



# 媒体云播放器

## Web SDK API 参考文档

V1.0

发布日期： 2013年6月25日

百度开发者中心

（版权所有，翻版必究）

目录

第 1 章 控制接口 ..... 4

1. 1. CYBERPLAYER ..... 4

1. 2. CYBERPLAYER::SETUP ..... 4

1. 3. CYBERPLAYER::REMOVE ..... 5

1. 4. CYBERPLAYER::GETDURATION ..... 6

1. 5. CYBERPLAYER::GETFULLSCREEN ..... 6

1. 6. CYBERPLAYER::GETHEIGHT ..... 6

1. 7. CYBERPLAYER::GETMETA ..... 6

1. 8. CYBERPLAYER::GETMUTE ..... 6

1. 9. CYBERPLAYER::GETPLAYLIST ..... 6

1. 10. CYBERPLAYER::GETPOSITION ..... 7

1. 11. CYBERPLAYER::GETRENDERINGMODE ..... 7

1. 12. CYBERPLAYER::GETSTATE ..... 7

1. 13. CYBERPLAYER::GETVOLUME ..... 7

1. 14. CYBERPLAYER::GETWIDTH ..... 7

1. 15. CYBERPLAYER::SETFULLSCREEN ..... 8

1. 16. CYBERPLAYER::SETMUTE ..... 8

1. 17. CYBERPLAYER::PLAYLISTITEM ..... 8

1. 18. CYBERPLAYER::PLAYLISTPREV ..... 8

1. 19. CYBERPLAYER::PLAYLISTNEXT ..... 8

1. 20. CYBERPLAYER::RESIZE ..... 8

1. 21. CYBERPLAYER::PLAY ..... 9

1. 22. CYBERPLAYER::PAUSE ..... 9

1. 23. CYBERPLAYER::STOP ..... 9

1. 24. CYBERPLAYER::SEEK ..... 9

1. 25. CYBERPLAYER::SETVOLUME ..... 9

第 2 章 事件响应接口 .....10

2. 1. CYBERPLAYER::ONBUFFER ..... 10

2. 2. CYBERPLAYER::ONBUFFERCHANGE ..... 10

2. 3. CYBERPLAYER::ONBUFFERFULL ..... 10

2. 4. CYBERPLAYER::ONCOMPLETE ..... 10

2. 5. CYBERPLAYER::ONERROR ..... 10

2. 6. CYBERPLAYER::ONFULLSCREEN ..... 11

2. 7. CYBERPLAYER::ONIDLE ..... 11

2. 8. CYBERPLAYER::ONMETA ..... 11

2. 9. CYBERPLAYER::ONMUTE..... 11

2. 10. CYBERPLAYER::ONPAUSE ..... 12

2. 11. CYBERPLAYER::ONPLAY ..... 12

2. 12. CYBERPLAYER::ONPLAYLIST ..... 12

2. 13. CYBERPLAYER::ONPLAYLISTITEM..... 12

2. 14. CYBERPLAYER::ONREADY ..... 12

2. 15. CYBERPLAYER::ONRESIZE ..... 13

2. 16. CYBERPLAYER::ONSEEK ..... 13

2. 17. CYBERPLAYER::ONTIME ..... 13

2. 18. CYBERPLAYER::ONVOLUME ..... 13

# 第1章 控制接口

## 1.1. cyberplayer

参数列表: String - 播放器 DIV 容器的 id, 用于盛放播放器  
返回类型: Object  
接口功能: 在指定的 DIV 容器内创建一个播放器实例。  
代码示例:

```
var myPlayer = cyberplayer("playerContainer").setup({
    flashplayer:"cyberplayer.swf",
    width:600,
    height:450,
    file:"http://ip:port/playlist.m3u8",
    image: "xxx.jpg"
});
```

## 1.2. cyberplayer::setup

参数列表: Object  
返回类型: 无  
接口功能: 创建一个播放器。  
代码示例:

```
var myPlayer = cyberplayer("playerContainer").setup({
    flashplayer:"player.swf",
    width:600,
    height:450,
    file:" http://ip:port/playlist.m3u8",
    image: "xxx.jpg"
});
```

