VMware HTML Console SDK
Guía de programación

Para vSphere 5.5 y posteriores (disponibilidad general)
y vCloud Director (vista previa técnica)

Febrero de 2016

VMware HTML Console SDK
Guía de programación

Índice

HTML Console SDK (WMKS) Descripción general

HTML Console SDK es una biblioteca JavaScript que se implementa mediante webmks (WMKS), y que permite el control y procesamiento de ratones, teclados y entradas táctiles, así como actualizaciones de pantalla y modificaciones de cursor en una sesión de escritorio desde un navegador. Las aplicaciones que se ejecutan en el navegador pueden usar JavaScript para acceder a las funciones de consola de la máquina virtual mediante la API de HTML Console SDK.

La API de HTML Console SDK proporciona varios métodos de conexión y comunicación con una máquina virtual. También activa eventos para notificar a los usuarios los cambios de estado de la máquina virtual. Puede utilizar estos métodos y devoluciones de llamada para proporcionar a los usuarios finales la capacidad de gestionar de forma remota una máquina virtual desde cualquier sistema con el navegador web y sistema operativo apropiados.

Navegadores compatibles (incluidos en iOS y Android)

- Internet Explorer 10+
- Firefox 24+
- Chrome 30+
- Safari 6.1+

Pasos para usar la API de HTML Console SDK

Para usar la API de HTML Console SDK, una aplicación basada en web debe normalmente hacer lo siguiente:

1. Cargar los archivos de la librería JavaScript de HTML Console SDK.
2. Crear el objeto núcleo de WMKS usando el método predeterminado createWMKS() con un ID de elemento div proporcionado en el DOM del documento y las opciones de creación.
3. Registrar las devoluciones de llamada JavaScript para responder a los eventos activados por WMKS.
4. Usar el método connect() de HTML Console SDK para conectarse a una máquina virtual de destino.
5. Usar los métodos de HTML Console SDK para interactuar con la máquina virtual conectada.
6. Usar el método disconnect() de HTML Console SDK para desconectar la conexión (si está activa) y eliminar el widget del elemento asociado.

Colocación de la consola en una página web

El HTML Console SDK está disponible en la sección de descargas de www.vmware.com/es/: Controladores y herramientas para vSphere, vCloud Director, vRealize Automation, y vCloud Air.

En cada compilación se proporcionan un wmks.min.js minificado y un wmks.js sin minificar. El HTML Console SDK emplea el widget jQuery para mostrar la consola VM, por lo que la biblioteca relacionada con jQuery necesita incluirse primero. Aquí se muestra un ejemplo:

1. Incluya los componentes jQuery y jQuery UI Javascript en etiquetas de script y el archivo jQuery css correspondiente en la etiqueta de hoja de estilos de vínculos.
2. Incluya wmks.js (o wmks.min.js) en una etiqueta de script y wmks-all.css la etiqueta de hoja de estilos de vínculos.
3. Proporcione un div con un ID apropiado.
4. Cree el objeto núcleo de WMKS con el método predeterminado WMKS.createWMKS().
5. Vincule las devoluciones de llamada de los eventos de WMKS.
6. Conecte con las URL de destino apropiadas.

```
 <link rel="stylesheet" type="text/css" href="wmks-all.css" />

<script type="text/javascript" src="jquery-1.8.3.min.js"></script>
 <script type="text/javascript" src="jquery-ui.1.8.16.min.js"></script>
 <script type="text/javascript" src="wmks.js" type="text/javascript"></script>
<script>
        var wmks = WMKS.createWMKS("wmksContainer",{})
                    .register(WMKS.CONST.Events.CONNECTION_STATE_CHANGE,
                    function(event,data){
                        if(data.state == cons.ConnectionState.CONNECTED)
                        {
                         console.log("connection state change : connected");
                        }
                    });
        wmks.connect("ws://127.0.0.1:8080");
</script >
<div  id="wmksContainer" style="position:absolute;width:100%;height:100%"></div>
```

Conexión con una máquina virtual remota

Modos de conectar con una máquina virtual remota:

WMKS también puede utilizarse como componente para ofrecer una conexión de consola basada totalmente en web con máquinas virtuales controladas por vCenter Server usando un proxy intermediario que gestione los requisitos de autenticación.

