

Player API 开发指南

版本: 1.0.6

日期: 2016-03-16

北京梦之窗数码科技有限公司

目 录

| | |
|-------------------------------|---|
| 1. 概述..... | 1 |
| 2. 接口说明..... | 1 |
| 3. 配置说明..... | 2 |
| 4. 插件介绍..... | 4 |
| 4.1 字幕插件 | 4 |
| 4.2 弹幕插件 | 5 |
| 附录 1. Flash和Javascript交互..... | 6 |
| 附录 2. 播放器错误码 | 8 |

1. 概述

利用 Player API 与 CC视频播放器进行交互，可以改变播放器界面、实现一些增值功能。

目前仅限于Flash播放器，FlashPlayer插件版本要求为10.1及以上，播放器页面代码中必须允许播放器与页面脚本通信。

2. 接口说明

接口分为两种类型：方法、回调；

方法：播放器开放的JavaScript函数，播放器的句柄需要用Object标签的id属性或embed的name属性来获取；

回调：播放器运行过程中尝试调用页面JS函数，可理解为事件。

使用方法可参考[附录1. Flash与JavaScript交互](#)；

| 接口 | 类型 | 说明 |
|------------------------------|----|--|
| start() | 方法 | 开始播放 |
| pause() | 方法 | 暂停播放 |
| resume() | 方法 | 恢复播放 |
| seek(time) | 方法 | 定位至指定时间 参数 time 单位 秒 |
| getDuration() | 方法 | 获取视频片长，返回值单位 秒 |
| getPosition() | 方法 | 获取当前播放时间，返回值单位 秒 |
| getQualities() | 方法 | 获取当前视频可用清晰度列表，返回类型 Array， 格式如下： [{ value:"0", label:"普通" }, { value:"1", label:"清晰" }] |
| getTitle() | 方法 | 获取当前播放视频的标题,返回值文本 |
| setQuality(quality) | 方法 | 设置清晰度 参数 quality 为 getQualities方法获取的value值 |
| setVolume(volume) | 方法 | 设置音量 参数 volume 取值范围0 - 1 |
| normalScreen() | 方法 | 退出全屏 |
| setConfig(config) | 方法 | 配置播放器 参数 config 为配置对象，详见 3. 配置说明 |
| on_cc_player_init(vid,swfID) | 回调 | 播放器界面元素初始化时 vid：视频唯一标识 swfID：页面中播放器的id |

| 接口 | 类型 | 说明 |
|--------------------------------|----|----------------------------------|
| on_cc_player_pluginready(name) | 回调 | 插件准备就绪，意味着该插件相关接口可用 name：插件名称 |

3. 配置说明

播放器配置可以改变播放器的界面，开启更多的回调接口，还可以使用一些特殊的功能，如自定义全屏、字幕功能。

配置的方法是在回调接口 on_cc_player_init 触发时，调用 setConfig 方法传递一个 Object 对象，此对象每个 key 都是一个配置项，对应的 value 就是所配置的值。

播放器所支持的配置项如下：

| 配置项 | 说明 |
|---------------------|---|
| control_enable | 是否显示控制条 默认1，设置为0则不显示控制条 |
| progressbar_enable | 是否可操作进度条 默认1，设置为0则进度条不可操作 |
| loadingpic_enable | 是否启用缓冲片头 默认1，设置为0则不会出现缓冲片头 |
| loading_enable | 是否显示loading图标 默认1，设置为0则不显示loading图标 |
| bigbutton_enable | 是否显示非自动播放时出现的开始按钮和暂停时的大播放按钮 默认1，设置为0则不显示 如需要单独控制请使用 startbutton_enable, resumebutton_enable |
| startbutton_enable | 是否显示非自动播放时出现的开始按钮 默认1，设置为0则不显示 |
| resumebutton_enable | 是否显示暂停时的大播放按钮 默认1，设置为0则不显示 |
| tip_enable | 是否可显示播放器顶部的提示条 默认1，设置为0则不再出现提示条 |
| alert_enable | 是否可显示播放器中间的提示面板 默认1，设置为0则不再出现提示面板 |
| recommend_enable | 是否启用播放结束后的推荐界面 默认1，设置为0则播放结束后不显示推荐界面 |
| keyboard_enable | 是否启用按键操作 默认1，设置为0则按键操作不可用 |

