

# MULTISCREEN.DEVICE

This class provides methods of device settings. Like getting current device instance, creating a channel, initialization, connection for using Nservice.

Add the following line for multiscreen.Device class into a html file your own :

```
<script type="text/javascript" src="$MANAGER_WIDGET/Common/webapi/1.0/webapis.js"></script>
```

You can declare multiscreen.Device class like this :

```
ex) var multiscreenDevice = webapis.multiscreen.Device;
```

## Contents

### Methods

[getCurrent](#)

[createChannel](#)

[getChannel](#)

[showPinCode](#)

[openChannel](#)

[ready](#)

## Methods

### getCurrent

Description

Retrieves the current devices instance

Parameters

- callback
  - Function
  - The callback handler

[callback parameter]

- \* error
  - An error instance if one occurred during the async call
- \* device
  - The device instance

Return

■Void

Emulator Support

Y

SDK Constraint

.

Example

```
webapis.multiscreen.Device.getCurrent(function (error, device) {  
  // device is the local device instance  
});
```

## createChannel

### Description

Creates a channel instance

### Parameters

- id
    - String
    - The id of the application
  - callback
    - Function
    - The callback handler
- [callback parameter]
- \* error
    - An error instance if one occurred during the async call
  - \* channel
    - The channel instance

### Return

■Void

### Emulator Support

Y

### SDK Constraint

.

### Example

```
var device = webapis.multiscreen.Device;  
device.createChannel(id, function (error, channel) {  
  
});
```

## getChannel

### Description

get instance of channel by id

### Parameters

- id
    - String
    - The id of the application
  - callback
    - Function
    - The callback handler
- [callback parameter]
- \* error
    - An error instance if one occurred during the async call
  - \* channel
    - The channel instance

### Return

■Void

### Emulator Support

Y

### SDK Constraint

.

### Example

```
var device = webapis.multiscreen.Device;  
device.getChannel(id, function (error, channel) {  
  
});
```

## showPinCode

### Description

Displays the current pin code for the device using the devices system UI	
Parameters	<ul style="list-style-type: none"> <li>■ callback <ul style="list-style-type: none"> <li>- Function</li> <li>- The callback handler</li> </ul> </li> </ul> <p>[callback parameter]</p> <ul style="list-style-type: none"> <li>* error <ul style="list-style-type: none"> <li>- An error instance if one occurred during the async call</li> </ul> </li> </ul>
Return	■Void
Emulator Support	Y
SDK Constraint	.
Example	
<pre>var device = webapis.multiscreen.Device; device.showPincode(function (error) {  });</pre>	

## openChannel

### Description

Creates, configures, and connects to a channel

Parameters	<ul style="list-style-type: none"> <li>■ id <ul style="list-style-type: none"> <li>- String</li> <li>- The id of the application</li> </ul> </li> <li>■ attributes <ul style="list-style-type: none"> <li>- Object</li> <li>- attributes for opening channel</li> </ul> </li> <li>■ callback <ul style="list-style-type: none"> <li>- Function</li> <li>- The callback handler</li> </ul> </li> </ul> <p>[callback parameter]</p> <ul style="list-style-type: none"> <li>* error <ul style="list-style-type: none"> <li>- An error instance if one occurred during the async call</li> </ul> </li> <li>* channel <ul style="list-style-type: none"> <li>- The channel instance</li> </ul> </li> </ul>
Return	■Void
Emulator Support	Y
SDK Constraint	.
Example	
<pre>var device = webapis.multiscreen.Device; device.openChannel(id, {name:"Joe"},function (error, channel) {  });</pre>	

## ready

### Description

Tells the system that application is ready to receive events

Parameters	none
Return	■Void
Emulator Support	Y
SDK Constraint	.

Example
<pre>var device = webapis.multiscreen.Device; device.ready();</pre>