

Box It Now™ Widget

Intro

The Box It Now Widget is a great way to take advantage of Box It Now's advanced Design Studio capabilities directly from within your own website. This allows you to retain the styling and flexibility of your own page without having to drastically transition the user from one interface to another. This document is intended to be a quick introduction to the code that allows you to embed our widget in your site.

CSS

The Box It Now widget requires some CSS to be loaded in the page. This CSS file should be included at the top of your page, before the bootstrap js file or the widget snippet. The file is remotely loaded from the Box It Now servers. Certain parts of the widget styling can be overridden to allow you to further customize the look of the widget, however that is beyond the scope of this document. The following code will add the CSS to your page.

```
<!-- BoxItNow CSS -->
<link rel="stylesheet" type="text/css" href="http://localhost/css/bin.cache.min.css" />
```

JS Loader

The JS loader contains the bulk of what you will need to be able to add a Box It Now widget to your page. This file should be included after the CSS file mentioned above is included.

```
<!-- BoxItNow JS -->
<script type="text/javascript" src="http://localhost/js/bin.bootstrap.min.js"></script>
```

Base JS Snippet

The base JS Snippet is where the bulk of the configuration is done. Please see the "example-widget" folder in the GitHub repo for an example of this.

The parameters are described below:

- api: url for the Box It Now widget server. Generally api.boxitnow.com or widget.boxitnow.com
- tag: the ID of the div that you would like the widget to reside in on your page
- design: information about the design you're loading is located in this array
 - type: is the embedded design coming from our standard categories or from a saved design made by a user
 - id: the ID number of the design you want to embed
 - userToken: (optional) required only if you are loading a saved design. This token is received when logging the user in via the api
 - parameters: (optional) design parameters such as length, width, depth, allowance, etc can be outlined and passed here.
- Options: settings for enabling or disabling features are in this array. If this is not included, all features default to True/Enabled

- resizer: enable/disable the entire resizer functionality tab, or specific modules of the resizer
 - graphics: enable/disable the entire graphics functionality tab, or specific graphics capabilities modules
 - export: enable/disable the entire export tab, or specific modules of the export section
- onLoad: (optional) include custom functionality that you would like to automatically perform once the widget is successfully loaded
- onResize: (optional) include custom functionality the you would like to automatically perform ever time a resize event occurs
- onSave: (required) include custom functionality that dictates what actions are taken when the user chooses to save their design
- onExport (required) include custom functionality that dictates what actions are taken when the user attempts to export a design.