google.cn/reference/android/webkit/PermissionRequest.html#deny())。如果此方法未被重写，则权限被拒绝。

参数：

request PermissionRequest：来自当前网页内容的PermissionRequest。

* onPermissionRequestCanceled(level 21)

void onPermissionRequestCanceled([PermissionRequest](https://developer.android.google.cn/reference/android/webkit/PermissionRequest.html) request)

通知应用程序已经取消了给定的权限请求。因此任何相关的用户界面应该隐藏。

参数：

request PermissionRequest：需要被取消的PermissionRequest。

* onProgressChanged(level 1)

void onProgressChanged([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， int newProgress)

通知应用程序加载页面的当前进度。

参数：

view WebView：启动回调的WebView。

newProgress int：当前页面加载进度，由0到100之间的整数表示。

* onReachedMaxAppCacheSize(level 7)

void onReachedMaxAppCacheSize(long requiredStorage，long quota

, [WebStorage.QuotaUpdater](https://developer.android.google.cn/reference/android/webkit/WebStorage.QuotaUpdater.html) quotaUpdater)

**此方法在API级别19中已弃用。**  
此方法不再被调用; WebView现在使用HTML5 / JavaScript配额管理API。

通知应用程序应用程序缓存已达到最大。客户端必须通过调用[WebStorage.QuotaUpdater](https://developer.android.google.cn/reference/android/webkit/WebStorage.QuotaUpdater.html)实例提供的[updateQuota(long)](https://developer.android.google.cn/reference/android/webkit/WebStorage.QuotaUpdater.html#updateQuota(long)) 方法进行响应 。可以为新配额设置的最小值是当前配额。默认实现以当前配额响应，若是选择默认，则配额不会增加。

参数：

requiredStorage long：触发​​此通知的应用程序缓存操作所需的存储量(以字节为单位)。

quota long：当前最大应用程序缓存大小(以字节为单位)

quotaUpdater WebStorage.QuotaUpdater：必须使用[WebStorage.QuotaUpdater](https://developer.android.google.cn/reference/android/webkit/WebStorage.QuotaUpdater.html)实例通知WebView新配额。

* onReceivedIcon(level 1)

void onReceivedIcon([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， Bitmap icon)

通知当前页面收到新的图片。

参数：

view WebView：启动回调的WebView。

icon Bitmap：包含当前页面的图标的位图。

* onReceivedTitle(level 1)

void onReceivedTitle([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， String title)

通知应用程序的当前页面中的标题更改。

参数：

view WebView：启动回调的WebView。

title String：包含文档新标题的字符串。

* onReceivedTouchIconUrl(level 7)

void onReceivedTouchIconUrl([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [String](https://developer.android.google.cn/reference/java/lang/String.html) url，boolean precomposed)

通知应用程序 apple-touch-icon的 url 参数：

view WebView：启动回调的WebView。

url String：图标网址。

precomposed boolean：如果true，网址用于预先分解的触摸图标。

* onRequestFocus(level 1)

void onRequestFocus([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view)

显示此WebView并为其获取焦点。这可能是由于另一个WebView在此WebView中打开一个链接并要求显示此WebView。

参数：

view WebView：需要焦点的WebView。

* onShowCustomView(level 14)

void onShowCustomView([View](https://developer.android.google.cn/reference/android/view/View.html) view， int requestedOrientation，

[WebChromeClient.CustomViewCallback](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.CustomViewCallback.html) callback)

**此方法在API级别18中已被弃用。**  
此方法支持过时的插件机制，将来不会调用该方法

通知应用程序：当前页面希望以特定方向显示自定义视图。

参数：

view View：是要显示的视图对象。

requestedOrientation int：用于的定向常数 。[ActivityInfo.screenOrientation](https://developer.android.google.cn/reference/android/content/pm/ActivityInfo.html#screenOrientation)

callback WebChromeClient.CustomViewCallback：是在视图被解散时调用的回调函数。

* onShowCustomView(level 7)

void onShowCustomView(View view，

[WebChromeClient.CustomViewCallback](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.CustomViewCallback.html) callback)

通知应用程序当前页面已进入全屏模式。应用程序必须以全屏模式显示包含网页内容(视频或其他HTML内容)的自定义视图。另请参阅“全屏支持”文档 [WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html)。

参数：

view View：是要显示的视图对象。

callback WebChromeClient.CustomViewCallback：调用此回调来请求页面退出全屏模式。

* onShowFileChooser(level 21)

boolean onShowFileChooser([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) webView，

[ValueCallback](https://developer.android.google.cn/reference/android/webkit/ValueCallback.html) < [Uri []](https://developer.android.google.cn/reference/android/net/Uri.html) > filePathCallback，

[WebChromeClient.FileChooserParams](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.FileChooserParams.html) fileChooserParams)

告诉客户显示文件选择器。这被称为用'文件'输入类型处理HTML表单，以响应用户按下“选择文件”按钮。要取消请求，请致电filePathCallback.onReceiveValue(null)并返回true。

参数：

webView WebView：发起请求的WebView实例。

filePathCallback ValueCallback：调用此回调以提供要上传或null取消的文件的路径列表。只有在实现返回时才能调用 。[onShowFileChooser(WebView, ValueCallback, WebChromeClient.FileChooserParams)](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.html#onShowFileChooser(android.webkit.WebView,%20android.webkit.ValueCallback%3Candroid.net.Uri[]%3E,%20android.webkit.WebChromeClient.FileChooserParams))true

fileChooserParams WebChromeClient.FileChooserParams：描述要打开的文件选择器的模式以及与之一起使用的选项

返回：

boolean 如果将调用filePathCallback，返回true，若是返回false，使用默认处理。

**相关函数：**

* [WebChromeClient.FileChooserParams](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.FileChooserParams.html)

1.17.1 WebChromeClient.CustomViewCallback

* public static interface WebChromeClient.CustomViewCallback

应用程序使用的回调接口通知当前页面的自定义视图已被解除。

**public**

* onCustomViewHidden(level 7)

abstract void onCustomViewHidden( )

在应用程序关闭自定义视图时调用。

## 1.18 WebHistoryItem

* public abstract class WebHistoryItem

用于访问WebView的后退/前进列表中条目中字段的便捷类。每个WebHistoryItem都是请求的历史记录项的快照。每个历史项目可能会在加载页面期间更新。

**相关函数：**

* [WebBackForwardList](https://developer.android.com/reference/android/webkit/WebBackForwardList.html)

**Public constructor**

* WebHistoryItem(level 22)

WebHistoryItem( )

**public**

* getFavicon(level 1)

abstract Bitmap getFavicon( )

返回这个历史项目的图标，没有找到图标返回null。

参数：

[Bitmap](https://developer.android.com/reference/android/graphics/Bitmap.html) 包含此历史项目的图标的位图，若是没有，则返回null。注意：VM确保32位读/写操作，因此我们不必同步此方法

* getOriginalUrl(level 3)

[String](https://developer.android.com/reference/java/lang/String.html) getOriginalUrl( )

返回此历史项目的原始网址。这是请求的网址，最终的网址可能会有所不同，因为加载网站时可能有重定向。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 此历史项目的原始网址。

* getTitle(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getTitle( )

返回此历史项目的文档标题。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 此历史项目的文档标题。注意：VM确保32位读/写操作，因此我们不必同步此方法

* getUrl(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getUrl ()

返回此历史项目的网址。 该网址是此历史记录的基本网址。 请参阅getTargetUrl( )以获取此历史记录项目的实际目标网址。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 此历史记录的基本网址，注意：VM确保32位读/写操作，因此我们不必同步此方法

**Protected方法**

* clone(level 1)

[WebHistoryItem](https://developer.android.com/reference/android/webkit/WebHistoryItem.html) clone ()

克隆历史记录。

## 1.19 WebIconDatabase

* public abstract class WebIconDatabase

**此类在API级别18中已被弃用。**  
此类仅在运行于最多至的设备上时才需要[**JELLY\_BEAN\_MR2**](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN_MR2)

用于操作WebView使用的图标数据库。这些函数需要在被调用之前需要构造一个WebView，并且用WebView.getIconDatabase( )将返回一个WebIconDatabase对象。这个WebIconDatabase对象是一个单例的实例，所有的方法都在这个单例的对象上运行。此类的主要用例是[open(String)](https://developer.android.com/reference/android/webkit/WebIconDatabase.html#open(java.lang.String)) 在此过程中调用以在所有WebView实例上启用favicon功能。

**Public constructor**

* WebIconDatabase(level 22)

WebIconDatabase( )

**public**

* Close(level 1)

void close( )

关闭图标数据库的共享实例。

* getInstance(level 1)

[WebIconDatabase](https://developer.android.com/reference/android/webkit/WebIconDatabase.html) getInstance( )

获取WebIconDatabase的全局实例。

返回：

[WebIconDatabase](https://developer.android.com/reference/android/webkit/WebIconDatabase.html) WebIconDatabase的单个实例。每次调用此方法时，它将是当前进程的相同实例。

* Open(level 1)

void open([String](https://developer.android.com/reference/java/lang/String.html) path)

打开图标数据库并将图标存储在给定路径中。

参数：

path String：图标数据库将被存储的目录路径。

* releaseIconForPageUrl(level 1)

void releaseIconForPageUrl([String](https://developer.android.com/reference/java/lang/String.html) url)

释放给定页面网址的图标。

参数：

url tring：该网页的网址。

* removeAllIcons(level 1)

void removeAllIcons( )

删除数据库中的所有图标。

* requestIconForPageUrl(level 1)

void requestIconForPageUrl([String](https://developer.android.com/reference/java/lang/String.html) url，

[WebIconDatabase.IconListener](https://developer.android.com/reference/android/webkit/WebIconDatabase.IconListener.html) listener)

请求给定页面url的图标。如果图标存在，监听器将被调用并显示结果。

参数：

url String：该网页的网址。  
 listener WebIconDatabase.IconListener：IconListener上的一个实现来接收结果。

* retainIconForPageUrl(level 1)

void retainIconForPageUrl([String](https://developer.android.com/reference/java/lang/String.html) url)

保留给定页面网址的图标。

参数：

url String：该网页的网址。

1.19.1 WebIconDatabase.IconListener

* public static interface WebIconDatabase.IconListener

**此接口在API级别18中已弃用。**  
此接口已过时。

从数据库接收图标的接口。

**public**

* onReceivedIcon(level 1)

abstract void onReceivedIcon([String](https://developer.android.google.cn/reference/java/lang/String.html) url， [Bitmap](https://developer.android.com/reference/android/graphics/Bitmap.html) icon)

当从数据库中检索图标并且结果非空时调用。该方法是虚方法。

参数：

url String：请求中传递的网址。

icon Bitmap：给定网址的图标。

## 1.20 WebResourceRequest

* public interface WebResourceRequest

**public**

* getMethod(level 21)

abstract [String](https://developer.android.com/reference/java/lang/String.html) getMethod( )

获取与请求关联的方法，例如“GET”。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 与请求相关联的方法。

* getRequestHeaders(level 21)

abstract [Map](https://developer.android.com/reference/java/util/Map.html) < [String](https://developer.android.com/reference/java/lang/String.html)， [String](https://developer.android.com/reference/java/lang/String.html) > getRequestHeaders( )

获取与请求关联的标题。这些表示为标题名称到标题值的映射。

返回：

[Map](https://developer.android.com/reference/java/util/Map.html)<[String](https://developer.android.com/reference/java/lang/String.html), [String](https://developer.android.com/reference/java/lang/String.html)> 与请求关联的标头。

* getURL(level 21)

abstract [Uri](https://developer.android.com/reference/android/net/Uri.html) getUrl( )

获取进行资源请求的URL。该方法是虚方法。

返回：

Uri 为其提出资源请求的URL。

* hasGesture(level 21)

abstract boolean hasGesture( )

获取手势(如点击)是否与请求相关联。出于安全原因，在某些情况下，false即使导致请求被创建的事件序列由用户手势启动，该方法也可能返回。该方法是虚方法。

返回：

boolean 手势是否与请求相关联。

* isForMainFrame(level 21)

abstract boolean isForMainFrame( )

获取：请求是否是针对主框架进行的。该方法是虚方法。

返回:

boolean 是否要求为主框架。

* isRedirect(level 24)

abstract boolean isRedirect( )

获取：请求是否是服务器端重定向的结果。该方法是虚方法。

返回：

boolean 请求是否是服务器端重定向的结果。

## 1.21 WebResourceResponse

* public class WebResourceResponse

封装资源响应。应用程序可以用[shouldInterceptRequest(WebView, WebResourceRequest)](https://developer.android.com/reference/android/webkit/WebViewClient.html#shouldInterceptRequest(android.webkit.WebView,%20android.webkit.WebResourceRequest))返回此类的实例，以在WebView请求特定资源时提供自定义响应。

**Public constructor**

* WebResourceResponse(level 11)

WebResourceResponse(String mimeType， String code， [InputStream](https://developer.android.com/reference/java/io/InputStream.html) data)

使用给定的MIME类型、编码和输入流构造资源响应。调用者必须实现 [InputStream.read(byte[])](https://developer.android.com/reference/java/io/InputStream.html#read(byte[]))。

参数：

mimeType String：资源响应的MIME类型，例如text / html

encoding String：资源响应的编码

data InputStream：提供资源响应数据的输入流。绝对不能是一个StringBufferInputStream。

* WebResourceResponse(level 21)

WebResourceResponse(String mimeType， String code， int statusCode，

[String](https://developer.android.com/reference/java/lang/String.html) reasonPhrase，

[Map](https://developer.android.com/reference/java/util/Map.html) < [String](https://developer.android.com/reference/java/lang/String.html)，  [String](https://developer.android.com/reference/java/lang/String.html) > responseHeaders，

[InputStream](https://developer.android.com/reference/java/io/InputStream.html) data)

用给定的参数构造资源响应。调用者必须实现[InputStream.read(byte[])](https://developer.android.com/reference/java/io/InputStream.html#read(byte[]))。

参数：

mimeType String：资源响应的MIME类型，例如text / html

encoding String：资源响应的编码

statusCode int：状态码需要在[100,299]，[400,599]的范围内。不支持通过指定3xx代码引起重定向。

reasonPhrase String：描述状态码的短语，例如“OK”。必须是非空的。这个值绝对不能null

responseHeaders Map：资源响应的标题表示为标题名称 - >标题值的映射。

data InputStream：提供资源响应数据的输入流。一定不能是一个StringBufferInputStream。

**public**

* getData(level 11)

[InputStream](https://developer.android.com/reference/java/io/InputStream.html) getData( )

获取提供资源响应数据的输入流。

返回：

[InputStream](https://developer.android.com/reference/java/io/InputStream.html) 提供资源响应数据的输入流

* getEncoding(level 11)

[String](https://developer.android.com/reference/java/lang/String.html) getEncoding( )

获取资源响应的编码。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 资源响应的编码

* getMimeType(level 11)

[String](https://developer.android.com/reference/java/lang/String.html) getMimeType( )

获取资源响应的MIME类型。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 资源响应的MIME类型

* getReasonPhrase(level 21)

[String](https://developer.android.com/reference/java/lang/String.html) getReasonPhrase( )

获取资源响应状态码的描述。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 资源响应状态码的描述。

* getResponseHeaders(level 21)

[Map](https://developer.android.com/reference/java/util/Map.html) < [String](https://developer.android.com/reference/java/lang/String.html)，  [String](https://developer.android.com/reference/java/lang/String.html) > getResponseHeaders( )

获取资源响应的标题。

返回：

[Map](https://developer.android.com/reference/java/util/Map.html)<[String](https://developer.android.com/reference/java/lang/String.html), [String](https://developer.android.com/reference/java/lang/String.html)> 资源响应的标题。

* getStatusCode(level 21)

int getStatusCode( )

获取资源响应的状态码。

返回：

int 资源响应的状态码。

* setData(level 11)

void setData([InputStream](https://developer.android.com/reference/java/io/InputStream.html) data)

设置提供资源响应数据的输入流。调用者必须实现[InputStream.read(byte[])](https://developer.android.com/reference/java/io/InputStream.html#read(byte[]))。

参数：

data InputStream：提供资源响应数据的输入流。一定不能是一个StringBufferInputStream。

* setEncoding(level 11)

void setEncoding([String](https://developer.android.com/reference/java/lang/String.html) encoding)

设置资源响应的编码，例如“UTF-8”。这用于解码来自输入流的数据。

参数：

encoding String：资源响应的编码

* setMimeType(level 11)

void setMimeType([String](https://developer.android.com/reference/java/lang/String.html) mimeType)

设置资源响应的MIME类型，例如“text / html”。

参数：

mimeType String：资源响应的MIME类型

* setResponseHeaders(level 21)

void setResponseHeaders([Map](https://developer.android.com/reference/java/util/Map.html) < [String](https://developer.android.com/reference/java/lang/String.html)，  [String](https://developer.android.com/reference/java/lang/String.html) > headers)

设置资源响应的标题。

参数：

headers Map：标题名称映射 - >标题值。

* setStatusCodeAndReasonPhrase(level 21)

void setStatusCodeAndReasonPhrase(int statusCode， [String](https://developer.android.com/reference/java/lang/String.html) reasonPhrase)

设置资源响应的状态码和原因词组。

参数：

statusCode int：状态码需要在[100,299]，[400,599]的范围内。不支持通过指定3xx代码引起重定向。

reasonPhrase String：描述状态码的短语，例如“OK”。必须是非空的。这个值绝对不能null。

## 1.22 WebSettings

* public abstract class WebSettings

管理WebView的设置。首次创建WebView时，它会获取一组默认设置。这些默认设置将从任何getter调用中返回。WebSettings从WebView的[getSettings()](https://developer.android.com/reference/android/webkit/WebView.html#getSettings())中获得的对象 ，其与WebView的生命周期息息相关。如果一个WebView已经被销毁，调用WebSettings的任何方法都会抛出一个 [IllegalStateException](https://developer.android.com/reference/java/lang/IllegalStateException.html)。

**Public构造方法：**

[WebSettings](https://developer.android.com/reference/android/webkit/WebSettings.html#WebSettings())()

**Public**

* enableSmoothTransition(level 11)

boolean enableSmoothTransition( )

**此方法在API级别17中已弃用。**  
此方法现在已过时，并且将来将成为禁用操作。

获取WebView在平移或缩放时是否平滑过渡。

返回：

boolean

**相关函数：**

* [setEnableSmoothTransition(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setEnableSmoothTransition(boolean))
* getAllowContentAccess(level 11)

boolean getAllowContentAccess( )

获取此WebView是否支持内容URL访问。

返回：

boolean

**相关函数：**

* [setAllowContentAccess(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setAllowContentAccess(boolean))
* getAllowFileAccess(level 3)

boolean getAllowFileAccess( )

获取此WebView是否支持文件访问。

返回:

boolean

相关函数：

* [setAllowFileAccess(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setAllowFileAccess(boolean))
* getAllowFileAccessFromFileURLs(level 16)

boolean getAllowFileAccessFromFileURLs( )

获取在文件系统URL上下文中运行的JavaScript是否可以访问来自其他文件URL系统的内容。

返回;

boolean 在文件系统URL上下文中运行的JavaScript是否可以访问来自其他文件系统URL的内容

**相关函数：**

* [setAllowFileAccessFromFileURLs(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setAllowFileAccessFromFileURLs(boolean))
* getAllowUniversalAccessFromFileURLs(level 16)

boolean getAllowUniversalAccessFromFileURLs( )

获取在文件系统URL上下文中运行的JavaScript是否可以访问任何来源的内容。这包括访问来自其他文件系统URL的内容。

返回：

boolean 在文件系统URL的上下文中运行的JavaScript是否可以访问任何来源的内容

**相关函数：**

* [setAllowUniversalAccessFromFileURLs(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setAllowUniversalAccessFromFileURLs(boolean))
* getBlockNetworkImage(level 1)

boolean getBlockNetworkImage( )

获取WebView是否不从网络加载图像资源。

返回：

boolean 如果true  ，WebView不从网络加载图像资源

**相关函数：**

* [setBlockNetworkImage(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setBlockNetworkImage(boolean))
* getBlockNetworkLoads(level 8)

boolean getBlockNetworkLoads( )

获取WebView是否不从网络加载任何资源。

返回：

boolean 如果true  ，WebView不从网络加载任何资源

**相关函数：**

* [setBlockNetworkLoads(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setBlockNetworkLoads(boolean))
* getBuiltInZoomControls(level 3)

boolean getBuiltInZoomControls( )

获取是否正在使用WebView中内置的缩放机制。

返回：

boolean 如果true  ，正在使用WebView中内置的缩放机制

**相关函数：**

* [setBuiltInZoomControls(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setBuiltInZoomControls(boolean))
* getCacheMode(level 1)

int getCacheMode( )

获取当前缓存模式的设置。

返回：

int 返回值是[LOAD\_DEFAULT](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_DEFAULT)，[LOAD\_NORMAL](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_NORMAL)，[LOAD\_CACHE\_ELSE\_NETWORK](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_CACHE_ELSE_NETWORK)，[LOAD\_NO\_CACHE](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_NO_CACHE)或[LOAD\_CACHE\_ONLY](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_CACHE_ONLY)

**相关函数：**

* [setCacheMode(int)](https://developer.android.com/reference/android/webkit/WebSettings.html#setCacheMode(int))
* getCursiveFontFamily(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getCursiveFontFamily( )

获取草书字体系列名称。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 草书字体系列的名称作为字符串

**相关函数：**

* [setCursiveFontFamily(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setCursiveFontFamily(java.lang.String))
* getDatabaseEnabled(level 5)

boolean getDatabaseEnabled( )

获取是否已启用数据库存储。

返回：

boolean  如果已启用数据库存储，返回true

**相关函数：**

* [setDatabaseEnabled(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setDatabaseEnabled(boolean))
* getDatabasePath(level 5)

[String](https://developer.android.com/reference/java/lang/String.html) getDatabasePath( )

**此方法在API级别19中已被弃用。**  
数据库路径由实施管理，此方法已过时。

获取数据库存储数的路径。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 数据库存储的字符串路径

**相关函数：**

* [setDatabasePath(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setDatabasePath(java.lang.String))
* getDefaultFixedFontSize(level 1)

int getDefaultFixedFontSize( )

获取默认的固定字体大小。

返回：

int 1到72之间的非负整数

**相关函数：**

* [setDefaultFixedFontSize(int)](https://developer.android.com/reference/android/webkit/WebSettings.html#setDefaultFixedFontSize(int))
* getDefaultFontSize(level 1)

int getDefaultFontSize( )

获取默认字体大小。

返回:

int 1到72之间的非负整数

**相关函数：**

* [setDefaultFontSize(int)](https://developer.android.com/reference/android/webkit/WebSettings.html#setDefaultFontSize(int))
* getDefaultTextEncodingName(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getDefaultTextEncodingName( )

获取默认的文本编码名称。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 默认的文本编码名称

**相关函数：**

* [setDefaultTextEncodingName(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setDefaultTextEncodingName(java.lang.String))
* getDefaultUserAgent(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getDefaultUserAgent(Context context)

返回WebView使用的默认用户代理。如果用[setUserAgentString(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setUserAgentString(java.lang.String))，WebView的一个实例可以使用不同的用户代理。

参数：

context Context：用于访问应用程序上下文的Context对象

返回：

[String](https://developer.android.com/reference/java/lang/String.html)

* getDefaultZoom(level 7)

[WebSettings.ZoomDensity](https://developer.android.com/reference/android/webkit/WebSettings.ZoomDensity.html) getDefaultZoom( )

**此方法在API级别19中已被弃用。**  
只会返回默认值。

该方法在UI线程中使用。获取页面的默认缩放密度。这应该从UI线程中调用。不建议将此设置用于新应用程序。

返回：

[WebSettings.ZoomDensity](https://developer.android.com/reference/android/webkit/WebSettings.ZoomDensity.html) 缩放密度

**相关函数：**

* [setDefaultZoom(WebSettings.ZoomDensity)](https://developer.android.com/reference/android/webkit/WebSettings.html#setDefaultZoom(android.webkit.WebSettings.ZoomDensity))
* getDisabledActionModeMenuItems(level 24)

int getDisabledActionModeMenuItems( )

获取禁用的动作模式菜单项，用整数字段标志表示。默认值是[MENU\_ITEM\_NONE](https://developer.android.com/reference/android/webkit/WebSettings.html#MENU_ITEM_NONE)

返回：

int 值可以是0或[MENU\_ITEM\_NONE](https://developer.android.com/reference/android/webkit/WebSettings.html#MENU_ITEM_NONE)，[MENU\_ITEM\_SHARE](https://developer.android.com/reference/android/webkit/WebSettings.html#MENU_ITEM_SHARE)，[MENU\_ITEM\_WEB\_SEARCH](https://developer.android.com/reference/android/webkit/WebSettings.html#MENU_ITEM_WEB_SEARCH) 或[MENU\_ITEM\_PROCESS\_TEXT](https://developer.android.com/reference/android/webkit/WebSettings.html#MENU_ITEM_PROCESS_TEXT)的组合。

* getDisplayZoomControls(level 11)

boolean getDisplayZoomControls( )

获取WebView使用内置缩放机制时是否显示屏幕缩放控件。

返回：

boolean  如果WebView在使用内置缩放机制时显示屏幕缩放控件，返回true。

**相关函数：**

* [setDisplayZoomControls(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setDisplayZoomControls(boolean))
* getDomStorageEnabled(level 7)

boolean getDomStorageEnabled( )

