

YouTube Embedded Players and Player Parameters

bookmark_border

-
-
-
-
-
-
-
-
-
-

Tổng quan

Tài liệu này giải thích cách nhúng trình phát YouTube vào ứng dụng của bạn, đồng thời xác định các tham số có trong trình phát YouTube được nhúng.

Bằng cách thêm các tham số vào URL IFrame, bạn có thể tùy chỉnh trải nghiệm phát trong ứng dụng của mình. Ví dụ: bạn có thể tự động phát video bằng cách sử dụng tham số [autoplay](#) hoặc khiến video phát lặp lại bằng cách sử dụng tham số [loop](#). Bạn cũng có thể dùng tham số [enablejsapi](#) để cho phép kiểm soát trình phát thông qua [API Trình phát IFrame](#).

Trang này hiện xác định tất cả các tham số được hỗ trợ trong bất kỳ trình phát được nhúng nào của YouTube. Mỗi định nghĩa tham số đều xác định những trình phát hỗ trợ tham số tương ứng.

Lưu ý: Trình phát nhúng phải có khung nhìn ít nhất là 200px x 200px. Nếu trình phát hiển thị các nút điều khiển, thì trình phát đó phải đủ lớn để hiển thị đầy đủ các nút điều khiển mà không làm giảm kích thước khung hiển thị xuống dưới kích thước tối thiểu. Bạn nên sử dụng trình phát có tỷ lệ khung hình 16:9 với chiều rộng tối thiểu là 480 pixel và chiều cao tối thiểu là 270 pixel.

Nhúng trình phát YouTube

Bạn có thể sử dụng bất kỳ phương thức nào sau đây để nhúng trình phát YouTube vào ứng dụng và chỉ định các thông số của trình phát. Xin lưu ý rằng hướng dẫn dưới đây

minh họa cách nhúng một trình phát tải một video. Phần sau đây giải thích cách định cấu hình trình phát để tải một danh sách phát hoặc video do người dùng tải lên.

Nhúng trình phát bằng thẻ `<iframe>`

Xác định thẻ `<iframe>` trong ứng dụng của bạn, trong đó URL `src` chỉ định nội dung mà trình phát sẽ tải cũng như mọi thông số trình phát khác mà bạn muốn đặt. Các tham số `height` và `width` của thẻ `<iframe>` chỉ định kích thước của trình phát.

Nếu tự tạo phần tử `<iframe>` (thay vì dùng IFrame Player API để tạo phần tử này), bạn có thể trực tiếp thêm các tham số trình phát vào cuối URL. URL có định dạng như sau:

```
https://www.youtube.com/embed/VIDEO_ID
```

Thẻ `<iframe>` bên dưới sẽ tải một trình phát có kích thước 640x360px để phát video M7lc1UVf-VE trên YouTube. Vì URL đặt tham số `autoplay` thành 1, nên video sẽ tự động phát sau khi trình phát tải xong.

```
<iframe id="ytplayer" type="text/html" width="640" height="360"
  src="https://www.youtube.com/embed/M7lc1UVf-
  VE?autoplay=1&origin=http://example.com"
  frameborder="0"></iframe>
```

Nhúng trình phát bằng IFrame Player API

Làm theo [hướng dẫn về IFrame Player API](#) để chèn một trình phát video vào trang web hoặc ứng dụng của bạn sau khi mã JavaScript của Player API đã tải. Tham số thứ hai trong hàm khởi tạo của trình phát video là một đối tượng chỉ định các lựa chọn của trình phát. Trong đối tượng đó, thuộc tính `playerVars` xác định các tham số của trình phát.

Đoạn mã HTML và JavaScript bên dưới minh họa một ví dụ đơn giản về cách chèn trình phát YouTube vào phần tử trang có giá trị `id` là `ytplayer`.

