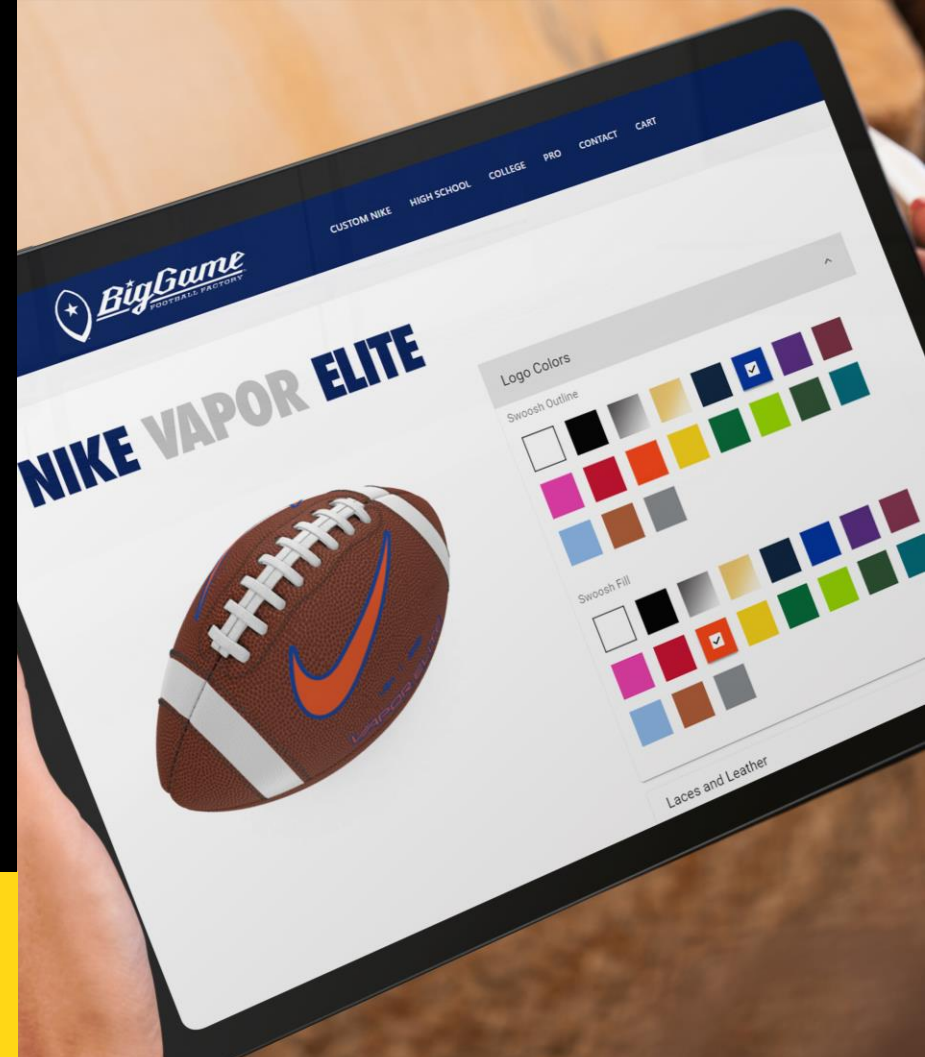


Quick start guide to **Expivi** Website Integration



expivi.com



Expivi

Expivi Website Integration

Browser support

Expivi's 3D Configurator requires WebGL support on client's browser and hardware (supported by all major browsers and desktop/mobile devices).

The minimum required browser version includes

Desktop:

- Chrome 8+
- Firefox 4+
- Safari 8+
- Opera 12.1+
- Edge 12+
- IE 11

Mobile:

- Safari 8+
- Android browser 67+
- Blackberry browser 10+
- Opera mobile 12+
- Chrome 71+
- Firefox 64+
- IE 11+

Expivi Website Integration

Webpage script inclusion and containers

In order to create an instance of the configurator, the webpage must have included required scripts and provide containers to place the 3D viewer and the options inside.