参数详解:  
setup 只有一个 Object 类型的参数, 该参数是一个参数集合, 该参数集合所包含的参数元素及使用  
方法如下描述:

参数名称	参数解释	备注
flashplayer	指定 flash 播放器 swf 文件的存放地址, 可以是相对路径, 也可以以 http 协议开头的绝对路径。	必须
width	指定要创建的播放窗口的宽度。	必须
height	指定要创建的播放窗口的高度。	必须
playlist	设置播放列表的显示模式和位置: 1.none; 2.right; 3.left; 4.bottom; 5.top; 6.over	可选
playlistfile	设置播放列表文件, 每个视频文件的播放参数在播放列表文件里单独设置, 播放列表文件格式如下:  <rss version="2.0" xmlns:cbplayer=" <a href="http://cyberplayer.baidu.com/">http://cyberplayer.baidu.com/</a> "	可选 ( 设置 playlist 参数时为必选项)

	<pre>encoding="utf-8"&gt; &lt;channel&gt;   &lt;title&gt;Example playlist&lt;/title&gt;    &lt;item&gt;     &lt;title&gt;标题 1&lt;/title&gt;     &lt;description&gt;描述&lt;/description&gt;     &lt;cbplayer:image&gt;snap1.jpg&lt;/cbplayer:image&gt;     &lt;cbplayer:file&gt;http://ip:port/playlist1.m3u8&lt;/cbpl ayer:file&gt;   &lt;/item&gt;   &lt;item&gt;     &lt;title&gt;标题 2&lt;/title&gt;     &lt;description&gt;描述&lt;/description&gt;     &lt;cbplayer:image&gt;snap2.jpg&lt;/cbplayer:image&gt;     &lt;cbplayer:file&gt;http://ip:port/playlist2.m3u8&lt;/cbpl ayer:file&gt;   &lt;/item&gt; &lt;/channel&gt; &lt;/rss&gt;</pre>	
item	设置播放列表起始播放位置，默认是 0 表示从第一个开始播放	可选
autoStart	设置是否在播放器载入后自动播放：true：自动播放；false：不自动播放。	可选
repeat	设置视频的重复播放模式，重复模式分为：1.none:无重复；2.single:单一重复模式；3.list:整个列表重复一次；4.always:始终重复（不管是列表还是单条目）	可选
streamer	设置 cb 服务器访问地址，形式为：rtmp://ip:port/app	可选
file	设置媒体流名称或文件名或 M3U8 播放列表地址	必选
image	设置媒体流的预览图	可选
volume	设置播放器音量大小	可选
controlbar	设置播放器控制条的显示模式，显示模式分为：1.none:不显示；2.over:悬浮（鼠标无操作时自动隐藏）；3.top:上方固定（不隐藏）；4.bottom:下方固定（不隐藏）	可选
skin	设置播放器皮肤包	可选

1. 3. cyberplayer::remove

参数列表：无  
返回类型：无  
接口功能：移除当前播放器。  
代码示例：  
myPlayer.remove();

