

# Technical Specifications for Preparing Creatives

# Curtain

- Description
  - Main supported formats
  - Mobile devices
- Parameters
- API

## Description

This ad format display billboard which after cursor hovering or click (depends from settings) slide and reveal hidden image.

## Main supported formats

- HTML5
- Image
- Video
  - MP4/OGG/Webm
  - Flv

## Mobile devices

- Mobile web browsers
- Mobile web applications

## Parameters

Name	Description
General parameters	
Width*	Width in pixels.
Height*	Height in pixels.
Responsive	If checked, standard responsiveness will be enabled. (if the width set exceeds the width of the container in which the advertisement is located, the creative assumes the width of the container, and the height is appropriately rescaled).
Revealed element	
HTML5 file	HTML5 file for revealed element.
Image file	Image file for revealed element.
Curtain general parameters	
Image	Determines whether one or two images will be used as curtain.
The direction of opening	Horizontal/Vertical
Open event	Event which triggers opening of curtain. Hover or click.
Closeable	If checked, when hover device will leave container and 5 seconds will pass, creative will be closed.

Left/Right curtain parameters	
Image file	Image of curtain.
Background color	Color of curtain.
Image size	Values will be the same as background-size of CSS. Dimensions can be set as: width height, width or use one of keyword: contain, cover, auto. When dimensions will be set as pixels you should add "px" shortcut after values, eg.: 500px 250px, percentage values should be used in the same way with sign"%": "100% 100%". Set only width value will set only this dimension, height will be properly scaled.
Image position	Values will be the same as background-size of CSS. Position can be set as two values signifying in order horizontal and vertical. When positions will be set as pixels, percents or as one of keyword: left, center, right, top, bottom. Eg.: center center, 100px 200px, 5% 20px etc.
Image repeat	Specifies whether the image should be repeated in the banner area (to be used when image not filling whole container). Eg. of values: repeat (repeat on both axis), repeat-x (repeat only on axis x-horizontal), repeat-y (repeat only on y-vertical), no-repeat - don't repeat.
Text parameters	
Value	Text to be displayed in the middle of curtain.
Size	Size of text in pixels.
Color	Text color.
Shadow color	Color of shadow around text.
Other	
Extra JavaScript Code	Additional JS code executed when the ad creative is fully loaded.
Extra HTML Code	Additional HTML code executed when the ad creative is fully loaded

\* - obligatory parameter

## API

api can be used in "Extra JavaScript Code" field

```
api.onReady(callback: Function): void
```

Execute Callback when api is ready, eg. when there is a possibility to execute other api methods.

```
api.onReady(function() {
  console.log('I am ready');
});
```

```
api.onCurtainOpen(callback: Function): void
```

Execute Callback when curtain will start to open.

```
api.onCurtainClose(callback: Function): void
```

Execute Callback when curtain will start to close.

```
api.setImage(imageSource: String): void
```

Set 'src' of revealed image.

```
api.setOrientation(orientation: String): void
```

Set the direction of opening: api.VERTICAL or api.HORIZONTAL

```
api.setOrientation(api.VERTICAL);
```

```
api.setText(text: String, config?: {size?: string|number, color?: string, shadow?: string}): void
```

Set text which will be displayed in the middle of curtain. Optionally can set font size, color and shadow color.

Example of API usage:

```
const srcs = [  
  'image1.jpg',  
  'image2.jpg',  
  'image3.jpg'  
];  
let openCount = 0;  
let closeCount = 0;  
let isVertical = false;  
api.onReady(() => {  
  api.onCurtainOpen(() => {  
    if (openCount === 0) {  
      api.setImage(srcs[0]);  
    } else if (openCount === 1) {  
      api.setImage(srcs[1]);  
    } else if (openCount === 2) {  
      api.setImage(srcs[2]);  
    } else if (openCount === 3) {  
      api.setOrientation(!isVertical ? api.VERTICAL : api.HORIZONTAL);  
      isVertical = !isVertical;  
    }  
    ++openCount;  
  });  
  api.onCurtainClose(() => {  
    ++closeCount;  
    if (closeCount === 1) {  
      api.setText('Second text');  
    } else if (closeCount === 2) {  
      api.setText('Third text');  
    } else if (closeCount === 3) {  
      api.setText('Fourth text', {size: '36'});  
    } else if (closeCount === 4) {  
      api.setText('Fifth text');  
    } else if (closeCount === 5) {  
      api.setText('First text');  
      openCount = closeCount = 0;  
    }  
  });  
});
```