获取是否启用DOM Storage API。

返回：

boolean  如果DOM Storage被启用，返回true。

**相关函数：**

* [setDomStorageEnabled(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setDomStorageEnabled(boolean))
* getFantasyFontFamily(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getFantasyFontFamily( )

获取幻想字体系列的名称。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 幻想字体系列名称作为一个字符串返回

**相关函数：**

* [setFantasyFontFamily(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setFantasyFontFamily(java.lang.String))
* getFixedFontFamily(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getFixedFontFamily( )

获取固定的字体系列名称。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 固定的字体系列名称作为字符串返回

**相关函数：**

* [setFixedFontFamily(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setFixedFontFamily(java.lang.String))
* getJavaScriptCanOpenWindowsAutomatically(level 1)

boolean getJavaScriptCanOpenWindowsAutomatically( )

获取JavaScript是否可以自动打开窗口。

返回：

boolean  如果JavaScript可以自动打开窗口 ，返回true。

**相关函数：**

* [setJavaScriptCanOpenWindowsAutomatically(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setJavaScriptCanOpenWindowsAutomatically(boolean))
* getJavaScriptEnabled(level 1)

boolean getJavaScriptEnabled( )

获取是否启用JavaScript。

返回：

boolean 如果启用了JavaScript ，返回true 。

**相关函数：**

* [setJavaScriptEnabled(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setJavaScriptEnabled(boolean))
* getLayoutAlgorithm(level 1)

[WebSettings.LayoutAlgorithm](https://developer.android.com/reference/android/webkit/WebSettings.LayoutAlgorithm.html) getLayoutAlgorithm( )

获取当前的布局算法。

返回：

[WebSettings.LayoutAlgorithm](https://developer.android.com/reference/android/webkit/WebSettings.LayoutAlgorithm.html) 返回使用布局算法

**相关函数：**

* [setLayoutAlgorithm(WebSettings.LayoutAlgorithm)](https://developer.android.com/reference/android/webkit/WebSettings.html#setLayoutAlgorithm(android.webkit.WebSettings.LayoutAlgorithm))
* getLightTouchEnabled(level 1)

boolean getLightTouchEnabled( )

**此方法在API级别18中已弃用。**  
此设置已过时。

获取是否启用轻触。

返回：

boolean 若果启动了轻触，返回true。

**相关函数：**

* [setLightTouchEnabled(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setLightTouchEnabled(boolean))
* getLoadWithOverviewMode(level 7)

boolean getLoadWithOverviewMode( )

获取此WebView是否以概览模式加载页面。

返回：

boolean 此WebView以概览模式加载页面，返回true。

**相关函数：**

* [setLoadWithOverviewMode(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setLoadWithOverviewMode(boolean))
* getLoadsImagesAutomatically(level 1)

boolean getLoadsImagesAutomatically( )

获取WebView是否加载图片资源。这包括使用数据URI方案嵌入的图片。

返回：

boolean 如果WebView加载图片资源，返回true

**相关函数：**

* [setLoadsImagesAutomatically(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setLoadsImagesAutomatically(boolean))
* getMediaPlaybackRequiresUserGesture(level 17)

boolean getMediaPlaybackRequiresUserGesture( )

获取WebView是否需要用户手势才能播放媒体。

返回：

boolean 如果WebView需要用户手势播放媒体，返回true 。

**相关函数：**

* [setMediaPlaybackRequiresUserGesture(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setMediaPlaybackRequiresUserGesture(boolean))
* getMinimumFontSize(level 1)

int getMinimumFontSize( )

获取最小字体大小。

返回：

int 1到72之间的非负整数

**相关函数：**

* [setMinimumFontSize(int)](https://developer.android.com/reference/android/webkit/WebSettings.html#setMinimumFontSize(int))
* getMinimumLogicalFontSize(level 1)

int getMinimumLogicalFontSize( )

获取最小的逻辑字体大小。

返回：

int 1到72之间的非负整数

**相关函数：**

* [setMinimumLogicalFontSize(int)](https://developer.android.com/reference/android/webkit/WebSettings.html#setMinimumLogicalFontSize(int))
* getMixedContentMode(level 21)

int getMixedContentMode( )

获取WebView从安全来源加载不安全内容的当前行为。

返回：

int [MIXED\_CONTENT\_NEVER\_ALLOW](https://developer.android.com/reference/android/webkit/WebSettings.html#MIXED_CONTENT_NEVER_ALLOW)， [MIXED\_CONTENT\_ALWAYS\_ALLOW](https://developer.android.com/reference/android/webkit/WebSettings.html#MIXED_CONTENT_ALWAYS_ALLOW)或[MIXED\_CONTENT\_COMPATIBILITY\_MODE](https://developer.android.com/reference/android/webkit/WebSettings.html#MIXED_CONTENT_COMPATIBILITY_MODE)中的一个。

* getOffscreenPreRaster(level 23)

boolean getOffscreenPreRaster( )

获取此WebView是否应当在屏幕上显示栅格。

返回:

boolean 如果此WebView将在屏幕上显示栅格，返回true。

* getPluginState(level 8)

[WebSettings.PluginState](https://developer.android.com/reference/android/webkit/WebSettings.PluginState.html) getPluginState( )

**此方法在API级别18中已弃用。**  
将来不会支持插件，因此不应使用插件。

获取插件的当前状态。

返回：

[WebSettings.PluginState](https://developer.android.com/reference/android/webkit/WebSettings.PluginState.html) [WebSettings.PluginState](https://developer.android.com/reference/android/webkit/WebSettings.PluginState.html)插件状态作为一个值返回。

**相关函数：**

* [setPluginState(WebSettings.PluginState)](https://developer.android.com/reference/android/webkit/WebSettings.html#setPluginState(android.webkit.WebSettings.PluginState))
* getSafeBrowsingEnabled(level 26)

boolean getSafeBrowsingEnabled( )

获取是否启用安全浏览。通过[setSafeBrowsingEnabled(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setSafeBrowsingEnabled(boolean))设置的。

返回：

boolean 如果安全浏览已启用，返回true。

* getSansSerifFontFamily(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getSansSerifFontFamily( )

获取sans-serif字体系列的名称。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) sans-serif字体系列的名称作为字符串返回。

**相关函数：**

* [setSansSerifFontFamily(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setSansSerifFontFamily(java.lang.String))
* getSaveFormData(level 1)

boolean getSaveFormData( )

获取WebView是否保存表单数据。

返回：

boolean WebView保存表单数据，返回true。

**相关函数：**

* [setSaveFormData(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setSaveFormData(boolean))
* getSavePassword(level 1)

boolean getSavePassword( )

**此方法在API级别18中已弃用**

将来的版本不支持在WebView中保存密码。

获取WebView是否保存密码。

返回：

boolean 若果WebView保存密码，返回true。

**相关函数：**

* [setSavePassword(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setSavePassword(boolean))
* getSerifFontFamily(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getSerifFontFamily( )

获取serif字体系列名称。默认是“serif”。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) serif字体系列名称作为字符串返回。

**相关函数：**

* [setSerifFontFamily(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setSerifFontFamily(java.lang.String))
* getStandardFontFamily(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getStandardFontFamily( )

获取标准字体系列名称。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 标准字体系列名称作为字符串返回。

**相关函数：**

* [setStandardFontFamily(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setStandardFontFamily(java.lang.String))
* getTextSize(level 1)

[WebSettings.TextSize](https://developer.android.com/reference/android/webkit/WebSettings.TextSize.html) getTextSize( )

**这种方法在API级别被废弃14**  
使用[**getTextZoom()**](https://developer.android.com/reference/android/webkit/WebSettings.html#getTextZoom())来代替。

获取页面的文本大小。如果文本大小先前用[setTextZoom(int)](https://developer.android.com/reference/android/webkit/WebSettings.html#setTextZoom(int))以百分比形式指定，则会返回[WebSettings.TextSize](https://developer.android.com/reference/android/webkit/WebSettings.TextSize.html)的最接近的匹配项。

返回：

[WebSettings.TextSize](https://developer.android.com/reference/android/webkit/WebSettings.TextSize.html) 文字大小用[WebSettings.TextSize](https://developer.android.com/reference/android/webkit/WebSettings.TextSize.html)的一个值返回

**相关函数：**

* [setTextSize(WebSettings.TextSize)](https://developer.android.com/reference/android/webkit/WebSettings.html#setTextSize(android.webkit.WebSettings.TextSize))
* getTextZoom(level 14)

int getTextZoom( )

获取页面的文本缩放百分比。

返回：

int 页面的文本缩放百分比

**相关函数：**

* [setTextZoom(int)](https://developer.android.com/reference/android/webkit/WebSettings.html#setTextZoom(int))
* getUseWideViewPort(level 1)

boolean getUseWideViewPort( )

获取WebView是否支持“viewport”HTML meta标签。

返回：

boolean 如果WebView支持视图meta标签，返回true。

**相关函数：**

* [setUseWideViewPort(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setUseWideViewPort(boolean))
* getUserAgentString(level 3)

[String](https://developer.android.com/reference/java/lang/String.html) getUserAgentString( )

获取WebView的用户代理字符串。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) WebView的用户代理作为字符串返回。

**相关函数：**

* [setUserAgentString(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setUserAgentString(java.lang.String))
* setAllowContentAccess(level 11)

void setAllowContentAccess(boolean allow)

在WebView中启用或禁用内容URL访问。内容URL访问允许WebView从系统中安装的内容提供者中加载内容。默认已启用。

参数：

allow boolean

* setAllowFileAccess(level 3)

void setAllowFileAccess(boolean allow)

在WebView中启用或禁用文件访问。文件访问默认启用。请注意，这只启用或禁用文件系统访问。资源仍可通过file：/// android\_asset和file：/// android\_res访问。

参数：

allow boolean

* setAllowFileAccessFromFileURLs(level 16)

void setAllowFileAccessFromFileURLs(boolean flag)

设置是否允许在文件系统URL上下文中运行的JavaScript访问来自其他文件系统URL的内容。若是需要最严格的安全策略，应禁用此设置。请注意，如果[getAllowUniversalAccessFromFileURLs()](https://developer.android.com/reference/android/webkit/WebSettings.html#getAllowUniversalAccessFromFileURLs())的值是true，此设置的值被忽略。还要注意，该设置仅影响对文件系统资源的JavaScript访问，其他的对这些资源的访问(例如，来自图像HTML元素)不受影响。为了防止在[ICE\_CREAM\_SANDWICH\_MR1](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ICE_CREAM_SANDWICH_MR1)和更早版本时可能违反同一域策略，您应该明确地将此值设置为false。

[ICE\_CREAM\_SANDWICH\_MR1](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ICE_CREAM_SANDWICH_MR1)和更低版本上默认值为true， [JELLY\_BEAN](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN) 和更高版本上，默认值是false。

参数：

flag boolean：是否允许在文件方案URL上下文中运行的JavaScript访问来自其他文件方案URL的内容

* setAllowUniversalAccessFromFileURLs(level 16)

void setAllowUniversalAccessFromFileURLs(boolean flag)

设置是否允许在文件系统URL上下文中运行的JavaScript访问任何来源的内容。这包括访问来自其他文件格式URL的内容。这是通过 [setAllowFileAccessFromFileURLs(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setAllowFileAccessFromFileURLs(boolean))设置的。若是需要最严格的安全策略，应禁用此设置。请注意，此设置仅影响对文件系统资源的JavaScript访问。其他的对这些资源的访问(例如，来自图像HTML元素)不受影响。为了防止在[ICE\_CREAM\_SANDWICH\_MR1](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ICE_CREAM_SANDWICH_MR1)和更早版本时可能违反同一域策略，您应该明确地将此值设置为false。

[ICE\_CREAM\_SANDWICH\_MR1](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ICE_CREAM_SANDWICH_MR1)和更低版本上默认值为true， [JELLY\_BEAN](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN) 和更高版本上，默认值是false。

参数：

flag boolean：是否允许在文件系统URL上下文中运行的JavaScript访问任何来源的内容

* setAppCacheEnabled(level 7)

void setAppCacheEnabled(boolean flag)

设置是否启用应用程序缓存。默认是false。请注意，为了启用应用程序缓存，还必须通过[setAppCachePath(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setAppCachePath(java.lang.String))提供有效的数据库路径。

参数：

flag boolean：如果WebView应该启用应用程序缓存，则返回true。

* setAppCacheMaxSize(level 7)

void setAppCacheMaxSize(long appCacheMaxSize)

**此方法在API级别18中已弃用。**  
未来的配额将自动进行管理。

设置应用程序缓存内容的最大容量。传递的值将四舍五入到数据库可以支持的最接近的值，因此应该将其视为指导，而不是硬限制。将大小设置为小于当前数据库大小的值不会导致数据库被修剪。默认大小是[MAX\_VALUE](https://developer.android.com/reference/java/lang/Long.html#MAX_VALUE)。建议将[MAX\_VALUE](https://developer.android.com/reference/java/lang/Long.html#MAX_VALUE)设置为默认值。

参数：

appCacheMaxSize long：以字节为单位的最大容量

* setAppCachePath(level 7)

void setAppCachePath([String](https://developer.android.com/reference/java/lang/String.html) appCachePath)

设置应用程序缓存文件的路径。为了启用应用程序缓存，这个方法必须被调用，同事还必须给出一个可以被写入的路径。该方法只能被调用一次：重复的调用会被忽略。

参数;

appCachePath String：应用程序缓存文件的目录的路径

**相关函数：**

* [setAppCacheEnabled(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setAppCacheEnabled(boolean))
* setBlockNetworkImage(level 1)

void setBlockNetworkImage(boolean flag)

设置WebView是否不应从网络加载图片资源(通过http和https URI方案访问的资源)。请注意，除非[getLoadsImagesAutomatically()](https://developer.android.com/reference/android/webkit/WebSettings.html#getLoadsImagesAutomatically())返回 true，否则此方法不起作用。另请注意，即使此标志设置为false ，但用[setBlockNetworkLoads(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setBlockNetworkLoads(boolean)) 禁用所有使用的网络的加载也会不能加载图片。当该设置的值从true到false，通过当前web视图引用网络图像资源会被自动加载。该设置默认是false。

参数：

flag boolean：当为true时， WebView不能从网络加载图片资源。

**相关函数：**

* [setBlockNetworkLoads(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setBlockNetworkLoads(boolean))
* setBlockNetworkLoads(level 8)

void setBlockNetworkLoads(boolean flag)

设置WebView是否不应从网络加载资源。使用[setBlockNetworkImage(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setBlockNetworkImage(boolean))仅不加载图片资源。请注意，如果此设置的值从更改true为false，则WebView当前显示的内容引用的网络资源在调用[reload()](https://developer.android.com/reference/android/webkit/WebView.html#reload())之前不会被加载。如果应用程序没有 [INTERNET](https://developer.android.com/reference/android/Manifest.permission.html#INTERNET)权限，则设置值为false将导致[SecurityException](https://developer.android.com/reference/java/lang/SecurityException.html) 抛出。如果应用程序有[INTERNET](https://developer.android.com/reference/android/Manifest.permission.html#INTERNET)权限，默认值是false。

参数：

flag boolean：当为true时， WebView不能从网络加载资源。

**相关函数：**

* [reload()](https://developer.android.com/reference/android/webkit/WebView.html#reload())
* setBuiltInZoomControls(level 3)

void setBuiltInZoomControls(boolean enabled)

设置WebView是否应使用其内置的缩放机制。内置的缩放机制包括：WebView内容上显示的屏幕缩放控件，以及使用手势控制缩放。[setDisplayZoomControls(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setDisplayZoomControls(boolean))可以控制是否显示这些屏幕控件可以使用，并且默认是false。

内置机制是目前唯一支持的缩放机制，因此建议始终启用此设置。

参数：

enabled boolean：WebView是否应该使用其内置的缩放机制

* setCacheMode(level 1)

void setCacheMode(int mode)

重写缓存的使用方式。缓存的使用方式基于导航类型。对于正常的页面加载，检查缓存并根据需要重新验证内容。在返回时，内容不会重新生效，而是仅从缓存中检索内容。该方法允许客户端通过从[LOAD\_DEFAULT](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_DEFAULT)， [LOAD\_CACHE\_ELSE\_NETWORK](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_CACHE_ELSE_NETWORK)，[LOAD\_NO\_CACHE](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_NO_CACHE)或 [LOAD\_CACHE\_ONLY](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_CACHE_ONLY)这几个类型中指定的一个，以重写这种行为。默认值是[LOAD\_DEFAULT](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_DEFAULT)。

参数：

mode int：使用的模式的值是：[LOAD\_DEFAULT](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_DEFAULT)，[LOAD\_NORMAL](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_NORMAL)，[LOAD\_CACHE\_ELSE\_NETWORK](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_CACHE_ELSE_NETWORK)，[LOAD\_NO\_CACHE](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_NO_CACHE)或[LOAD\_CACHE\_ONLY](https://developer.android.com/reference/android/webkit/WebSettings.html#LOAD_CACHE_ONLY).

* setCursiveFontFamily(level 1)

void setCursiveFontFamily([String](https://developer.android.com/reference/java/lang/String.html) font)

设置草书字体系列名称。默认是“cursive”。

参数：

font String：一个字体系列名称

* setDatabaseEnabled(level 5)

void setDatabaseEnabled(boolean flag)

设置是否启用数据库存储。默认值是false。另请参阅[setDatabasePath(String)](https://developer.android.com/reference/android/webkit/WebSettings.html#setDatabasePath(java.lang.String))来看如何正确设置数据库存储。这个设置是全局的，在一个进程中的所有WebView实例中都是有效的。请注意，在给定进程中加载任何 WebView页面之前，应该只修改此设置，因为WebView实现可能会在此点之后忽略此设置的更改。

参数：

flag boolean：如果是true， WebView应该使用数据库存储

* setDatabasePath(level 5)

void setDatabasePath([String](https://developer.android.com/reference/java/lang/String.html) databasePath)

**此方法在API级别19中已弃用。**  
数据库路径由实现管理，调用此方法将不起作用。

设置数据库存的保存到的路径。为了使数据库存储正常工作，必须调用此方法，并且给出可以写入的路径。该方法只能被调用一次：重复调用会被忽略。

参数；

databasePath String：应该保存数据库的的路径。

* setDefaultFixedFontSize(level 1)

void setDefaultFixedFontSize(int size)

设置默认的固定字体大小。默认值是16。

参数：

size int：1到72之间的非负整数。指定范围之外的任何数字将被锁定。

* setDefaultFontSize(level 1)

void setDefaultFontSize(int size)

设置默认字体大小。默认值是16。

参数：

size int：1到72之间的非负整数。指定范围之外的任何数字将被锁定

* setDefaultTextEncodingName(level 1)

void setDefaultTextEncodingName([String](https://developer.android.com/reference/java/lang/String.html) encoding)

设置解码HTML页面时使用的默认文本编码名称。默认值是“UTF-8”。

参数：

encoding String：文本编码名称

* setDefaultZoom(level 7)

void setDefaultZoom([WebSettings.ZoomDensity](https://developer.android.com/reference/android/webkit/WebSettings.ZoomDensity.html) zoom)

**此方法在API级别19中已弃用。**  
此方法不再受支持，请参阅推荐替代方法的功能文档。

设置页面的默认缩放密度。这必须从UI线程调用。默认是[MEDIUM](https://developer.android.com/reference/android/webkit/WebSettings.ZoomDensity.html#MEDIUM)。不建议将此设置用于新应用程序。如果WebView用于显示面向移动设备的页面，则可以通过调整页面 “meta viewport”标签的“宽度”和“初始缩放”属性来实现预期的效果。对于缺少标签的网页，可以使用[setInitialScale(int)](https://developer.android.com/reference/android/webkit/WebView.html#setInitialScale(int)) 和[setUseWideViewPort(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setUseWideViewPort(boolean))达到效果。

参数：

zoom WebSettings.ZoomDensity：缩放密度

* setDisabledActionModeMenuItems(level 24)

void setDisabledActionModeMenuItems(int menuItems)

根据menuItems标志位禁用动作模式菜单项。

参数：

menuItems int：要禁用的菜单项的整数字段标志位。值可以是0，也可以是 [MENU\_ITEM\_NONE](https://developer.android.com/reference/android/webkit/WebSettings.html#MENU_ITEM_NONE)，[MENU\_ITEM\_SHARE](https://developer.android.com/reference/android/webkit/WebSettings.html#MENU_ITEM_SHARE)，[MENU\_ITEM\_WEB\_SEARCH](https://developer.android.com/reference/android/webkit/WebSettings.html#MENU_ITEM_WEB_SEARCH)或[MENU\_ITEM\_PROCESS\_TEXT](https://developer.android.com/reference/android/webkit/WebSettings.html#MENU_ITEM_PROCESS_TEXT)的组合。

* setDisplayZoomControls(level 11)

void setDisplayZoomControls(boolean enabled)

设置WebView在使用内置缩放机制([setBuiltInZoomControls(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setBuiltInZoomControls(boolean)))时是否应显示屏幕缩放控件。默认是true。

参数：

enabled boolean：如果true，WebView显示屏幕缩放控件

* setDomStorageEnabled(level 7)

void setDomStorageEnabled(boolean flag)

设置是否启用DOM存储API。默认值是false。

参数：

flag boolean：如果true ，WebView使用DOM存储

* setEnableSmoothTransition(level 11)

void setEnableSmoothTransition(boolean enable)

**此方法在API级别17中已弃用。**  
此方法现在已过时，并且将来将成为禁用操作。

设置WebView是否在平移或缩放时启用平滑过渡，和当托管WebView的窗口没有焦点。如果是true，WebView将选择一个解决方案来最大限度地提高性能。例如WebView的内容在转换过程中可能不会更新。如果它是false，WebView将保持其保真度。默认值是false。

参数：

enable boolean

* setFantasyFontFamily(level 1)

void setFantasyFontFamily([String](https://developer.android.com/reference/java/lang/String.html) font)

设置幻想字体系列名称。默认是“fantasy”。

参数：

font String：一个字体系列名称

* setFixedFontFamily(level 1)

void setFixedFontFamily([String](https://developer.android.com/reference/java/lang/String.html) font)

设置固定的字体系列名称。默认是“monospace”。

参数：

font String：一个字体系列名称

* setGeolocationDatabasePath(level 5)

void setGeolocationDatabasePath([String](https://developer.android.com/reference/java/lang/String.html) databasePath)

**此方法在API级别24中已弃用。**  
地理位置数据库由实施管理，调用此方法将不起作用。

设置地理位置数据库应该保存的路径。为了保存Geolocation权限和缓存位置，必须调用此方法，同时给出可写入的路径。

参数：

databasePath String：应该保存数据库的路径。

* setGeolocationEnabled(level 5)

void setGeolocationEnabled(boolean flag)

设置是否启用地理位置。默认是true。

请注意，为了WebView中的页面可以使用Geolocation API，必须满足以下要求：

1.应用程序必须有权限访问该设备的位置，看看[ACCESS\_COARSE\_LOCATION](https://developer.android.com/reference/android/Manifest.permission.html#ACCESS_COARSE_LOCATION)， [ACCESS\_FINE\_LOCATION](https://developer.android.com/reference/android/Manifest.permission.html#ACCESS_FINE_LOCATION);

2.应用程序必须提供[onGeolocationPermissionsShowPrompt(String, GeolocationPermissions.Callback)](https://developer.android.com/reference/android/webkit/WebChromeClient.html#onGeolocationPermissionsShowPrompt(java.lang.String,%20android.webkit.GeolocationPermissions.Callback))回调的实现 ，用以接收通过JavaScript地理位置API发出的请求访问位置的页面通知。

参数:

flag boolean：是否启用地理位置

* setJavaScriptCanOpenWindowsAutomatically(level 1)

void setJavaScriptCanOpenWindowsAutomatically(boolean flag)

设置JavaScript自动打开窗口。这适用于JavaScript功能window.open()。默认是false。

参数：

flag boolean：如果为true ，JavaScript可以自动打开窗口

* setJavaScriptEnabled(level 1)

void setJavaScriptEnabled(boolean flag)

设置WebView启用执行JavaScript。 默认是false。

参数:

flag boolean：如果为true， WebView会执行JavaScript

* setLayoutAlgorithm(level 1)

void setLayoutAlgorithm([WebSettings.LayoutAlgorithm](https://developer.android.com/reference/android/webkit/WebSettings.LayoutAlgorithm.html) l)

设置底层布局算法。这将导致WebView重新布局。默认是[NARROW\_COLUMNS](https://developer.android.com/reference/android/webkit/WebSettings.LayoutAlgorithm.html#NARROW_COLUMNS)。

参数;

l WebSettings.LayoutAlgorithm：使用的布局算法.