Scripts & Styling

Include the following javascript & css files inside the header tag (<header></header>) of HTML that is going to contain the configurator.

```
<link href="{api_asset_url}/ExpiviComponent.css"
rel="stylesheet" />

<script
src="{api_asset_url}/vendor.lib.js"></script>
<script src="{api_asset_url}/index.js"></script>
<script
src="{api_asset_url}/ExpiviComponent.js"></script>
```

Replace {api_asset_url} with Expivi's provided asset directory.

NOTE: Please contact Expivi to receive the Asset_URL.

Expivi Website Integration

Containers

The configurator requires 3 container elements inside the body tag (<body></body>):

- 1 to render the 3D viewer inside.
- 1 to render the product's options inside.
- 1 to render price into.

The added elements must contain an id attribute to be passed to configurator's constructor method.

Example container:

```
<div id="viewerContainer" style="width: 70%;height: 600px;float: left"></div>
<div style="width: 30%;float:left">
  <div id="priceContainer"></div>
  <div id="optionsContainer"></div>
</div>
```

NOTE: The viewer and options will fill their parenting container's size. The position and responsiveness of containers is left to webpage implementation.

Expivi Website Integration

3D Viewer and Options

With the script and containers included into the HTML page (see: Webpage script inclusion and containers), an instance of configurator can be created to initialize and render the Expivi's elements into provided containers.

Creating an instance

Create a new instance of `ExpiviComponent` object after the window has loaded.

```
<script type="text/javascript">
  window.addEventListener('load', function () {
    var expiviInstance = new ExpiviComponent.default({
      catalogueId: {catalogue_id},
      viewerContainer: {view_container_selector},
      optionContainer: {option_container_selector},
      priceSelectors: {price_container_selector},
      currency: {currency},
      locale: {locale},
      token: {api_token},
      preset: {preset}
    });
  });
</script>
```

Add the script tag at the end of body tag.

Expivi Website Integration

Constructor options

The configurator must be initialized with the following options:

```
{
  catalogueId: {catalogue_id},
  viewerContainer: {view_container_selector},
  optionContainer: {option_container_selector},
  priceSelectors: {price_container_selector},
  currency: {currency},
  locale: {locale},
  token: {api_token},
  configuration?: {serialized_object},
  preset?: {preset}
}
```

{catalogue_id}: The id of product, inside catalogue, to be loaded by the configurator.

{view_container_selector}: The element's selector to place the 3D Viewer inside (eg. '#viewerContainer' as defined in Containers).

{option_container_selector}: The element's selector to place the product options inside (eg. '#optionsContainer' as defined in Containers).

{price_container_selector}: The element's selector to place the product price inside (eg. '#priceContainer' as defined in Containers).

{currency}: Pricing currency to display (eg. 'USD').

{locale}: Locale to use for localization (eg. 'en');

{api_token}: API Token with read permission created from Expivi's admin panel.

{serialized_object}: Serialized JSON object to reload a product to a pre-configured state **-optional parameter**

{preset}: Presets are preconfigured options of a product. These are defined in the Expivi's admin panel.

Expivi Website Integration

Serialization

The configuration created by a user can be serialized to a JSON object to be communicated to webshop cart; or to be saved to storage to reload a product, to a pre-configured state.

Saving configuration

Saving the configuration will create a thumbnail of 3D rendered product (which can be used by webshop cart), provide representable specifications (visible and selected attributes by user) and a snapshot of attribute's state for internal reloading purposes.

Method

```
window.expivi.saveConfiguration(width, height) :  
Promise<ConfiguredProductOptions[]>;
```

Parameters

- width: the width of thumbnail to be created.
- height: the height of thumbnail to be created.