| 配置项 | 说明 |
|------------------------|--|
| rightmenu_enable | 是否启用右侧菜单 默认1，设置为0则不显示右侧菜单 |
| fullscreen_enable | 是否使用自定义全屏操作 默认0，设置为1可以替代Flash全屏 |
| play_start_time | 设置开播的时间点，单位秒，如果超出视频时长会按0处理，此配置项能够使接着上次观看功能失效 |
| fullscreen_function | 替代Flash全屏的JS函数的名称，需 fullscreen_enable 设置为1，这个函数需要返回一个值以确定当前状态(0: 正常, 1: 全屏) |
| video_marks | 配置视频打点数据，此配置项的值是一个数组，格式如下： [{ time : time , desc : desc }] time 为视频时间，单位秒；desc 为对应时间的打点说明 |
| video_marks_json_url | 配置视频打点数据接口地址，优先级高于 video_marks，接口必须返回JSON数据，格式如下： { value : [{ time : time , desc : desc }] } |
| player_plugins | 配置播放器插件，详见 4. 插件介绍 |
| on_player_stop | 播放结束后回调JS函数的名称 默认 on_spark_player_stop |
| on_player_pause | 暂停播放时回调JS函数的名称 默认 on_spark_player_pause |
| on_player_resume | 暂停后继续播放时回调JS函数的名称 默认 on_spark_player_resume |
| on_player_ready | 播放器就绪回调JS函数的名称 默认 on_spark_player_ready |
| on_player_start | 开始播放时回调JS函数的名称 默认 on_spark_player_start |
| on_player_seek | 指定时间定位时回调JS函数的名称 回调时的参数： from : 定位之前的时间点，单位秒 to : 定位指定的时间点，单位秒 |
| on_player_buffering | 缓冲开始或结束时回调JS函数的名称 回调时的参数： flag : 是否缓冲中 (0: 否, 1: 是) |
| on_player_setquality | 清晰度改变时回调JS函数的名称 回调时的参数： quality : 当前清晰度 |
| on_player_volumechange | 音量改变时回调JS函数的名称 回调时的参数： vol : 当前音量 (范围0-1) |

| 配置项 | 说明 |
|----------------------|--|
| on_player_playerror | 当播放失败时回调JS函数的名称 回调时的参数： code：错误码，详见 附录2.播放器错误码 |
| on_player_fullscreen | 全屏或退出全屏时回调JS函数的名称 回调时的参数： flag：是否全屏 (0：否,1：是) |

以下代码为配置示例

```

<script type="text/javascript">

function customFullScreen(){
    //TODO 自定义全屏
}

function onPlayPaused(){
    //已暂停播放
}

function on_cc_player_init( vid, swfID ){
    var config = {};
    config.fullscreen_enable = 1;
    config.fullscreen_function = "customFullScreen"; //设置自定义全屏函数的名称
    config.on_player_pause = "onPlayPaused"; //设置当暂停播放时的回调函数的名称

    var player = getSWF( swfID );
    player.setConfig( config );
}

function getSWF( swfID ) {
    if( window.document[ swfID ] ) {
        return window.document[ swfID ];
    } else if( navigator.appName.indexOf( "Microsoft" ) == -1 ) {
        if( document.embeds && document.embeds[ swfID ] ) {
            return document.embeds[ swfID ];
        }
    } else {
        return document.getElementById( swfID );
    }
}

</script>

```

4. 插件介绍

插件是指基于播放器实现可插拔的独立功能块，用户通过播放器配置启用插件，从而可以使用相应的功能。要使用插件时，在 player_plugins 配置项中添加插件的名称和参数。