* setLightTouchEnabled(level 1)

void setLightTouchEnabled(boolean enabled)

**此方法在API级别18中已被弃用。**  
从[JELLY\_BEAN](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN)起，该设置已过时并且不起作用。

启用轻触式进行选择并激活鼠标悬停。

参数：

enabled boolean

* setLoadWithOverviewMode(level 7)

void setLoadWithOverviewMode(boolean overview)

设置WebView是否以概览模式加载页面，即缩小宽度以适合屏幕的内容。默认是false。

参数：

overview boolean

* setLoadsImagesAutomatically(level 1)

void setLoadsImagesAutomatically(boolean flag)

设置WebView是否自动加载图像资源。请注意，此方法控制所有图像的加载，包括使用数据URI方案嵌入的图像。请注意，如果此设置的值从更改false为true，则由WebView当前显示的内容引用的所有图像资源都会自动加载。默认是true。

参数：

flag boolean：WebView是否应加载图片资源

* setMediaPlaybackRequiresUserGesture(level 17)

void setMediaPlaybackRequiresUserGesture(boolean require)

设置WebView是否需要用户手势才能播放媒体文件。默认是true。

参数：

require boolean：WebView是否需要用户手势来播放媒体文件

* setMinimumFontSize(level 1)

void setMinimumFontSize(int size)

设置最小字体大小。默认值是8。

参数：

size int：1到72之间的非负整数。指定范围之外的任何数字将被锁定。

* setMinimumLogicalFontSize(level 1)

void setMinimumLogicalFontSize(int size)

设置最小逻辑字体大小。默认值是8。

参数：

size int：1到72之间的非负整数。指定范围之外的任何数字将被锁定。

* setMixedContentMode(level 21)

void setMixedContentMode(int mode)

在安全来源尝试从不安全来源加载资源时，配置WebView的行为。默认情况下， [KITKAT](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#KITKAT)或更低版本的默认为[MIXED\_CONTENT\_ALWAYS\_ALLOW](https://developer.android.com/reference/android/webkit/WebSettings.html#MIXED_CONTENT_ALWAYS_ALLOW)。[LOLLIPOP](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#LOLLIPOP)版本默认为 [MIXED\_CONTENT\_NEVER\_ALLOW](https://developer.android.com/reference/android/webkit/WebSettings.html#MIXED_CONTENT_NEVER_ALLOW)。强烈建议不要使用[MIXED\_CONTENT\_ALWAYS\_ALLOW](https://developer.android.com/reference/android/webkit/WebSettings.html#MIXED_CONTENT_ALWAYS_ALLOW)，[MIXED\_CONTENT\_NEVER\_ALLOW](https://developer.android.com/reference/android/webkit/WebSettings.html#MIXED_CONTENT_NEVER_ALLOW)是使用WebView的首选和最安全的操作模式 。

参数：

mode int：其中之一：[MIXED\_CONTENT\_NEVER\_ALLOW](https://developer.android.com/reference/android/webkit/WebSettings.html#MIXED_CONTENT_NEVER_ALLOW)，[MIXED\_CONTENT\_ALWAYS\_ALLOW](https://developer.android.com/reference/android/webkit/WebSettings.html#MIXED_CONTENT_ALWAYS_ALLOW)或[MIXED\_CONTENT\_COMPATIBILITY\_MODE](https://developer.android.com/reference/android/webkit/WebSettings.html#MIXED_CONTENT_COMPATIBILITY_MODE)。

* setNeedInitialFocus(level 1)

void setNeedInitialFocus(boolean flag)

通知WebView在调用[requestFocus(int, android.graphics.Rect)](https://developer.android.com/reference/android/webkit/WebView.html#requestFocus(int,%20android.graphics.Rect))时是否需要设置焦点 。默认值是true。

参数：

flag boolean：如果是true，WebView需要设置节点

* setOffscreenPreRaster(level 23)

void setOffscreenPreRaster(boolean enabled)

当此WebView屏幕外，但是附加到窗口上，设置此WebView是否应显示光栅格。打开此功能可避免在屏幕上对外屏WebView进行动画制作时出现伪影。此模式下的外屏WebView使用更多内存。默认值是false。  
请遵循以下准则来限制内存使用情况：

1.WebView大小不应大于设备屏幕大小。

2.将此模式限制在少量的WebView中。

参数：

enabled boolean

* setPluginState(level 8)

void setPluginState([WebSettings.PluginState](https://developer.android.com/reference/android/webkit/WebSettings.PluginState.html) state)

**此方法在API级别18中已弃用。**  
将来不会支持插件，因此不应使用插件。

通知WebView根据需要启用、禁用或插件。按需模式意味着如果存在可处理内容的插件，则会显示占位符图标而不是插件。当点击占位符时，插件将被启用。默认是 [OFF](https://developer.android.com/reference/android/webkit/WebSettings.PluginState.html#OFF)。

参数：

state WebSettings.PluginState：一个PluginState值

* setRenderPriority(level 1)

void setRenderPriority([WebSettings.RenderPriority](https://developer.android.com/reference/android/webkit/WebSettings.RenderPriority.html) priority)

**此方法在API级别18中已被弃用。**  
不建议调整线程优先级，并且在将来的版本中将不支持此方法。

设置渲染线程的优先级。与其他设置不同，这个设置只需在每个进程中调用一次。默认值是 [NORMAL](https://developer.android.com/reference/android/webkit/WebSettings.RenderPriority.html#NORMAL)。

参数：

priority WebSettings.RenderPriority：渲染线程的优先级。

* setSafeBrowsingEnabled(level 26)

void setSafeBrowsingEnabled(boolean enabled)

设置是否启用安全浏览。安全浏览功能允许WebView通过验证链接来防范恶意软件和网络钓鱼攻击。使用清单标记可以禁用所有WebView的[安全浏览功能](https://developer.android.com/reference/android/webkit/WebView.html)(请阅读[一般安全浏览信息](https://developer.android.com/reference/android/webkit/WebView.html))。清单标记的优先级低于此API。

对于支持它的设备，默认启用安全浏览功能。

参数：

enabled boolean：如果为true，则启用安全浏览功能。

* setSansSerifFontFamily(level 1)

void setSansSerifFontFamily([String](https://developer.android.com/reference/java/lang/String.html) font)

设置sans-serif字体系列名称。默认值是“sans-serif”。

参数：

font String：一个字体系列名称

* setSaveFormData(level 1)

void setSaveFormData(boolean save)

设置WebView是否应保存表单数据。在Android O中，该平台实现了功能齐全的自动填充功能来存储表单数据。因此，Webview表单数据保存功能被禁用。请注意，该功能将继续像旧版Android一样受到支持。这个功能没有任何作用。

参数：

save boolean

* setSavePassword(level 1)

void setSavePassword(boolean save)

**此方法在API级别18中已弃用**  
。将来的版本不支持在WebView中保存密码。

设置WebView是否应保存密码。默认是true。

参数:

save boolean

* setSerifFontFamily(level 1)

void setSerifFontFamily([String](https://developer.android.com/reference/java/lang/String.html) font)

设置serif字体系列名称。默认值是“sans-serif”。

参数：

font String：一个字体系列名称

* setStandardFontFamily(level 1)

void setStandardFontFamily([String](https://developer.android.com/reference/java/lang/String.html) font)

设置标准字体系列名称。默认值是“sans-serif”。

参数：

font String：一个字体系列名称

* setSupportMultipleWindows(level 1)

void setSupportMultipleWindows(boolean support)

设置WebView是否支持多个窗口。如果设置为true，则应用程序必须实现[onCreateWindow(WebView, boolean, boolean, Message)](https://developer.android.com/reference/android/webkit/WebChromeClient.html#onCreateWindow(android.webkit.WebView,%20boolean,%20boolean,%20android.os.Message))。默认是false。

参数：

support boolean：是否支持多个窗口

* setSupportZoom(level 1)

void setSupportZoom(boolean support)

设置WebView是否支持使用屏幕缩放控件和手势进行缩放。应该使用[setBuiltInZoomControls(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setBuiltInZoomControls(boolean))来设置的特定缩放机制。此设置不影响使用[zoomIn()](https://developer.android.com/reference/android/webkit/WebView.html#zoomIn())和 [zoomOut()](https://developer.android.com/reference/android/webkit/WebView.html#zoomOut())方法执行的缩放。默认是true。

参数：

support boolean：若是true，则WebView支持缩放

* setTextSize(level 1)

void setTextSize([WebSettings.TextSize](https://developer.android.com/reference/android/webkit/WebSettings.TextSize.html) t)

**这种方法在API级别被废弃14**  
使用[**setTextZoom(int)**](https://developer.android.com/reference/android/webkit/WebSettings.html#setTextZoom(int))来代替。

设置页面的文字大小。默认是[NORMAL](https://developer.android.com/reference/android/webkit/WebSettings.TextSize.html#NORMAL)。

参数：

t WebSettings.TextSize：文字大小作为[WebSettings.TextSize](https://developer.android.com/reference/android/webkit/WebSettings.TextSize.html)的一个值

* setTextZoom(level 14)

void setTextZoom(int textZoom)

以百分比设置页面的文本缩放。默认值是100。

参数：

textZoom int：文字放大百分比

* setUseWideViewPort(level 1)

void setUseWideViewPort(boolean use)

设置WebView是应该启用对“视图”HTML meta标签的支持还是应该使用宽视图。当设置的值为时false，布局宽度始终设置为设备无关(CSS)。当值为true且页面包含视图meta标签时，将使用标记中指定的宽度值。如果页面不包含标签或不提供宽度，则将使用宽视图。

参数：

use boolean：是否启用对视图meta标签的支持

* setUserAgentString(level 3)

void setUserAgentString([String](https://developer.android.com/reference/java/lang/String.html) ua)

设置WebView的用户代理。如果字符串是null或空的，使用系统默认值。请注意，从[KITKAT](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#KITKAT)的Android版本开始，在加载网页时更改用户代理会导致WebView再次启动加载。

参数：

ua String：新的用户代理，这个值可能是null。

* supportMultipleWindows(level 1)

boolean supportMultipleWindows( )

获取WebView是否支持多个窗口。

返回：

boolean 如果WebView支持多个窗口，返回true。

**相关函数：**

* [setSupportMultipleWindows(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setSupportMultipleWindows(boolean))
* supportZoom(level 1)

boolean supportZoom( )

获取WebView是否支持缩放。

返回：

boolean 如果WebView支持缩放，则返回true。

**相关函数：**

* [setSupportZoom(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setSupportZoom(boolean))

## 1.23 WebStorage

* public class WebStorage

这个类用于管理由[WebView](https://developer.android.com/reference/android/webkit/WebView.html)提供的JavaScript存储 。它管理应用程序缓存、Web SQL数据库和HTML5 Web存储。应用程序缓存为离线Web应用程序提供了一种机制来创建和维护应用程序缓存。应用程序缓存无法设置每个来源的配额。请注意，每个应用程序只能有一个应用程序缓存。Web SQL数据库提供对给定来源私有的存储。与应用程序缓存类似，使用Web SQL数据库可以归因于一个来源。也可以设置每个来源的配额。

**public**

* deleteAllData(level 7)

void deleteAllData( )

清除JavaScript存储当前正在使用的所有存储。这包括应用程序缓存，Web SQL数据库和HTML5 Web存储。

* deleteOrigin(level 7)

void deleteOrigin([String](https://developer.android.com/reference/java/lang/String.html) origin)

按给定的来源清除Application Cache和Web SQL数据库当前正在使用的存储。

参数：

origin String

* getInstance(level 7)

[WebStorage](https://developer.android.com/reference/android/webkit/WebStorage.html) getInstance( )

获取此类的单例实例。

返回：

[WebStorage](https://developer.android.com/reference/android/webkit/WebStorage.html) 单例[WebStorage](https://developer.android.com/reference/android/webkit/WebStorage.html)实例

* getOrigins(level 7)

void getOrigins([ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) < [Map](https://developer.android.com/reference/java/util/Map.html) > callback)

获取当前使用的应用程序缓存或Web SQL数据库的来源。该方法异步运行，结果通过 [ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html)提供。

参数：

callback ValueCallback

* getQuotaForOrigin(level 7)

void getQuotaForOrigin([String](https://developer.android.com/reference/java/lang/String.html) origin， [ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) < [Long](https://developer.android.com/reference/java/lang/Long.html) > callback)

获取给定来源的Web SQL数据库的存储配额。配额以字节为单位给出。该方法异步运行，结果通过 [ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html)提供。请注意，应用程序缓存不针对每个来源实施配额。

参数：

origin String

* getUsageForOrigin(level 7)

void getUsageForOrigin([String](https://developer.android.com/reference/java/lang/String.html) origin， [ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) < [Long](https://developer.android.com/reference/java/lang/Long.html) > callback)

根据给定的来源获取Application Cache和Web SQL数据库当前正在使用的存储量。数量以字节为单位给出。该方法异步运行，结果通过 [ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html)提供。

参数：

origin String

callback ValueCallback

* setQuotaForOrigin(level 7)

void setQuotaForOrigin([String](https://developer.android.com/reference/java/lang/String.html) origin，long quota)

**此方法在API级别18中已被弃用**  
。将来不会支持控制每个来源的配额。

设置给定来源的Web SQL数据库的存储配额。配额以字节为单位指定，并且使用其字符串表示来指定来源。请注意，应用程序缓存不针对每个来源实施配额。

1.23.1 WebStorage.QuotaUpdater

* public static interface WebStorage.QuotaUpdater

**此接口在API级别19中已弃用。**  
此类已过时并不再使用。

封装一个回调函数，用于为JavaScript存储提供新的配额。见 [onExceededDatabaseQuota(String, String, long, long, long, WebStorage.QuotaUpdater)](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.html#onExceededDatabaseQuota(java.lang.String,%20java.lang.String,%20long,%20long,%20long,%20android.webkit.WebStorage.QuotaUpdater))和 [onReachedMaxAppCacheSize(long, long, WebStorage.QuotaUpdater)](https://developer.android.google.cn/reference/android/webkit/WebChromeClient.html#onReachedMaxAppCacheSize(long,%20long,%20android.webkit.WebStorage.QuotaUpdater))。

**public**

* updateQuota(level 5)

abstract void updateQuota(long newQuota)

提供新的配额，以字节为单位指定。

参数：

newQuota long：新配额，以字节为单位

## 1.24 webview

* public class WebView

用来显示网页。这个类是您可以推出自己的网络浏览器或仅在您的活动中显示一些在线内容的基础。它使用WebKit渲染引擎来显示网页，并包括在历史中向前和向后导航，放大和缩小，执行文本搜索等等的方法。

请注意，为了让您的活动访问Internet并在WebView中加载网页，您必须将INTERNET权限添加到Android Manifest文件中：

<uses-permission android:name="android.permission.INTERNET" />

有关更多信息，请阅读 [在WebView中构建Web应用程序](https://developer.android.com/guide/webapps/webview.html)。

**基本用法**

默认情况下，WebView不提供类似浏览器的小部件，不启用JavaScript并忽略网页错误。如果你的目标只是将一些HTML作为你的用户界面的一部分显示出来，这可能是好的; 除了阅读外，用户不需要与网页交互，网页也不需要与用户交互。如果你真的想要一个完整的Web浏览器，那么你可能想用URL Intent调用浏览器应用程序，而不是用WebView显示它。例如：

Uri uri = Uri.parse(“https://www.example.com”);

Intent intent = new Intent(Intent.ACTION\_VIEW，uri);

startActivity(intent);

查看[Intent](https://developer.android.com/reference/android/content/Intent.html)更多信息。

要在您自己的Activity中提供WebView，请<WebView>在布局中包含一个WebView ，或在以下期间将整个Activity窗口设置为WebView [onCreate()](https://developer.android.com/reference/android/app/Activity.html#onCreate(android.os.Bundle))：

 WebView webview = new WebView (this);

 setContentView (webview );

然后加载所需的网页：

//最简单的用法：注意不会抛出异常

//如果加载此页面时发生错误(请参阅下文)。

webview.loadUrl( “https://example.com/”);

//或者，您也可以从HTML字符串中加载：

String summary =“<html> <body>您取得了<b> 192 </ b>分。</ body> </ html>”;

webview.loadData(摘要，“text / html”，null);

// ...但请注意，此HTML可以执行的操作有限制。

//见[loadData(String, String, String)](https://developer.android.com/reference/android/webkit/WebView.html#loadData(java.lang.String,%20java.lang.String,%20java.lang.String))及[loadDataWithBaseURL(String, String, String, String, String)](https://developer.android.com/reference/android/webkit/WebView.html#loadDataWithBaseURL(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))更多信息。

//另请参阅[loadData(String, String, String)](https://developer.android.com/reference/android/webkit/WebView.html#loadData(java.lang.String,%20java.lang.String,%20java.lang.String))有关特殊编码的信息

WebView有几个定制点，您可以在其中添加自己的行为。这些是：

* 创建和设置一个[WebChromeClient](https://developer.android.com/reference/android/webkit/WebChromeClient.html)子类。当某些可能影响浏览器用户界面的事件发生时会调用此类，例如，在此处发送进度更新和JavaScript警报(请参阅[调试任务](https://developer.android.com/guide/developing/debug-tasks.html#DebuggingWebPages))。
* 创建和设置一个[WebViewClient](https://developer.android.com/reference/android/webkit/WebViewClient.html)子类。当影响内容呈现的事情发生时，它会被调用，例如错误或表单提交。您也可以在此拦截URL加载(通过[shouldOverrideUrlLoading()](https://developer.android.com/reference/android/webkit/WebViewClient.html#shouldOverrideUrlLoading(android.webkit.WebView,%20java.lang.String)))。
* 修改[WebSettings](https://developer.android.com/reference/android/webkit/WebSettings.html)，如启用JavaScript [setJavaScriptEnabled()](https://developer.android.com/reference/android/webkit/WebSettings.html#setJavaScriptEnabled(boolean))。
* 使用该[addJavascriptInterface(Object, String)](https://developer.android.com/reference/android/webkit/WebView.html#addJavascriptInterface(java.lang.Object,%20java.lang.String))方法将Java对象注入WebView 。此方法允许您将Java对象注入到页面的JavaScript上下文中，以便可以通过页面中的JavaScript访问它们。

**放大**

要启用内置缩放，请设置 [WebSettings](https://developer.android.com/reference/android/webkit/WebView.html#getSettings())中的[setBuiltInZoomControls(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setBuiltInZoomControls(boolean)) (在API级别引入[CUPCAKE](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#CUPCAKE))。

注意：如果高度或宽度设置为，[WRAP\_CONTENT](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams.html#WRAP_CONTENT)则使用缩放 可能会导致未定义的行为，应避免。

**Cookie和窗口管理**

出于明显的安全原因，您的应用程序拥有自己的缓存，Cookie存储等 - 它不共享浏览器应用程序的数据。

默认情况下，由HTML打开新窗口的请求被忽略。您可以自定义您 [WebChromeClient](https://developer.android.com/reference/android/webkit/WebChromeClient.html)的行为以打开多个窗口，并以您想要的任何方式呈现它们。

当设备方向或任何其他配置发生变化时，活动的标准行为将被销毁并重新创建。这将导致WebView重新加载当前页面。如果你不想这样做，你可以设置你的Activity来处理orientation和keyboardHidden 更改，然后只留下WebView。它会自动适当地重新定位自己。阅读[处理运行时更改](https://developer.android.com/guide/topics/resources/runtime-changes.html)以获取有关如何在运行时处理配置更改的更多信息。

**构建网页以支持不同的屏幕密度**

设备的屏幕密度基于屏幕分辨率。低密度屏幕的每英寸可用像素较少，其中高密度屏幕的每英寸像素更多。屏幕的密度很重要，因为在其他条件相同的情况下，屏幕像素定义其高度和宽度的UI元素(例如按钮)将在较低密度屏幕上显示较大而在较高密度屏幕上显示较小。为了简单起见，Android将所有实际屏幕密度折叠为三种一般密度：高，中和低。

默认情况下，WebView会缩放网页，以便以与中等密度屏幕上的默认外观相匹配的尺寸绘制网页。因此，它在高密度屏幕上应用1.5倍缩放(因为其像素较小)，在低密度屏幕上缩放0.75倍(因为其像素较大)。从API级别[ECLAIR](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ECLAIR)开始，WebView支持DOM，CSS和元标记功能，以帮助您以不同屏幕密度的屏幕为目标。

以下是可用于处理不同屏幕密度的功能摘要：

该window.devicePixelRatioDOM属性。此属性的值指定用于当前设备的默认比例因子。例如，如果值为window.devicePixelRatio“1.0”，则该设备被视为中等密度(mdpi)设备，并且默认缩放未应用于网页; 如果该值是“1.5”，则该设备被认为是高密度设备(hdpi)，并且页面内容被缩放为1.5倍; 如果该值是“0.75”，则该设备被认为是低密度设备(ldpi)，并且该内容被缩放为0.75x。

该-webkit-device-pixel-ratioCSS媒体查询。使用此选项指定要使用此样式表的屏幕密度。相应的值应该是“0.75”，“1”或“1.5”，以表明这些样式分别适用于低密度，中等密度或高密度屏幕的设备。例如：

<link rel =“stylesheet”media =“screen and(-webkit-device-pixel-ratio：1.5)”href =“hdpi.css”/>

该hdpi.css样式表仅用于的屏幕的像素比例是1.5时。

**HTML5视频支持**

为了在您的应用程序中支持HTML5视频，您需要打开硬件加速。

**全屏支持**

为了支持全屏 ， 对于视频或其他HTML内容 ，您需要设置 [WebChromeClient](https://developer.android.com/reference/android/webkit/WebChromeClient.html)，并实现[onShowCustomView(View, WebChromeClient.CustomViewCallback)](https://developer.android.com/reference/android/webkit/WebChromeClient.html#onShowCustomView(android.view.View,%20android.webkit.WebChromeClient.CustomViewCallback)) 和[onHideCustomView()](https://developer.android.com/reference/android/webkit/WebChromeClient.html#onHideCustomView())。如果缺少这两种方法中的任何一种，则Web内容将不被允许进入全屏。可选地，您可以实现 [getVideoLoadingProgressView()](https://developer.android.com/reference/android/webkit/WebChromeClient.html#getVideoLoadingProgressView())自定义视频加载时显示的视图。

**HTML5地理位置API支持**

对于面向Android N及更高版本(API级别> [M](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#M))的应用程序，地理位置api仅在安全来源(如https)上受支持。对于这样的应用程序，在非安全源地址上对地理位置api的请求会自动被拒绝，而不会调用相应的[onGeolocationPermissionsShowPrompt(String, GeolocationPermissions.Callback)](https://developer.android.com/reference/android/webkit/WebChromeClient.html#onGeolocationPermissionsShowPrompt(java.lang.String,%20android.webkit.GeolocationPermissions.Callback)) 方法。

**布局大小**

建议将WebView布局高度设置为固定值或 [MATCH\_PARENT](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams.html#MATCH_PARENT)不使用 [WRAP\_CONTENT](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams.html#WRAP_CONTENT)。当[MATCH\_PARENT](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams.html#MATCH_PARENT) 用于高度时，WebView的父视图都不应使用 [WRAP\_CONTENT](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams.html#WRAP_CONTENT)布局高度，因为这可能导致视图的大小不正确。

设置WebView的高度以[WRAP\_CONTENT](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams.html#WRAP_CONTENT) 启用以下行为：

* HTML主体布局高度设置为固定值。这意味着具有相对于HTML正文的高度的元素的尺寸可能不正确。
* 对于应用程序定位[KITKAT](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#KITKAT)和较早的SDK，为了保持向后兼容性，HTML视图标记将被忽略。

[WRAP\_CONTENT](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams.html#WRAP_CONTENT)不支持 使用布局宽度。如果使用这样的宽度，WebView将尝试使用父视图宽度。

**度量**

当用户同意时，WebView可能会将匿名诊断数据上传到Google。这些数据有助于Google改进WebView。针对实例化WebView的每个应用程序，这会基于每个应用程序收集数据。一个单独的应用程序可以通过在其manifest的<application>元素中加入以下标签来退出此功能：

<manifest>

< application >

...

<meta-data android：name =“android.webkit.WebView.MetricsOptOut”

android：value =“true”/>

</ application >

</manifest>

只有在用户同意并且应用程序未选择退出的情况下，数据才会上传。

**安全浏览**

借助安全浏览功能，WebView将阻止恶意网址并向用户显示警告用户界面，以便他们能够安全地返回。

安全浏览默认情况下在支持它的设备上启用。如果您的应用需要禁用所有WebView的安全浏览，则可以在manifest的<application>元素中执行此操作：

< manifest >

< application >

...

