

Variable Template Previews

Variable Template Previews are thumbnail images, generated in real time, in which a Template Customization Object is applied to a specific Template. The following examples show how these images can be generated via the Design Huddle Javascript SDK.

SDK Initialization

When initializing the Javascript SDK, a Design Huddle instance domain must first be supplied:

```
DSHDLib.configure({
  domain: "dh1.designhuddle.com"
});
```

In addition, a previously attained Access Token must be supplied or the designation of the current user as a Visitor or Guest, to obtain a token implicitly with the Client ID of a Private App:

```
//Option 1
DSHDLib.configure({
  access_token: "abcd"
});

//Option 2
DSHDLib.configure({
  visitor: true,
  client_id: "abcd"
});

//Option 3
DSHDLib.configure({
  guest: true,
  client_id: "abcd"
});
```

Generating a Variable Template Preview Image URL

To generate a Variable Template Preview image, the following SDK method may be used to retrieve an image URL. This URL should be referenced directly when displaying the image to the user. These image requests must originate from a domain that is white-listed within the same Private App used to generate the supplied Access Token (this occurs automatically when the Access Token is generated implicitly for Visitor and Guest users).

```

DSHDLib.getVariableTemplatePreviewURL({
  template_id: 123,
  customizations_hash: "abcd"
}, function(error, image_url){
  if (!error) {
    var img = document.createElement("img");
    img.src = image_url;
  }
});

```

In the previous example only two parameters were supplied, but the entire list consists of:

Parameter	Data Type	Example Value
template_id	integer	123
template_code	string	"abcd"
page_number	integer	2
scene_number	integer	3
customizations_hash	string	"a1b2c3"
customizations	object	{ "elements": { "header": { "text": "My Title" } } }
width	integer	400
height	integer	300
fit	string	"top"

Note that the only required properties are a Template identifier (either ID or Code) and a Template Customization Object, which can be configured globally or supplied in each individual request (examples of these options are provided at the end of this document).

Template ID

The *template_id* parameter references the Design Huddle generated Template identifier. It is required unless a Template Code is supplied instead.

Template Code

The *template_code* parameter references an optional user generated Template identifier. It is required unless a Template ID is supplied instead.

Page/Scene Number

The *page_number* and *scene_number* parameters optionally allow specifying a page or scene to return other than the first (default) in the project.

Template Customization Object Hash

The *customizations_hash* parameter references a previously attained Template Customization Object Hash. It is required unless a Template Customization Object is supplied instead, or if an Object/Hash was configured globally.

Template Customization Object

The *customizations* parameter references a Template Customization Object. It is required unless a Template Customization Object Hash is supplied instead, or if an Object/Hash was configured globally.

Width

The *width* parameter is used to supply the desired width in pixels of the resulting image. If it is not supplied, the supplied height is used to calculate the width, maintaining the Template's aspect ratio. If a height is not supplied, then a default width of 320 pixels is used. If the width supplied or the resulting calculated height is greater than 800 pixels, the width/height will be sized down automatically (maintaining the aspect ratio) to ensure neither dimension is greater than 800 pixels.

Height

The *height* parameter is used to supply the desired height in pixels of the resulting image. If it is not supplied, the supplied width is used to calculate the height, maintaining the Template's aspect ratio. If a width is not supplied, then a default width of 320 pixels is used to calculate the height, maintaining the Template's aspect ratio. If the height supplied or the resulting calculated width is greater than 800 pixels, the width/height will be sized down automatically (maintaining the aspect ratio) to ensure neither dimension is greater than 800 pixels.

Fit

The fit parameter is used to determine how the resulting image is cropped when both a width and height are supplied that don't match the Template's aspect ratio. The following options are available:

Position	Alias	Description
"center"	"centre"	Will retain the horizontal/vertical center of the full aspect ratio thumbnail image
"northwest"	"top left"	Will retain the top left region
"north"	"top"	Will retain the top center region
"northeast"	"top right"	Will retain the top right region
"west"	"left"	Will retain the center left region
"east"	"right"	Will retain the center right region
"southwest"	"bottom left"	Will retain the bottom left region
"south"	"bottom"	Will retain the bottom center region
"southeast"	"bottom right"	Will retain the bottom right region
"entropy"		Will retain the region with the highest "Shannon" entropy
"attention"		Will retain the region with the highest luminance frequency, color saturation, and presence of skin tones

Note that if *fit* is not supplied and a *width* and *height* are supplied, "center" will be used by default.

The following section describes how Template Customization objects can be configured globally to remove the need to supply them in every Variable Template Preview request.

Template Customization Objects

If a Template Customization Object was previously stored with its generated Object Hash retained, it can be configured globally as follows:

```
DSHDLib.configure({
  template_customization_object_hash: "abcd"
});
```

If this is the first use of this object or its generated hash was not retained from a previous use, it can also be supplied in full:

```
DSHDLib.configure({
  template_customization_object: {
    elements: {
      "customer_name": {
        text: "ABCD Inc."
      }
    },
    classes: {
      "logo": {
        url: "data:image/svg+xml;utf8,<svg ... </svg>"
      },
      "logo_topleft": {
        position_relative_to_original: {
          horizontal_align: "left",
          vertical_align: "top"
        }
      }
    }
  }
});
```

Note that after its first use, the Javascript SDK will cache its generated Object Hash internally for possible re-use, avoiding unnecessary API calls/backend deduplication. However this cache is lost upon browser reload.

To swap out the current Template Customization Object in use, simply call the *configure* method of the SDK again with a new object.

Storing a Template Customization Object Explicitly

If an Object Hash is desired to retain for later use, the following SDK method may be used:

```
DSHDLib.storeTemplateCustomizationObject({
  object: {
    elements: {
      "customer_name": {
        text: "ABCD Inc."
      }
    }
  }
}, function(error, data){
  if (!error) {
    var hash = data.object_hash;
  }
});
```