4.1 字幕插件

字幕插件可使播放器在播放视频的同时显示相关文字，启用时在配置中添加插件名称 **Subtitle**，参数如下：

| 参数 | 说明 |
|---------------|--------------------------------|
| url | 字幕文件地址，仅支持srt格式的字幕文件，此参数为必需参数 |
| size | 字体大小，默认 20 |
| color | 字体颜色，默认 0xFFFFFFFF |
| surroundColor | 文字环绕颜色，默认 0x000000 |
| font | 字体名称，默认 Times New Roman |
| bottom | 文本与播放区域底部的距离占播放区域高的百分比，默认 0.15 |
| code | 字幕文件编码，支持 utf-8 、gbk，默认 utf-8 |
| bgColor | 字幕的背景色,默认值是0xfffff |
| bgAlpha | 字幕的背景透明度,取值0到1,默认值是0 |
| leading | 字幕的行与行间距,默认值是0,可赋值整数 |

注意事项：字幕文件所在的服务器根目录需要部署一个安全策略文件(crossdomain.xml)，使播放器能够跨域访问字幕文件。

了解更多关于安全策略文件可访问 http://help.adobe.com/zh_CN/as3/dev/WS5b3ccc516d4fbf351e63e3d118a9b90204-7c85.html#WS5b3ccc516d4fbf351e63e3d118a9b90204-7e08

以下代码为字幕插件配置示例

```
<script type="text/javascript">  
  
function on_cc_player_init( vid, objectID){  
    var config = {};  
    config.player_plugins = {  
        Subtitle : {  
            url : "http://yourdomain/subtitle.srt"  
            , size : 20  
            , color : 0xCCCCCC  
            , surroundColor : 0x0000FF  
            , bottom : 0.15  
            , font : "微软雅黑"  
            , code : "gbk"  
        }  
    };  
  
    var player = getSWF( objectID );  
    player.setConfig( config );  
}  
  
</script>
```

4.2 弹幕插件

启用时在配置中添加插件名称 **BulletCurtain**，参数如下：

| 参数 | 说明 |
|----------|---------------------------|
| size | 默认字体大小，默认 30 |
| color | 默认字体颜色，默认 0xFFFFFFFF |
| font | 默认字体名称，默认 Times New Roman |
| duration | 所有弹幕在展示的时长，单位秒，默认 10 |
| alpha | 所有弹幕的不透明度，取值范围0-1，默认 1 |
| visible | 所有弹幕是否可见，1 为可见，0 不可见，默认 1 |

弹幕插件启用后还开放有如下接口：

| 接口 | 说明 |
|----------------------------------|---|
| setBulletCurtainAlpha(alpha) | 设置所有弹幕的不透明度 |
| setBulletCurtainVisible(visible) | 设置所有弹幕是否可见 |
| addBullet(data, menu) | 立刻显示一行弹幕 参数 data 是一个对象，于此接口有意义的 key 如下： id：唯一标识符 text：文字内容，此为必要数据 type：弹幕类型，0 为从右至左，1 为顶端，2 为底端，默认 0 size：字体大小 color：字体颜色 borderColor：边框颜色，默认 -1 不显示边框 参数 menu 为一个数组，用来构建弹幕的右键菜单，默认为空 数组元素需为一个对象 label：菜单项显示的文字 callback：菜单项点击后回调的JS函数名称，回调时参数为 data |
| removeBullet(id) | 移除一行弹幕 |

使用弹幕插件时，需了解此插件仅提供了把文字渲染至视频画面上(包括滚动和排列)的功能，播放器不管理各个弹幕与视频时间的对应关系，调用 addBullet 时在当前时刻显示一个弹幕并在 duration 秒之后自动移除，弹幕从画面中移除后(设置 alpha 和 visible 不会移除)，其相关数据都会丢弃，所以 seek 到一个之前播放过的时间点，需要重新调用 addBullet 以显示之前出现过的弹幕，另外只有对于正显示着的弹幕调用 removeBullet 才有意义。