Hàm `onYouTubePlayerAPIReady()` được chỉ định ở đây sẽ tự động được gọi khi mã IFrame Player API đã tải. Mã này không xác định bất kỳ thông số nào của trình phát và cũng không xác định các trình xử lý sự kiện khác.

```
<div id="ytplayer"></div>
```

```
<script>
  // Load the IFrame Player API code asynchronously.
  var tag = document.createElement('script');
  tag.src = "https://www.youtube.com/player_api";
  var firstScriptTag = document.getElementsByTagName('script')[0];
  firstScriptTag.parentNode.insertBefore(tag, firstScriptTag);
```

```
// Replace the 'ytplayer' element with an <iframe> and
// YouTube player after the API code downloads.
var player;
function onYouTubePlayerAPIReady() {
  player = new YT.Player('ytplayer', {
    height: '360',
    width: '640',
    videoId: 'M7lc1UVf-VE'
  });
}
</script>
```

Chọn nội dung để phát

Bạn có thể định cấu hình trình phát được nhúng để tải một video, danh sách phát hoặc video do người dùng tải lên.

Danh sách sau đây giải thích các lựa chọn này:

- **Tải video lên**

Đối với một thành phần nhúng IFrame, mã video trên YouTube của video mà bạn muốn tải được chỉ định trong URL `src` của IFrame.

`https://www.youtube.com/embed/VIDEO_ID`

Nếu đang sử dụng YouTube Data API (phiên bản 3), bạn có thể tạo các URL này theo cách lập trình bằng cách truy xuất mã video từ [kết quả tìm kiếm](#), [tài nguyên mục trong danh sách phát](#), [tài nguyên video](#) hoặc các tài nguyên khác. Sau khi lấy được mã video, hãy thay thế văn bản `VIDEO_ID` trong các URL ở trên bằng giá trị đó để tạo URL của trình phát.

- **Tải danh sách phát**

Đặt tham số `listType` của trình phát thành `playlist`. Ngoài ra, hãy đặt tham số trình phát `list` thành mã nhận dạng danh sách phát trên YouTube mà bạn muốn tải.

`https://www.youtube.com/embed?listType=playlist&list=PLAYLIST_ID`

Xin lưu ý rằng bạn cần thêm các chữ cái `PL` vào trước mã nhận dạng danh sách phát như trong ví dụ sau:

`https://www.youtube.com/embed?listType=playlist&list=PLC77007E23FF423C6`

Nếu đang sử dụng YouTube Data API (phiên bản 3), bạn có thể lập trình để tạo các URL này bằng cách truy xuất mã danh sách phát từ [kết quả tìm kiếm](#), [tài nguyên kênh](#) hoặc [tài nguyên hoạt động](#). Sau khi lấy được mã nhận dạng danh sách phát, hãy thay thế văn bản `PLAYLIST_ID` trong URL ở trên bằng giá trị đó.

- **Tải video do người dùng tải lên**

Đặt tham số [listType](#) của trình phát thành `user_uploads`. Ngoài ra, hãy đặt tham số trình phát [list](#) thành tên người dùng YouTube có video tải lên mà bạn muốn tải.

`https://www.youtube.com/embed?listType=user_uploads&list=USERNAME`

Các tham số được hỗ trợ

Tất cả các thông số sau đây đều không bắt buộc.

Thông số

`autoplay`

Tham số này chỉ định xem video ban đầu có tự động bắt đầu phát khi trình phát tải hay không. Giá trị được hỗ trợ là `0` hoặc `1`. Giá trị mặc định là `0`.

Nếu bạn bật tính năng Tự động phát, thì quá trình phát sẽ diễn ra mà người dùng không cần tương tác với trình phát; do đó, quá trình thu thập và chia sẻ dữ liệu phát sẽ diễn ra khi trang tải.

`cc_lang_pref`

Tham số này chỉ định ngôn ngữ mặc định mà trình phát sẽ dùng để hiển thị phụ đề. Đặt giá trị tham số thành [mã ngôn ngữ gồm hai chữ cái theo tiêu chuẩn ISO 639-1](#).