Return

A promise which resolves an array of `ConfiguredProductOptions`

ConfiguredProductOptions

```
interface ConfiguredProductOptions {  
  version : string; //interface version  
  catalogue_id : string; //product's id  
  thumbnail? : string; //thumbnail as base64 data  
  configuration : {  
    attributes : AttributeSerializedValue[]//configuration state  
    for internal usage  
  };  
  attributes : AttributeSerializedValue[]; //representable  
  product specifications  
  articles? : ProductArticleModel[]; //bill of materials  
}
```

Continues on the next page...

Expivi Website Integration

AttributeSerializedValue

```
interface AttributeSerializedValue {
  attribute_id : number; //attribute's id in database
  attribute_name? : string; //attribute's name (product
specification name)
  attribute_value_name?: string; //attribute value (product
specification value)
}
```

Example call

```
window.expivi.saveConfiguration(256,
256).then(function(aSaveResults){
  console.log("products thumbnail: ", aSaveResults[0].thumbnail);
  console.log("products specifications: ",
aSaveResults[0].attributes);
  console.log("products articles: ", aSaveResults[0].articles);
});
```

ProductArticleModel

```
interface ProductArticleModel {
  models_id? : number; //component's id in database
  models_sku : string; //part's sku
  material_group_id? : number; //applied material's id in
database
  material_group_sku? : string; //sku of applied material on part
  price_sku? : string; //price identifier to look into pricing
matrix
  price_fields? : {[key : string]: number}; //calculated total
price for each predefined field
}
```


Expivi Website Integration

Sample code

The following sample contains a responsive fullscreen example layout with the configurator.

```
<html>
  <head>
    <title>Expivi - 3D Configurator sample code</title>
    <link href="ExpiviComponent.css" rel="stylesheet" />
    <script src="vendor.lib.js"></script>
    <script src="index.js"></script>
    <script src="ExpiviComponent.js"></script>
    <style>
      .expivi-single-product-container {
        display: table;
        height: 100%;
        width: 100%;
      }

      .expivi-single-product-container .expivi-viewer-
      container {
        display: table-cell;
        width: 64%;
        padding-right: 15px;
        vertical-align: top;
      }

      .expivi-single-product-container .expivi-option-
      container {
        height: 100%;
        display: table-cell;
        width: 36%;
      }
    </style>
  </head>
  <body>
    <div class="expivi-single-product-container">
      <div class="expivi-viewer-container">
        <div id="viewerContainer" style="position:
        sticky;top:0;width:100%;height:100vh;"></div>
      </div>
      <div class="expivi-option-container">
        <div id="priceContainer"></div>
        <div id="optionsContainer"></div>
      </div>
    </div>
  </body>
</html>
```

```
@media (max-width: 600px) {
  .expivi-single-product-container {
    display: block;
  }

  .expivi-single-product-container .expivi-viewer-
  container {
    height: 40vh;
    display: block;
    width: 100%;
    padding-right: 0;
  }

  .expivi-single-product-container .expivi-option-
  container {
    height: initial;
    display: block;
    width: 100%;
  }
}
</style>
</head>
<body style="width: 100%;height: 100%;margin:0;padding:0">
  <div class="expivi-single-product-container">
    <div class="expivi-viewer-container">
      <div id="viewerContainer" style="position:
      sticky;top:0;width:100%;height:100vh;"></div>
    </div>
    <div class="expivi-option-container">
      <div id="priceContainer"></div>
      <div id="optionsContainer"></div>
    </div>
  </div>
</body>
</html>
```

```
<div class="expivi-option-container">
  <div id="priceContainer"></div>
  <div id="optionsContainer"></div>
</div>

<script type="text/javascript">
  window.addEventListener('load', function () {
    var expiviInstance = new ExpiviComponent.default({
      catalogueId: 154,
      viewerContainer: "#viewerContainer",
      optionContainer: "#optionsContainer",
      priceSelectors: "#priceContainer",
      currency: "EUR",
      locale: "nl",
      token: "ABC565656565656565",
      preset: "preset-product-red"
    });
  });
</script>
</body>
</html>
```