<meta-data android：name =“android.webkit.WebView.EnableSafeBrowsing”

android：value =“false”/>

</ application >

</manifest>

否则，请参阅[setSafeBrowsingEnabled(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setSafeBrowsingEnabled(boolean))。

**概要**

嵌套类：

interface [WebView.FindListener](https://developer.android.com/reference/android/webkit/WebView.FindListener.html)

接口来监听查找结果。

class [WebView.HitTestResult](https://developer.android.com/reference/android/webkit/WebView.HitTestResult.html)

interface [WebView.PictureListener](https://developer.android.com/reference/android/webkit/WebView.PictureListener.html)

此接口在API级别12中已弃用。此接口现在已过时。

class [WebView.VisualStateCallback](https://developer.android.com/reference/android/webkit/WebView.VisualStateCallback.html)

提供回调接口[postVisualStateCallback(long, WebView.VisualStateCallback)](https://developer.android.com/reference/android/webkit/WebView.html#postVisualStateCallback(long,%20android.webkit.WebView.VisualStateCallback))用于接收关于视觉状态的通知

class [WebView.WebViewTransport](https://developer.android.com/reference/android/webkit/WebView.WebViewTransport.html)

用于跨线程边界返回WebView的对象。

Public constructor：

[WebView](https://developer.android.com/reference/android/webkit/WebView.html#WebView(android.content.Context))([Context](https://developer.android.com/reference/android/content/Context.html) context)

用活动上下文对象构造一个新的WebView。

[WebView](https://developer.android.com/reference/android/webkit/WebView.html#WebView(android.content.Context,%20android.util.AttributeSet))([Context](https://developer.android.com/reference/android/content/Context.html) context, [AttributeSet](https://developer.android.com/reference/android/util/AttributeSet.html) attrs)

用布局参数构造一个新的WebView。

[WebView](https://developer.android.com/reference/android/webkit/WebView.html#WebView(android.content.Context,%20android.util.AttributeSet,%20int))([Context](https://developer.android.com/reference/android/content/Context.html) context, [AttributeSet](https://developer.android.com/reference/android/util/AttributeSet.html) attrs, int defStyleAttr)

使用布局参数和默认样式构造一个新的WebView。

[WebView](https://developer.android.com/reference/android/webkit/WebView.html#WebView(android.content.Context,%20android.util.AttributeSet,%20int,%20int))([Context](https://developer.android.com/reference/android/content/Context.html) context, [AttributeSet](https://developer.android.com/reference/android/util/AttributeSet.html) attrs, int defStyleAttr, int defStyleRes)

使用布局参数和默认样式构造一个新的WebView。

[WebView](https://developer.android.com/reference/android/webkit/WebView.html#WebView(android.content.Context,%20android.util.AttributeSet,%20int,%20int))([Context](https://developer.android.com/reference/android/content/Context.html) context, [AttributeSet](https://developer.android.com/reference/android/util/AttributeSet.html) attrs, int defStyleAttr, int defStyleRes)

使用布局参数和默认样式构造一个新的WebView。

**常量**

* RENDERER\_PRIORITY\_BOUND(level 26)

int RENDERER\_PRIORITY\_BOUND

与此WebView关联的呈现器与服务的默认优先级绑定。使用[setRendererPriorityPolicy(int, boolean)](https://developer.android.com/reference/android/webkit/WebView.html#setRendererPriorityPolicy(int,%20boolean))。

常数值：1(0x00000001)

* RENDERER\_PRIORITY\_IMPORTANT(level 26)

int RENDERER\_PRIORITY\_IMPORTANT

与此WebView关联的呈现器与绑定 [BIND\_IMPORTANT](https://developer.android.com/reference/android/content/Context.html#BIND_IMPORTANT)。使用[setRendererPriorityPolicy(int, boolean)](https://developer.android.com/reference/android/webkit/WebView.html#setRendererPriorityPolicy(int,%20boolean))。

常量值：2(0x00000002)

* RENDERER\_PRIORITY\_WAIVED(level 26)

int RENDERER\_PRIORITY\_WAIVED

与此WebView关联的呈现器与绑定 [BIND\_WAIVE\_PRIORITY](https://developer.android.com/reference/android/content/Context.html#BIND_WAIVE_PRIORITY)。在此优先级下， [WebView](https://developer.android.com/reference/android/webkit/WebView.html)渲染器将​​成为内存不足的强大目标。使用[setRendererPriorityPolicy(int, boolean)](https://developer.android.com/reference/android/webkit/WebView.html#setRendererPriorityPolicy(int,%20boolean))。

常量值：0(0x00000000)

* SCHEME\_GEO(level 1)

String SCHEME\_GEO

地图地址的URI方案。

常量值：“geo：0,0？q =”

* SCHEME\_MAILTO(level 1)

String SCHEME\_MAILTO

电子邮件地址的URI方案。

常量值：“mailto：”

* SCHEME\_TEL(level 1)

String SCHEME\_TEL

电话号码的URI方案。

常数值：“tel：

**public**

* addJavascriptInterface(level 1)

void addJavascriptInterface([Object](https://developer.android.com/reference/java/lang/Object.html) object，[String](https://developer.android.com/reference/java/lang/String.html) name)

将提供的Java对象注入此WebView。该对象使用提供的名称注入主框架的JavaScript上下文中。这允许从JavaScript访问Java对象的方法。对于面向API级别[JELLY\_BEAN\_MR1](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN_MR1) 和更高级别的应用程序，只能[JavascriptInterface](https://developer.android.com/reference/android/webkit/JavascriptInterface.html)使用JavaScript访问注释过的public 。对于针对API级别[JELLY\_BEAN](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN)或更低级别的应用程序，可以访问所有public(包括继承的方法)，有关含义，请参阅下面的重要安全说明。请注意，注入的对象在下次(重新)加载页面之前不会出现在JavaScript中。在注入对象之前应该启用JavaScript。例如：

class JsObject {

@JavascriptInterface

public String toString() { return "injectedObject"; }

}

webview.getSettings().setJavaScriptEnabled(true);

webView.addJavascriptInterface(new JsObject(), "injectedObject");

webView.loadData("", "text/html", null);

webView.loadUrl("javascript:alert(injectedObject.toString())");

参数：

object Object：注入此WebView的JavaScript上下文的Java对象。null值被忽略

name String：用于在JavaScript中公开对象的名称

* canGoBack(level 1)

boolean canGoBack( )

获取此WebView是否具有回退历史项目。

返回：

boolean 如果此WebView具有后退历史项目，返回true

* canGoBackOrForward(level 1)

boolean canGoBackOrForward(int steps)

获取页面是否可以返回。

返回：

boolean

参数：

steps int：历史的步骤数量

* canGoForward(level 1)

boolean canGoForward( )

获取此WebView是否具有向前历史记录项目。

返回：

boolean true 如果此WebView具有向前历史记录项目

* canZoomIn(level 11)

***boolean canZoomIn( )***

**此方法在API级别17中已弃用。**  
由于Web渲染和UI线程之间的竞争条件，此方法容易出现不准确性;可以用 [**onScaleChanged(WebView, float, float)**](https://developer.android.com/reference/android/webkit/WebViewClient.html#onScaleChanged(android.webkit.WebView, float, float))**代替**。

获取此WebView是否可以放大。

返回：

boolean 如果这个WebView可以放大，返回true。

* canZoomOut(level 11)

boolean canZoomOut( )

**此方法在API级别17中已弃用。**  
由于Web渲染和UI线程之间的竞争条件，此方法容易出现不准确性; 可以用 [**onScaleChanged(WebView, float, float)**](https://developer.android.com/reference/android/webkit/WebViewClient.html#onScaleChanged(android.webkit.WebView, float, float))**代替**。

获取此WebView是否可以缩小。

返回：

boolean 如果这个WebView可以缩小，返回true。

* capturePicture(level 1)

[Picture](https://developer.android.com/reference/android/graphics/Picture.html) capturePicture( )

**这种方法在API级别19已被废弃**  
使用[**onDraw(Canvas)**](https://developer.android.com/reference/android/webkit/WebView.html#onDraw(android.graphics.Canvas))获得的WebView的位图快照，或用 [**saveWebArchive(String)**](https://developer.android.com/reference/android/webkit/WebView.html#saveWebArchive(java.lang.String))**。**

获取此WebView当前内容的新图片。该图片并不限于此WebView当前显示的区域。此外，图片是一个静态副本，不受对所显示内容的更改的影响。请注意，由于内部更改，对于介于[HONEYCOMB](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB)和 [ICE\_CREAM\_SANDWICH](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#ICE_CREAM_SANDWICH)之间的API级别 ，图片不包含固定位置元素和可滚动div。

返回：

Picture 获取此WebView当前内容的图片

* clearCache(level 1)

void clearCache(boolean includeDiskFiles)

清除资源缓存。请注意，缓存是按应用程序进行的，因此这将清除整个应用程序的缓存。

参数：

includeDiskFiles boolean：如果false只清除RAM缓存

* clearClientCertPreferences(level 21)

void clearClientCertPreferences([Runnable](https://developer.android.com/reference/java/lang/Runnable.html) onCleared)

清除客户端证书首选项。请注意，WebView在接收到[ACTION\_STORAGE\_CHANGED](https://developer.android.com/reference/android/security/KeyChain.html#ACTION_STORAGE_CHANGED)意图时会自动清除这些首选项 。首选项由应用程序创建的所有WebView共享。

参数：

onCleared Runnable：清除客户端证书时将调用。runnable将在UI线程中调用。这个值可能是null。

* clearFormData(level 1)

void clearFormData( )

如果存在，从当前关注的表单域中移除自动填充弹出框。请注意，这只会影响自动填充弹出窗口的显示，它不会从此WebView的存储中移除任何已保存的表单数据。

* clearHistory(level 1)

void clearHistory( )

清除该WebView后退/前进列表。

* clearMatches(level 3)

void clearMatches( )

清除由[findAllAsync(String)](https://developer.android.com/reference/android/webkit/WebView.html#findAllAsync(java.lang.String))创建的文本匹配 。

* clearSslPreferences(level 1)

void clearSslPreferences( )

清除存储的SSL首选项表，用以响应处理SSL证书错误。

* ClearView(level 1)

void clearView( )

**此方法在API级别18中已弃用。**  
使用WebView.loadUrl(“about：blank”)可靠地重置视图状态并释放页面资源(包括任何正在运行的JavaScript)。

清除此WebView，以便onDraw( )只会绘制白色背景，如果MeasureSpec不是MeasureSpec.EXACTLY，onMeasure( )将返回0。

* computeScroll(level 1)

void computeScroll( )

由父级调用，以请求子级在必要时更新mScrollX和mScrollY的值。这通常会在子级使用[Scroller](https://developer.android.com/reference/android/widget/Scroller.html) 对象动画滚动时完成。

* copyBackForwardList(level 1)

[WebBackForwardList](https://developer.android.com/reference/android/webkit/WebBackForwardList.html) copyBackForwardList( )

获取此WebView的WebBackForwardList。这包含用于查询历史堆栈的每个项目的后退/前进列表。这是私有WebBackForwardList的副本，因此它只包含当前状态的快照。多次调用此方法可能会返回不同的对象。从此方法返回的对象将不会更新以反映任何新状态。

返回：

WebBackForward

* createPrintDocumentAdapter(level 21)

[PrintDocumentAdapter](https://developer.android.com/reference/android/print/PrintDocumentAdapter.html) createPrintDocumentAdapter([String](https://developer.android.com/reference/java/lang/String.html) documentName)

创建一个PrintDocumentAdapter对象来提供此WebView的打印内容。该适配器通过将WebView内容转换为PDF流来工作。WebView无法在转换过程中绘制，任何此类绘制均未定义。建议使用专用的屏幕外WebView进行打印。如有必要，应用程序可能会暂时隐藏可见的WebView。

参数：

documentName String：打印文档的面向用户的名称。

返回：

[PrintDocumentAdapter](https://developer.android.com/reference/android/print/PrintDocumentAdapter.html)

* createPrintDocumentAdapter(level 19)

[PrintDocumentAdapter](https://developer.android.com/reference/android/print/PrintDocumentAdapter.html) createPrintDocumentAdapter( )

**此方法在API级别21中已弃用。**  
使用[**createPrintDocumentAdapter(String)**](https://developer.android.com/reference/android/webkit/WebView.html#createPrintDocumentAdapter(java.lang.String))**，**该方法需要用户提供打印文档名称。

返回：

[PrintDocumentAdapter](https://developer.android.com/reference/android/print/PrintDocumentAdapter.html)

* createWebMessageChannel(level 23)

[WebMessagePort []](https://developer.android.com/reference/android/webkit/WebMessagePort.html) createWebMessageChannel( )

创建与JS通信的消息通道并返回表示此消息通道端点的消息端口。[这里](https://html.spec.whatwg.org/multipage/comms.html#messagechannel)描述[了](https://html.spec.whatwg.org/multipage/comms.html#messagechannel)HTML5消息通道的功能。返回的消息通道已处于启动状态。

返回：

[WebMessagePort[]](https://developer.android.com/reference/android/webkit/WebMessagePort.html) 构成消息通道的两个消息端口。

* destroy(level 1)

void destroy( )

销毁这个WebView的内部状态。在从视图系统中删除此WebView后，应该调用此方法。销毁后，此WebView上不会调用其他任何方法。

* dispatchKeyEvent(level 1)

boolean dispatchKeyEvent([KeyEvent](https://developer.android.com/reference/android/view/KeyEvent.html) event)

将事件分派到焦点路径上的下一个视图。此路径从视图树的顶部向下延伸到当前聚焦的视图。如果这个视图已经被聚焦，它会发送给自己。否则，它将沿着焦点路径调度下一个节点。这个方法也会触发所有的事件的监听器。

参数：

event KeyEvent：要发送的事件。

返回：

boolean 如果处理该事件，则为true，否则为false。

* documentHasImages(level 1)

void documentHasImages([Message](https://developer.android.com/reference/android/os/Message.html) response)

查询文档以查看它是否包含任何片的引用。如果找到图像，则会将arg1设置为1，如果文档未引用任何图像，则将调度消息对象。

参数：

response Message：将与结果一起发送的消息。

* enableSlowWholeDocumentDraw(level 21)

void enableSlowWholeDocumentDraw( )

对于面向L版本的应用程序，WebView具有新的默认行为，通过智能地选择需要绘制的HTML文档的来减少内存占用并提高性能。这些优化对开发者来说是透明的。但是，在某些情况下，应用程序开发人员可能希望禁用它们：

1.当应用程序使用[onDraw(Canvas)](https://developer.android.com/reference/android/webkit/WebView.html#onDraw(android.graphics.Canvas))自己的绘图并访问页面可见部分之外的部分页面时。

2.当应用程序[capturePicture()](https://developer.android.com/reference/android/webkit/WebView.html#capturePicture())用来捕获非常大的HTML文档时。请注意，capturePicture是一个弃用的API。

启用绘制整个HTML文档具有显着的性能成本。应该在创建任何WebView之前调用此方法。

* evaluateJavascript(level 19)

void evaluateJavascript([String](https://developer.android.com/reference/java/lang/String.html) script， [ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) < [String](https://developer.android.com/reference/java/lang/String.html) > resultCallback)

在当前显示的页面上下文中异步评估JavaScript。如果非null，| resultCallback | 将被该执行中返回的任何结果调用。必须在UI线程上调用此方法，并且将在UI线程上进行回调。

兼容性说明。应用程序定位[N](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#N)或更高版本，来自空WebView的JavaScript状态不再像导航一样持续存在[loadUrl(String)](https://developer.android.com/reference/android/webkit/WebView.html#loadUrl(java.lang.String))。例如， [loadUrl(String)](https://developer.android.com/reference/android/webkit/WebView.html#loadUrl(java.lang.String))之前定义的全局变量和函数将不存在于加载的页面中。

参数：

script String：要执行的JavaScript。

resultCallback ValueCallback：当脚本执行完成并执行结果(如果有)时要调用的回调。如果需要通知，则可以为null。

* findAddress(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) findAddress([String](https://developer.android.com/reference/java/lang/String.html) addr)

获取由物理位置的地址组成的第一个子字符串。目前，只有在美国的地址被检测到，并且包括：

一个房子号码

一个街道名称

街道类型(道路，圆形等)，不论是拼写还是缩写

一个城市的名字

一个州或领土，要么拼写出来，要么是两个字母的缩写

一个可选的5位或9位数的邮政编码

所有名称必须正确大写，并且邮政编码(如果存在)必须对该州有效。街道类型必须是标准的USPS拼写或缩写。国家或地区也必须使用USPS标准拼写或缩写。门牌号码不得超过五位数字。

参数：

addr String: 搜索地址的字符串

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 地址，或者如果没有找到地址， null

* findAll(level 3)

int findAll([String](https://developer.android.com/reference/java/lang/String.html) find)

**此方法在API级别16中**  
[**findAllAsync(String)**](https://developer.android.com/reference/android/webkit/WebView.html#findAllAsync(java.lang.String))已**被弃用**。

在页面上查找所有需要查找实例并突出显示它们。通知注册过[WebView.FindListener](https://developer.android.com/reference/android/webkit/WebView.FindListener.html)的地方。

参数：

find String: 要查找的字符串

返回：

Int 找到的字符串“find”的出现次数

**相关函数：**

* [setFindListener(WebView.FindListener)](https://developer.android.com/reference/android/webkit/WebView.html#setFindListener(android.webkit.WebView.FindListener))
* findAllAsync(level 16)

void findAllAsync([String](https://developer.android.com/reference/java/lang/String.html) find)

在页面上查找所有要查找的实例，并异步突出显示它们。通知任何注册[WebView.FindListener](https://developer.android.com/reference/android/webkit/WebView.FindListener.html)的地方。对此的连续调用将取消所有待处理的搜索。

参数：

find String: 要查找的字符串

**相关函数：**

* [setFindListener(WebView.FindListener)](https://developer.android.com/reference/android/webkit/WebView.html#setFindListener(android.webkit.WebView.FindListener))
* findFocus(level 1)

[View](https://developer.android.com/reference/android/view/View.html) findFocus( )

在当前拥有焦点的视图中的层次结构中查找视图。

* findNext(level 3)

void findNext(boolean forward)

突出显示并滚动到找到的下一个匹配项 [findAllAsync(String)](https://developer.android.com/reference/android/webkit/WebView.html#findAllAsync(java.lang.String))，并根据需要环绕页面边界。通知任何注册[WebView.FindListener](https://developer.android.com/reference/android/webkit/WebView.FindListener.html)。如果[findAllAsync(String)](https://developer.android.com/reference/android/webkit/WebView.html#findAllAsync(java.lang.String)) 还没有被调用，或者[clearMatches()](https://developer.android.com/reference/android/webkit/WebView.html#clearMatches())自上次查找操作后被调用，则此函数不执行任何操作。

参数：

forward boolean：搜索的方向

**相关函数：**

* [setFindListener(WebView.FindListener)](https://developer.android.com/reference/android/webkit/WebView.html#setFindListener(android.webkit.WebView.FindListener))
* flingScroll(level 1)

void flingScroll(int vx， int vy)

参数：

vx int

vy int

* freeMemory(level 7)

void freeMemory( )

**此方法在API级别19中已弃用。**  
内存缓存在不再需要时会自动丢弃，并且会响应系统内存压力。

通知此WebView内存不足以释放任何可用内存。

* getAccessibilityClassName(level 23)

[CharSequence](https://developer.android.com/reference/java/lang/CharSequence.html) getAccessibilityClassName( )

返回此对象的类的名称用于辅助功能。子类正在实现的东西应该被视为一个全新的视图类，当它被可访问性使用时，子类只应该重写这个类，与其它父类无关。这用于填写 [AccessibilityNodeInfo.setClassName](https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.html#setClassName(java.lang.CharSequence))。

返回：

[CharSequence](https://developer.android.com/reference/java/lang/CharSequence.html)

* getAccessibilityNodeProvider(level 16)

[AccessibilityNodeProvider](https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeProvider.html) getAccessibilityNodeProvider( )

获取用于管理以此视图为根的虚拟视图层次结构的提供者，并将其报告为[AccessibilityService](https://developer.android.com/reference/android/accessibilityservice/AccessibilityService.html)探索窗口内容。

如果此方法返回实例，则此实例负责管理 [AccessibilityNodeInfo](https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.html)描述以此View为根的虚拟子树，包括代表View本身的虚拟子树。同样，返回的实例负责在任何虚拟视图或根视图本身上执行辅助功能操作。

如果[View.AccessibilityDelegate](https://developer.android.com/reference/android/view/View.AccessibilityDelegate.html)已经通过调用 [setAccessibilityDelegate(AccessibilityDelegate)](https://developer.android.com/reference/android/view/View.html#setAccessibilityDelegate(android.view.View.AccessibilityDelegate))指定它[getAccessibilityNodeProvider(View)](https://developer.android.com/reference/android/view/View.AccessibilityDelegate.html#getAccessibilityNodeProvider(android.view.View)) 负责处理这个呼叫。

返回：

[AccessibilityNodeProvider](https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeProvider.html) 提供者。

* getCertificate(level 1)

[SslCertificate](https://developer.android.com/reference/android/net/http/SslCertificate.html) getCertificate( )

获取主要顶层页面的SSL证书或者null没有证书(该网站不安全)。

返回：

[SslCertificate](https://developer.android.com/reference/android/net/http/SslCertificate.html) 顶层页面的SSL证书，这个值可能是null。

* getContentHeight(level 1)

int getContentHeight( )

获取HTML内容的高度。

返回：

int 获取HTML内容的高度

* getCurrentWebViewPackage(level 26)

[PackageInfo](https://developer.android.com/reference/android/content/pm/PackageInfo.html) getCurrentWebViewPackage( )

如果WebView已经加载到当前进程中，则此方法将返回用于加载它的程序包。否则，返回将在现在加载WebView时使用的包; 这不会导致WebView被加载，所以这些信息可能会随时过时。WebView程序包在当前WebView程序包更新，禁用或卸载时发生更改。它也可以通过开发者的设置进行更改。如果WebView程序包发生更改，则加载WebView的任何应用程序进程都将被终止。下次应用程序启动并加载WebView时，它将使用新的WebView包。

* getFavicon(level 1)

Bitmap getFavicon( )

获取当前页面的图标。调用WebViewClient.onReceivedIcon之前，获取的都是当前页面的图标。

返回：

[Bitmap](https://developer.android.com/reference/android/graphics/Bitmap.html) 当前页面的图标

* getHandler(level 1)

[Handler](https://developer.android.com/reference/android/os/Handler.html) getHandler( )

返回：

[Handler](https://developer.android.com/reference/android/os/Handler.html) 与运行View的线程关联的处理程序。该处理程序可用于在UI事件队列中抽取事件。

* getHitTestResult(level 1)

[WebView.HitTestResult](https://developer.android.com/reference/android/webkit/WebView.HitTestResult.html) getHitTestResult( )

基于当前游标节点获取HitTestResult。如果找到HTML :: a标记并且锚点具有非JavaScript URL，则HitTestResult类型将设置为SRC\_ANCHOR\_TYPE，并在“extra”字段中设置URL。如果锚点没有URL或者它是JavaScript URL，则该类型将为UNKNOWN\_TYPE，并且，URL必须通过异步检索 [requestFocusNodeHref(Message)](https://developer.android.com/reference/android/webkit/WebView.html#requestFocusNodeHref(android.os.Message))。如果找到HTML :: img标签，则将HitTestResult类型设置为IMAGE\_TYPE，并在“extra”字段中设置URL。一种类型的SRC\_IMAGE\_ANCHOR\_TYPE指示具有将图像作为子节点的URL的锚点。如果找到电话号码，则将HitTestResult类型设置为PHONE\_TYPE，并在HitTestResult的“extra”字段中设置电话号码。如果找到地图地址，则HitTestResult类型设置为GEO\_TYPE，地址在HitTestResult的“extra”字段中设置。如果找到电子邮件地址，则HitTestResult类型设置为EMAIL\_TYPE，并且电子邮件设置在HitTestResult的“extra”字段中。否则，HitTestResult类型设置为UNKNOWN\_TYPE。

返回：

[WebView.HitTestResult](https://developer.android.com/reference/android/webkit/WebView.HitTestResult.html)

* getHttpAuthUsernamePassword(level 1)

[String []](https://developer.android.com/reference/java/lang/String.html) getHttpAuthUsernamePassword([String](https://developer.android.com/reference/java/lang/String.html) host， [String](https://developer.android.com/reference/java/lang/String.html) realm)

**这种方法在API级别被废弃26.**  
使用[**getHttpAuthUsernamePassword(String, String)**](https://developer.android.com/reference/android/webkit/WebViewDatabase.html#getHttpAuthUsernamePassword(java.lang.String, java.lang.String))代替

从[WebViewDatabase](https://developer.android.com/reference/android/webkit/WebViewDatabase.html)实例中检索给定主机和领域的HTTP身份验证凭证。

参数：

host String：凭据适用的主机

realm String：证书适用的领域

返回：

[String[]](https://developer.android.com/reference/java/lang/String.html) 作为字符串数组的凭据(如果找到)。第一个元素是用户名，第二个元素是密码。如果没有找到，返回null。

* getOriginalUrl(level 3)

[String](https://developer.android.com/reference/java/lang/String.html) getOriginalUrl( )

获取当前页面的原始URL。这并不总是与传递给WebViewClient.onPageStarted的URL相同，因为尽管该URL的加载已经开始，但当前页面可能没有更改。此外，可能会有重定向导致与最初请求的URL不同。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 最初为当前页面请求的网址

* getProgress(level 1)

int getProgress( )

获取当前页面的进度。

int 当前页面在0到100之间的进度.