---

#### 1.4. **cyberplayer::getDuration**

参数列表：无

返回类型：Number - 当前播放的媒体文件的总时长

接口功能：获取当前播放的媒体文件的总时长。

代码示例：

```
var duration = myPlayer.getDuration();
```

---

#### 1.5. **cyberplayer::getFullscreen**

参数列表：无

返回类型：Boolean - true : 全屏 ; false : 非全屏

接口功能：获取当前播放器是否处于全屏状态。

代码示例：

```
var fullscreen = myPlayer.getFullscreen();
```

---

#### 1.6. **cyberplayer::getHeight**

参数列表：无

返回类型：Number - 播放器高度

接口功能：获取当前播放器显示窗口高度。

代码示例：

```
var height = myPlayer.getHeight();
```

---

#### 1.7. **cyberplayer::getMeta**

参数列表：无

返回类型：Object - 元数据对象

接口功能：获取当前播放的媒体文件的元数据对象。

代码示例：

```
var meta = myPlayer.getMeta();
```

---

#### 1.8. **cyberplayer::getMute**

参数列表：无

返回类型：Boolean - true : 声音打开; false : 声音关闭

接口功能：获取当前播放器声音是否打开。

代码示例：

```
var mute = myPlayer.getMute();
```

---

#### 1.9. **cyberplayer::getPlaylist**

参数列表：无

返回类型：Object - 播放列表

接口功能：获取当前播放器的播放列表。

代码示例：

```
var duration = myPlayer.getDuration();
```

---

### 1. 10. **cyberplayer::getPosition**

参数列表：无

返回类型：Number

接口功能：获取当前播放的媒体文件的播放位置。

代码示例：

```
var position = myPlayer.getPosition();
```

---

### 1. 11. **cyberplayer::getRenderingMode**

参数列表：无

返回类型：String - 渲染模式

接口功能：获取当前播放器的渲染模式。

代码示例：

```
var mode = myPlayer.getRenderingMode();
```

---

### 1. 12. **cyberplayer::getState**

参数列表：无

返回类型：Number

接口功能：获取当前播放器状态。

代码示例：

```
var state = myPlayer.getState();
```

---

### 1. 13. **cyberplayer::getVolume**

参数列表：无

返回类型：Number

接口功能：获取当前播放器的音量。

代码示例：

```
var volume = myPlayer.getVolume();
```

---

### 1. 14. **cyberplayer::getWidth**

参数列表：无

返回类型：Number

接口功能：获取当前播放器窗口宽度。

代码示例：

```
var width = myPlayer.getWidth();
```

---

### 1. 15. **cyberplayer::setFullscreen**

参数列表: Boolean - true : 全屏; false : 非全屏

返回类型: 无

接口功能: 设置当前播放器是否全屏。

代码示例:

```
myPlayer.setFullscreen(true);
```

---

### 1. 16. **cyberplayer::setMute**

参数列表: Boolean - true : 打开声音; false : 关闭声音

返回类型: 无

接口功能: 设置当前播放器的声音开关。

代码示例:

```
myPlayer.setMute(true);
```

---

### 1. 17. **cyberplayer::playlistItem**

参数列表: Number

返回类型: Object

接口功能: 获取指定的播放条目。

代码示例:

```
var item = myPlayer.playlistItem(2);
```

---

### 1. 18. **cyberplayer::playlistPrev**

参数列表: 无

返回类型: Object

接口功能: 播放当前播放条目的前一项，并返回该项对象。

代码示例:

```
var item = myPlayer.playlistPrev();
```

---

### 1. 19. **cyberplayer::playlistNext**

参数列表: 无

返回类型: Object

接口功能: 播放当前播放条目的后一项，并返回该项对象。

代码示例:

```
var item = myPlayer.playlistNext();
```

---

### 1. 20. **cyberplayer::resize**

参数列表: Number - 播放器宽, Number - 播放器高

返回类型: 无

接口功能: 调整播放器大小。



代码示例：

```
myPlayer.resize(600, 400);
```

---

### 1. 21. **cyberplayer::play**

参数列表：无

返回类型：无

接口功能：开始播放当前媒体内容。

代码示例：

```
myPlayer.play();
```

---

### 1. 22. **cyberplayer::pause**

参数列表：无

返回类型：Number

接口功能：暂停播放当前媒体内容。

代码示例：

```
myPlayer.pause();
```

---

### 1. 23. **cyberplayer::stop**

参数列表：无

返回类型：无

接口功能：停止播放当前媒体内容。

代码示例：

```
myPlayer.stop();
```

---

### 1. 24. **cyberplayer::seek**

参数列表：Number - 目标播放时间

返回类型：无

接口功能：定位当前媒体内容开始播放的位置。

代码示例：

```
myPlayer.seek(102);
```

---

### 1. 25. **cyberplayer::setVolume**

参数列表：Number - 音量大小（0-100）

返回类型：无

接口功能：设置播放器音量。

代码示例：

```
myPlayer.setVolume(90);
```

## 第2章 事件响应接口

### 2.1. cyberplayer::onBuffer

参数列表: Function

返回类型: 无

接口功能: 设置播放器缓存事件监听函数。

代码示例:

```
myPlayer.onBuffer(function(event) {  
    log("onBuffer");  
});
```

---

### 2.2. cyberplayer::onBufferChange

参数列表: Function

返回类型: void

接口功能: 设置播放器缓存变化监听函数。

代码示例:

```
myPlayer.onBufferChange(function(event) {  
    log("buffer percent : " + event.bufferPercent + "; position : " + event.position);  
});
```