Conéctese a la máquina virtual mediante el siguiente método:

```
wmks.connect(url);
```

La URL debería tener el siguiente esquema:

```
<ws | wss> :// <host:port>/ <path> /? <authentication info>
```

Desconexión y destrucción

Realice esta llamada cuando haya terminado con el componente WMKS. Al destruir el componente WMKS también desconectará (si está activo) y eliminará el widget del elemento asociado.

El método predeterminado de WMKS

createWMKS()

Este debería ser el primer método cuando utiliza el HTML Console SDK. Al usar este método se genera el widget, que puede mostrar la pantalla remota y, a continuación, devolver el objeto núcleo de WMKS, que puede utilizar todas las API de HTML Console SDK para conectarse a una máquina virtual y realizar operaciones.

Método	createWMKS(id, options)
Parámetro 1	**id**(string): el ID del elemento div; este elemento sería el contenedor del lienzo empleado para mostrar la pantalla remota.
Parámetro 2	**options**:(json object): las opciones de creación afectarían al comportamiento de WMKS. En blanco significa “usar todas las opciones predeterminadas”.
Valor devuelto	El objeto núcleo de WMKS, que podría usar todas las API de WMKS para conectarse a una máquina virtual y realizar operaciones.
Llamada de ejemplo	var wmks =WMKS.createWMKS(“container”,{});

Opciones de creación

Como webMKS, incrustado en vSphere Web Client, WMKS también ofrece varias opciones para controlar su comportamiento. Aquí se muestra la comparación entre SDK y webMKS incrustado en vSphere Web Client:

rescale

Booleano: el valor predeterminado es true, e indica si se cambia la escala de la pantalla remota para que se ajuste al tamaño asignado al contenedor.

position

Enumerado: podría ser cualquier valor de WMKS.CONST.Position, el valor predeterminado es WMKS.CONST.Position.CENTER. Indica la posición (central o superior izquierda) en la que la pantalla remota debería situarse en el contenedor.

changeResolution

Booleano: el valor predeterminado es true. Cuando el valor para la opción changeResolution es true, WMKS envía la solicitud de cambio de resolución a la máquina virtual conectada; la resolución solicitada (anchura y altura) coincide con el tamaño asignado al contenedor.

Estas tres opciones siguen un orden específico en función de su prioridad.

1. Compruebe **changeResolution**; si su valor es true, webMKS envía la solicitud de cambio de resolución a la máquina virtual conectada.
2. Si la solicitud falla, pueden utilizarse las operaciones **rescale** y **position** para realizar los cambios correspondientes.

audioEncodeType

Enumerado: podría ser cualquier valor de WMKS.CONST.AudioEncodeType. Indica qué tipo de método de codificación de audio se está empleando: vorbis, opus o aac.

useNativePixels

Booleano: el valor predeterminado es false. Permite el uso de tamaños de píxeles nativos en el dispositivo. En iPhone 4+ o iPad 3+, su activación habilitará el "modo Retina", que proporciona más espacio en pantalla para el invitado y reduce el tamaño del contenido.

useUnicodeKeyboardInput

Booleano: el valor predeterminado es false. Para todas las entradas del usuario, WMKS podría enviar dos tipos de mensaje al servidor: código de tecla o Unicode. Aquí *true* significa que el cliente debe utilizar Unicode si es posible.

useVNCHandshake

Booleano: el valor predeterminado es true. Habilita un protocolo de enlace VNC estándar. Debería utilizarse cuando el terminal emplea autenticación VNC estándar. Establézcalo en false si se conecta a un proxy que autentique a través de authd y no realice un protocolo de enlace VNC.

sendProperMouseWheelDeltas

Booleano: el valor predeterminado es false. Las versiones anteriores de la biblioteca normalizan las diferencias de eventos de rueda del ratón en uno de los siguientes valores: [-1, 0, 1]. Cuando se establece en *true*, se envían las diferencias reales del navegador al servidor.

