



Документация по Site3D Configurator API

Оглавление

Доступ к виджету.....	2
Загрузка виджетов	2
Получение выбор пользователя	2
Получение доступа к нужному элементу конфигуратора и его настройкам.....	3
Изменение цвета.....	3
Загрузка собственной текстуры из внешнего источника.....	3
Создание скриншота.....	3
Не нашли ответ на свой вопрос?.....	4

Доступ к виджету

Для начала работы с API Site3D Configurator необходимо получить доступ к объекту 3D виджета (Site3dWidget) нужного конфигуратора. Это можно сделать несколькими способами:

- Через свойство объекта `window['site3dWidget_${id}']`, где `id` – идентификатор виджета (его можно увидеть на последнем шаге создания проекта)
- Через массив всех виджетов, доступный в свойстве объекта `window['site3dWidgets']`

Для проверки, загружен ли виджет в настоящий момент, служит свойство `isLoad`.

Для дальнейшего изучения возможностей класса Site3dWidget необходимо обратиться к документации библиотеки Site3D по адресу <https://doc.site3d.site>.

Загрузка виджетов

Для обработки процесса загрузки всех виджетов на странице необходимо определить функцию `site3dConfiguratorLoadStatus` глобального объекта `window`, где `info` – объект с результатом текущего этапа загрузки виджетов (`loaded` – истина, если процесс загрузки завершен, `message` – описание этапа загрузки, `percent` – процент завершения текущего этапа (может отсутствовать)).

Ниже представлен пример загрузчика в React:

```
useEffect(() =>
{
  window.site3dConfiguratorLoadStatus = info =>
  {
    if (info.loaded)
    {
      dispatch(loader(false)); // Скрыть загрузчик
      return;
    }

    dispatch(loader(true, info.message, info.percent)); // Показать загрузчик
  };
}, []);
```

Получение выбор пользователя

Для того, чтобы определить, какие настройки применил пользователь достаточно обратиться к методу `getConfiguratorInfo` объекта 3D виджета. Он возвращает объект с перечнем элементов конфигуратора и информацией об их настройках.

Ниже приведен пример вызова и возвращаемого данным методом объекта:

```
const widget = window['site3dWidgets'][0];
console.log(widget.getConfiguratorInfo());
```

1. `items: Array(1)`
 1. `0:`
 1. `caption: "Загородный дом"`
 2. `settings: Array(7)`
 1. `0: {caption: 'Показать крышу', value: true}`
 2. `1: {caption: 'Материал фасада', value: 'material_walls_out_1.jpg'}`

3. 2: {caption: 'Цвет черепицы', value: '#ffffff'}
4. 3: {caption: 'Вид черепицы', value: 'material_roof_tiles_1.jpeg'}
5. 4: {caption: 'Вид балкона', value: 'model_balcony_1.glb'}
6. 5: {caption: 'Цвет декоративных элементов', value: '#e7e7e7'}

Получение доступа к нужному элементу конфигуратора и его настройкам

Ниже приведен пример получения доступа к элементу конфигуратора с именем **main** (такое имя у элемента, которому соответствует модель, загружаемая на первом шаге создания или редактирования проекта), а далее, к его настройкам в виде объекта Map:

```
const item = widget.configurator.items.get('main');
const settingsItems = item.settings.items;
```

Изменение цвета

Для редактирования значения настройки цвета, можно вызвать у неё метод **setValue**, передав ему цвет в формате HEX.

Ниже представлен пример изменения цвета настройки, привязанной к части модели с именем **core**:

```
const colorSetting = settingsItems.get('color_core');
colorSetting.setValue('#ff0000');
```

Если же нам нужно сменить цвет у настройки материала, то можно вызвать у неё метод **setColor**, например:

```
const materialSetting = settingsItems.get('material_image');
materialSetting.setColor('#ff0000');
```

Загрузка собственной текстуры из внешнего источника

Чтобы загрузить свою текстуру для настройки материала, необходимо вызвать у неё метод **setTextureFromBlob**, передав ему изображение в виде Blob-объекта.

Ниже продемонстрирован пример загрузки текстуры **blob** для настройки материала, которая привязана к части модели с именем **image**:

```
const materialSetting = settingsItems.get('material_image');
materialSetting.setTextureFromBlob(blob);
```

Создание скриншота

Для того, чтобы получить файл изображения сцены в формате png или jpeg, можно вызвать метод **exportImage** объекта 3D виджета, передав ему нужные параметры в виде объекта (смотрите в примере ниже). Данный метод возвращает **Blob-объект** скриншота или **null** в случае невозможности создания изображения.

```
await widget.exportImage({
  fileName: 'image',
  format: 'png',
  width: 1920,
  height: 1080,
```

```
isWatermark: false,  
isDownload: true  
});
```

Не нашли ответ на свой вопрос?

Обратитесь, пожалуйста, в нашу [поддержку](#), поможем с интеграцией сервиса с вашим сайтом.