

Sommario ▼

 [Edit Source on GitHub \(https://github.com/apache/cordova-docs/tree/master/www/docs/it/3.1.0/cordova/inappbrowser/inappbrowser.md\)](https://github.com/apache/cordova-docs/tree/master/www/docs/it/3.1.0/cordova/inappbrowser/inappbrowser.md)

Italiano ▼ 3.1.0 ▼

Questa non è la versione più recente della documentazione! Click here for the latest released version. (/docs/it/latest/cordova/inappbrowser/inappbrowser.html) ×

InAppBrowser

Il **InAppBrowser** è una vista di browser web che visualizzi quando chiamata `[window.open](window.open.html)` `()` , o quando un link di apertura formata come `` .

```
var ref = window.open('http://apache.org', '_blank', 'location=yes');
```

Nota: La finestra di InAppBrowser si comporta come un browser web standard e non può accedere a Cordova APIs.

Descrizione

L'oggetto restituito da una chiamata a `[window.open](window.open.html)` .

Metodi

- `addEventListener`
- `removeEventListener`
- `close`
- `show`
- `executeScript`
- `insertCSS`

La funzionalità di accesso

A partire dalla versione 3.0, Cordova implementa le API a livello di dispositivo come *plugin*. Utilizzare la CLI **plugin** comando, descritto in Command-Line Interface, aggiungere o rimuovere questa funzionalità per un progetto:

```
$ cordova plugin add org.apache.cordova.inappbrowser
$ cordova plugin ls
[ 'org.apache.cordova.inappbrowser' ]
$ cordova plugin rm org.apache.cordova.inappbrowser
```

Questi comandi si applicano a tutte le piattaforme mirate, ma modificano le impostazioni di configurazione specifiche della piattaforma descritte di seguito:

- Android (in `app/res/xml/config.xml`)

```
<feature name="InAppBrowser">
  <param name="android-package" value="org.apache.cordova.InAppBrowser" />
</feature>
```

- iOS (in `config.xml`)

```
<feature name="InAppBrowser">
  <param name="ios-package" value="CDVInAppBrowser" />
</feature>
```

- Windows Phone 7 e 8 (in `config.xml`)

```
<feature name="InAppBrowser" />
```

Alcune piattaforme possono supportare questa funzionalità senza richiedere alcuna configurazione speciale. Vedere *Supporto piattaforma* nella sezione panoramica.

addEventListener

Aggiunge un listener per un evento dal **InAppBrowser** .

```
ref.addEventListener(eventname, callback);
```

- Rif: fare riferimento alla **InAppBrowser** finestra (*InAppBrowser*)
- EventName: l'evento per l'ascolto (*String*)
 - loadstart: evento viene generato quando il **InAppBrowser** comincia a caricare un URL.
 - loadstop: evento viene generato quando il **InAppBrowser** termina il caricamento di un URL.
 - LoadError: evento viene generato quando il **InAppBrowser** rileva un errore durante il caricamento di un URL.
 - uscita: evento viene generato quando il **InAppBrowser** finestra è chiusa.
- richiamata: la funzione che viene eseguito quando viene generato l'evento. La funzione viene passata un **InAppBrowserEvent** oggetto come parametro.

Piattaforme supportate

- Android
- BlackBerry
- iOS
- Windows Phone 7 e 8

Esempio rapido

```
var ref = window.open('http://apache.org', '_blank', 'location=yes');
ref.addEventListener('loadstart', function() { alert(event.url); });
```

Esempio completo

```
<!DOCTYPE html>
<html>
  <head>
    <title>InAppBrowser.addEventListener Example</title>

    <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
    <script type="text/javascript" charset="utf-8">

      // Wait for device API Libraries to load
      //
      document.addEventListener("deviceready", onDeviceReady, false);

      // device APIs are available
      //
      function onDeviceReady() {
        var ref = window.open('http://apache.org', '_blank', 'location=yes');
        ref.addEventListener('loadstart', function(event) { alert('start: ' + event.url); });
        ref.addEventListener('loadstop', function(event) { alert('stop: ' + event.url); });
        ref.addEventListener('loaderror', function(event) { alert('error: ' + event.message); });
        ref.addEventListener('exit', function(event) { alert(event.type); });
      }

    </script>
  </head>
  <body>
  </body>
</html>
```

removeEventListener

Rimuove un listener per un evento dal **InAppBrowser** .

```
ref.removeEventListener(eventname, callback);
```

- Rif: fare riferimento alla **InAppBrowser** finestra. (*InAppBrowser*)
- EventName: interrompere l'attesa per l'evento. (*String*)
 - loadstart: evento viene generato quando il **InAppBrowser** comincia a caricare un URL.
 - loadstop: evento viene generato quando il **InAppBrowser** termina il caricamento di un URL.
 - LoadError: evento viene generato quando il **InAppBrowser** rileva un errore di caricamento di un URL.
 - uscita: evento viene generato quando il **InAppBrowser** finestra è chiusa.
- richiamata: la funzione da eseguire quando viene generato l'evento. La funzione viene passata un **InAppBrowserEvent** oggetto.