reverseScrollY

Booleano: el valor predeterminado es false. Si este indicador se establece en *true*, se envía el valor de rueda del ratón opuesto a la máquina virtual conectada.

retryConnectionInterval

Entero: el valor predeterminado es -1. El intervalo (en milisegundos) para volver a intentar conectarse cuando falla el primer intento de establecer una conexión entre el cliente web y el servidor. Un valor inferior a 0 significa “no se volverá a intentar”.

ignoredRawKeyCodes

matriz: el valor predeterminado es en blanco. Se ignoran todos estos códigos de clave y no se envían al servidor.

fixANSIEquivalentKeys

Booleano: el valor predeterminado es false. Permite corregir cualquier código de clave de diseño que no sea ANSI para que coincida con el código de clave de diseño ANSI equivalente. Intenta corregir cualquier pulsación de tecla cuando el diseño de teclado internacional del cliente cuenta con una tecla que también está presente en el teclado ANSI pero en una ubicación distinta o que no coincide con el estado de activación de Mayús en un teclado ANSI.

keyboardLayoutId

Cadena: el valor predeterminado es "en-US" (inglés de EE. UU.). Esto proporciona configuraciones de teclado distintas según el idioma del invitado. Los usuarios necesitan determinar el tipo de diseño del teclado y mantener el mismo diseño para el teclado del escritorio remoto y el del escritorio local. En esta versión de HTML Console SDK, VMware no proporciona compatibilidad con dispositivos móviles. Se envían dos tipos de código al sistema operativo del invitado. Para la asignación internacional ofrecemos compatibilidad con vScancode.

Windows es compatible con Internet Explorer/Firefox/Chrome y Mac, con Chrome/Safari. Añada una lista de selección en html o utilice algún otro medio para que los usuarios seleccionen el idioma empleado. En el archivo js, utilice la API setOption como se indica a continuación:

```
<select id="selectLanguage">
        <option value="en-US">English</option>
        <option value="ja-JP_106/109">Japanese</option>
        <option value="de-DE">German</option>
        <option value="it-IT">Italian</option>
        <option value="es-ES">Spanish</option>
        <option value="pt-PT">Portuguese</option>
</select>
```

```
$('#selectLanguage').change(function(){
   if(!wmks) return;
   var keyboardLayoutId = $(this).find(":selected").val();
   wmks.setOption('keyboardLayoutId',keyboardLayoutId);
 });
```

VCDProxyHandshakeVmxPath

Cadena: el valor predeterminado es null. Pase al protocolo VNC al conectarse. vncProtocol responderá a una solicitud de conexión de una ruta VMX con la ruta VCDProxyHandshakeVmxPath proporcionada al conectarse a **vCloud Director**.

enableUint8Utf8

Booleano: el valor predeterminado es false. Si el proyecto no cuenta con compatibilidad de trabajo con el protocolo binario, será necesaria la opción enableUint8Utf8 para habilitar el protocolo uint8utf8 antiguo.

Control de eventos de WMKS

Lista de eventos de WMKS

WebMKS genera eventos cuando cambia el estado de la máquina virtual conectada en ese momento, o en respuesta a los mensajes de dicha máquina virtual. Todos los eventos de WMKS se enumeran en WMKS.CONST.Events.

Todos los controladores de eventos tienen dos parámetros: **event** y **data**. **Event** es un evento jQuery y todos los parámetros especiales de los eventos se almacenan en **data**.

connectionstatechange

El evento connectionstatechange se genera en respuesta a un cambio en el estado de la conexión, como de "desconectado" a "conectando" o de "conectando" a "conectado".