* getRendererPriorityWaivedWhenNotVisible(level 26)

boolean getRendererPriorityWaivedWhenNotVisible( )

返回此WebView是否请求[RENDERER\_PRIORITY\_WAIVED](https://developer.android.com/reference/android/webkit/WebView.html#RENDERER_PRIORITY_WAIVED)不可见的优先级 。

返回：

boolean 该WebView是否请求[RENDERER\_PRIORITY\_WAIVED](https://developer.android.com/reference/android/webkit/WebView.html#RENDERER_PRIORITY_WAIVED)不可见的优先级 。

* getRendererRequestedPriority(level 26)

int getRendererRequestedPriority( )

为此WebView获取渲染器优先级。

返回：

int 请求的渲染器优先级策略。值是[RENDERER\_PRIORITY\_WAIVED](https://developer.android.com/reference/android/webkit/WebView.html#RENDERER_PRIORITY_WAIVED)，[RENDERER\_PRIORITY\_BOUND](https://developer.android.com/reference/android/webkit/WebView.html#RENDERER_PRIORITY_BOUND)或[RENDERER\_PRIORITY\_IMPORTANT](https://developer.android.com/reference/android/webkit/WebView.html#RENDERER_PRIORITY_IMPORTANT)

* getSafeBrowsingPrivacyPolicyUrl(level 27)

[Uri](https://developer.android.com/reference/android/net/Uri.html) getSafeBrowsingPrivacyPolicyUrl( )

返回指向安全浏览报告隐私政策的网址。

返回：

[Uri](https://developer.android.com/reference/android/net/Uri.html) 指向隐私策略文档的url可以显示给用户。这个值永远不会null。

* getScale(level 1)

float getScale( )

**此方法在API级别17中已弃用。**  
由于Web渲染和UI线程之间的竞争条件，此方法容易出现不准确性; 可以用 [**onScaleChanged(WebView, float, float)**](https://developer.android.com/reference/android/webkit/WebViewClient.html#onScaleChanged(android.webkit.WebView, float, float))**代替**。

获取此WebView的当前比例。

返回：

float 当前的比例。

* getSettings(level 1)

[WebSettings](https://developer.android.com/reference/android/webkit/WebSettings.html) getSettings( )

获取用于控制此WebView设置的WebSettings对象。

返回:

[WebSettings](https://developer.android.com/reference/android/webkit/WebSettings.html) 一个WebSettings对象，可用于控制此WebView的设置

* getTitle(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getTitle( )

获取当前页面的标题。这是调用WebViewClient.onReceivedTitle之前当前页面的标题。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 获取当前页面的标题

* getURL(level 1)

[String](https://developer.android.com/reference/java/lang/String.html) getUrl( )

获取当前页面的URL。这并不总是与传递给WebViewClient.onPageStarted的URL相同，因为尽管该URL的加载已经开始，但当前页面可能没有更改。

返回：

[String](https://developer.android.com/reference/java/lang/String.html) 当前页面的URL

* getWebChromeClient(level 26)

[WebChromeClient](https://developer.android.com/reference/android/webkit/WebChromeClient.html) getWebChromeClient( )

获取chrome处理程序。

返回：

[WebChromeClient](https://developer.android.com/reference/android/webkit/WebChromeClient.html) WebChromeClient，当尚未设置时，返回null

**相关函数：**

* [setWebChromeClient(WebChromeClient)](https://developer.android.com/reference/android/webkit/WebView.html#setWebChromeClient(android.webkit.WebChromeClient))
* getWebViewClient(level 26)

[WebViewClient](https://developer.android.com/reference/android/webkit/WebViewClient.html) getWebViewClient( )

获取WebViewClient实例。

返回：

[WebViewClient](https://developer.android.com/reference/android/webkit/WebViewClient.html) WebViewClient或默认客户端(如果尚未设置)

**相关函数：**

* [setWebViewClient(WebViewClient)](https://developer.android.com/reference/android/webkit/WebView.html#setWebViewClient(android.webkit.WebViewClient))
* goBack(level 1)

void goBack( )

在这个WebView的历史中返回。

* goBackOrForward(level 1)

void goBackOrForward(int steps)

离开当前项目的步数。

参数：

steps int：在后退前进列表中取回或前进的步数

* goForward(level 1)

void goForward( )

在这个WebView的历史中前进。

* invokeZoomPicker(level 1)

void invokeZoomPicker( )

为此WebView调用图形缩放选择器小部件。这将导致缩放小部件出现在屏幕上以控制此WebView的缩放级别。

* isPrivateBrowsingEnabled(level 11)

boolean isPrivateBrowsingEnabled( )

获取是否在此WebView中启用隐私浏览。

返回：

boolean

* loadData(level 1)

void loadData([String](https://developer.android.com/reference/java/lang/String.html) data，[String](https://developer.android.com/reference/java/lang/String.html) mimeType， [String](https://developer.android.com/reference/java/lang/String.html) encoding)

使用“数据”方案URL将给定数据加载到此WebView中。

请注意，JavaScript的相同源策略意味着使用此方法加载的页面中运行的脚本将无法访问使用'data'之外的任何方案加载的内容，包括'http(s)'。为避免此限制，请使用[loadDataWithBaseURL()](https://developer.android.com/reference/android/webkit/WebView.html#loadDataWithBaseURL(java.lang.String, java.lang.String, java.lang.String, java.lang.String, java.lang.String))适当的基本URL。

该encoding参数指定数据是base64还是URL编码。如果数据是base64编码的，则编码参数的值必须是'base64'。HTML可以[encodeToString(byte[], int)](https://developer.android.com/reference/android/util/Base64.html#encodeToString(byte[], int))像这样编码：

字符串unencodedHtml =

“<html> <body>'％28'是'('</ body> </ html>”;

String encodedHtml = Base64.encodeToString(unencodedHtml.getBytes( )，Base64.NO\_PADDING);

webView.loadData(encodedHtml，“text / html”，“base64”);

对于encoding(包括null)的所有其他值，假定数据对安全URL字符范围内的八位字节使用ASCII编码，并对该范围外的八位字节使用标准的％xx十六进制URL编码。有关更多信息，请参阅[RFC 3986](https://tools.ietf.org/html/rfc3986#section-2.2)。

该mimeType参数指定数据的格式。如果WebView无法处理指定的MIME类型，它将下载数据。如果null，默认为'text / html'。

由该方法形成的'data'scheme URL使用默认的US-ASCII字符集。如果您需要设置不同的字符集，则应该形成一个“数据”方案URL，它在URL的媒体类型部分中明确指定charset参数，然后调用[loadUrl(String)](https://developer.android.com/reference/android/webkit/WebView.html#loadUrl(java.lang.String))。请注意，从数据URL的媒体类型部分获取的字符集总是会重写HTML或XML文档本身中指定的字符集。

参数：

data String：给定编码中的一串数据

mimeType String：数据的MIME类型，例如'text / html'。这个值可能是null。

encoding String：数据的编码。这个值可能是null。

* loadDataWithBaseURL(level 1)

void loadDataWithBaseURL([String](https://developer.android.com/reference/java/lang/String.html) baseUrl，[String](https://developer.android.com/reference/java/lang/String.html) data，[String](https://developer.android.com/reference/java/lang/String.html) mimeType，

[String](https://developer.android.com/reference/java/lang/String.html) encoding，[String](https://developer.android.com/reference/java/lang/String.html) historyUrl)

将给定的数据加载到此WebView中，使用baseUrl作为内容的基本URL。基本URL既用于解析相对URL，也用于应用JavaScript的相同源策略。historyUrl用于历史记录。

该mimeType参数指定数据的格式。如果WebView无法处理指定的MIME类型，它将下载数据。如果null，默认为'text / html'。

请注意，只有在baseUrl指定了'http'，'https'，'ftp'，'ftps'，'about'或'javascript'以外的方案时，以此方式指定的内容才可以访问本地设备文件(通过“文件”方案URL) ”。

如果基本URL使用数据方案，则此方法与调用相同，[loadData()](https://developer.android.com/reference/android/webkit/WebView.html#loadData(java.lang.String, java.lang.String, java.lang.String))并且historyUrl将被忽略，并且数据将被视为data：URL的一部分。如果基本URL使用任何其他方案，那么数据将作为普通字符串(即不是数据URL的一部分)加载到WebView中，并且字符串中的任何URL编码实体都不会被解码。

请注意，在请求使用此方法加载的页面的子资源(图像等)时，baseUrl在“Referer”HTTP标头中发送。

参数：

baseUrl String：用作网页基本网址的网址。如果null默认为'about：blank'。

data String：给定编码中的一串数据

mimeType String：数据的MIME类型，例如'text / html'。这个值可能是null。

Encoding String：数据的编码，这个值可能是null。

historyUrl String：用作历史记录条目的URL。如果null默认为'about：blank'。如果非null，则这必须

* loadURL(leve 1)

void loadUrl([String](https://developer.android.com/reference/java/lang/String.html) url)

加载给定的URL。

另请参阅兼容性说明。[evaluateJavascript(String, ValueCallback)](https://developer.android.com/reference/android/webkit/WebView.html#evaluateJavascript(java.lang.String, android.webkit.ValueCallback<java.lang.String>))

参数：

url String：要加载的资源的URL

* loadURL(leve 8)

void loadUrl([String](https://developer.android.com/reference/java/lang/String.html) url，[Map](https://developer.android.com/reference/java/util/Map.html) < [String](https://developer.android.com/reference/java/lang/String.html)，  [String](https://developer.android.com/reference/java/lang/String.html) > additionalHttpHeaders)

使用指定的额外HTTP标头加载给定的URL。

另请参阅兼容性说明。[evaluateJavascript(String, ValueCallback)](https://developer.android.com/reference/android/webkit/WebView.html#evaluateJavascript(java.lang.String, android.webkit.ValueCallback<java.lang.String>))

参数：

url String：要加载的资源的URL

additionalHttpHeaders Map：在此URL的HTTP请求中使用的附加标头，指定为从名称到值的映射。请注意，如果此映射包含此WebView默认设置的任何标头，例如控制缓存，接受类型或User-Agent的标头，则其值可能会被此WebView的默认值覆盖。

* onChildViewAdded(leve 1)

void onChildViewAdded([View](https://developer.android.com/reference/android/view/View.html) parent， [View](https://developer.android.com/reference/android/view/View.html) child)

**此方法在API级别8中已弃用**  
.WebView不再需要实现ViewGroup.OnHierarchyChangeListener。这种方法现在什么都不做。

当新的子视图被添加到父视图时调用。

参数：

parent View：添加子视图的视图

child View：在层次结构中添加新的子视图

* onChildViewRemoved(leve 1)

void onChildViewRemoved([View](https://developer.android.com/reference/android/view/View.html) p， [View](https://developer.android.com/reference/android/view/View.html) child)

**此方法在API级别8中已弃用**  
.WebView不再需要实现ViewGroup.OnHierarchyChangeListener。这种方法现在什么都不做。

当子视图从父视图中移除时调用。

参数：

p View：子视图被删除的视图

child View：从视图中移除子视图

* onCreateInputConnection(leve 1)

[InputConnection](https://developer.android.com/reference/android/view/inputmethod/InputConnection.html) onCreateInputConnection([EditorInfo](https://developer.android.com/reference/android/view/inputmethod/EditorInfo.html) outAttrs)

为InputMethod创建一个新的InputConnection用于与WebView进行交互。这与[onCreateInputConnection(EditorInfo)](https://developer.android.com/reference/android/view/View.html#onCreateInputConnection(android.view.inputmethod.EditorInfo))WebView 类似，但请注意WebView在UI线程以外的线程上调用InputConnection方法。如果这些方法被重写，那么在调用View方法或访问数据时，重写方法应该遵守线程限制。

参数：

outAttrs EditorInfo：填写关于连接的属性信息。

返回：

[InputConnection](https://developer.android.com/reference/android/view/inputmethod/InputConnection.html)

* onDragEvent(leve 11)

boolean onDragEvent([DragEvent](https://developer.android.com/reference/android/view/DragEvent.html) event)

处理由系统发送的拖动事件： [startDragAndDrop()](https://developer.android.com/reference/android/view/View.html#startDragAndDrop(android.content.ClipData, android.view.View.DragShadowBuilder, java.lang.Object, int))。

当系统调用这个方法时，它传递一个 [DragEvent](https://developer.android.com/reference/android/view/DragEvent.html)对象。调用来 [getAction()](https://developer.android.com/reference/android/view/DragEvent.html#getAction())返回DragEvent中定义的动作类型常量之一。该方法使用这些来确定拖放操作中发生了什么。

参数：

event DragEvent：[DragEvent](https://developer.android.com/reference/android/view/DragEvent.html)由系统发送。该[getAction()](https://developer.android.com/reference/android/view/DragEvent.html#getAction())方法返回在DragEvent中定义的操作类型常量，指示由此对象表示的拖动事件的类型。

返回：

boolean true如果方法成功，否则false。

该方法应该返回true以响应[ACTION\_DRAG\_STARTED](https://developer.android.com/reference/android/view/DragEvent.html#ACTION_DRAG_STARTED)接收当前操作的拖动事件的操作类型 。

* onFinishTemporaryDetach(leve 3)

void onFinishTemporaryDetach( )

[onStartTemporaryDetach()](https://developer.android.com/reference/android/view/View.html#onStartTemporaryDetach())在容器完成更改视图后调用。

* onGenericMotionEvent(leve 12)

boolean onGenericMotionEvent([MotionEvent](https://developer.android.com/reference/android/view/MotionEvent.html) event)

实现此方法来处理通用运动事件。

通用运动事件描述操纵杆运动，鼠标悬停，触控板触摸，滚轮运动和其他输入事件的 [source](https://developer.android.com/reference/android/view/MotionEvent.html#getSource())运动事件的指定类的输入被接收。此方法的实现必须在处理事件之前检查源中的字节。以下代码示例显示了这是如何完成的。

具有源类的通用运动事件[SOURCE\_CLASS\_POINTER](https://developer.android.com/reference/android/view/InputDevice.html#SOURCE_CLASS_POINTER) 被传递到指针下的视图。所有其他通用运动事件都会传送到焦点所在的视图。

public boolean onGenericMotionEvent(MotionEvent event){

if(event.isFromSource(InputDevice.SOURCE\_CLASS\_JOYSTICK)){

if(event.getAction( )== MotionEvent.ACTION\_MOVE){

//处理游戏杆运动...

返回true;

}

}

if(event.isFromSource(InputDevice.SOURCE\_CLASS\_POINTER)){

switch(event.getAction( )){

大小写MotionEvent.ACTION\_HOVER\_MOVE：

//处理鼠标悬停运动...

返回true;

大小写MotionEvent.ACTION\_SCROLL：

//处理滚轮运动...

返回true;

}

}

返回super.onGenericMotionEvent(event);

}

参数;

event MotionEvent：正在处理的通用运动事件。

返回：

boolean 如果处理该事件，则为true，否则为false。

* onGlobalFocusChanged(leve 1)

void onGlobalFocusChanged([View](https://developer.android.com/reference/android/view/View.html) oldFocus， [View](https://developer.android.com/reference/android/view/View.html) newFocus)

**此方法在API级别3中已弃用**  
.WebView不应该实现ViewTreeObserver.OnGlobalFocusChangeListener。这种方法现在什么都不做。

当焦点在视图树中更改时调用的回调方法。当视图树从触摸模式转换为非触摸模式时，oldFocus为空。当视图树从非触摸模式转换为触摸模式时，newFocus为null。当焦点在非触摸模式下改变(没有从或转换到触摸模式时)，oldFocus或newFocus可以为空。

参数：

oldFocus View：以前的重点观点，如果有的话。

newFocus View：新聚焦的视图，如果有的话。

* onHoverEvent(leve 14)

boolean onHoverEvent([MotionEvent](https://developer.android.com/reference/android/view/MotionEvent.html) event)

实现此方法来处理悬停事件。

只要指针悬停在视图边界之上或之外，并且视图当前未被触摸，就会调用此方法。悬停事件被表示为具有动作指针事件[ACTION\_HOVER\_ENTER](https://developer.android.com/reference/android/view/MotionEvent.html#ACTION_HOVER_ENTER)，[ACTION\_HOVER\_MOVE](https://developer.android.com/reference/android/view/MotionEvent.html#ACTION_HOVER_MOVE)或[ACTION\_HOVER\_EXIT](https://developer.android.com/reference/android/view/MotionEvent.html#ACTION_HOVER_EXIT)。

1.[ACTION\_HOVER\_ENTER](https://developer.android.com/reference/android/view/MotionEvent.html#ACTION_HOVER_ENTER) 当指针进入视图边界时，视图会收到带有动作的悬停事件。

2.[ACTION\_HOVER\_MOVE](https://developer.android.com/reference/android/view/MotionEvent.html#ACTION_HOVER_MOVE) 当指针已经进入视图边界并且已经移动时，视图接收到带有动作的悬停事件。

3.[ACTION\_HOVER\_EXIT](https://developer.android.com/reference/android/view/MotionEvent.html#ACTION_HOVER_EXIT) 当指针退出视图边界时，或者由于按钮点击，点击或导致视图被触摸的类似用户操作而导致指针即将关闭时，视图会收到带有动作的悬停事件。

4.该视图应实现此方法返回true以指示它正在处理悬停事件，例如通过更改其可绘制状态。

[setHovered(boolean)](https://developer.android.com/reference/android/view/View.html#setHovered(boolean))如果视图已启用且可点击，则在收到悬停输入或悬停退出事件时 ，默认实现调用更新视图的悬停状态。默认实现也发送悬停可访问性事件。

参数：

event MotionEvent：描述悬停的运动事件。

返回:

boolean 如果视图处理悬停事件，则为真.

* onkeydown(leve 1)

boolean onKeyDown(int keyCode， [KeyEvent](https://developer.android.com/reference/android/view/KeyEvent.html) event)

默认实现[KeyEvent.Callback.onKeyDown()](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyDown(int, android.view.KeyEvent))：当 视图被启用并且可点击时，在视图[KEYCODE\_DPAD\_CENTER](https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_DPAD_CENTER)或视图[KEYCODE\_ENTER](https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_ENTER)被释放时执行按下事件。

软件键盘中的按键通常不会触发这个监听器，尽管在某些情况下有些人可能会选择这样做。不要依靠这个来捕捉软件按键。

参数：

keyCode int代表按下按钮的关键代码 [KeyEvent](https://developer.android.com/reference/android/view/KeyEvent.html)

event KeyEvent：定义按钮操作的KeyEvent对象

返回:

boolean 如果您处理了该事件，请返回true。如果你想让事件由下一个接收者处理，返回false。

* onKeyMultiple(leve 1)

boolean onKeyMultiple(int keyCode， int repeatCount， [KeyEvent](https://developer.android.com/reference/android/view/KeyEvent.html) event)

默认实现[KeyEvent.Callback.onKeyMultiple()](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyMultiple(int, int, android.view.KeyEvent))：始终返回false(不处理事件)。

软件键盘中的按键通常不会触发这个监听器，尽管在某些情况下有些人可能会选择这样做。不要依靠这个来捕捉软件按键。

参数：

keyCode int：代表按下按钮的关键代码 [KeyEvent](https://developer.android.com/reference/android/view/KeyEvent.html)。

repeatCount int：动作的次数。

event KeyEvent：定义按钮操作的KeyEvent对象。

返回：

boolean 如果您处理了该事件，请返回true。如果你想让事件由下一个接收者处理，返回false。

* onKeyUp(leve 1)

boolean onKeyUp(int keyCode， [KeyEvent](https://developer.android.com/reference/android/view/KeyEvent.html) event)

默认实现[KeyEvent.Callback.onKeyUp()](https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyUp(int, android.view.KeyEvent))：当执行视图的点击[KEYCODE\_DPAD\_CENTER](https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_DPAD_CENTER)，[KEYCODE\_ENTER](https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_ENTER) 或者[KEYCODE\_SPACE](https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_SPACE)被释放。

软件键盘中的按键通常不会触发这个监听器，尽管在某些情况下有些人可能会选择这样做。不要依靠这个来捕捉软件按键。

参数：

keyCode int：代表按下按钮的关键代码 [KeyEvent](https://developer.android.com/reference/android/view/KeyEvent.html)。

event KeyEvent：定义按钮操作的KeyEvent对象。

返回：

boolean 如果您处理了该事件，请返回true。如果你想让事件由下一个接收者处理，返回false。

* 在onPause(leve 11)

void onPause( )

可以安全暂停的任何处理，例如动画和地理位置。请注意，此调用不会暂停JavaScript。要全局暂停JavaScript，请使用 [pauseTimers()](https://developer.android.com/reference/android/webkit/WebView.html#pauseTimers())。要恢复WebView，请使用[onResume()](https://developer.android.com/reference/android/webkit/WebView.html#onResume())。

* onProvideVirtualStructure(leve 23)

void onProvideVirtualStructure([ViewStructure](https://developer.android.com/reference/android/view/ViewStructure.html) structure)

在从视图中检索辅助结构时调用，作为[Activity.onProvideAssistData](https://developer.android.com/reference/android/app/Activity.html#onProvideAssistData(android.os.Bundle))在此视图下生成其他虚拟结构的一部分 。defaullt实现[getAccessibilityNodeProvider()](https://developer.android.com/reference/android/view/View.html#getAccessibilityNodeProvider())用来尝试从视图的虚拟可访问性节点(如果有的话)生成这个。您可以重写这个方法，提供此数据的更优化实施。

参数：

structure ViewStructure

* onResume(leve 11)

void onResume( )

在先前调用[onPause()](https://developer.android.com/reference/android/webkit/WebView.html#onPause())之后恢复WebView 。

* onStartTemporaryDetach(leve 3)

void onStartTemporaryDetach( )

这是在一个容器临时将子视图分开时调用的 [ViewGroup.detachViewFromParent](https://developer.android.com/reference/android/view/ViewGroup.html#detachViewFromParent(android.view.View))。当容器完成时，必须实现[onFinishTemporaryDetach()](https://developer.android.com/reference/android/view/View.html#onFinishTemporaryDetach())或 [onDetachedFromWindow()](https://developer.android.com/reference/android/view/View.html#onDetachedFromWindow())。

* onTouchEvent(leve 1)

boolean onTouchEvent([MotionEvent](https://developer.android.com/reference/android/view/MotionEvent.html) event)

实现此方法来处理触摸屏幕动作事件。

如果此方法用于检测点击操作，建议通过实现 [performClick()](https://developer.android.com/reference/android/view/View.html#performClick())来执行操作。这将确保一致的系统行为，包括：

1.遵守点击偏好

2.分派OnClickListener调用

3.[ACTION\_CLICK](https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.html#ACTION_CLICK)在启用辅助功能时处理

参数：

event MotionEvent：动作事件。

返回：

boolean 如果处理该事件，则为true，否则为false。

* onTrackballEvent(leve 1)

boolean onTrackballEvent([MotionEvent](https://developer.android.com/reference/android/view/MotionEvent.html) event)

实现这个方法来处理轨迹球运动事件。可以通过[MotionEvent.getX()](https://developer.android.com/reference/android/view/MotionEvent.html#getX())和[MotionEvent.getY()](https://developer.android.com/reference/android/view/MotionEvent.html#getY())跟踪上次事件以来球的相对移动。这些都是标准化的，因此移动1对应于用户按下一个DPAD键(所以它们通常是小数值，表示轨迹球中可用的更细粒度的移动信息)。

参数：

event MotionEvent：动作事件。

返回：

boolean 如果处理该事件，则为true，否则为false。

* onWindowFocusChanged(level 1)

void onWindowFocusChanged(boolean hasWindowFocus)

当包含此视图的窗口获得或失去焦点时调用。请注意，这与视图焦点不同：要接收事件，视图和其窗口都必须具有焦点。如果一个窗口显示在需要输入焦点的窗口上，那么您自己的窗口将失去焦点，但视图焦点将保持不变。

参数：

hasWindowFocus boolean：如果包含此视图的窗口现在具有焦点，则为true，否则为false。

* overlayHorizo​​ntalScrollbar(level 1)

boolean overlayHorizo​​ntalScrollbar( )

**此方法在API级别23中已弃用。**  
此方法现在已过时。

获取水平滚动条是否具有叠加样式。

返回：

boolean true

* overlayVerticalScrollbar(level 1)

boolean overlayVerticalScrollbar( )

**此方法在API级别23中已弃用。**  
此方法现在已过时。

获取垂直滚动条是否具有叠加样式。

返回：

boolean false

* PageDown(level 1)

Boolean pageDown(boolean bottom)

将此WebView的内容向下滚动页面大小的一半。

参数：

boolean boolean：true跳转到页面底部

返回：

boolean 如果页面滚动，返回true

* PageUp(level 1)

boolean pageUp(boolean up)

将此WebView的内容向上滚动一半视图大小。

参数：

top boolean：true跳转到页面顶部

返回：

boolean  如果页面滚动，返回true

* pauseTimers(level 1)

void pauseTimers( )

暂停所有WebView的布局、解析和JavaScript计时器。这是一个全局请求，并不局限于这个WebView。如果应用程序已暂停，这可能很有用。

* performLongClick(level 1)

boolean performLongClick( )

调用此视图的OnLongClickListener(如果已定义)。如果OnLongClickListener没有使用该事件，则调用上下文菜单。

返回：

boolean 如果接收者之一消费了该事件，返回true。

* postUrl(level 5)

void postUrl([String](https://developer.android.com/reference/java/lang/String.html) url， byte [] postData)

使用“POST”方法将postData加载到此WebView中。如果url不是网络URL，它将用[loadUrl(String)](https://developer.android.com/reference/android/webkit/WebView.html#loadUrl(java.lang.String)) 加载，而忽略postData参数。

参数：

url String：要加载的资源的URL

postData byte：数据将被传递给“POST”请求，该请求必须是“application / x-www-form-urlencoded”编码。

* postVisualStateCallback(level 23)

void postVisualStateCallback(long requestId， [WebView.VisualStateCallback](https://developer.android.com/reference/android/webkit/WebView.VisualStateCallback.html) callback)

当WebView的当前状态准备好被绘制时调用。

因为对DOM的更新是异步处理的，所以对DOM的更新可能不会立即通过后续[onDraw(Canvas)](https://developer.android.com/reference/android/webkit/WebView.html#onDraw(android.graphics.Canvas))调用直观地反映出来。在[WebView.VisualStateCallback](https://developer.android.com/reference/android/webkit/WebView.VisualStateCallback.html)提供了一种机制来通知主叫方当前时间的DOM的内容是准备好下一次要绘制[WebView](https://developer.android.com/reference/android/webkit/WebView.html) 。回调完成后的下一个绘制保证反映所有对DOM的更新，直到[WebView.VisualStateCallback](https://developer.android.com/reference/android/webkit/WebView.VisualStateCallback.html)发布的点，但它也可能包含在回调发布后应用的更新。

此API涵盖的DOM的状态包括以下内容：

原始HTML元素(div，img，span等)。

图片

CSS动画

WebGL的

画布

它不包括以下状态：

视频标签

为了保证在[onComplete(long)](https://developer.android.com/reference/android/webkit/WebView.VisualStateCallback.html#onComplete(long))被调用后，[WebView](https://developer.android.com/reference/android/webkit/WebView.html)成功渲染第一帧，必须满足一系列条件：

如果[WebView](https://developer.android.com/reference/android/webkit/WebView.html)的可见性设置为[VISIBLE](https://developer.android.com/reference/android/view/View.html#VISIBLE)则[WebView](https://developer.android.com/reference/android/webkit/WebView.html)必须连接到视图层级结构。