Nếu bạn sử dụng tham số này và cũng đặt tham số [cc_load_policy](#) thành `1`, thì trình phát sẽ hiển thị phụ đề bằng ngôn ngữ đã chỉ định khi trình phát tải. Nếu bạn không đặt tham số [cc_load_policy](#), phụ đề không xuất hiện theo mặc định, nhưng sẽ xuất hiện bằng ngôn ngữ đã chỉ định nếu người dùng bật phụ đề.

`cc_load_policy`

Việc đặt giá trị của tham số thành `1` sẽ khiến phụ đề xuất hiện theo mặc định, ngay cả khi người dùng đã tắt phụ đề. Hành vi mặc định dựa trên lựa chọn ưu tiên của người dùng.

`color`

Tham số này chỉ định màu sẽ được dùng trong thanh tiến trình video của trình phát để làm nổi bật video mà người xem đã xem. Các giá trị tham số hợp lệ là `red` và `white`. Theo mặc định, trình phát sử dụng màu đỏ trong thanh tiến trình của video. Hãy xem [blog về API YouTube](#) để biết thêm thông tin về các lựa chọn màu sắc.

`controls`

Tham số này cho biết liệu các nút điều khiển trình phát video có hiển thị hay không:

`controls=0` – Các nút điều khiển trình phát không xuất hiện trong trình phát.

`controls=1` (mặc định) – Các nút điều khiển trình phát xuất hiện trong trình phát.

`disablekb`

Việc đặt giá trị của tham số thành `1` khiến trình phát không phản hồi các chế độ điều khiển bằng bàn phím. Giá trị mặc định là `0`, tức là các chế độ điều khiển bằng bàn phím được bật. Các chế độ điều khiển bằng bàn phím hiện được hỗ trợ là:

Thông số

	<p>Phím cách hoặc phím [k]: Phát / Tạm dừng</p> <p>Mũi tên trái: Tua lại 5 giây trong video hiện tại</p> <p>Mũi tên phải: Tua đi 5 giây trong video hiện tại</p> <p>Mũi tên lên: Tăng âm lượng</p> <p>Mũi tên xuống: Giảm âm lượng</p> <p>[f]: Bật/tắt chế độ hiển thị toàn màn hình</p> <p>[j]: Tua lại 10 giây trong video hiện tại</p> <p>[l]: Tua đi 10 giây trong video hiện tại</p> <p>[m]: Tắt hoặc bật tiếng video</p> <p>[0-9]: Chuyển đến một thời điểm trong video. 0 chuyển đến phần đầu video, 1 chuyển đến điểm 25% video, 2 chuyển đến điểm 50% video, v.v.</p>
<code>enablejsapi</code>	<p>Việc đặt giá trị của tham số thành 1 cho phép kiểm soát trình phát thông qua các lệnh gọi API. Giá trị mặc định là 0, tức là người chơi không thể điều khiển bằng API đó.</p> <p>Để biết thêm thông tin về IFrame API và cách sử dụng API này, hãy xem tài liệu về IFrame API.</p>
<code>end</code>	<p>Tham số này chỉ định thời gian (tính bằng giây kể từ khi video bắt đầu) mà trình phát sẽ dừng video. Giá trị tham số là một số nguyên dương.</p> <p>Xin lưu ý rằng thời gian được đo từ đầu video chứ không phải từ giá trị của tham số trình phát <code>start</code> hoặc tham số <code>startSeconds</code>. Tham số này được dùng trong các hàm của Video API để tải hoặc xếp hàng video.</p>
<code>fs</code>	<p>Việc đặt tham số này thành 0 sẽ ngăn nút toàn màn hình xuất hiện trong trình phát. Giá trị khác 0 cho phép người dùng nhấn nút để kích hoạt nút toàn màn hình xuất hiện.</p>
<code>hl</code>	<p>Đặt ngôn ngữ giao diện của trình phát. Giá trị tham số là mã ngôn ngữ ISO 639-1 gồm 2 chữ cái của một ngôn ngữ được chỉ định đầy đủ. Ví dụ: <code>fr</code> và <code>fr-ca</code> đều là giá trị hợp lệ. Các mã ngôn ngữ khác, chẳng hạn như thẻ ngôn ngữ IETF (BCP 47) cũng có thể được xử lý đúng cách.</p> <p>Ngôn ngữ giao diện được dùng cho chú thích công cụ trong trình phát và cũng ảnh hưởng đến đề xuất đề mặc định. Xin lưu ý rằng YouTube có thể chọn một ngôn ngữ phụ đề khác cho một video dựa trên lựa chọn ưu tiên về ngôn ngữ của người dùng đó và khả năng cung cấp phụ đề.</p>