Piattaforme supportate

- Android
- BlackBerry
- iOS
- Windows Phone 7 e 8

Esempio rapido

```
var ref = window.open('http://apache.org', '_blank', 'location=yes');
var myCallback = function() { alert(event.url); }
ref.addEventListener('loadstart', myCallback);
ref.removeEventListener('loadstart', myCallback);
```

Esempio completo

```
<!DOCTYPE html>
<html>
  <head>
    <title>InAppBrowser.removeEventListener Example</title>

    <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
    <script type="text/javascript" charset="utf-8">

      // Wait for device API libraries to load
      //
      document.addEventListener("deviceready", onDeviceReady, false);

      // Global InAppBrowser reference
      var iabRef = null;

      function iabLoadStart(event) {
        alert(event.type + ' - ' + event.url);
      }

      function iabLoadStop(event) {
        alert(event.type + ' - ' + event.url);
      }

      function iabLoadError(event) {
        alert(event.type + ' - ' + event.message);
      }

      function iabClose(event) {
        alert(event.type);
        iabRef.removeEventListener('loadstart', iabLoadStart);
        iabRef.removeEventListener('loadstop', iabLoadStop);
        iabRef.removeEventListener('loaderror', iabLoadError);
        iabRef.removeEventListener('exit', iabClose);
      }

      // device APIs are available
      //
      function onDeviceReady() {
        iabRef = window.open('http://apache.org', '_blank', 'location=yes');
        iabRef.addEventListener('loadstart', iabLoadStart);
        iabRef.addEventListener('loadstop', iabLoadStop);
        iabRef.removeEventListener('loaderror', iabLoadError);
        iabRef.addEventListener('exit', iabClose);
      }

    </script>
  </head>
  <body>
  </body>
</html>
```

chiudere

Chiude la **InAppBrowser** finestra.

```
ref.close();
```

- Rif: fare riferimento alla **InAppBrowser** finestra (*InAppBrowser*)

Piattaforme supportate

- Android
- BlackBerry
- iOS
- Windows Phone 7 e 8

Esempio rapido

```
var ref = window.open('http://apache.org', '_blank', 'location=yes');
ref.close();
```

Esempio completo

```
<!DOCTYPE html>
<html>
  <head>
    <title>InAppBrowser.close Example</title>

    <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
    <script type="text/javascript" charset="utf-8">

      // Wait for device API Libraries to Load
      //
      document.addEventListener("deviceready", onDeviceReady, false);

      // device APIs are available
      //
      function onDeviceReady() {
        var ref = window.open('http://apache.org', '_blank', 'location=yes');
        // close InAppBrowser after 5 seconds
        setTimeout(function() {
          ref.close();
        }, 5000);
      }

    </script>
  </head>
  <body>
  </body>
</html>
```

Visualizza

Visualizza una finestra di InAppBrowser che è stato aperto nascosta. Questa chiamata non ha effetto se la InAppBrowser era già visibile.

```
ref.show();
```

- ref: riferimento per il InAppBrowser finestra (**InAppBrowser**)

Piattaforme supportate

- Android
- BlackBerry
- iOS

Esempio rapido

```
var ref = window.open('http://apache.org', '_blank', 'hidden=yes');
ref.show();
```

Esempio completo

```
<!DOCTYPE html>
<html>
  <head>
    <title>InAppBrowser.show Example</title>

    <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
    <script type="text/javascript" charset="utf-8">

      // Wait for Cordova to Load
      //
      document.addEventListener("deviceready", onDeviceReady, false);

      // Cordova is ready
      //
      function onDeviceReady() {
        var ref = window.open('http://apache.org', '_blank', 'hidden=yes');
        ref.addEventListener('loadstop', function(event) {
          alert('background window loaded');
        });
        // close InAppBrowser after 5 seconds
        setTimeout(function() {
          ref.close();
        }, 5000);
      }

    </script>
  </head>
  <body>
  </body>
</html>
```

executeScript

Inserisce il codice JavaScript nella **InAppBrowser** finestra

```
ref.executeScript(details, callback);
```

- Rif: fare riferimento alla **InAppBrowser** finestra. (*InAppBrowser*)
- injectDetails: dettagli dello script da eseguire, specificando un **file** o **code** chiave. (*Oggetto*)
 - file: URL dello script da iniettare.
 - codice: testo dello script da iniettare.
- richiamata: la funzione che viene eseguito dopo che il codice JavaScript viene iniettato.
 - Se lo script iniettato è di tipo **code** , il callback viene eseguita con un singolo parametro, che è il valore restituito del copione, avvolto in un **Array** . Per gli script multi-linea, questo è il valore restituito dell'ultima istruzione, o l'ultima espressione valutata.