如果[WebView](https://developer.android.com/reference/android/webkit/WebView.html)可见性设置为[INVISIBLE](https://developer.android.com/reference/android/view/View.html#INVISIBLE) 那么[WebView](https://developer.android.com/reference/android/webkit/WebView.html)必须将其附加到视图层次结构，并且必须 [VISIBLE](https://developer.android.com/reference/android/view/View.html#VISIBLE)由该[onComplete(long)](https://developer.android.com/reference/android/webkit/WebView.VisualStateCallback.html#onComplete(long))方法创建。

如果[WebView](https://developer.android.com/reference/android/webkit/WebView.html)可见性设置为[GONE](https://developer.android.com/reference/android/view/View.html#GONE)，那么 [WebView](https://developer.android.com/reference/android/webkit/WebView.html)必须将其附加到视图层次结构，并且其 [LayoutParams](https://developer.android.com/reference/android/widget/AbsoluteLayout.LayoutParams.html)宽度和高度需要设置为固定值，并且必须[VISIBLE](https://developer.android.com/reference/android/view/View.html#VISIBLE)由该 [onComplete(long)](https://developer.android.com/reference/android/webkit/WebView.VisualStateCallback.html#onComplete(long))方法创建。

在使用此API时，如果[WebView](https://developer.android.com/reference/android/webkit/WebView.html)屏幕处于关闭状态以避免闪烁，还建议启用预栅格化。请参阅[setOffscreenPreRaster(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setOffscreenPreRaster(boolean))更多细节，并考虑其注意事项。

参数：

requestId long：将在回调中返回的ID，以允许呼叫者使用回叫匹配请求。

callback WebView.VisualStateCallback：要调用的回调。

* postWebMessage(level 23)

void postWebMessage([WebMessage](https://developer.android.com/reference/android/webkit/WebMessage.html) message， [Uri](https://developer.android.com/reference/android/net/Uri.html) target Origin)

发布消息到主框架。应用程序可以将消息限制到某个目标来源。有关如何使用目标来源的信息，请参阅 [HTML5规范](https://html.spec.whatwg.org/multipage/comms.html#posting-messages)。

目标来源可以设置为通配符(“\*”)。不过不推荐这样做。这样会有有关安全问题，请参阅上面的页面。

参数：

message WebMessage：WebMessage

targetOrigin Uri：目标来源

* reLoad(level 1)

void reload( )

重新加载当前的URL。

* removeJavascriptInterface(level 11)

void removeJavascriptInterface([String](https://developer.android.com/reference/java/lang/String.html) name)

从此WebView中删除以前注入的Java对象。请注意，除非下一次(重新)加载页面，否则删除将不会反映在JavaScript中。看[addJavascriptInterface(Object, String)](https://developer.android.com/reference/android/webkit/WebView.html#addJavascriptInterface(java.lang.Object, java.lang.String))。

参数:

name String：用于在JavaScript中公开对象的名称. 这个值绝对不能null。

* requestChildRectangleOnScreen(level 1)

boolean requestChildRectangleOnScreen(View child， [Rect](https://developer.android.com/reference/android/graphics/Rect.html) rect，boolean immediate)

当该组的子视图想要将特定矩形定位到屏幕上时调用。

参数：

child View：提出请求的直接子视图。

rect Rect：子视图希望在屏幕上显示的子视图坐标中的矩形。

immediate boolean：正确禁止动画或延迟滚动，否则返回false

返回：

boolean 该组是否滚动来处理操作

* requestFocus(level 1)

boolean requestFocus(int direction，[Rect](https://developer.android.com/reference/android/graphics/Rect.html) previouslyFocusedRect)

调用此方法可将焦点放在特定视图或其后面的某个子视图上，并提供关于该焦点的特定矩形的提示。

参数：

direction int：FOCUS\_UP，FOCUS\_DOWN，FOCUS\_LEFT和FOCUS\_RIGHT之一

previouslyFocusedRect Rect：矩形(在这个视图的坐标系中)给出关于焦点来自哪里的更精细的提示。如果没有提示，可能为空。

返回：

boolean 这个视图或是它的子视图是否是获取了了焦点

* requestFocusNodeHref(level 1)

void requestFocusNodeHref([Message](https://developer.android.com/reference/android/os/Message.html) hrefMsg)

请求最后一个点击点处的定位点或图片元素网址。如果hrefMsg是null，则此方法立即返回并且不会将hrefMsg发送到其目标。如果点击到了图像，锚点或锚点中的图像，则该消息将其数据中指定键中的字符串关联起来。与键配对的值可能是空字符串。

参数：

hrefMsg Message：与请求结果一起发送的消息。消息数据包含三个密钥。“url”返回锚点的href属性。“标题”返回锚点的文本。“src”返回图像的src属性。这个值可能是null。

* requestImageRef(level 1)

void requestImageRef([Message](https://developer.android.com/reference/android/os/Message.html) msg)

请求用户最后触摸的图像的URL。msg将把表示URL的字符串作为对象发送到其目标。

参数：

msg Message：以“url”为关键字，将请求的结果作为数据成员发送的消息。结果可以null。

* restoreState(level 1)

[WebBackForwardList](https://developer.android.com/reference/android/webkit/WebBackForwardList.html) restoreState([Bundle](https://developer.android.com/reference/android/os/Bundle.html) inState)

从给定的Bundle中恢复此WebView的状态。此方法旨在用于[onRestoreInstanceState(Bundle)](https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)) ，还用来调用以恢复此WebView的状态。如果在此WebView有机会构建状态(加载页面，创建后退/前进列表等)之后调用它，则可能会产生不良副作用。请注意，此方法不再恢复此WebView的显示数据。

参数：

instate Bundle：传入状态的Bundle

返回:

[WebBackForwardList](https://developer.android.com/reference/android/webkit/WebBackForwardList.html) 恢复的后退/前进列表,当为null restoreState表示失败。

* resumeTimers(level 1)

void resumeTimers( )

恢复所有WebView的所有布局，解析和JavaScript计时器。这将恢复调度所有计时器。

* 保存密码(level 1)

void savePassword([String](https://developer.android.com/reference/java/lang/String.html) host， [String](https://developer.android.com/reference/java/lang/String.html) username， [String](https://developer.android.com/reference/java/lang/String.html) password)

**此方法在API级别18中已被弃用**  
。将来的版本不支持在WebView中保存密码。

为指定的主机设置用户名和密码对。WebView使用此数据自动填充Web表单中的用户名和密码字段。请注意，这与用于HTTP身份验证的凭证无关。

参数：

host String：需要凭证的主机

username String：给定主机的用户名

password String：给定主机的密码

**相关函数：**

* [clearUsernamePassword()](https://developer.android.com/reference/android/webkit/WebViewDatabase.html#clearUsernamePassword())
* [hasUsernamePassword()](https://developer.android.com/reference/android/webkit/WebViewDatabase.html#hasUsernamePassword())
* saveState(level 1)

[WebBackForwardList](https://developer.android.com/reference/android/webkit/WebBackForwardList.html) saveState([Bundle](https://developer.android.com/reference/android/os/Bundle.html) outState)

保存在[onSaveInstanceState(Bundle)](https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle))中使用的WebView状态。请注意，此方法不再存储此WebView的显示数据。如果[restoreState(Bundle)](https://developer.android.com/reference/android/webkit/WebView.html#restoreState(android.os.Bundle))从未被调用，以前的行为可能会泄漏文件。

参数：

outstate Bundle：Bundle来存储这个WebView的状态

返回：

[WebBackForwardList](https://developer.android.com/reference/android/webkit/WebBackForwardList.html) 用于保存状态的后退/前进列表的副本。如果方法失败返回null

* saveWebArchive(level 11)

void saveWebArchive([String](https://developer.android.com/reference/java/lang/String.html) filename)

将当前视图保存为Web档案。

参数：

filename String：应该放置档案的文件名

* saveWebArchive(level 11)

void saveWebArchive([String](https://developer.android.com/reference/java/lang/String.html) basename，

boolean autoname， [ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html) < [String](https://developer.android.com/reference/java/lang/String.html) > callback)

将当前视图保存为Web档案。

参数：

basename String：放置档案的文件名

autoname boolean：if false，以basename为文件。如果true， basename被假定为根据当前页面的URL选择文件。

callback ValueCallback：在网络存档被保存后调用。onReceiveValue的参数可以是保存文件的文件名，也可以是保存null的失败的文件名。

* setBackgroundColor(level 1)

void setBackgroundColor(int color)

设置此视图的背景颜色。

参数：

color int：背景的颜色

* setCertificate(level 1)

void setCertificate([SslCertificate](https://developer.android.com/reference/android/net/http/SslCertificate.html) certificate)

**此方法在API级别17中已被弃用。**  
调用此函数没有用处，并且在将来的发行版中将被忽略。

设置主要顶层页面的SSL证书。

参数;

certificate SslCertificate

* setDownloadListener(level 1)

void setDownloadListener([DownloadListener](https://developer.android.com/reference/android/webkit/DownloadListener.html) listener)

当渲染引擎无法处理内容时,注册的要使用的接口，并且应该下载。这将取代当前的处理程序。

参数：

Listener DownloadListener：DownloadListener的实现

* setFindListener(level 16)

void setFindListener([WebView.FindListener](https://developer.android.com/reference/android/webkit/WebView.FindListener.html) listener)

注册监听器以通知页面查找操作进度。这将取代当前的监听器。

参数：

listener WebView.FindListener：执行 [WebView.FindListener](https://developer.android.com/reference/android/webkit/WebView.FindListener.html)

* setHorizo​​ntalScrollbarOverlay(level 1)

void setHorizo​​ntalScrollbarOverlay(boolean overlay)

**此方法在API级别23中已弃用。**  
此方法不起作用。

指出水平滚动条是否具有叠加样式。

参数：

overlay boolean：true如果水平滚动条应该有叠加样式

* setHttpAuthUsernamePassword(level 1)

void setHttpAuthUsernamePassword([String](https://developer.android.com/reference/java/lang/String.html) host， [String](https://developer.android.com/reference/java/lang/String.html) realm，

[String](https://developer.android.com/reference/java/lang/String.html) usrname， [String](https://developer.android.com/reference/java/lang/String.html) password)

**这种方法在API级别被废弃26.**  
使用[**setHttpAuthUsernamePassword(String, String, String, String)**](https://developer.android.com/reference/android/webkit/WebViewDatabase.html#setHttpAuthUsernamePassword(java.lang.String, java.lang.String, java.lang.String, java.lang.String))代替

将给定主机和领域的HTTP身份验证凭证存储到[WebViewDatabase](https://developer.android.com/reference/android/webkit/WebViewDatabase.html) 实例。

参数：

host String：凭据适用的主机

realm String：证书适用的领域

username String：用户名

password String： 密码

* setInitialScale(level 1)

void setInitialScale(int scaleInPercent)

设置此WebView的初始缩放比例。0表示默认。默认缩放行为取决于状态 [getUseWideViewPort()](https://developer.android.com/reference/android/webkit/WebSettings.html#getUseWideViewPort())和[getLoadWithOverviewMode()](https://developer.android.com/reference/android/webkit/WebSettings.html#getLoadWithOverviewMode())。如果内容符合WebView控件的宽度，则缩放设置为100％。对于宽泛的内容，行为取决于状态[getLoadWithOverviewMode()](https://developer.android.com/reference/android/webkit/WebSettings.html#getLoadWithOverviewMode())。如果它的值是true，则内容将被缩小为适合宽度的WebView控件，否则不会。如果初始比例大于0，WebView将以此值作为初始比例。请注意，与视图标记中的比例属性不同，此方法不考虑屏幕密度。

参数：

scaleInPercent int：以百分比表示的初始比例

* setLayerType(level 11)

void setLayerType(int layerType， [Paint](https://developer.android.com/reference/android/graphics/Paint.html) paint)

指定支持此视图的图层的类型。该图层可以是 [LAYER\_TYPE\_NONE](https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_NONE)，[LAYER\_TYPE\_SOFTWARE](https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_SOFTWARE)或者[LAYER\_TYPE\_HARDWARE](https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_HARDWARE)。

一个图层与一个可选[Paint](https://developer.android.com/reference/android/graphics/Paint.html) 实例相关联，该实例控制图层在屏幕上的组成方式。组成图层时考虑以下涂料属性：

[Translucency (alpha)](https://developer.android.com/reference/android/graphics/Paint.html#getAlpha())

[Blending mode](https://developer.android.com/reference/android/graphics/Paint.html#getXfermode())

[Color filter](https://developer.android.com/reference/android/graphics/Paint.html#getColorFilter())

如果此视图通过调用[setAlpha(float)](https://developer.android.com/reference/android/view/View.html#setAlpha(float))将alpha值设置为<1.0 ，则此图层的绘画的Alpha值将被此视图的Alpha值取代。

请参阅的文件[LAYER\_TYPE\_NONE](https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_NONE)， [LAYER\_TYPE\_SOFTWARE](https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_SOFTWARE)以及[LAYER\_TYPE\_HARDWARE](https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_HARDWARE) 有关何时以及如何使用层的更多信息。

参数：

layerType int：类型层的与该视图使用，必须是中的一个 [LAYER\_TYPE\_NONE](https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_NONE)，[LAYER\_TYPE\_SOFTWARE](https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_SOFTWARE)或[LAYER\_TYPE\_HARDWARE](https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_HARDWARE)

paint Paint：用于组成图层的油漆。该参数是可选的，可以为null。当图层类型为[LAYER\_TYPE\_NONE](https://developer.android.com/reference/android/view/View.html#LAYER_TYPE_NONE)时，它将被忽略。

* setLayoutParams(level 1)

void setLayoutParams([ViewGroup.LayoutParams](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams.html) params)

设置与此视图关联的布局参数。这些为父视图的提供参数指定应如何安排布局。ViewGroup.LayoutParams有许多子类，它们对应ViewGroup的不同子类，它们负责安排子布局。

参数：

params ViewGroup.LayoutParams：此视图的布局参数不能为空

* setMapTrackballToArrowKeys(level 1)

void setMapTrackballToArrowKeys(boolean setMap)

**此方法在API级别17中已弃用**。未来版本中仅支持  
缺省情况**true**。

参数：

setMap boolean

* setNetworkAvailable(level 3)

void setNetworkAvailable(boolean networkUp)

给WebView通知网络状态。这用于设置JavaScript属性window.navigator.isOnline，并生成HTML5 , sec. 5.7.7中指定的联机/脱机事件。

参数：

networkUp boolean：表示网络是否可用，true表示可用

* setOverScrollMode(level 9)

void setOverScrollMode(int mode)

为此视图设置过卷模式。有效的过滚动模式是 [OVER\_SCROLL\_ALWAYS](https://developer.android.com/reference/android/view/View.html#OVER_SCROLL_ALWAYS)(默认)，[OVER\_SCROLL\_IF\_CONTENT\_SCROLLS](https://developer.android.com/reference/android/view/View.html#OVER_SCROLL_IF_CONTENT_SCROLLS)(只有在视图内容大于容器时才允许过滚动)，或[OVER\_SCROLL\_NEVER](https://developer.android.com/reference/android/view/View.html#OVER_SCROLL_NEVER)。仅当视图能够滚动时才能设置视图的滚动模式。

参数：

mode int：这个视图的新的滚动模式。

* setPictureListener(level 1)

void setPictureListener([WebView.PictureListener](https://developer.android.com/reference/android/webkit/WebView.PictureListener.html) listener)

**此方法在API级别12中已弃用。**  
此方法现在已过时。

设置图片监听器。这是一个用于接收新图片通知的界面。

参数：

listener WebView.PictureListener：WebView.PictureListener的实现

* setRendererPriorityPolicy(level 26)

void setRendererPriorityPolicy(int rendererRequestedPriority，

boolean waivedWhenNotVisible)

为此[WebView](https://developer.android.com/reference/android/webkit/WebView.html)设置渲染器优先级策略。优先级策略将用于确定是否应将进程外渲染器视为OOM查杀的目标。由于渲染器可以与多个WebView关联，因此计算的最终优先级为所有WebView的最高优先级。当一个WebView被销毁时，计算渲染器优先级时将不再考虑这个WebView。一旦没有WebView与渲染器保持关联，渲染器的优先级将降低到[RENDERER\_PRIORITY\_WAIVED](https://developer.android.com/reference/android/webkit/WebView.html#RENDERER_PRIORITY_WAIVED)。默认策略是将优先级设置为[RENDERER\_PRIORITY\_IMPORTANT](https://developer.android.com/reference/android/webkit/WebView.html#RENDERER_PRIORITY_IMPORTANT)，并且不管可见性如何，除非调用者还用 [onRenderProcessGone(WebView, RenderProcessGoneDetail)](https://developer.android.com/reference/android/webkit/WebViewClient.html#onRenderProcessGone(android.webkit.WebView, android.webkit.RenderProcessGoneDetail))处理渲染崩溃，否则不应更改此优先级。任何其他设置都会导致WebView渲染器比应用更容易的被系统杀死。

参数：

rendererRequestedPriority int：此WebView要求渲染进程绑定的最低优先级。

[RENDERER\_PRIORITY\_WAIVED](https://developer.android.com/reference/android/webkit/WebView.html#RENDERER_PRIORITY_WAIVED)，[RENDERER\_PRIORITY\_BOUND](https://developer.android.com/reference/android/webkit/WebView.html#RENDERER_PRIORITY_BOUND) [RENDERER\_PRIORITY\_IMPORTANT](https://developer.android.com/reference/android/webkit/WebView.html#RENDERER_PRIORITY_IMPORTANT)。

waivedWhenNotVisible boolean：如果为true，这个标志指定当这个WebView不可见时，它将被视为它已经请求了优先级[RENDERER\_PRIORITY\_WAIVED](https://developer.android.com/reference/android/webkit/WebView.html#RENDERER_PRIORITY_WAIVED)。

* setScrollBarStyle(level 1)

void setScrollBarStyle(int style)

指定滚动条的样式。滚动条可以重叠或嵌入。滚动条可以在填充区域或视图的边缘内绘制。例如，如果视图具有可绘制背景，并且您想要在drawable指定的填充内绘制滚动条，则可以使用SCROLLBARS\_INSIDE\_OVERLAY或SCROLLBARS\_INSIDE\_INSET。如果您希望它们出现在视图边缘，忽略填充，则可以使用SCROLLBARS\_OUTSIDE\_OVERLAY或SCROLLBARS\_OUTSIDE\_INSET。

参数：

stytle int：滚动条的样式。

* setTextClassifier(level 27)

void setTextClassifier([TextClassifier](https://developer.android.com/reference/android/view/textclassifier/TextClassifier.html) textClassifier)

为此WebView 设置[TextClassifier](https://developer.android.com/reference/android/view/textclassifier/TextClassifier.html)。

参数：

textClassifier TextClassifier ，这个值可能是null。

* setVerticalScrollbarOverlay(level 1)

void setVerticalScrollbarOverlay(boolean overlay)

**此方法在API级别23中已弃用。**  
此方法不起作用。

指定垂直滚动条是否具有叠加样式。

参数：

overlay boolean：如果true，垂直滚动条有覆盖样式。

* setWebChromeClient(level 1)

void setWebChromeClient([WebChromeClient](https://developer.android.com/reference/android/webkit/WebChromeClient.html)客户端)

设置chrome处理程序。这是WebChromeClient的一个实现，用于处理JavaScript对话框，网站图标，标题和进度。这将取代当前的处理程序。

参数：

client WebChromeClient：WebChromeClient的实现

相关函数：

* [getWebChromeClient()](https://developer.android.com/reference/android/webkit/WebView.html#getWebChromeClient())
* setWebContentsDebuggingEnabled(level 19)

void setWebContentsDebuggingEnabled(boolean enabled)

启用加载到此应用程序的任何WebView中的Web内容(HTML / CSS / JavaScript)调试。可以启用此标志以便于调试WebViews中运行的Web布局和JavaScript代码。有关调试指南，请参阅WebView文档。默认是false。

参数：

enabled boolean：是否启用网页内容调试

* setWebViewClient(level 1)

void setWebViewClient([WebViewClient](https://developer.android.com/reference/android/webkit/WebViewClient.html) client)

设置WebViewClient接收各种通知和请求。这将取代当前的处理程序。

参数：

client WebViewClient：WebViewClient的实现

**相关函数：**

* [getWebViewClient()](https://developer.android.com/reference/android/webkit/WebView.html#getWebViewClient())
* shouldDelayChildPressedState(level 14)

boolean shouldDelayChildPressedState( )

如果应该延迟此ViewGroup的子项或后代的按下状态，则返回true。这可以防止在用户实际尝试滚动内容时出现按下状态。出于兼容性原因，默认实现返回true。不滚动的子类通常应该重写此方法并返回false。

返回：

boolean

* showFindDialog(level 11)

boolean showFindDialog(String text，boolean showIme)

**此方法在API级别18中已弃用。**  
此方法在所有Android版本上无法可靠运行; 使用WebView.findAllAsync( )实现自定义查找对话框提供了更强大的解决方案。

启动一个ActionMode来查找此WebView中的文本。仅当此WebView被添加到视图系统时才有效。

参数：

text String：如果非null，要搜索的初始文本。否则，将使用此WebView中搜索的最后一个String

showIme boolean：如果true，显示IME，用户将开始输入。如果false并且text不为空，则执行全部查找。

返回：

boolean 加入查找的对话框显示了，返回true，否则，返回false。

* startSafeBrowsing(level 27)

void startSafeBrowsing([Context](https://developer.android.com/reference/android/content/Context.html) context, [ValueCallback](https://developer.android.com/reference/android/webkit/ValueCallback.html)<[Boolean](https://developer.android.com/reference/java/lang/Boolean.html)> callback)

开始安全浏览初始化。只有当callback返回true时，才能保证URL加载受到安全浏览的保护。有些不支持安全浏览功能。对于这些设备callback将收到false。

如果安全浏览已被禁用或[setSafeBrowsingEnabled(boolean)](https://developer.android.com/reference/android/webkit/WebSettings.html#setSafeBrowsingEnabled(boolean))，则不应调用此功能。要使用应用程序的context来调用(并且始终使用应用程序的context来执行其工作)。

参数：

context Context

callback ValueCallback：true如果初始化成功，将在UI线程上调用，false否则。

这个值可能是null。

* stopLoading(level 1)

void stopLoading( )

停止当前的加载。

* zoomBy(level 21)

void zoomBy(float zoomFactor)

在此WebView中执行缩放操作。

参数：

zoomFactor float：要应用的缩放因子。缩放系数将被限制到WebView的缩放限制。该值必须在0.01到100.0

* zoomIn(level 1)

布尔zoomIn( )

在此WebView中执行放大。

返回：

boolean 如果放大成功，返回true，如果没有缩放变化，返回false

* zoomOut(level 1)

boolean zoomOut( )

在此WebView中执行缩小。

返回：

boolean 如果缩小成功，返回true，如果没有缩放变化，返回false

**protected方法：**

* computeHorizo​​ntalScrollOffset(level 1)

int computeHorizo​​ntalScrollOffset( )

计算水平滚动条在水平范围内的水平偏移量。该值用于计算滚动条轨道内的滚动位置。默认偏移量是该视图的滚动偏移量。

返回：

int 滚动条的水平偏移量

* computeHorizo​​ntalScrollRange(level 1)

int computeHorizo​​ntalScrollRange( )

计算水平滚动条滚动的水平范围。默认范围是当前视图的宽度。

返回：

int 水平滚动条滚动的水平范围

* computeVerticalScrollExtent(level 1)

int computeVerticalScrollExtent( )

计算垂直滚动条拇指在垂直范围内的范围。该值用于计算滚动条轨道内的拇指长度。默认范围是此视图的高度。

返回：

int 滚动条大拇指的垂直范围

* computeVerticalScrollOffset(level 1)

int computeVerticalScrollOffset( )

计算垂直滚动条拇指在水平范围内的垂直偏移量。该值用于计算滚动条轨道内的拇指位置。范围内任意单位必须由[computeVerticalScrollRange()](https://developer.android.com/reference/android/view/View.html#computeVerticalScrollRange())和[computeVerticalScrollExtent()](https://developer.android.com/reference/android/view/View.html#computeVerticalScrollExtent())所使用的单位表示。默认偏移量是该视图的滚动偏移量。

返回：

int 滚动条大拇指的垂直偏移量

* computeVerticalScrollRange(level 1)

int computeVerticalScrollRange( )

计算垂直滚动条代表的垂直范围。

范围内任意单位必须由[computeVerticalScrollExtent()](https://developer.android.com/reference/android/view/View.html#computeVerticalScrollExtent())和[computeVerticalScrollOffset()](https://developer.android.com/reference/android/view/View.html#computeVerticalScrollOffset())所使用的单位表示。默认范围是此视图的高度

返回：

int 垂直滚动条的垂直范围

* dispatchDraw(level 1)

void dispatchDraw([Canvas](https://developer.android.com/reference/android/graphics/Canvas.html) canvas)

绘制子视图。这可能会被派生类重写，以便在子对象被绘制之前获得控制权(这些发生在绘制自己的视图之后)。

参数：

canvas Canvas：绘制视图的画布

* onAttachedToWindow(level 1)

void onAttachedToWindow( )

这个函数在视图被附加到窗口时被调用。此时它有一个Surface并将开始绘制。请注意，此函数一定要在[onDraw(android.graphics.Canvas)](https://developer.android.com/reference/android/view/View.html#onDraw(android.graphics.Canvas))之前被调用，但可以在第一次onDraw之前的任何时候调用，包括在[onMeasure(int, int)](https://developer.android.com/reference/android/view/View.html#onMeasure(int, int))之前或之后。