Thông số

`iv_load_policy`

Việc đặt giá trị của tham số thành 1 sẽ khiến chú thích video xuất hiện theo mặc định, còn thành 3 sẽ khiến chú thích video không xuất hiện theo mặc định. Giá trị mặc định là 1.

`list`

Tham số `list`, cùng với tham số [listType](#), xác định nội dung sẽ tải trong trình phát.

Nếu giá trị tham số [listType](#) là `user_uploads`, thì giá trị tham số `list` sẽ xác định kênh video được tải lên sẽ được tải.

Nếu giá trị tham số [listType](#) là `playlist`, thì giá trị tham số `list` sẽ chỉ định một mã danh sách phát trên YouTube. Trong giá trị tham số, bạn cần thêm các chữ cái `PL` vào trước mã danh sách phát như trong ví dụ bên dưới.

```
https://www.youtube.com/embed?
    listType=playlist
    &list=PLC77007E23FF423C6
```

Nếu giá trị tham số [listType](#) là `search`, thì giá trị tham số `list` sẽ chỉ định cụm từ tìm kiếm. **Lưu ý: Chức năng này đã ngừng hoạt động và sẽ không được hỗ trợ nữa kể từ ngày 15 tháng 1 năm 2020.**

Lưu ý: Nếu bạn chỉ định giá trị cho các tham số `list` và `listType`, thì URL nhúng `IFrame` sẽ chỉ định mã video.

`listType`

Tham số `listType`, cùng với tham số [list](#), xác định nội dung sẽ tải trong trình phát. Giá trị mặc định là `playlist` và `user_uploads`.

Nếu bạn chỉ định giá trị cho tham số [list](#) và `listType`, thì URL nhúng `IFrame` không cần tham số `listType`.

Lưu ý: Giá trị tham số thứ ba, `search`, đã ngừng hoạt động và sẽ không còn được hỗ trợ nữa kể từ ngày 15 tháng 1 năm 2020.

`loop`

Trong trường hợp trình phát video đơn lẻ, chế độ cài đặt 1 sẽ khiến trình phát phát đi phát đi liên tục. Trong trường hợp trình phát danh sách phát (hoặc trình phát tùy chỉnh), trình phát sẽ phát hết danh sách phát rồi bắt đầu lại ở video đầu tiên.

Các giá trị được hỗ trợ là 0 và 1, còn giá trị mặc định là 0.

Lưu ý: Tham số này có chế độ hỗ trợ hạn chế trong các thành phần nhúng `IFrame`. Để phát video, hãy đặt giá trị tham số `loop` thành 1 và đặt giá trị tham số [playlist](#) thành mã video hoặc mã danh sách phát được chỉ định trong URL Player API:

```
https://www.youtube.com/embed/VIDEO_ID?playlist=VIDEO_ID&loop=1
```