Parámetros de **data**:

- state: puede ser cualquier valor de WMKS.CONST. ConnectionState puede ser:
  1. "conectando"
  2. "conectado"
  3. "desconectado"
- Si el estado es WMKS.CONST. ConnectionState. CONECTANDO, hay dos o más parámetros ***vvc*** y ***vvcSession*** en data.
- Si el estado es WMKS.CONST. ConnectionState. DESCONECTADO, hay dos o más parámetros ***reason*** y ***code*** en data.

screensizechange

El evento screensizechange se genera en respuesta a cambios en el tamaño de la pantalla de la máquina virtual conectada en ese momento.
Parámetros de data:

- width: la anchura (en píxeles) de la máquina virtual conectada en ese momento.
- height: la altura (en píxeles) de la máquina virtual conectada en ese momento.

fullscreenchange

El evento fullscreenChange se genera cuando la consola WMKS sale del modo de pantalla completa o entra en él.
Parámetros de data:

- isFullScreen: Booleano, true significa entrar en pantalla completa y false significa salir de pantalla completa.

error

El evento error se genera cuando se produce un error como un fallo de autenticación, un error de WebSocket o un error de protocolo.

Parámetros de data:

- errorType: puede ser cualquier valor de WMKS.CONST.Events.ERROR:
  1. AUTHENTICATION_FAILED: "authenticationfailed",
  2. WEBSOCKET_ERROR: "websocketerror",
  3. PROTOCOL_ERROR: "protocolerror"

Keyboardledschanged

El evento keyboardledschanged se genera cuando cambia el estado de bloqueo LED del teclado de la máquina virtual remota.

Parámetros de data:

- Data es un entero: 1 significa Bloq Despl, 2 significa Bloq Num, 4 significa Bloq Mayús.

heartbeat

El evento heartbeat se genera cuando se recibe un mensaje de que se produce un latido en el servidor.

- Data es un entero que indica el intervalo del latido.

audio

El evento audio se genera cuando se recibe un mensaje de audio del servidor.
Parámetros de data:

- sampleRate
- numChannels
- containerSize
- sampleSize
- length
- audioTimestampLo
- audioTimestampHi
- frameTimestampLo
- frameTimestampHi
- flags
- data

copy

El evento copy se genera cuando el servidor envía un evento de corte de texto.

- Data muestra la cadena copiada.

toggle

El evento toggle se genera al mostrar u ocultar el teclado, el teclado ampliado o el panel táctil.
Parámetros de data:

- type: puede ser “KEYBOARD”, "EXTENDED_KEYPAD", "TRACKPAD"
- visibility: Booleano, true significa “mostrar” y false significa “ocultar”.

Configuración de controladores para los eventos de WMKS

En el HTML Console SDK, proporcione el método register() y unregister para añadir y eliminar controladores para los eventos de WMKS.

register()

Este método se emplea para registrar controladores de eventos de WMKS.

Método	register(eventName, eventHandler)
Parámetro 1	eventName(constante, cualquier valor de WebMKS.Events)
Parámetro 2	eventHandler(función de devolución de llamada JavaScript)
Valor devuelto	Ninguno
Llamada de ejemplo	var connectionStateHandler = function(event, data){}; wmks.register(WebMKS.Events.CONNECTION_STATE_CHANGE, connectionStateHandler);

unregister()

Este método se emplea para cancelar el registro de los controladores de eventos de WMKS.
Si este método no tiene parámetros, elimina todos los controladores de eventos.
Si solo hay un parámetro, elimina todos los controladores de eventos de ese eventName en concreto.

Método	unregister(eventName, eventHandler)
Parámetro 1	eventName (constante, cualquier valor de WebMKS.Events)
Parámetro 2	eventHandler(función de devolución de llamada JavaScript)
Valor devuelto	Ninguno
Llamada de ejemplo	wmks.unregister(WebMKS.Events.CONNECTION_STATE_CHANGE, connectionStateHandler);

Métodos del objeto de HTML Console SDK

Tras ejecutar el método createWMKS(), se obtiene un objeto núcleo de WMKS que puede utilizar las API de WMKS para conectarse a una máquina virtual y realizar operaciones. En este ejemplo, nombraremos el objeto de WMKS "wmks".

API generales

Los métodos de API de uso general proporcionan información acerca de WMKS. Se pueden realizar llamadas a estos métodos en cualquier momento, incluso antes de conectarse a la máquina virtual de destino.

getVersion()

Recupera el número de versión actual de HTML Console SDK.

Método	getVersion()
Parámetros	Ninguno
Valor devuelto	Cadena. Contiene el número de versión completo de HTML Console SDK.
Llamada de ejemplo	var version = wmks.getVersion();

getConnectionState()

Recupera el estado de conexión en ese momento.

