

河马云web接口文档

版本: 1.1.7

杭州龙境科技有限公司

目录

- LongeneClient
 - Function: createAppPlayer([mountPoint][, options])
 - Class: AppPlayer
 - Property: state
 - Property: muted
 - Property: volume
 - Property: orientation
 - Function: open([parameters])
 - Function: close()
 - Function: destroy()
 - Function: updateInterface()
 - Function: setMuted([value])
 - Function: setVolume([value])
 - Function: setOrientation([value])
 - Event: 'error'
 - Event: 'statechange'
 - Event: 'mutedchange'
 - Event: 'volumechange'
 - Event: 'orientationchange'
 - Class: LoggingControl
 - Function: getLogLevel()
 - Function: setLogLevel([value])
 - Function: addLogListener([callback])
 - Function: removeLogListener([callback])

LongeneClient

河马云的JS库，用于在Web端显示河马云手机

```
<div id="mountPoint"></div>
<script src="http://h5.longene.com.cn/static/js/longene-client-1.1.7.js">
</script>
<script>
  // 在挂载点为mountPoint的div下创建云手机
  var player = LongeneClient.createAppPlayer('mountPoint');
  // 启动云端应用
  player.open({ ip, domain, port, token });
</script>
```

Function: createAppPlayer([mountPoint][, options])

- **mountPoint** (*String or HTMLElement*) DOM挂载点(**required**)
- **options** (*Object*) 配置选项
- **Returns** 成功返回 **AppPlayer** , 失败返回 **null**

创建一个云手机的实例

```
// 通过div的id名创建云手机
var player = LongeneClient.createAppPlayer('mountPoint');
// 通过div的dom对象创建云手机
var player = LongeneClient.createAppPlayer(document.getElementById('mountPoint'));
if (!player) {
    console.log('AppPlayer创建失败! ');
}
```

可配置选项列表

选项名字	描述	默认值
muted	静音属性, 值为 boolean	false
volume	音量属性, 值为 (0 - 1) 的 float 值	1
keyboard	键盘开关, 值为 boolean	false
mediaType	媒体类型, 值为 [auto, video, image] 的 string 值	auto
orientation	定位方向, 值为 [landscape, portrait] 的 string 值	0

设置云手机的音量和静音属性

```
// 创建初始为静音的云手机
var player = LongeneClient.createAppPlayer('mountPoint', {
    muted: true
});
// 创建初始为50%音量的云手机
var player = LongeneClient.createAppPlayer('mountPoint', {
    volume: 0.5
});
```

设置云手机的支持键盘输入的开关

```
// 创建支持键盘输入的云手机
```

```
var player = LongeneClient.createAppPlayer('mountPoint', {
  keyboard: true
});
```

设置云手机的媒体类型

云手机的媒体类型有两种 `video` 和 `image`，`auto` 为自动匹配，在浏览器支持的情况下优先选择 `video` 类型。

`video` 类型的云手机通过视频标签 `video` 进行播放，图像清晰，画面流畅，效果较好。

`image` 类型的云手机通过图像标签 `canvas` 进行播放，图像较为模糊，且不支持声音的播放。

```
// 自动匹配类型
var player = LongeneClient.createAppPlayer('mountPoint', {
  mediaType: 'auto'
});
// 创建video类型的云手机，不成功返回null
var player = LongeneClient.createAppPlayer('mountPoint', {
  mediaType: 'video'
});
// 创建image类型的云手机，不成功返回null
var player = LongeneClient.createAppPlayer('mountPoint', {
  mediaType: 'image'
});
```

设置云手机的定位方向，`landscape` 表示横屏播放，`portrait` 表示竖屏播放。

```
// 设置初始为横屏的云手机
var player = LongeneClient.createAppPlayer('mountPoint', {
  orientation: 'landscape'
});
// 设置初始为竖屏的云手机
var player = LongeneClient.createAppPlayer('mountPoint', {
  orientation: 'portrait'
});
```

Class: AppPlayer

由 `createAppPlayer()` 返回的实例

Property: state

- (*String*) 获取云手机的播放状态，只读属性

状态类型描述

名字	描述
new	空闲状态
connecting	正在连接服务
connected	连接成功
playing	开始播放
closed	关闭状态

```
var state = player.state;
if (state === 'connecting') {
    console.log('正在连接中, 请稍后...');
}
if (state === 'playing') {
    console.log('应用已启动');
}
if (state === 'closed') {
    console.log('应用已关闭');
}
```

Property: muted

- *(Boolean)* 获取或设置云手机的静音状态，读写属性

```
var muted = player.muted;
if (muted) {
    console.log('当前静音');
    // 关闭静音
    player.muted = false;
}
```

Property: volume

- *(Float)* 获取或设置云手机的音量大小，读写属性

```
var volume = player.volume;
console.log('当前音量为: ' + volume);
// 设置音量到80%
player.volume = 0.8;
```

Property: orientation

- *(Number)* 获取或设置云手机的定位方向，读写属性

```

var orientation = player.orientation;
if (orientation === 'landscape') {
    console.log('当前处于横屏');
    // 改变定位方向为竖屏
    player.orientation = 'portrait';
}

```

Function: open([parameters])

- **parameters** (*Object*) 启动参数
 - **ip** (*String*) 由河马云返回的云服务IP地址或域名(**required**)
 - **domain** (*String*) 由河马云返回的云服务域名(**required**)
 - **port** (*Number*) 由河马云返回的连接端口(**required**)
 - **token** (*String*) 由河马云返回的验证字段(**required**)