Thông số

<code>modestbranding</code>	Lưu ý: Tham số này không được dùng nữa và không có hiệu lực. Hãy xem thông báo về dùng nữa để biết thêm thông tin.
<code>origin</code>	Tham số này cung cấp thêm một biện pháp bảo mật cho IFrame API và chỉ được hỗ trợ ở phần nhúng IFrame. Nếu đang sử dụng IFrame API (tức là bạn đang đặt giá trị tham số enablejsapi thành 1), thì bạn phải luôn chỉ định miền của mình làm giá trị tham số <code>origin</code> .
<code>playlist</code>	Tham số này chỉ định danh sách mã video được phân tách bằng dấu phẩy để phát. Nếu không đặt một giá trị, video đầu tiên phát sẽ là <code>VIDEO_ID</code> được chỉ định trong đường dẫn URL và chỉ định trong tham số <code>playlist</code> sẽ phát sau đó.
<code>playsinline</code>	<p>Tham số này kiểm soát việc video phát ở chế độ nội tuyến hay toàn màn hình trên iOS. Có hai lựa chọn:</p> <p>0: Kết quả ở chế độ phát toàn màn hình. Đây hiện là giá trị mặc định, mặc dù giá trị mặc định sẽ thay đổi.</p> <p>1: Kết quả là chế độ phát trong dòng cho trình duyệt di động và cho <code>WebViews</code> được tạo bằng tính năng allowsInlineMediaPlayback được đặt thành <code>YES</code>.</p>
<code>rel</code>	<p>Lưu ý: Tham số này sẽ thay đổi từ ngày 25 tháng 9 năm 2018 trở đi.</p> <p>Trước khi thay đổi, tham số này cho biết liệu trình phát có nên hiển thị video có liên quan hay không.</p> <p>Nếu giá trị của tham số được đặt thành 1 (giá trị mặc định), thì trình phát sẽ hiển thị các video có liên quan.</p> <p>Nếu giá trị của tham số được đặt thành 0, thì trình phát sẽ không hiển thị video có liên quan.</p> <p>Sau khi thay đổi, bạn sẽ không thể tắt video có liên quan. Thay vào đó, nếu thông số <code>rel</code> được đặt thành 0, thì các video có liên quan sẽ đến từ cùng một kênh với video vừa được phát.</p>
<code>start</code>	Tham số này khiến trình phát bắt đầu phát video tại số giây đã cho tính từ đầu video. Giá trị phải là một số nguyên dương. Xin lưu ý rằng tương tự như hàm seekTo , trình phát sẽ tìm kiếm vị trí gần nhất với thời gian mà bạn chỉ định. Điều này có nghĩa là đôi khi đầu phát có thể tìm đến thời gian được yêu cầu, thường là không quá khoảng 2 giây.
<code>widget_referrer</code>	Tham số này xác định URL nơi trình phát được nhúng. Giá trị này được dùng trong báo cáo YouTube Analytics khi trình phát YouTube được nhúng vào một tiện ích, rồi tiện ích đó được nhúng vào một web hoặc ứng dụng. Trong trường hợp đó, thông số origin sẽ xác định miền của nhà cung cấp tiện ích, nhưng YouTube Analytics không nên xác định nhà cung cấp tiện ích là nguồn lưu lượng truy cập thực tế. Thay vào đó, YouTube Analytics sử dụng giá trị tham số <code>widget_referrer</code> để xác định nguồn lưu lượng truy cập liên kết với nguồn lưu lượng truy cập.

Nhật ký sửa đổi

August 15, 2023

Note: This is a deprecation announcement for the `modestbranding` parameter.

The [modestbranding](#) parameter is deprecated and will have no effect. To align with YouTube's branding requirements, the player now determines the appropriate branding treatment based on a combination of factors, including player size, other API parameters (e.g. `controls`), and additional signals.

April 27, 2021

The [playsinline](#) parameter definition has been updated slightly to explain that the parameter pertains to both mobile browsers on iOS and iOS WebViews.

October 15, 2020

The [autoplay](#) parameter definition has been updated to explain that if you enable Autoplay, playback will occur without any user interaction with the player; playback data collection and sharing will therefore occur upon page load.

October 13, 2020

Note: This is a deprecation announcement for the embedded player functionality that lets you configure the player to load search results. This announcement affects the player's [list](#) and [listType](#) parameters.

This change will become effective on or after 15 November 2020. After that time, setting the `listType` parameter value to `search` will generate a 4xx response code, such as 404 (Not Found) or 410 (Gone).

As an alternative, you can use the YouTube Data API's [search.list](#) method to retrieve search results and then load selected videos in the player.

In addition to the change described above, the `showinfo` parameter, which was [deprecated](#) in 2018, has been removed from this document.