Método	getConnectionState()
Parámetros	Ninguno
Valor devuelto	Constante. Cualquier valor en WMKS. CONST.ConnectionState.
Llamada de ejemplo	var state = wmks.getConnectionState();

API relacionadas con el ciclo de vida

connect()

Conecta WMKS a una máquina virtual remota mediante la URL de WebSocket y establece la interfaz de usuario.

Método	connect()
Parámetro	URL de WebSocket, el tipo formato es una cadena: <ws | wss> :// <host:port>/ <path> /? <authentication info>
Valor devuelto	Ninguno
Llamada de ejemplo	wmks.connect("ws://localhost:8080");

disconnect()

Se desconecta de la máquina virtual remota y anula la interfaz de usuario.

Método	disconnect()
Parámetro	Ninguno
Valor devuelto	Ninguno
Llamada de ejemplo	wmks.disconnect();

destroy()

Finaliza la conexión (si está activa) con la máquina virtual y elimina el widget del elemento asociado. Los consumidores deben realizar esta llamada antes de eliminar el elemento del DOM.

Método	destroy()
Parámetro	Ninguno
Valor devuelto	Ninguno
Llamada de ejemplo	wmks.destroy();

API relacionadas con la pantalla

setRemoteScreenSize()

Envía una solicitud para establecer la resolución de la pantalla de la máquina virtual conectada en ese momento. Si los parámetros de anchura y altura proporcionados son superiores a los del tamaño asignado al widget de WMKS, el tamaño se normaliza.

Nota: este método solo funciona correctamente cuando la opción changeResolution se establece en *true*.

Método	setRemoteScreenSize(anchura, altura)
Parámetro 1	width(int) representa la anchura deseada para la máquina virtual conectada, en píxeles
Parámetro 2	height(int) representa la altura deseada para la máquina virtual conectada, en píxeles
Valor devuelto	Ninguno
Llamada de ejemplo	wmks.setRemoteScreeSize(800,600);

getRemoteScreenSize()

Recupera la anchura y la altura en píxeles de la pantalla de la máquina virtual conectada en ese momento.

Método	getRemoteScreenSize()
Parámetro	Ninguno
Valor devuelto	Objeto en formato de {width:, height:}
Llamada de ejemplo	var size = wmks.getRemoteScreenSize();

updateScreen()

Cambia la resolución o la escala de la pantalla remota para que se ajuste al tamaño asignado en ese momento.

El comportamiento de updateScreen depende de la opción changeResolution, rescale, y position:

1) Cuando el valor para la opción changeResolution es true, se envía la solicitud de cambio de resolución a la máquina virtual conectada; la resolución solicitada (anchura y altura) coincide con el tamaño asignado al contenedor.

2) Compruebe la opción rescale: si el valor es true, cambie la escala de la pantalla remota para que se ajuste al tamaño asignado al contenedor.

3) Compruebe la opción position: si el tamaño de la pantalla remota no coincide con el tamaño asignado al contenedor, la pantalla remota se colocará en la parte central o superior izquierda del contenedor en función de su valor.

Método	updateScreen()
Parámetro	Ninguno
Valor devuelto	Ninguno
Llamada de ejemplo	wvmrc.updateScreen();

API relacionadas con la pantalla completa

canFullScreen()

Indica si la función de pantalla completa está habilitada en el navegador. El modo de pantalla completa está deshabilitado en Safari y no es compatible con las entradas del teclado en pantalla completa por motivos de seguridad.

Método	canFullScreen()
Parámetro	Ninguno
Valor devuelto	Booleano. True significa que se puede; false significa que no.
Llamada de ejemplo	wmks.canFullScreen();

isFullScreen()

Informa de si el navegador está en modo de pantalla completa.

Método	isFullScreen()
Parámetro	Ninguno
Valor devuelto	Booleano. True significa en modo de pantalla completa, false significa que no lo está.
Llamada de ejemplo	wmks.isFullScreen();

enterFullScreen()

Fuerza la entrada del navegador en modo de pantalla completa si es compatible. En el modo de pantalla completa, solo se muestra la pantalla remota.