附录 1. Flash和Javascript交互

播放器的HTML代码中

allowScriptAccess 是一个与Javascript交互所必需的属性，其值必须为 always；

object 标签中 id 属性 是IE内核浏览器中获取播放器句柄的关键，在非IE内核浏览器中则是 embed 标签的 name 属性；它们的值不能为空，并且不能和页面其他元素重名。

可使用下例中的 getSWF 函数来获取播放器句柄；另外播放器回调接口 on_cc_player_init 会传递参数 swfID，在这个回调接口中使用 getSWF(swfID) 也可获取到播放器句柄。

```
<object ... id="cc_A5FEB0EA05F2100B9C33DC5901307461">
  ...
  <param name="allowFullScreen" value="true" />
  <param name="allowScriptAccess" value="always" />
  <embed ... name="cc_A5FEB0EA05F2100B9C33DC5901307461" allowFullScreen="true"
allowScriptAccess="always" ... />
</object>

<script type="text/javascript">

  function on_cc_player_init( vid, objectID ){
    var player = getSWF( objectID );
    player.start();
  }

  function getSWF( swfID ) {
    if( window.document[ swfID ] ) {
      return window.document[ swfID ];
    } else if( navigator.appName.indexOf( "Microsoft" ) == -1 ) {
      if( document.embeds && document.embeds[ swfID ] ) {
        return document.embeds[ swfID ];
      }
    } else {
      return document.getElementById( swfID );
    }
  }
}

</script>
```

另外，对于使用SWFObject的用户，可参考以下代码把播放器嵌入您的网页。

```

// for 1.5
var swfobj = new SWFObject( [PlayerURL], [PlayerID], [PlayerWidth], [PlayerHeight], "10" );

//如果 [PlayerURL] 不是带参数的，就至少必须增加以下参数：
swfobj.addVariable( "siteid", [UID] );
swfobj.addVariable( "vid", [VID] );

swfobj.addParam( "allowScriptAccess", "always" );
swfobj.addParam( "allowFullScreen", "true" );

swfobj.write( [ContainerID] );

// for 2.x
var flashvars = {
  "siteid": [UID],
  "vid": [VID]
};
var params = {
  "allowScriptAccess": "always",
  "allowFullScreen": "true"
};
var attributes = {
  "name": [PlayerID],
  "id": [PlayerID]
};
swfobject.embedSWF( [PlayerURL], [ContainerID], [PlayerWidth], [PlayerHeight], "10.1.0",
[ExpressInstallSwfurl], flashvars, params, attributes );

```

[PlayerURL] 播放器地址；

[PlayerID] 指定页面上播放器的id，可用于播放器API交互；

[PlayerWidth] 播放器宽；

[PlayerHeight] 播放器高；

[UID] 用户ID；

[VID] 视频ID；

[ContainerID] 将添加播放器的容器的ID；

[ExpressInstallSwfurl] 指定 express install SWF 的 URL 并激活 Adobe express install，不指定时可用 false。

附录 2. 播放器错误码

| 代码 | 说明 |
|-----|--------------|
| 102 | 视频列表为空 |
| 103 | 无效的视频列表 |
| 108 | 列表模式视频信息加载失败 |
| 109 | 视频信息加载失败 |
| 110 | 视频审核中 |
| 111 | 用户流量用尽 |
| 112 | 用户账号被锁 |

| 代码 | 说明 |
|-----|-----------|
| 113 | 视频处理中 |
| 114 | 视频被屏蔽 |
| 115 | 视频信息错误 |
| 116 | 无视频文件 |
| 205 | 当前域名不允许播放 |
| 207 | 授权播放验证未通过 |
| 304 | 测速失败 |
| 305 | 切换清晰度失败 |
| 306 | 播放中播放失败 |