August 23, 2018

Note: This is a deprecation announcement for the `showinfo` parameter. In addition, the behavior for the `rel` parameter is changing. Titles, channel information, and related videos are an important part

of YouTube's core user experience, and these changes help to make the YouTube viewing experience consistent across different platforms.

- The behavior for the `rel` parameter is changing on or after September 25, 2018. The effect of the change is that you will not be able to disable related videos. However, you will have the option of specifying that the related videos shown in the player should be from the same channel as the video that was just played.

To be more specific:

- Prior to the change, if the parameter's value is set to `0`, then the player does not show related videos.
- After the change, if the `rel` parameter is set to `0`, the player will show related videos that are from the same channel as the video that was just played.
- The `showinfo` parameter, which indicates whether the player should display information like the video title and uploader before the video starts playing, is also being deprecated. Following the change, the channel avatar and video title will always display before playback begins, when playback is paused, and when playback ends. The avatar being displayed is new behavior that will be consistent across all embedded players.

These changes will become effective on or after September 25, 2018. After that time, the `showinfo` parameter will be ignored. The behavior following the changes is consistent with the current default behavior for embedded players with the exception of the channel avatar changes mentioned above.

August 16, 2018

The [cc_lang_pref](#) parameter can be used to specify the default language that the player will use to display captions. This parameter can be used in conjunction with the [cc_load_policy](#) parameter to automatically show captions in the specified language during playback.

September 15, 2017

The [controls](#) parameter's definition has been updated to remove references to the deprecated Flash (AS3) player. The value `2` has been deprecated as it was originally designed to provide a performance improvement for embeds that loaded a Flash player.

June 12, 2017

The new [widget_referrer](#) parameter helps to enable more accurate YouTube Analytics reporting when the YouTube player is embedded in a widget, and that widget is then embedded in a web page or application.

November 1, 2016

This document has been updated to remove references to the deprecated Flash (AS3) player as well as to parameters only supported by that player. The YouTube Flash player was deprecated in January 2015.

October 20, 2016

This update contains the following changes:

- The [disablekb](#) parameter definition has been corrected to note that the default value is `0`, which means that keyboard controls are enabled. A value of `1` indicates that keyboard controls should be disabled.

The list of keyboard controls that the player supports has also been updated to include the following:

- `[f]`: Toggle full-screen display
- `[j]`: Jump back 10 seconds in the current video
- `[k]`: Play / Pause
- `[l]`: Jump ahead 10 seconds in the current video
- `[m]`: Mute or unmute the video
- `[0-9]`: Jump to a point in the video. `0` jumps to the beginning of the video, `1` jumps to the time 10% into the video, `2` jumps to the point 20% into the video, and so forth.

In addition, the effect of pressing the `[arrow left]` or `[arrow right]` keys has changed. These keys now jump 5 seconds back (arrow left) or ahead (arrow right) in the current video.

August 11, 2016

This update contains the following changes:

- The newly published YouTube API Services Terms of Service ("the Updated Terms"), discussed in detail on the [YouTube Engineering and Developers Blog](#), provides a rich set of updates to the current Terms of Service. In addition to the [Updated Terms](#), which will go into effect as of February 10, 2017, this update includes several supporting documents to help explain the policies that developers must follow.

The full set of new documents is described in the [revision history for the Updated Terms](#). In addition, future changes to the Updated Terms or to those supporting documents will also be explained in that revision history. You can subscribe to an RSS feed listing changes in that revision history from a link in that document.

December 18, 2015

European Union (EU) laws require that certain disclosures must be given to and consents obtained from end users in the EU. Therefore, for end users in the European Union, you must comply with the [EU User Consent Policy](#). We have added a notice of this requirement in our [YouTube API Terms of Service](#).

August 19, 2015

- The [autohide](#) parameter has been deprecated for the HTML5 player. In HTML5 players, the video progress bar and player controls display or hide automatically. That behavior corresponds to an `autohide` setting of 1.
- The `theme` parameter has been deprecated for the HTML5 player. HTML5 players now always use the dark theme.