Método	enterFullScreen()
Parámetro	Ninguno
Valor devuelto	Ninguno
Llamada de ejemplo	wmks.enterFullScreen();

exitFullScreen()

Fuerza la salida del modo de pantalla completa del navegador.

Método	exitFullScreen()
Parámetro	Ninguno
Valor devuelto	Ninguno
Llamada de ejemplo	wmks.exitFullScreen();

API relacionadas con la entrada

sendInputString()

Envía al servidor una cadena como entrada de teclado.

Método	sendInputString(string)
Parámetros	Cadena
Valor devuelto	Ninguno.
Llamada de ejemplo	wmks.sendInputString(“test”);

sendKeyCodes()

Envía una serie de códigos de clave a la máquina virtual. Recupera una matriz de códigos de clave especiales y envía eventos KeyDown a cada uno en el orden enumerado. A continuación envía eventos KeyUp a cada uno en el orden inverso. Esto puede utilizarse para enviar combinaciones de teclas como Ctrl-Alt-Supr.

Método	sendKeyCodes ()
Parámetros	Matriz. Cada elemento de la matriz puede ser un código de clave o un código Unicode. El código clave es para claves no imprimibles, Unicode es para caracteres imprimibles. Si utiliza Unicode, hágalo negativo, como en -118: “v”
Valor devuelto	Ninguno.
Llamada de ejemplo	wmks.sendKeyCodes([17,18,46]) ; // Ctrl + Alt + Suprimir

sendCAD()

Envía una secuencia de teclas Ctrl-Alt-Supr a la máquina virtual conectada en ese momento.

Método	sendCAD()
Parámetros	Ninguno.
Valor devuelto	Ninguno.
Llamada de ejemplo	wmks.sendCAD();

API relacionadas con la capacidad móvil

enableInputDevice()

Habilita el dispositivo de entrada (teclado, teclado ampliado, panel táctil) del dispositivo móvil y lo inicia para el uso.

Método	enableInputDevice (deviceType)
Parámetros	deviceType:(constante) cualquier valor en WMKS.CONST.InputDeviceType.
Valor devuelto	Ninguno.
Llamada de ejemplo	wmks.enableInputDevice(WMKS. CONST .InputDeviceType. KEYBOARD) ;

disableInputDevice()

Deshabilita el dispositivo de entrada (teclado, teclado ampliado, panel táctil) del dispositivo móvil y lo destruye.

Método	disableInputDevice (deviceType)
Parámetros	deviceType:(constante) cualquier valor en WMKS.CONST.InputDeviceType.
Valor devuelto	Ninguno.
Llamada de ejemplo	wmks.disableInputDevice(WMKS. CONST .InputDeviceType. KEYBOARD) ;

showKeyboard()

Muestra el teclado de un dispositivo móvil.

Método	showKeyboard()
Parámetros	Ninguno
Valor devuelto	Ninguno.
Llamada de ejemplo	wmks.showKeyboard();

hideKeyboard()

Oculta el teclado de un dispositivo móvil.

Método	hideKeyboard()
Parámetros	Ninguno.
Valor devuelto	Ninguno.
Llamada de ejemplo	wmks.hideKeyboard();

toggleExtendedKeypad()

Muestra/oculta el teclado ampliado del dispositivo móvil en función de la visibilidad en ese momento.

Método	toggleExtendedKeypad()
Parámetros	Una asignación, puede incluir minToggleTime(ms) como {minToggleTime: 50} si el usuario intenta realizar la llamada a este método de alternancia con demasiada frecuencia, en intervalos inferiores a lo establecido en minToggleTime, no se ejecuta.
Valor devuelto	Ninguno.
Llamada de ejemplo	wmks.toggleExtendedKeypad();

toggleTrackpad()

Muestra/oculta el panel táctil del dispositivo móvil en función de la visibilidad en ese momento.