Piattaforme supportate

- Android
- BlackBerry
- iOS

Esempio rapido

```
var ref = window.open('http://apache.org', '_blank', 'location=yes');
ref.addEventListener('loadstop', function() {
  ref.executeSript({file: "myscript.js"});
});
```

Esempio completo

```
<!DOCTYPE html>
<html>
  <head>
    <title>InAppBrowser.executeScript Example</title>

    <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
    <script type="text/javascript" charset="utf-8">

      // Wait for device API Libraries to Load
      //
      document.addEventListener("deviceready", onDeviceReady, false);

      // Global InAppBrowser reference
      var iabRef = null;

      // Inject our custom JavaScript into the InAppBrowser window
      //
      function replaceHeaderImage() {
        iabRef.executeScript({
          code: "var img=document.querySelector('#header img'); img.src='http://cordova.apache.org/images/cordova_bot.png';"
        }, function() {
          alert("Image Element Successfully Hijacked");
        })
      }

      function iabClose(event) {
        iabRef.removeEventListener('loadstop', replaceHeaderImage);
        iabRef.removeEventListener('exit', iabClose);
      }

      // device APIs are available
      //
      function onDeviceReady() {
        iabRef = window.open('http://apache.org', '_blank', 'location=yes');
        iabRef.addEventListener('loadstop', replaceHeaderImage);
        iabRef.addEventListener('exit', iabClose);
      }

    </script>
  </head>
  <body>
  </body>
</html>
```

insertCSS

Inietta CSS nella **InAppBrowser** finestra.

```
ref.insertCSS(details, callback);
```

- Rif: fare riferimento alla **InAppBrowser** finestra (*InAppBrowser*)
- injectDetails: dettagli dello script da eseguire, specificando un **file** o **code** chiave. (*Oggetto*)
 - file: URL del foglio di stile per iniettare.
 - codice: testo del foglio di stile per iniettare.
- richiamata: la funzione che viene eseguito dopo che il CSS viene iniettato.

Piattaforme supportate

- Android
- BlackBerry
- iOS

Esempio rapido

```
var ref = window.open('http://apache.org', '_blank', 'location=yes');
ref.addEventListener('loadstop', function() {
  ref.insertCSS({file: "mystyles.css"});
});
```

Esempio completo

```
<!DOCTYPE html>
<html>
  <head>
    <title>InAppBrowser.insertCSS Example</title>

    <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
    <script type="text/javascript" charset="utf-8">

      // Wait for device API Libraries to Load
      //
      document.addEventListener("deviceready", onDeviceReady, false);

      // Global InAppBrowser reference
      var iabRef = null;

      // Inject our custom CSS into the InAppBrowser window
      //
      function changeBackgroundColor() {
        iabRef.insertCSS({
          code: "body { background: #ffff00"
        }, function() {
          alert("Styles Altered");
        })
      }

      function iabClose(event) {
        iabRef.removeEventListener('loadstop', changeBackgroundColor);
        iabRef.removeEventListener('exit', iabClose);
      }

      // device APIs are available
      //
      function onDeviceReady() {
        iabRef = window.open('http://apache.org', '_blank', 'location=yes');
        iabRef.addEventListener('loadstop', changeBackgroundColor);
        iabRef.addEventListener('exit', iabClose);
      }

    </script>
  </head>
  <body>
  </body>
</html>
```

InAppBrowserEvent

L'oggetto che viene passato alla funzione di richiamata da un **addEventListener** chiamare su un **InAppBrowser** oggetto.

Proprietà

- tipo: il eventname, o **loadstart** , **loadstop** , **loaderror** , o **exit** . *(String)*
- URL: l'URL che è stato caricato. *(String)*
- codice: il codice di errore, solo nel caso di **loaderror** . *(Numero)*
- messaggio: il messaggio di errore, solo nel caso di **loaderror** . *(String)*

More Resources

- GENERAL
 - Apache Project Page (<https://projects.apache.org/project.html?cordova>)
 - Source Distribution (<http://www.apache.org/dyn/closer.cgi/cordova>)
 - License (<http://www.apache.org/licenses/LICENSE-2.0>)
 - Artwork ([/artwork](#))
- DEVELOPMENT
 - Source Code (<https://github.com/apache?utf8=%E2%9C%93&q=cordova->)
 - Issue Tracker (<https://issues.apache.org/jira/browse/CB/>)

Stack Overflow (<http://stackoverflow.com/questions/tagged/cordova>)

Mailing List (</contact>)

Nightly builds (/contribute/nightly_builds.html)

APACHE SOFTWARE FOUNDATION

About ASF (<http://www.apache.org/>)

Become a Sponsor (<http://www.apache.org/foundation/sponsorship.html>)

Thanks (<http://www.apache.org/foundation/thanks.html>)

Security (<http://www.apache.org/security/>)

Contribute

Help Cordova move forward!

Report bugs, improve the docs, or contribute to the code.

[Learn More \(/contribute\)](/contribute)

Segui [@apachecordova](#)

Slack

4063

Copyright © 2012, 2013, 2015 The Apache Software Foundation, Licensed under the Apache License, Version 2.0 (<http://www.apache.org/licenses/LICENSE-2.0>).
Apache and the Apache feather logos are trademarks (<http://www.apache.org/foundation/marks/list/>) of The Apache Software Foundation.
'Raleway' font used under license. For details see the attributions page (</attributions/>).