启动云手机, **parameters** 为云手机的调用参数

```

var player = LongeneClient.createAppPlayer('mountPoint');
player.open({
    ip: 'lcloud.longene.com.cn',
    domain: 'lcloud.longene.com.cn',
    port: 20000,
    token: '123456'
});

```

Function: close()

关闭云手机的显示

```

// 连接云手机的界面
player.open(parameters);
// 关闭云手机的界面
player.close();
// 再次连接
player.open(parameters);

```

Function: destroy()

销毁创建的云手机实例

```

// 创建云手机实例
var player = LongeneClient.createAppPlayer('mountPoint');
// 连接云手机
player.open(parameters);
// 销毁云手机实例

```

```
player.destroy();  
player = null;
```

Function: updateInterface()

在挂载点 `mountPoint` 的宽高发生变化，更新云手机的用户界面，适应新的宽高

```
<div id="mountPoint" style="height: 200px;width: 200px;"></div>  
<script>  
  var player = LongeneClient.createAppPlayer('mountPoint');  
  // 改变挂载点的宽高  
  document.getElementById("mountPoint").style.height = '800px';  
  document.getElementById("mountPoint").style.widht = '800px';  
  // 更新player的宽高  
  player.updateInterface();  
  // 在回调函数里更新player的宽高  
  window.onresize = function() {  
    player.updateInterface();  
  }  
</script>
```

Function: setMuted([value])

- `value` (Boolean) (required)

设置云手机的静音状态

```
player.setMuted(true);  
player.on('mutedchange', function() {  
  if (player.muted) {  
    console.log('静音开启');  
  }  
});
```

Function: setVolume([value])

- `value` (Float) (required)

设置云手机的音量大小

```
player.setVolume(0.55);  
player.on('volumechange', function() {  
  var volume = Math.round(player.volume * 100)  
  console.log('当前音量为: ' + volume + '%');  
});
```

Function: setOrientation([value])

- **value** (Number) (**required**)

设置云手机的定位方向

```
<div id="mountPoint"></div>
<script>
  var player = LongeneClient.createAppPlayer('mountPoint', {
    orientation: 'landscape'
  });
  // 改变player的定位方向
  player.setOrientation('portrait');
  // 在回调函数里更新player的定位方向
  window.addEventListener("orientationchange", function() {
    if (window.orientation === 0 || window.orientation === 180) {
      // 竖屏时
      player.setOrientation('portrait');
    } else {
      // 横屏时
      player.setOrientation('landscape');
    }
  });
</script>
```

Event: 'error'

- **error** (Object)
 - **code** (Number) 错误代号
 - **message** (String) 错误信息

云手机的错误事件

code	message
1000	连接超时或服务出错
1002	ICE协商出错
1007	远端连接断开
1008	SDP协商出错
1015	参数不正确
1026	服务无响应
1027	验证失败
1029	发生未知错误


```
player.on('error', function(error) {  
    console.log('发生错误: ' + error.code + ' ' + error.message);  
});
```

Event: 'statechange'

云手机状态属性 `state` 发生改变时的事件

```
player.on('statechange', function() {  
    console.log('状态发生改变: ' + player.state);  
});
```

Event: 'mutedchange'

云手机静音属性 `muted` 发生改变时的事件

```
player.on('mutedchange', function() {  
    if (player.muted) {  
        console.log('静音开启');  
    } else {  
        console.log('静音关闭');  
    }  
});
```

Event: 'volumechange'

云手机音量属性 `volume` 发生改变时的事件

```
player.on('volumechange', function() {  
    console.log('音量发生改变: ' + player.volume);  
});
```

Event: 'orientationchange'

云手机定位属性 `orientation` 发生改变时的事件

```
player.on('orientationchange', function() {  
    if (player.orientation === 'landscape') {  
        console.log('横屏');  
    } else if (player.orientation === 'portrait') {
```

```
        console.log('竖屏');
    }
});
```

Class: LoggingControl

全局的日志控制对象

共有5个等级 `[verbose, debug, info, warn, error]` 依次递升

如果日志等级为 `info`，则类型为 `info warn error` 的日志都会输出到控制台

默认全局日志等级为 `warn`

Function: LoggingControl.getLogLevel()

- **Returns** `[verbose, debug, info, warn, error]`

获取当前的日志等级

```
var level = LongeneClient.LoggingControl.getLogLevel();
console.log('当前日志等级为: ' + level);
```

Function: LoggingControl.setLogLevel([value])

- **value** (*String*) `[verbose, debug, info, warn, error]`

设置当前的日志等级

```
// 设置日志等级为info
LongeneClient.LoggingControl.setLogLevel('info');
```

Function: LoggingControl.addLogListener([callback])

- **callback** (*Function*) 回调函数
 - **type** (*String*) 日志的类型 `[verbose, debug, info, warn, error]`
 - **message** (*String*) 日志的信息

日志默认是输入到控制台 `console`，在打不开控制台的时候调用该接口，添加回调函数监听日志输出

```
LongeneClient.LoggingControl.addLogListener(function(type, message) {
    if (type === 'error') {
        console.log('错误: ' + message);
    }
})
```

```
if (type === 'warn') {  
  console.log('警告: ' + message);  
}  
if (type === 'info') {  
  console.log('信息: ' + message);  
}  
});
```

Function: LoggingControl.removeLogListener([callback])

- `callback` (*Function*) 回调函数

将监听日志的回调函数移除

```
`js  
function logListener(type, message) {  
  console.log('日志: ' + message);  
}  
// 增加监听器  
LongeneClient.LoggingControl.addLogListener(logListener);;  
// 移除监听器  
LongeneClient.LoggingControl.removeLogListener(logListener);;
```