* onConfigurationChanged(level 8)

void onConfigurationChanged([Configuration](https://developer.android.com/reference/android/content/res/Configuration.html) newConfig)

当应用程序使用的资源的当前配置发生更改时调用。您可以使用它来决定何时重新加载可根据方向和其他配置特性更改的资源。如果您不依赖于在[Activity](https://developer.android.com/reference/android/app/Activity.html)配置更改时重新创建activity实例的正常机制，则只需要使用它。

参数：

newConfig Configuration：新的资源配置。

* onDraw(level 1)

void onDraw(Canvas canvas)

实现这个方法来绘制你需要绘制的内容。

参数：

canvas Canvas：将在其上绘制背景的画布

* onFocusChanged(level 1)

void onFocusChanged(boolean focused， int direction， [Rect](https://developer.android.com/reference/android/graphics/Rect.html) previouslyFocusedRect)

当视图的焦点状态改变时，由视图系统调用。当焦点更改事件是由方向导航导致的，direction和previouslyFocusedRect提供了焦点改变来自何处。重写时，一定要调用超类，以便进行标准的焦点处理。

参数：

focused boolean：如果视图具有焦点，则为true; 否则为false。

direction int：当调用requestFocus( )以使该视图焦点时，方向焦点已移动。值是 [FOCUS\_UP](https://developer.android.com/reference/android/view/View.html#FOCUS_UP)，[FOCUS\_DOWN](https://developer.android.com/reference/android/view/View.html#FOCUS_DOWN)，[FOCUS\_LEFT](https://developer.android.com/reference/android/view/View.html#FOCUS_LEFT)， [FOCUS\_RIGHT](https://developer.android.com/reference/android/view/View.html#FOCUS_RIGHT)，[FOCUS\_FORWARD](https://developer.android.com/reference/android/view/View.html#FOCUS_FORWARD)，或[FOCUS\_BACKWARD](https://developer.android.com/reference/android/view/View.html#FOCUS_BACKWARD)。它可能并不总是适用，在这种情况下使用默认值。

previouslyFocusedRect Rect：在此视图的坐标系中，以前焦点视图的矩形。如果适用，这将作为关于焦点来自何处的信息(除了方向)传递。否则为null。

* onMeasure(level 1)

void onMeasure(int widthMeasureSpec， int heightMeasureSpec)

测量视图及其内容以确定宽度和高度。此方法由[measure(int, int)](https://developer.android.com/reference/android/view/View.html#measure(int, int))调用并应由子类覆盖，以提供准确的内容和高效的度量。

如果此方法被覆盖，则子类的责任是确保测量的高度和宽度至少为视图的最小高度和宽度([getSuggestedMinimumHeight()](https://developer.android.com/reference/android/view/View.html#getSuggestedMinimumHeight())和 [getSuggestedMinimumWidth()](https://developer.android.com/reference/android/view/View.html#getSuggestedMinimumWidth()))。

参数：

widthMeasureSpec int：父类强加的水平空间要求。这些要求用[View.MeasureSpec](https://developer.android.com/reference/android/view/View.MeasureSpec.html)编码 。

heightMeasureSpec int：父类强加的垂直空间要求。这些要求用[View.MeasureSpec](https://developer.android.com/reference/android/view/View.MeasureSpec.html)编码 。

* onOverScrolled(level 9)

void onOverScrolled(int scrollX， int scrollY， boolean clampedX，boolean clampedY)

被[overScrollBy(int, int, int, int, int, int, int, int, boolean)](https://developer.android.com/reference/android/view/View.html#overScrollBy(int, int, int, int, int, int, int, int, boolean))调用以响应过度滚动操作。

参数：

scrollX int：以像素为单位的新的X滚动值

scroll int：以像素为单位的新的Y滚动值

clamped boolean：如果scrollX被限制到过度滚动边界，则为true

clamped boolean：如果scrollY被限制到过度滚动边界，则为true

* onScrollChanged(level 1)

void onScrollChanged(int l， int t， int oldl， int oldt)

这是为了响应此视图中的内部滚动而调用的(即视图滚动了其自己的内容)。这通常是由于 [scrollBy(int, int)](https://developer.android.com/reference/android/view/View.html#scrollBy(int, int))或[scrollTo(int, int)](https://developer.android.com/reference/android/view/View.html#scrollTo(int, int))曾经被调用。

参数：

l int：当前水平滚动来源。

t int：当前垂直滚动来源。

oldl int：上一个水平滚动来源。

oldt int：上一个垂直滚动来源。

* onSizeChanged(level 1)

void onSizeChanged(int w， int h，int ow， int oh)

当这个视图的大小发生变化时，这个函数会在布局期间被调用。如果您刚刚添加到视图层次结构中，则会使用旧值0。

参数：

w int：此视图的当前宽度。

h int：此视图的当前高度。

ow int：这个视图的旧宽度。

oh int：这个视图的旧高度。

* onVisibilityChanged(level 8)

void onVisibilityChanged([View](https://developer.android.com/reference/android/view/View.html) changedView，int visibility)

当视图的可见性或视图的父类的可见性已更改时调用。

参数：

changedView View：可见性改变的视图。可能是 this或父类的视图。

visibility int：新的可见性，一个[VISIBLE](https://developer.android.com/reference/android/view/View.html#VISIBLE)， [INVISIBLE](https://developer.android.com/reference/android/view/View.html#INVISIBLE)或[GONE](https://developer.android.com/reference/android/view/View.html#GONE)。

* onWindowVisibilityChanged(level 1)

void onWindowVisibilityChanged(int visibility)

当窗口可见性改变时调用([GONE](https://developer.android.com/reference/android/view/View.html#GONE)，[INVISIBLE](https://developer.android.com/reference/android/view/View.html#INVISIBLE)和[VISIBLE](https://developer.android.com/reference/android/view/View.html#VISIBLE))。请注意，这会告诉您窗口管理器是否使您的窗口可见；另外，即使您的窗口是可见的，这并不能告诉你：你的窗口是否被屏幕上的其他窗口遮挡。

参数：

visibility int：窗口的新可见性

1.24.1 WebView.FindListener

* public static interface WebView.FindListener

该接口用来监听查找结果。

**public**

* onFindResultReceived(level 16)

abstract void onFindResultReceived(int activeMatchOrdinal，

int numberOfMatches，boolean isDoneCounting)

把查找操作所取得的进展通知给监听者。

参数：

activeMatchOrdinal int：当前选择的是基于零的序数

numberOfMatches int：已找到多少匹配

isDoneCounting boolean：查找操作是否已完成。当操作正在进行时，可能会多次通知监听器，除非isDoneCounting为true，否则不应将numberOfMatches值视为最终值true。

1.24.2 WebView.PictureListener

* public static interface WebView.PictureListener

**此接口在API级别12中已弃用。**  
此接口现在已过时。

可随时更改新图片。

**public**

* onNewPicture(level)

abstract void onNewPicture([WebView](https://developer.android.com/reference/android/webkit/WebView.html) view， Picture picture)

**此方法在API级别12中**  
已弃用。由于内部更改已弃用。

用于提供WebView图片更改的通知。请参阅[capturePicture()](https://developer.android.com/reference/android/webkit/WebView.html#capturePicture())图片的详细信息。

参数：

view WebView：拥有图片的WebView

picture Picture：新的图片。 [JELLY\_BEAN\_MR2](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#JELLY_BEAN_MR2)或以上的应用程序将始终会收到null

## 1.25 WebViewClient

* public class WebViewClient

**常量**

* ERROR\_AUTHENTICATION(level 5)

int ERROR\_AUTHENTICATION

用户认证在服务器上失败

常量值：-4(0xfffffffc)

* ERROR\_BAD\_URL(level 5)

int ERROR\_BAD\_URL

格式错误的网址

常量值：-12(0xfffffff4)

* ERROR\_CONNECT(level 5)

int ERROR\_CONNECT

无法连接到服务器

常量值：-6(0xfffffffa)

* ERROR\_FAILED\_SSL\_HANDSHAKE(level 5)

int ERROR\_FAILED\_SSL\_HANDSHAKE

无法执行SSL握手

常量值：-11(0xfffffff5)

* ERROR\_FILE(level 5)

int ERROR\_FILE

通用文件错误

常量值：-13(0xfffffff3)

* ERROR\_FILE\_NOT\_FOUND(level 5)

int ERROR\_FILE\_NOT\_FOUND

文件未找到

常量值：-14(0xfffffff2)

* ERROR\_HOST\_LOOKUP(level 5)

int ERROR\_HOST\_LOOKUP

服务器或代理主机名查找失败

常量值：-2(0xfffffffe)

* ERROR\_IO(level 5)

int ERROR\_IO

无法读取或写入服务器

常量值：-7(0xfffffff9)

* ERROR\_PROXY\_AUTHENTICATION(level 5)

int ERROR\_PROXY\_AUTHENTICATION

代理上的用户身份验证失败

常量值：-5(0xfffffffb)

* ERROR\_REDIRECT\_LOOP(level 5)

int ERROR\_REDIRECT\_LOOP

太多的重定向

常量值：-9(0xfffffff7)

* ERROR\_TIMEOUT(level 5)

int ERROR\_TIMEOUT

连接超时

常量值：-8(0xfffffff8)

* ERROR\_TOO\_MANY\_REQUESTS(level 5)

int ERROR\_TOO\_MANY\_REQUESTS

在此负载期间请求过多

常量值：-15(0xfffffff1)

* ERROR\_UNKNOWN(level 5)

int ERROR\_UNKNOWN

一般错误

常量值：-1(0xffffffff)

* ERROR\_UNSAFE\_RESOURCE(level 26)

int ERROR\_UNSAFE\_RESOURCE

资源加载已被安全浏览取消

常数值：-16(0xfffffff0)

* ERROR\_UNSUPPORTED\_AUTH\_SCHEME(level 5)

int ERROR\_UNSUPPORTED\_AUTH\_SCHEME

不支持的认证方案(不是基本或摘要)

常量值：-3(0xfffffffd)

* ERROR\_UNSUPPORTED\_SCHEME(level 5)

int ERROR\_UNSUPPORTED\_SCHEME

不支持的URI方案

常量值：-10(0xfffffff6)

* SAFE\_BROWSING\_THREAT\_MALWARE(level 27)

int SAFE\_BROWSING\_THREAT\_MALWARE

资源被阻止，因为它包含恶意软件

常数值：1(0x00000001)

* SAFE\_BROWSING\_THREAT\_PHISHING(level 27)

int SAFE\_BROWSING\_THREAT\_PHISHING

资源被阻止，因为它包含欺骗性内容

常量值：2(0x00000002)

* SAFE\_BROWSING\_THREAT\_UNKNOWN(level 27)

int SAFE\_BROWSING\_THREAT\_UNKNOWN

该资源因未知原因被阻止

常量值：0(0x00000000)

* SAFE\_BROWSING\_THREAT\_UNWANTED\_SOFTWARE(level 27)

int SAFE\_BROWSING\_THREAT\_UNWANTED\_SOFTWARE

该资源被阻止，因为它包含不需要的软件

常量值：3(0x00000003)

**Public constructor**

* WebViewClient(level 1)

WebViewClient( )

**public**

* doUpdateVisitedHistory(level 1)

void doUpdateVisitedHistory([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [String](https://developer.android.google.cn/reference/java/lang/String.html) url，boolean isReload)

通知应用程序更新其访问的链接数据库。

参数：

view WebView：启动回调的WebView。

url String：正在访问的网址。

isReload boolean：true如果这个网址被重新加载。

* onFormResubmission(level 1)

void onFormResubmission([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view，[Message](https://developer.android.google.cn/reference/android/os/Message.html) dontResend， [Message](https://developer.android.google.cn/reference/android/os/Message.html) resend)

如果请求的页面是POST的结果，浏览器应该重新发送数据。默认是不重新发送数据。

参数：

view WebView：启动回调的WebView。

dontResend Message：如果浏览器不应该重新发送的消息

resend Message：浏览器应重新发送数据时发送的消息

* onLoadResource(level 1)

void onLoadResource([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [String](https://developer.android.google.cn/reference/java/lang/String.html) url)

通知应用程序，WebView将加载由给定url指定的资源。

参数：

view WebView：启动回调的WebView。

url String：WebView将加载的资源的URL。

* onPageCommitVisible(level 23)

void onPageCommitVisible([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view，[String](https://developer.android.google.cn/reference/java/lang/String.html) url)

通知应用程序，[WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html)从前一页导航中遗留的内容将不再绘制。

此回调可用于确定再循环[WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html)可见的安全点，以 确保没有显示陈旧的内容。它被称为在可以保证[onDraw(Canvas)](https://developer.android.google.cn/reference/android/webkit/WebView.html#onDraw(android.graphics.Canvas))不会再从以前的导航中提取任何内容的最早时刻。接下来的抽签将显示任一 [background color](https://developer.android.google.cn/reference/android/webkit/WebView.html#setBackgroundColor(int))的[WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html)，或者一些新加载页面的内容。

当HTTP响应的主体已经开始加载时，这个方法被调用，并且在DOM中被反映，并且在随后的绘制中可见。此回调在文档加载过程的早期发生，因此您应该预期链接的资源(例如CSS和图像)可能不可用。

有关视觉状态更新的更细粒度的通知，请参阅[postVisualStateCallback(long, WebView.VisualStateCallback)](https://developer.android.google.cn/reference/android/webkit/WebView.html#postVisualStateCallback(long,%20android.webkit.WebView.VisualStateCallback))。请注意，所有适用的条件和建议 [postVisualStateCallback(long, WebView.VisualStateCallback)](https://developer.android.google.cn/reference/android/webkit/WebView.html#postVisualStateCallback(long,%20android.webkit.WebView.VisualStateCallback))也适用于此API。

此回调仅用于主框架导航。

参数：

view WebView：[WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html)导航发生的地方。

url String：触发​​此回调的页面导航对应的URL。

* onPageFinished(level 1)

void onPageFinished([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view，[String](https://developer.android.google.cn/reference/java/lang/String.html) url)

通知应用程序页面已完成加载。这种方法仅适用于主框架。调用onPageFinished( )时，渲染图片可能尚未更新。要获取新图片的通知，请使用[onNewPicture(WebView, Picture)](https://developer.android.google.cn/reference/android/webkit/WebView.PictureListener.html#onNewPicture(android.webkit.WebView,%20android.graphics.Picture))。

参数：

view WebView：启动回调的WebView。

url String：网页的网址。

* onPageStarted(level 1)

void onPageStarted([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [String](https://developer.android.google.cn/reference/java/lang/String.html) url， Bitmap favicon)

通知应用程序一个页面已经开始加载。每个主框架加载时，这种方法都被调用一次，所以带有iframe或框架集的页面将为主框架调用onPageStarted一次。这也意味着，当嵌入帧的内容发生变化时，onPageStarted不会被调用，即点击一个目标是iframe的链接，它也不会被称为片段导航(导航到#fragment\_id)。

* onReceivedClientCertRequest(level 21)

void onReceivedClientCertRequest([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [ClientCertRequest](https://developer.android.google.cn/reference/android/webkit/ClientCertRequest.html) request)

**此方法在UI线程上调用**。通知应用程序处理SSL客户端证书请求。宿主应用程序负责根据需要显示UI并提供密钥。有三种方法应对：[proceed(PrivateKey, X509Certificate[])](https://developer.android.google.cn/reference/android/webkit/ClientCertRequest.html#proceed(java.security.PrivateKey,%20java.security.cert.X509Certificate[]))，[cancel()](https://developer.android.google.cn/reference/android/webkit/ClientCertRequest.html#cancel())或[ignore()](https://developer.android.google.cn/reference/android/webkit/ClientCertRequest.html#ignore())。如果[ignore()](https://developer.android.google.cn/reference/android/webkit/ClientCertRequest.html#ignore()) 被调用，Webview不存储响应。请注意，铬网络堆栈中的多个图层可能会缓存响应，因此忽略行为只是最好的情况。在回调期间，连接被暂停。对于大多数使用情况，应用程序应该实现 [KeyChainAliasCallback](https://developer.android.google.cn/reference/android/security/KeyChainAliasCallback.html)接口并将其传递给[choosePrivateKeyAlias(Activity, KeyChainAliasCallback, String[], Principal[], Uri, String)](https://developer.android.google.cn/reference/android/security/KeyChain.html#choosePrivateKeyAlias(android.app.Activity,%20android.security.KeyChainAliasCallback,%20java.lang.String[],%20java.security.Principal[],%20android.net.Uri,%20java.lang.String))，这样为用户启动一个活动来选择适当的别名。keychain将通过实现的接口中的回调方法提供别名。接下来，应用程序应创建一个异步任务来调用 [getPrivateKey(Context, String)](https://developer.android.google.cn/reference/android/security/KeyChain.html#getPrivateKey(android.content.Context,%20java.lang.String))以接收密钥。在[AOSP浏览器中](https://android.googlesource.com/platform/packages/apps/Browser/+/android-5.1.1_r1/src/com/android/browser/Tab.java)可以看到客户端证书的示例实现 默认是取消，不返回客户端证书。

参数：

view WebView：启动回调的WebView

request ClientCertRequest：一个 [ClientCertRequest](https://developer.android.google.cn/reference/android/webkit/ClientCertRequest.html)的实例

* onReceivedError(level 1)

void onReceivedError([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， int errorCode，

String description， [String](https://developer.android.google.cn/reference/java/lang/String.html) failingUrl)

**这种方法在API级别被废弃23**  
使用[**onReceivedError(WebView, WebResourceRequest, WebResourceError)**](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#onReceivedError(android.webkit.WebView,%20android.webkit.WebResourceRequest,%20android.webkit.WebResourceError))来代替。

向应用程序报告错误。这些错误是不可恢复的(即主资源不可用)。该errorCode参数对应于其中一个ERROR\_\*常量。

参数：

view WebView：启动回调的WebView。

errorCode int：对应于ERROR\_ \*值的错误代码。

description String：描述错误的字符串。

failingUrl String：无法加载的网址。

* onReceivedError(level 23)

void onReceivedError([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view，

[WebResourceRequest](https://developer.android.google.cn/reference/android/webkit/WebResourceRequest.html) request， [WebResourceError](https://developer.android.google.cn/reference/android/webkit/WebResourceError.html) error)

向应用程序报告Web资源加载错误。这些错误通常表明无法连接到服务器。请注意，与过时的回调版本不同，新版本将针对任何资源(iframe，图像等)而被调用，而不仅仅是主页面。因此，建议在此回调中执行最低限度的工作。

参数：

view WebView：启动回调的WebView。

request WebResourceRequest：发起请求。

error WebResourceError：有关发生错误的信息。

* onReceivedHttpAuthRequest(level 1)

void onReceivedHttpAuthRequest([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [HttpAuthHandler](https://developer.android.google.cn/reference/android/webkit/HttpAuthHandler.html) handler，

[String](https://developer.android.google.cn/reference/java/lang/String.html) host， [String](https://developer.android.google.cn/reference/java/lang/String.html) realm)

通知应用程序WebView收到HTTP认证请求。应用程序可以使用提供的 [HttpAuthHandler](https://developer.android.google.cn/reference/android/webkit/HttpAuthHandler.html)来设置WebView对请求的响应。默认行为是取消请求。

参数：

view WebView：启动回调的WebView

handler HttpAuthHandler：HttpAuthHandler用于设置WebView的响应

host String：需要认证的主机

realm String：需要认证的领域

**相关函数：**

* [getHttpAuthUsernamePassword(String, String)](https://developer.android.google.cn/reference/android/webkit/WebView.html#getHttpAuthUsernamePassword(java.lang.String,%20java.lang.String))
* onReceivedHttpError(level 23)

void onReceivedHttpError([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [WebResourceRequest](https://developer.android.google.cn/reference/android/webkit/WebResourceRequest.html) request，

[WebResourceResponse](https://developer.android.google.cn/reference/android/webkit/WebResourceResponse.html) errorResponse)

通知应用程序在加载资源时从服务器收到HTTP错误。HTTP错误的状态码大于等于400.这个回调将被调用任何资源(iframe，图像等)，而不仅仅是主页面。因此，建议在此回调中执行最低限度的工作。请注意，服务器响应的内容可能不会在errorResponse参数中提供。

参数：

view WebView：启动回调的WebView。

Request WebResourceRequest：发起请求。

errorResponse WebResourceResponse：有关发生错误的信息。

* onReceivedLoginRequest(level 12)

无效onReceivedLoginRequest([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view，String realm，

String account， String args)

通知应用程序已处理自动登录用户的请求。

参数：

view WebView：WebView请求登录。

realm String：用于查找帐户的帐户领域。

account String：可选帐户。如果不是null，则应根据设备上的帐户检查帐户。

* onReceivedSslError(level 8)

void onReceivedSslError([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [SslErrorHandler](https://developer.android.google.cn/reference/android/webkit/SslErrorHandler.html) handler，

[SslError](https://developer.android.google.cn/reference/android/net/http/SslError.html) error)

通知应用程序加载资源时发生SSL错误。应用程序必须调用handler.cancel( )或handler.proceed( )。请注意，该决定可能会保留以用于响应未来的SSL错误。默认行为是取消加载。

参数：

view WebView：启动回调的WebView。

handler SslErrorHandler：将处理用户响应的SslErrorHandler对象。

error SslError：SSL错误对象。

* onScaleChanged(level 1)

void onScaleChanged([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， float oldScale， float newScale)

通知应用程序WebView的比例已更改。

参数：

view WebView：启动回调的WebView。

oldScale float：旧的比例因子

newScale float：新的比例因子

* onTooManyRedirects(level 1)

void onTooManyRedirects([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [Message](https://developer.android.com/reference/android/os/Message.html) cancelMsg，

[Message](https://developer.android.com/reference/android/os/Message.html) continueMsg)

**此方法在API级别8中已弃用。**  
此方法不再被调用。当WebView遇到重定向循环时，它将取消加载。

通知应用程序有过多的HTTP重定向。默认行为是发送取消消息。

参数：

view WebView：启动回调的WebView。

cancelMsg Message：主机想要取消时发送的消息

continueMsg Message：如果主机想继续发送的消息

* onUnhandledKeyEvent(level 1)

void onUnhandledKeyEvent([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [KeyEvent](https://developer.android.google.cn/reference/android/view/KeyEvent.html) event)

通知应用程序按键未由WebView处理。除系统按键，WebView总是消耗正常流程中的按键或者[shouldOverrideKeyEvent(WebView, KeyEvent)](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#shouldOverrideKeyEvent(android.webkit.WebView,%20android.view.KeyEvent))返回true。这是从按键分发的地方异步调用的。它使应用程序有机会处理未处理的事件。

参数：

view WebView：启动回调的WebView。

event KeyEvent：事件。

* shouldInterceptRequest(level 21)

[WebResourceResponse](https://developer.android.google.cn/reference/android/webkit/WebResourceResponse.html) shouldInterceptRequest([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view，

[WebResourceRequest](https://developer.android.google.cn/reference/android/webkit/WebResourceRequest.html) request)

通知应用程序请求资源，并允许应用程序返回数据。如果返回值是null，WebView将继续像平常一样继续加载资源。否则，将使用返回数据响应。

**注意：**此方法在UI线程以外的线程上调用，因此客户端在访问私人数据或视图系统时应谨慎行事。

**注意：**启用安全浏览功能后，这些网址仍会进行安全浏览检查。如果这是不希望的，则将URL列入白名单或使用忽略该警告。[**setSafeBrowsingWhitelist(List, ValueCallback)**](https://developer.android.google.cn/reference/android/webkit/WebView.html#setSafeBrowsingWhitelist(java.util.List%3Cjava.lang.String%3E,%20android.webkit.ValueCallback%3Cjava.lang.Boolean%3E))[**onSafeBrowsingHit(WebView, WebResourceRequest, int, SafeBrowsingResponse)**](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#onSafeBrowsingHit(android.webkit.WebView,%20android.webkit.WebResourceRequest,%20int,%20android.webkit.SafeBrowsingResponse))

参数：

view WebView：[WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html)正在请求资源。

request WebResourceRequest：包含请求细节的对象。

返回：

[WebResourceResponse](https://developer.android.google.cn/reference/android/webkit/WebResourceResponse.html) 包含响应信息，如果web视图应该加载本身资源，返回null。

* shouldInterceptRequest(level 1)

[WebResourceResponse](https://developer.android.google.cn/reference/android/webkit/WebResourceResponse.html) shouldInterceptRequest([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html)视图，

[String](https://developer.android.google.cn/reference/java/lang/String.html) url)

**这种方法在API级别被废弃21**  
使用[**shouldInterceptRequest(WebView, WebResourceRequest)**](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#shouldInterceptRequest(android.webkit.WebView,%20android.webkit.WebResourceRequest))来代替。

通知应用程序请求资源，并允许应用程序返回数据。如果返回值是null，WebView将继续像平常一样继续加载资源。否则，将使用返回响应和数据。

**注意：**此方法在UI线程以外的线程上调用，因此客户端在访问私人数据或视图系统时应谨慎行事。

**注意：**启用安全浏览功能后，这些网址仍会进行安全浏览检查。如果这是不希望的，则将URL列入白名单或使用忽略该警告。[**setSafeBrowsingWhitelist(List, ValueCallback)**](https://developer.android.google.cn/reference/android/webkit/WebView.html#setSafeBrowsingWhitelist(java.util.List%3Cjava.lang.String%3E,%20android.webkit.ValueCallback%3Cjava.lang.Boolean%3E))[**onSafeBrowsingHit(WebView, WebResourceRequest, int, SafeBrowsingResponse)**](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#onSafeBrowsingHit(android.webkit.WebView,%20android.webkit.WebResourceRequest,%20int,%20android.webkit.SafeBrowsingResponse))