Método	toggleTrackpad()
Parámetros	Una asignación, puede incluir minToggleTime(ms) como {minToggleTime: 50} si el usuario intenta realizar la llamada a este método de alternancia con demasiada frecuencia, en intervalos inferiores a lo establecido en minToggleTime, no se ejecuta.
Valor devuelto	Ninguno.
Llamada de ejemplo	wmks.toggleTrackpad();

toggleRelativePad ()

Muestra/oculta el panel táctil de ratón relativo en la pantalla. Envía el evento de ratón relativo en lugar del evento de ratón absoluto tras abrir el panel relativo. Puede utilizarse cuando no está instalado VMware Tools en el sistema operativo del invitado remoto. Añada un botón toggleRelativeMouse en la interfaz de usuario y vincúlelo a la API toggleRelativePad(). Tras hacer clic en él, el panel táctil relativo se mostrará y enviará un evento de ratón relativo al invitado remoto. Esto resulta útil cuando el sistema operativo del invitado no instala VMware Tools y no puede aceptar un evento de ratón absoluto. Dado que si no hay cursor el servidor devuelve la imagen, necesitamos mostrar una imagen de cursor local. Por tanto, es necesario añadir wmks-all.css en las carpetas css e img del producto. Todas están contenidas en la carpeta wmkssdk.

Método	toggleRelativePad ()
Parámetros	Una asignación, puede incluir minToggleTime(ms) como {minToggleTime: 50} si el usuario intenta realizar la llamada a este método de alternancia con demasiada frecuencia, en intervalos inferiores a lo establecido en minToggleTime, no se ejecuta.
Valor devuelto	Ninguno.
Llamada de ejemplo	wmks. toggleRelativePad ();

API relacionadas con opciones

setOption()

Cambia el valor de la opción. Este método solo se puede aplicar a las siguientes opciones:

- rescale
- position
- changeResolution
- useNativePixels
- reverseScrollY
- fixANSIEquivalentKeys
- sendProperMouseWheelDeltas
- keyboardLayoutId

Método	setOption(optionName, optionValue)
Parámetro 1	optionName (cadena del nombre de la opción)
Parámetro 2	optionValue
Valor devuelto	Ninguno
Llamada de ejemplo	wmks.setOption(“changeResolution”,false);

Apéndice

Constantes utilizadas en WMKS

- **Position:** WMKS muestra la pantalla remota de la máquina virtual en la misma proporción. Por tanto, tras cambiar la escala, el tamaño de la pantalla remota podría seguir siendo distinto al tamaño del contenedor. Aquí, ***Position*** proporciona dos posibles opciones de colocación de la pantalla en el contenedor.

- **ConnectionState**: hay 3 estados de conexión posibles cuando se intenta conectar con la máquina virtual remota.

- **Events**: todos los nombres de eventos que WMKS puede activar se enumeran aquí.

- **ErrorType**: tipos de error posibles en el ciclo de vida de WMKS.

- **InputDeviceType**: HTML Console SDK es compatible con la visualización de consolas de máquinas virtuales en dispositivos móviles; este campo enumera los dispositivos de entrada posibles.

(Consulte en la página siguiente la lista de constantes)

```
Position: {
   CENTER:   0,
   LEFT_TOP: 1
},

ConnectionState: {
   CONNECTING: "connecting",
   CONNECTED: "connected",
   DISCONNECTED: "disconnected"
},

Events: {
   CONNECTION_STATE_CHANGE: "connectionstatechange",
   REMOTE_SCREEN_SIZE_CHANGE: "screensizechange",
   FULL_SCREEN_CHANGE: "fullscreenchange",
   ERROR: "error",
   KEYBOARD_LEDS_CHANGE: "keyboardledschanged",
   HEARTBEAT: "heartbeat",
   AUDIO: "audio",
   COPY: "copy",
   TOGGLE: "toggle"
},

ErrorType: {
   AUTHENTICATION_FAILED: "authenticationfailed",
   WEBSOCKET_ERROR: "websocketerror",
   PROTOCOL_ERROR: "protocolerror"
},

AudioEncodeType: {
   VORBIS: "vorbis",
   OPUS: "opus",
   AAC: "aac"
},

InputDeviceType: {
   KEYBOARD: 0,
   EXTENDED_KEYBOARD: 1,
   TRACKPAD: 2
}
```