March 9, 2015

- The document has been updated to reflect the fact that YouTube `<object>` embeds, the YouTube Flash Player API and the YouTube JavaScript Player API have all been [deprecated](#) as of January 27, 2015. A deprecation warning appears in several sections of this document to help point readers to the [IFrame Player API](#) as an alternative.
- The definition of the [autohide](#) parameter has been updated to clarify the meaning of the parameter's values. The default behavior (`autohide=2`) is that if the player has a 16:9 or 4:3 aspect ratio, the player's video progress bar and player controls display or hide automatically. Otherwise, those controls are visible throughout the video.
- The definition of the [hl](#) parameter has been updated to note that the parameter value could be an ISO 639-1 two-letter language code or a fully specified locale. For example, `fr` and `fr-ca` are both valid parameter values.
- The definition of the [enablejsapi](#) parameter has been reworded to clarify that the parameter enables a player to be controlled via API calls. The API could be either the [IFrame Player API](#) or the [JavaScript Player API](#).

October 14, 2014

- The HTML5 player now supports the [cc_load_policy](#), [disablekb](#), [end](#), [fs](#), [list](#), and [listType](#) parameters. The parameter descriptions have been updated accordingly.

July 18, 2014

- The new [hl](#) parameter can be used to set the player's interface language. The interface language is used for tooltips in the player and also affects the default caption track. The

selected caption track may also depend on the availability of caption tracks and user's individual language preferences.

The parameter's value is an [ISO 639-1 two-letter language code](#), though other language input codes, such as IETF language tags (BCP 47) may also be handled properly.

- The definition of the [playsinline](#) parameter, which only affects HTML5 players on iOS, has been modified slightly. The definition now notes that setting the parameter value to 1 causes inline playback only for `UIWebViews` created with the `allowsInlineMediaPlayback` property set to `TRUE`.

January 28, 2014

- The [playsinline](#) parameter controls whether videos play inline or fullscreen in an HTML5 player on iOS. Setting the value to 1 causes inline playback.
- The [Selecting content to play](#) section has been updated to explain how to find YouTube video IDs and playlist IDs using the YouTube Data API (v3) rather than the older API version.
- The [controls](#) parameter's definition has been updated to reflect the fact that the parameter value only affects the time that the Flash player actually loads in IFrame embeds. In addition, for IFrame embeds, the parameter value also determines when the controls display in the player. If you set the parameter's value to 2, then the controls display and the Flash player loads after the user initiates the video playback.

May 10, 2013

This update contains the following changes:

- YouTube no longer identifies experimental API features and services. Instead, we now provide a list of [YouTube APIs that are subject to the deprecation policy](#).

July 20, 2012

This update contains the following changes:

- The definition of the [controls](#) parameter has been updated to reflect support for a parameter value of 2 and to explain that, for AS3 players, the parameter value determines the time when the Flash player actually loads. If the [controls](#) parameter is set to 0 or 1, the Flash player loads immediately. If the parameter value is 2, the Flash player does not load until the video playback is initiated.

June 5, 2012

This update contains the following changes:

- The HTML5 player now supports the [color](#), [modestbranding](#), and [rel](#) parameters, and the definitions for these parameters have been updated accordingly.
- The definition of the [showinfo](#) parameter has been updated to explain how that if the player is loading a playlist, and you explicitly set the parameter value to `1`, then, upon loading, the player will also display thumbnail images for the videos in the playlist. Note that this functionality is only supported for the AS3 player since that is the only player that can load a playlist.

May 4, 2012

This update contains the following changes:

- The [showinfo](#) parameter's definition has been updated to reflect the fact that the HTML5 player supports this parameter.

May 3, 2012

This update contains the following changes:

- The new [end](#) parameter specifies the time, measured in seconds from the start of the video, when the player should stop playing a video. Note that the time when playback is stopped is measured from the beginning of the video and not from the value of either the `start` player parameter or the `startSeconds` parameter, which is used in YouTube Player API functions for loading or queueing a video.