参数：

view WebView：[WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html)正在请求资源。

url String：资源的原始网址。

返回：

[WebResourceResponse](https://developer.android.google.cn/reference/android/webkit/WebResourceResponse.html) 包含响应信息，如果web视图应该加载本身资源，返回null。

* shouldOverrideKeyEvent(level 1)

boolean shouldOverrideKeyEvent([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [KeyEvent](https://developer.android.google.cn/reference/android/view/KeyEvent.html) event)

让应用程序有机会同步处理事件。例如菜单快捷键事件需要以这种方式进行过滤。如果返回true，WebView将不会处理事件。如果返回false，WebView将始终处理事件，因此视图链中的超级用户都不会看到事件。默认行为返回false。

参数：

view WebView：启动回调的WebView。

event KeyEvent：事件。

返回：

boolean 如果应用程序想要自己处理事件，返回true ，则返回 false。

* shouldOverrideUrlLoading(level 24)

boolean shouldOverrideUrlLoading([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view，

[WebResourceRequest](https://developer.android.google.cn/reference/android/webkit/WebResourceRequest.html) request)

当一个新的URL即将被加载到当前的WebView中时，应用程序有机会接管控制权。如果未提供WebViewClient，则默认情况下，WebView将要求活动管理器为URL选择适当的处理程序。如果提供了WebViewClient，则返回true 意味着应用程序处理该url，而返回false意味着当前的WebView处理该url。

这种方法不会被使用POST“方法”的请求调用。

参数：

view WebView：启动回调的WebView。

request WebResourceRequest：包含请求细节的对象。

返回：

boolean 如果应用程序想要离开当前的WebView并自己处理url，返回true，否则返回false。

* shouldOverrideUrlLoading(level 1)

boolean shouldOverrideUrlLoading([WebView](https://developer.android.google.cn/reference/android/webkit/WebView.html) view， [String](https://developer.android.google.cn/reference/java/lang/String.html) url)

**这种方法在API级别被废弃24.**  
使用[**shouldOverrideUrlLoading(WebView, WebResourceRequest)**](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#shouldOverrideUrlLoading(android.webkit.WebView,%20android.webkit.WebResourceRequest))代替。

当一个新的URL即将被加载到当前的WebView中时，应用程序有机会接管控制权。如果未提供WebViewClient，则默认情况下，WebView将要求活动管理器为URL选择适当的处理程序。如果提供了WebViewClient，则返回true 意味着应用程序处理该url，而返回false意味着当前的WebView处理该url。这种方法不会被使用POST“方法”的请求调用。

参数:

view WebView：启动回调的WebView。

url String：要加载的网址。

返回：

boolean 如果应用程序想要离开当前的WebView并处理url本身，返回 true，否则返回false。

## 1.26 WebViewDatabase

* public abstract class WebViewDatabase

该类允许开发人员确定应用程序中使用的任何WebView是否存储了下面任何类型的浏览数据，并清除应用程序中所有WebView的此类存储数据。

* Web表单的用户名/密码对
* HTTP认证用户名/密码对
* 数据输入到文本字段(例如自动填充建议)

**Public constructor**

* WebViewDatabase(level 22)

WebViewDatabase( )

**public**

* clearFormData(level 1)

void clearFormData( )

清除所有保存的网页表单数据。

**相关函数：**

* [hasFormData()](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#hasFormData())
* clearHttpAuthUsernamePassword(level 1)

void clearHttpAuthUsernamePassword( )

清除所有保存的HTTP身份验证凭据。此方法仅清除存储在WebViewDatabase实例中的用户名和密码。用于http身份验证的用户名和密码可能会缓存在网络堆栈本身中，并且在调用此方法时不会被清除。客户端注销机制应由网站设计者(例如服务器发送用于使证书无效的HTTP 401)来实现。

**相关函数：**

* [getHttpAuthUsernamePassword(String, String)](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#getHttpAuthUsernamePassword(java.lang.String,%20java.lang.String))
* [setHttpAuthUsernamePassword(String, String, String, String)](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#setHttpAuthUsernamePassword(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))
* [hasHttpAuthUsernamePassword()](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#hasHttpAuthUsernamePassword())
* clearUsernamePassword(level 1)

void clearUsernamePassword( )

**此方法在API级别18中已弃用**  
。将来的版本不支持在WebView中保存密码。

清除所有保存的Web表单中的用户名/密码对。请注意，这些与HTTP身份验证凭证无关。

**相关函数：**

* [savePassword(String, String, String)](https://developer.android.google.cn/reference/android/webkit/WebView.html#savePassword(java.lang.String,%20java.lang.String,%20java.lang.String))
* [hasUsernamePassword()](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#hasUsernamePassword())
* getHttpAuthUsernamePassword(level 1)

[String []](https://developer.android.google.cn/reference/java/lang/String.html) getHttpAuthUsernamePassword([String](https://developer.android.google.cn/reference/java/lang/String.html) host，[String](https://developer.android.google.cn/reference/java/lang/String.html) realm)

从[WebViewDatabase](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html)实例中检索给定主机和领域的HTTP身份验证凭证。

参数：

host String：凭据适用的主机

realm String：证书适用的领域

返回：

[String[]](https://developer.android.google.cn/reference/java/lang/String.html) 作为字符串数组的凭据(如果找到)。第一个元素是用户名，第二个元素是密码。如果没有找到则为null。

**相关函数：**

* [setHttpAuthUsernamePassword(String, String, String, String)](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#setHttpAuthUsernamePassword(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))
* [hasHttpAuthUsernamePassword()](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#hasHttpAuthUsernamePassword())
* [clearHttpAuthUsernamePassword()](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#clearHttpAuthUsernamePassword())
* getInstance(level 1)

[WebViewDatabase](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html) getInstance(Context context)

参数：

context Context

返回：

[WebViewDatabase](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html) [WebViewDatabase](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html)的实例

* hasFormData(level 1)

boolean hasFormData( )

获取是否有任何保存的Web表单的数据。

返回：

boolean 是否有任何保存的网页表单数据

**相关函数：**

* [clearFormData()](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#clearFormData())
* hasHttpAuthUsernamePassword(level 1)

boolean hasHttpAuthUsernamePassword( )

获取是否有任何已保存的HTTP身份验证凭据。

返回：

boolean 是否有任何已保存的凭证

**相关函数：**

* [getHttpAuthUsernamePassword(String, String)](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#getHttpAuthUsernamePassword(java.lang.String,%20java.lang.String))
* [setHttpAuthUsernamePassword(String, String, String, String)](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#setHttpAuthUsernamePassword(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String))
* [clearHttpAuthUsernamePassword()](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#clearHttpAuthUsernamePassword())
* hasUsernamePassword(level 1)

boolean hasUsernamePassword( )

**此方法在API级别18中已弃用**  
。将来的版本不支持在WebView中保存密码。

获取是否有任何保存的Web表单的用户名/密码对。请注意，这些与HTTP身份验证凭证无关。

返回：

boolean 如果有任何保存的用户名/密码对，返回true 。

**相关函数：**

* [savePassword(String, String, String)](https://developer.android.google.cn/reference/android/webkit/WebView.html#savePassword(java.lang.String,%20java.lang.String,%20java.lang.String))
* [clearUsernamePassword()](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#clearUsernamePassword())
* setHttpAuthUsernamePassword(level 26)

void setHttpAuthUsernamePassword([String](https://developer.android.google.cn/reference/java/lang/String.html) host， [String](https://developer.android.google.cn/reference/java/lang/String.html) realm，

[String](https://developer.android.google.cn/reference/java/lang/String.html) username， [String](https://developer.android.google.cn/reference/java/lang/String.html) password)

将给定主机和领域的HTTP身份验证凭证存储到[WebViewDatabase](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html) 。

要使用HTTP身份验证，程序必须实现 [onReceivedHttpAuthRequest(WebView, HttpAuthHandler, String, String)](https://developer.android.google.cn/reference/android/webkit/WebViewClient.html#onReceivedHttpAuthRequest(android.webkit.WebView,%20android.webkit.HttpAuthHandler,%20java.lang.String,%20java.lang.String))并 使用正确的用户名和密码调用[proceed(String, String)](https://developer.android.google.cn/reference/android/webkit/HttpAuthHandler.html#proceed(java.lang.String,%20java.lang.String))。

应用程序可以选择任何方式获取用户名和密码，而不必使用[WebViewDatabase](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html)。

1.[WebViewDatabase](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html)仅作为便利来存储和检索http身份验证凭证。在HTTP认证期间，WebView不会从中读取它。

2.WebView不提供特殊的机制来清除实现客户端注销的HTTP身份验证凭证。客户端注销机制应由网站设计者(例如服务器发送用于使证书无效的HTTP 401)来实现。

参数：

host String：凭据适用的主机

realm String：证书适用的领域

username String：用户名

password String： 密码

**相关函数：**

* [getHttpAuthUsernamePassword(String, String)](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#getHttpAuthUsernamePassword(java.lang.String,%20java.lang.String))
* [hasHttpAuthUsernamePassword()](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#hasHttpAuthUsernamePassword())
* [clearHttpAuthUsernamePassword()](https://developer.android.google.cn/reference/android/webkit/WebViewDatabase.html#clearHttpAuthUsernamePassword())

# 2 扩展WebView方法

## 2.1 WebViewEx

* getProductVersion

String getProductVersion()

获取浏览器内核的版本信息

返回：

String 返回内核版本的编号信息

* loadUrl

void loadUrl(String url, Map<String, String> additionalHttpHeaders, long id,

boolean shouldReturnCacheDataDontLoad)

参数：

url String：加载的url

additionalHttpHeaders Map<String, String>:

id Long:id值

shouldReturnCacheDontLoad boolean：

* selectText

void selectText()

选择文本信息，在自由复制的时候用来选择文本信息。

* getContentBitmap

void getContentBitmap(float scale, Rect srcRect, boolean async,

final ValueCallback<Bitmap> callback)

内核截图功能实现。

参数：

scale float：缩放比例，默认是1

sreRect Rect：截图区域

async boolean：为true表示是异步的

callback ValueCallback<Bitmap> 通知上层的回调

* updateTopControls

void updateTopControls(boolean hide, boolean show, boolean animate)

更新顶部位置的状态。hide表示隐藏顶部位置，show表示控制显示顶部位置，animate表示显示隐藏的过程中做一个动画。注意：hide和show可以同时为true或是false，当同时为true时，表示标题栏可以随着上下推动而消失或是出现。

参数：

hide boolean：为true时，隐藏

show boolean:为true时，显示

animate boolean:为true是，做动画

* setTopControlsHeight

void setTopControlsHeight(float height)

从上层设置内核需要为标题栏留出的高度。这样可以方便的应对不同的浏览器对内核需要留出来不同的高度。

参数：

height float：为标题栏留出的大小，注意：该值为pix

* autofillText

void autofillText(String text, int type)

文本自动填充。

参数：

text String：文本内容

type String：文本样式

* dismissSelectToolbar

void dismissSelectToolbar()

消除自由复制的工具栏

* onPauseVideo

void onPauseVideo(int pauseReason)

视频暂停时调用。

参数：

pauseReason int：视频暂停的原因。

* resetDidFirstFrameOnResumeCounter

void resetDidFirstFrameOnResumeCounter()

左右滑屏时，回到当前界面时第一帧时，重设计数器。

* saveImage

void saveImage(String url, String referer)

保存图片时使用。

参数：

url String：图片网址

referrer String：引用页

* acquireDomInfo

void acquireDomInfo(final ValueCallback<String> callback)

获取DOM信息。

参数：

callback ValueCallback<String>：回调

* onSoftInputHeightChanged

void onSoftInputHeightChanged(int height)

在软键盘高度改变时调用

参数：

height int：软件键盘的高度位置

* onDnsPrefetch

void onDnsPrefetch(String[] hostList)

dns预解析

参数：

hostList String：域名列表

* detectFixedAdvertise

void detectFixedAdvertise(final ValueCallback<Boolean> callback)

检测是否有Fixed广告，并通过callback通知上层

参数

* displayImageForNoImageMode

void displayImageForNoImageMode(String url)

在无图模式下，显示单张图片。

返回：

url String：图片的链接

* hasTextSelected

boolean hasTextSelected()

是否已经选择了文本

返回：

boolean 若是已经选择了文本，返回true

* clearPinchZoomEnabledHostList

void clearPinchZoomEnabledHostList()

清楚允许手指缩放的域名列表

* detectDrawStatus

void detectDrawStatus()

检测是否是白屏。

* getVideoView

WebVideoView getVideoView()

获取视频WebVideoView的对象，用于处理一些特殊配置。

返回：

WebVideoView WebVideoView的对象

* getCommonExtension

CommonExtension getCommonExtension()

获取CommonExtension的对象，用于获取内核的一些信息。包括：appId，verNumber，verCode。

返回：

CommonExtension 返回CommonExtension的对象

* setInterceptJsUrl

void setInterceptJsUrl(String jsUrl)

设置js注入的url

参数：

JsUrl String：要注入的url

* blockAdvertiseByManual

void blockAdvertiseByManual()

手动屏蔽广告。

* appendReaderModeContent

void appendReaderModeContent(String content, String title, int id,

boolean hasNext, int error)

设置该webview为显示阅读模式的webview标志

参数：

content String：正文内容

title String：页面的标题

id int：页面的章数

hasNext boolean：是否有下一页

error int：加载失败类型

* setReaderModeNode

void setReaderModeNode(String nodeClass, String nodeId)

设置阅读模式的节点标识

参数：

nodeClass String：节点的class

nodeId String：节点的id

* getReaderModeInfo

void getReaderModeInfo()

获取阅读模式的信息。

* setReaderModeBackgroundColor

void setReaderModeBackgroundColor(int type)

设置阅读模式的背景颜色。

参数：

type int：颜色类型

* setReaderModeFontSize

void setReaderModeFontSize(int type)

设置阅读模式的默认字体大小，默认为normal

参数：

type int：默认字体大小，为normal

* registerServiceWorker

void registerServiceWorker(String url, String scriptUrl)

该接口可在内核启动后调用。调用后，内核会启动一个 ServiceWorker 实例，并注册该 scriptUrl。注册完成后会执行首次安装和更新，将 scriptUrl 脚本中所指定的资源提前离线到本地。待内核之后打开 url 网页时，会直接使用之前注册过的 ServiceWorker 实例，并使用离线数据展示网页。

参数：

url  String，待注册的网页完整 URL，scriptUrl 将作用于该网页，只支持 https；

scriptural String，待注册的 ServiceWorker 脚本，只支持 https。

* registerServiceWorker

void registerServiceWorker(String url, String scriptUrl,

final ValueCallback<Boolean> callback)

该接口可在内核启动后调用。调用后，内核会启动一个 ServiceWorker 实例，并注册该 scriptUrl。注册完成后会执行首次安装和更新，将 scriptUrl 脚本中所指定的资源提前离线到本地。待内核之后打开 url 网页时，会直接使用之前注册过的 ServiceWorker 实例，并使用离线数据展示网页。

参数：

url  String，待注册的网页完整 URL，scriptUrl 将作用于该网页，只支持 https；

scriptural String，待注册的 ServiceWorker 脚本，只支持 https。

Callback IRegisterServiceWorkerCallback（非必须），注册回调，由参数带回返回成功或失败。

* setEnabledShowWebviewInfo

void setEnabledShowWebviewInfo(boolean flag)

设置展示WebView的一些信息，包括：内存、帧率、gpu信息、gpu内存。注意：此函数只在新建的WebView里面生效。因此，若是需要在打开后再关闭显示，需要销毁WebView。

参数：

flag 当为true时，显示信息。

* clearPasswords

void clearPasswords()

清除浏览器中密码，这个方法是全局的。

## 2.2 WebSettingEx

Vivo扩展的设置功能。用来满足vivo的定制需求。注意：这些接口需要在ui线程上使用。

* setForceUserScalable

void setForceUserScalable(boolean forceEnableUserScalable)

设置是否能强制缩放

参数：

forceEnableUserScalable boolean：为true表示能强制缩放

* getForceUserScalable

boolean getForceUserScalable()

获取是否是能强制缩放。

返回：

boolean 为true，表示能强制缩放

* setOpenLinkInNewWebView

void setOpenLinkInNewWebView(boolean flag)

设置是否允许打开新的WebView。

参数：

flag boolean：为true，表示允许打开新的WebView

* getOpenLinkInNewWebView

boolean getOpenLinkInNewWebView()

获取是否允许打开新的新的WebView。

返回：

boolean 若是为true，则表示允许打开新的WebView

* setImageDownloadPath

void setImageDownloadPath(String path)

设置图片下载的保存路径。

参数：

path String：图片需要存储的路径

* getImageDownloadPath

String getImageDownloadPath()

获取图片下载的保存路径。

返回：

String 图片下载后的保存路径

* setAutofillTextType

void setAutofillTextType(int type)

设置自动填充文本的类型。

参数：

type int：自动填充文本的类型

* getAutofillTextType

int getAutofillTextType()

获取自动填充文本的类型

返回：

int 自动填充文本的类型

* isNightMode

boolean isNightMode()

获取当前是否是夜间模式。

返回：

boolean 若是夜间模式，返回true。

* setNightMode

void setNightMode(boolean flag)

设置夜间模式。

参数：

flag boolean：true，表示设置为夜间模式，false，设置为白天模式。

* setWifiRedirectUrl

void setWifiRedirectUrl(String url)

设置WIFI重定向的url。

参数：

url String：WIFI重定向的url

* getWifiRedirectUrl

String getWifiRedirectUrl()

获取WIFI重定向的url。

返回：

String WIFI重定向的url

* updateVideoSettings

void updateVideoSettings(String url)

更新与特定url相关的视频参数设置。与视频黑白名单相关。

参数：

url String：特定的网站

* setVivoProxyEnabled

void setVivoProxyEnabled(boolean enabled)

设置当前是否使用vivo代理。

参数：

enabled boolean：为true，表示允许使用vivo代理

* setReaderModeLoadNextPageFlag

void setReaderModeLoadNextPageFlag(boolean flag)

设置webview加载下一页的webview。

flag boolean：true时，表示允许

* getReaderModeLoadNextPageFlag

boolean getReaderModeLoadNextPageFlag()

获取是否加载下一页的webview

返回：

boolean true时，表示加载下一页的webview。

* setReaderModeShowPageFlag

void setReaderModeShowPageFlag(boolean flag)

设置该webview显示阅读模式的webview标志位。

参数：

flag boolean：true时，显示该标志位

* getReaderModeShowPageFlag

boolean getReaderModeShowPageFlag()

获取该webview是否显示阅读模式的标志位。

返回：

boolean 显示标志位时，为true。

* setBlockAdvertiseEnable

void setBlockAdvertiseEnable(boolean flag)

设置是否打开广告拦截。

参数：

flag boolean：为true时，表示打开广告拦截

* getBlockAdvertiseEnable

boolean getBlockAdvertiseEnable()

获取是否已经打开广告拦截

返回：

boolean 为true，表示已经打开了广告拦截

## 2.3 ExtensionClient

内核扩展的回调接口，用来处理一些vivo特殊的功能。注意，这些接口都需要在ui线程上使用。

* onSaveImageCompleted

void onSaveImageCompleted(String url, String mimeType, String filename,

String path, long contentLength)

保存图片完成时调用。

参数：

url String：网址

mimeType String：mime类型

filename String：文件名称

path String：文件路径

contentLength Long：目录长度

* onSaveImageFailed

void onSaveImageFailed(String url, String mimeType, String filename, String path)

保存图片失败是调用。

参数：

url String：

mimeType String：

filename String：

path String：

* openLinkInNewWebView

void openLinkInNewWebView(String url, String id, String downloadID)

在新的WebView上加载网页。

参数：

url String：要加载的网址

id

downloadID String：

* showFileChooser

void showFileChooser(ValueCallback<String[]> uploadFilePaths,

String acceptTypes, boolean capture)

上传文件时选择文件。

uploadFilePaths ValueCallback<String[]>：文件路径

acceptTypes String：接收类型

capture boolean：是否成功

* onTopControlsChanged

void onTopControlsChanged(float topControlsOffsetYPix, float contentOffsetYPix,

float overdrawBottomHeightPix)

顶部控制高度改变的时候调用。

参数：

topControlsOffsetYPix float:纵向纵向高度改变

contentOffsetYPix float:横向高度改变

overdrawBottomHeightPix 底部拉伸高度

* didFirstMessageForFrame

void didFirstMessageForFrame()

在页面渲染第一帧时调用。

* handleWebSearch

void handleWebSearch(String query)

自由复制时候，响应“搜索”按键。

参数：

query String：选择的字符创

* didFirstFrameOnResume

void didFirstFrameOnResume()

左右滑屏时，出现在界面上的第一帧时

* createInnerDialogBuilder

AlertDialog.Builder createInnerDialogBuilder(boolean[] canNotcheckedItems)

内核中创建对话框。

参数：

canNotcheckedItems boolean

* didLoadInSameDocument

void didLoadInSameDocument(String url, int type)

在当前url中加载，该加载不加载新的url。

* handleGotoUrl

void handleGotoUrl(String url)

自由复制的时候，若是链接，就跳转到相应的网址。

参数：

url String：选中的链接

* hasReaderMode

void hasReaderMode()

通知上层是否有阅读模式

* readerModeInfo

void readerModeInfo(String nextUrl, String title, String content, String nodeClass,

String nodeId, int error)

将阅读模式信息透传给外壳。

参数：

nextUrl String：下一个页面的url

title String：提取的标题

content String：提取的内容

nodeClass String：提取的正文节点

nodeId String：提取的正文节点id

error String：加载失败类型（0代表成功，1代表失败）

* displayReaderModeMenu

void displayReaderModeMenu(boolean flag)

通知上层显示阅读模式菜单

参数：

flag boolean：当为true时，显示。

* readerModeCurrentPage

void readerModeCurrentPage(int id)

通知上层当前用户阅读的章节。

参数：

id int：当前页面的id

* readerModeRetryLoad

void readerModeRetryLoad()

通知上层重新加载页面

* gotoReaderMode

void gotoReaderMode()

进入阅读模式

* callbackSetReaderModeBackgroundColor

void callbackSetReaderModeBackgroundColor(int type)

设置夜间模式背景色的回调

参数：

type int：颜色类型

* registerAutofillText

void registerAutofillText(int type)

注册自动填充文本类型。

参数：

type int：填充的文本类型

* documentAvailableInMainFrame

void documentAvailableInMainFrame()

文件在主frame中是否可用。

* onMainFrameHeadersReceived

void onMainFrameHeadersReceived(String url, boolean isMainFrame)

收到主frame表头时调用

url String：网址

isMainFrame String：为true时，表示是主frame

* onTouchEventAck

void onTouchEventAck(MotionEvent event, boolean consumed,

boolean pageOnLeftMost, boolean pageOnRightMost)

左右滑屏时调用。

参数：

event MotionEvent：触摸动作类型

consumed boolean：为true，表示事件在当前被消耗掉

pageOnLeftMost boolean:是否滑到最左端

pageOnRightMost boolean：是否滑到最右端

* onFillCodeSuccessed

void onFillCodeSuccessed(boolean isSuccess)

填充密码成功时调用

参数：

isSuccess boolean：为true，表示成功填充密码

* onReceivedResponseStatus

void onReceivedResponseStatus(int status\_code, int error\_code)

收到响应状态时调用。

参数：

status\_code int：状态码

error\_code int：错误码

## 2.4 WebVideoViewClient

视频相关的回调接口。

* shareVideoUrl

void shareVideoUrl(String url, String title)

点击视频分享按钮时，给应用的回调

参数：

url String：视频源的url(实际实现分享时，不会使用该值，应使用网页的url)

title String: 网页的标题

* isSupportDownload

**boolean isSupportDownload(boolean isFullscreen)**

全屏或者小屏是否支持视频下载(支持，则显示下载按钮)。默认不支持。

参数：

isFullscreen boolean：全屏/小屏

* downloadVideo

**void downloadVideo(String url, String title)**

点击视频下载按钮时的回调。

参数：

url String：视频源的url

title String：视频的标题

* getWebHost

**String getWebHost()**

获取网页host的方法，默认返回null。

* updateClarityData

**void updateClarityData()**

更新视频清晰度的回调（需要网站配合实现，浏览器目前只在爱奇艺站点使用）

* changeClarity

**void changeClarity(int clarity)**

改变视频清晰度的回调（需要网站配合实现，浏览器目前只在爱奇艺站点使用）

参数：

clarity int：要改变的清晰度的值

* playNextVideo

**void playNextVideo(int currentVideo)**

播放下一个视频的回调（需要网站配合实现，浏览器目前只在爱奇艺站点使用）

参数：

currentVideo int：当前播放的是第几集

* onHandleVCardEntry

void onHandleVCardEntry(boolean isFullscreen)

V粉卡（全屏或者小屏）点击“我要免流”之后跳转的回调

参数：

isFullscreen boolean：全屏/小屏

* onNotifyError

**void onNotifyError(int errorCode)**

视频播放出错时的回调

参数：

errorCode int：错误状态码

* onSetVideoUrl

**void onSetVideoUrl(String videoUrl, boolean isVideo)**

通知当前播放视频的url（外销浏览器埋点使用）

参数：

videoUrl String: 视频源的url

isVideo boolean：是否是视频