---

### 2.3. cyberplayer::onBufferFull

参数列表: Function

返回类型: 无

接口功能: 设置播放器缓存满载状态监听函数。

代码示例:

```
myPlayer.onBufferFull(function(event) {  
    log("onBufferFull");  
});
```

---

### 2.4. cyberplayer::onComplete

参数列表: Function

返回类型: 无

接口功能: 设置播放器完成事件监听函数。

代码示例:

```
myPlayer.onComplete(function(event) {  
    log("onComplete");  
});
```

---

### 2.5. cyberplayer::onError

参数列表: Function

返回类型: 无

接口功能：设置播放器出错监听函数。

代码示例：

```
myPlayer.onError(function(event) {  
    log("onError");  
});
```

---

## 2.6. **cyberplayer::onFullscreen**

参数列表：Function

返回类型：无

接口功能：设置播放器全屏状态变化监听函数。

代码示例：

```
myPlayer.onFullscreen(function(event) {  
    log("onFullscreen");  
});
```

---

## 2.7. **cyberplayer::onIdle**

参数列表：Function

返回类型：无

接口功能：设置播放器缓空闲事件监听函数。

代码示例：

```
myPlayer.onIdle(function(event) {  
    log("onIdle");  
});
```

---

## 2.8. **cyberplayer::onMeta**

参数列表：Function

返回类型：无

接口功能：设置播放器元数据接收函数。

代码示例：

```
myPlayer.onMeta(function(event) {  
    log("onMeta");  
});
```

---

## 2.9. **cyberplayer::onMute**

参数列表：Function

返回类型：无

接口功能：设置播放器音频开关变化监听函数。

代码示例：

```
myPlayer.onMute(function(event) {  
    log("onMute");  
});
```

---

## 2.10. cyberplayer::onPause

参数列表: Function

返回类型: 无

接口功能: 设置播放器暂停事件监听函数。

代码示例:

```
myPlayer.onPause(function(event) {  
    log("onPause");  
});
```

---

## 2.11. cyberplayer::onPlay

参数列表: Function

返回类型: 无

接口功能: 设置播放器播放事件监听函数。

代码示例:

```
myPlayer.onPlay(function(event) {  
    log("onPlay");  
});
```

---

## 2.12. cyberplayer::onPlaylist

参数列表: Function

返回类型: 无

接口功能: 设置播放器播放列表接收函数。

代码示例:

```
myPlayer.onPlaylist(function(event) {  
    log("buffer percent : " + event.bufferPercent + "; position : " + event.position);  
});
```

---

## 2.13. cyberplayer::onPlaylistItem

参数列表: Function

返回类型: 无

接口功能: 设置播放器当前播放条目变化监听函数。

代码示例:

```
myPlayer.onPlaylistItem(function(event) {  
    log("onPlaylistItem");  
});
```

---

## 2.14. cyberplayer::onReady

参数列表: Function

返回类型: 无

接口功能: 设置播放器就绪事件监听函数。

代码示例:

```
myPlayer.onReady(function(event) {
```

```
        log("onReady");
    });
```

---

## 2.15. cyberplayer::onResize

参数列表: Function

返回类型: 无

接口功能: 设置播放器窗口大小变化监听函数。

代码示例:

```
myPlayer.onResize(function(event) {
    log("onResize");
});
```

---

## 2.16. cyberplayer::onSeek

参数列表: Function

返回类型: 无

接口功能: 设置播放器拖动事件监听函数。

代码示例:

```
myPlayer.onSeek(function(event) {
    log("onSeek");
});
```

---

## 2.17. cyberplayer::onTime

参数列表: Function

回调参数列表: Object - Time 事件

返回类型: 无

接口功能: 设置播放器播放进度变化事件监听函数。

代码示例:

```
myPlayer.onTime(function(event) {
    log("onTime");
});
```

---

## 2.18. cyberplayer::onVolume

参数列表: Function

返回类型: 无

接口功能: 设置播放器音量变化事件监听函数。

代码示例:

```
myPlayer.onVolume(function(event) {
    log("onVolume");
});
```