March 29, 2012

This update contains the following changes:

- The new [Embedding a YouTube player](#) section explains different ways to embed a YouTube player in your application. This section covers manual IFrame embeds, IFrame embeds that use the IFrame Player API, and AS3 and AS2 object embeds. This section incorporates information from the old **Example usage** section, which has been removed.
- The new [Selecting content to play](#) section explains how to configure the player to load a video, a playlist, search results for a specified query, or uploaded videos for a specified user.
- The new [list](#) and [listType](#) parameters let you specify the content that the player should load. You can specify a playlist, a search query, or a particular user's uploaded videos.
- The definitions of the [fs](#) and [rel](#) parameters have been updated to more clearly explain the default parameter values for the AS3 player.
- The [border](#), [color1](#), [egm](#), [hd](#), and [showsearch](#) parameters, which are all only supported for the deprecated AS2 Player API, have been moved to a new section named [deprecated parameters only used in the AS2 Player API](#).
- The document no longer provides a way to filter the parameter list to only display parameters supported in either the AS3, AS2, or HTML5 player. Instead, each

parameter definition has been updated to identify the players that support that parameter.

August 11, 2011

This update contains the following changes:

- The new [theme](#) and [color](#) parameters let you customize the appearance of the embedded player's player controls. See the [YouTube API blog](#) for more information.

June 8, 2011

This update contains the following changes:

- The new [modestbranding](#) parameter lets you use a YouTube player that does not show a YouTube logo. As of this release, the parameter was only supported for the AS3 embedded player and for IFrame embeds that loaded the AS3 player. As of June 5, 2012, the HTML5 player also supported this parameter.

February 14, 2011

This update contains the following changes:

- The documentation has been updated to note that the AS2 Player API has been deprecated. The deprecation of the AS2 Player API was actually [announced](#) on October 14, 2009.

February 3, 2011

This update contains the following changes:

- The documentation has been updated to identify parameters supported in the HTML5 ([IFrame](#)) embedded player.
- The document now displays all of the parameters supported in any of YouTube's embedded players (HTML5, AS3, AS2).
- The following parameters are supported in the AS2 player but have been deprecated for the newer AS3 and HTML5 players: [border](#), [color1](#), [color2](#), [egm](#), [hd](#), and [showsearch](#).

In addition, the [loop](#) parameter has limited support in the AS3 player and in IFrame embeds, which could load either an AS3 or HTML player. Currently, the [loop](#) parameter only works in the AS3 player when used in conjunction with the [playlist](#) parameter. To loop a single video, set the [loop](#) parameter to 1 and set the [playlist](#) parameter value to the same video ID already specified in the Player API URL:

`https://www.youtube.com/v/VIDEO_ID?version=3&loop=1&playlist=VIDEO_ID`

Similarly, the [controls](#) and [playlist](#) parameters are supported in the AS3 and HTML5 players but are not and will not be supported in the AS2 player.

As noted above, IFrame embeds can load either an AS3 or HTML5 player. Though some parameters are not supported in both players, an IFrame embed that loads the AS3 player will support all parameters that work with that player and ignore all other parameters. Similarly, an IFrame embed that loads the HTML5 player will support all parameters that work with that player while ignoring all other parameters.

- The parameter list has been updated to include the [autohide](#) parameter, which indicates whether the player's video controls will automatically hide after a video begins playing.
- The parameter list has been updated to include the [controls](#) parameter, which indicates whether the player's video controls will display at all. (Player controls do display by default.)
- The parameter list has been updated to include the [playlist](#) parameter, which specifies a comma-separated list of video IDs to play.
- The definition of the [fs](#) has been updated to note that the fullscreen option will not work if you load the YouTube player into another SWF.
- The [example](#) at the end of the document has been updated to use the embedded AS3 player instead of the embedded AS2 player. The parameters used in the example have also been updated to only include parameters that the AS3 player supports.

In addition, the instructions for constructing the URLs that contain player parameters have been updated to reflect the URL formats used by the AS3 and AS2 embedded and chromeless players as well as by the HTML5 player.