



FREE eBook

LEARNING bigcommerce

Free unaffiliated eBook created from
Stack Overflow contributors.

#bigcommer

ce

Table of Contents

About	1
Chapter 1: Getting started with bigcommerce	2
Remarks.....	2
Examples.....	2
Getting Started with BigCommerce.....	2
Chapter 2: Using javascript to customize the storefront	3
Introduction.....	3
Examples.....	3
Obtaining product data and order details from finishorder.php.....	3
Change product image on hover on a category page in Stencil.....	6
Credits	7

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [bigcommerce](#)

It is an unofficial and free bigcommerce ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official bigcommerce.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with bigcommerce

Remarks

This section provides an overview of what bigcommerce is, and why a developer might want to use it.

It should also mention any large subjects within bigcommerce, and link out to the related topics. Since the Documentation for bigcommerce is new, you may need to create initial versions of those related topics.

Examples

Getting Started with BigCommerce

1. Go to the [BigCommerce](#) homepage.
2. Enter a store name in the field labeled **Create your store name**, then click **Start my free trial now**.

Your store name can be changed at any time after starting your trial.

3. Fill out the new trial store form, then click **Create my store now**.

Once your trial is created you'll be prompted to log in for the first time.

1. Enter the email address and password you set when creating your trial, then click **Log In**.

You can now use your trial store and new account to get started with the [Stencil frontend framework](#) or begin [building an OAuth app](#).

When you login to the Developer Portal to get OAuth credentials for the first time, you'll be prompted to authorize the Developer Portal application.

1. To obtain these credentials, click **My Apps** in this site's upper-right-hand corner, and log in to your BigCommerce account.
2. Once inside the Developer Portal, click **Create an app**. You can provide any name you'd like, such as "Test".
3. Click **View Client ID** in order to obtain your **Client Secret** and **Client ID**.

Read [Getting started with bigcommerce online](#):

<https://riptutorial.com/bigcommerce/topic/8117/getting-started-with-bigcommerce>

Chapter 2: Using javascript to customize the storefront

Introduction

Javascript examples to interact with existing store variables or scrape data that isn't easily exposed

Examples

Obtaining product data and order details from finishorder.php

It parses the data from the `%%GLOBAL_ConversionCode%%` template variable, and as such this script should be inserted in `order.html` immediately after the `%%GLOBAL_ConversionCode%%` variable. This was originally intended for the Blueprint theme framework and, as such, may not work on Stencil.

```
<script>
//----- Main -----//

/** Create the order data array from analytics script */
var data = parseAnalyticsData(getAnalyticsScript());
//console.log(data);

/**
 * Retrieve the order details as an object, properties are:
 * id           - The order ID.
 * shipping     - The order shipping cost.
 * tax          - The order tax cost.
 * shippingTax - The order shipping tax cost.
 * city         - The order shipping city.
 * state        - The order shipping state.
 * country      - The order shipping country.
 */
var orderDetails = getOrderDetails(data);

console.log("Order ID = %d", orderDetails.id);
console.log("Order shipping city = %s", orderDetails.city);
console.log("Order subtotal = %f", orderDetails.subtotal);

/**
 * Retrieve the order product details, as an array of product objects.
 * Properties are:
 * id           - The product ID.
 * description  - The product description.
 * tax          - The product tax cost.
 * price        - The product price per product.
 * qty          - The product quantity purchased.
 */
var products = getOrderProducts(data);
```

```

/** Loop through the products array to access each product */
console.log("Total number of products = %d", products.length);
for (x=0; x<products.length; x++) {
  console.log("-----");
  console.log("Item # ", x+1);
  console.log("Product ID = %f", products[x].id);
  console.log("Product QTY = %f", products[x].qty);
  console.log("Product Price = %f", products[x].price);
  console.log("-----");
}

//----- Functions -----//

/**
 * Parses the DOM to retrieve the order data analytics script.
 */
function getAnalyticsScript() {
  var scripts = document.getElementsByTagName('script');
  var thisScriptTag = scripts[scripts.length - 2];
  var data = thisScriptTag.textContent || thisScriptTag.innerText;
  return data;
}

/**
 * Parses the raw analytics script element to remove all script
 * text, and parse just the order related data into an array.
 * @param script <String> - The raw order analytics script.
 * @return <mixed> - Array containing the order data.
 */
function parseAnalyticsData(data) {
  String.prototype.replaceAll = function(search, replacement) {
    var target = this;
    return target.split(search).join(replacement);
  };
  // This is hacky, and probably inefficient, but it removes all
  // script related text, so the end result is just a comma separated
  // array of the order and product data.
  data = data.replace("if(typeof(pageTracker) != 'undefined') {", '');
  data = data.replaceAll( 'pageTracker._addTrans(', '');
  data = data.replaceAll( ' pageTracker._trackTrans();', '');
  data = data.replaceAll( 'pageTracker._addItem(', '');
  data = data.replaceAll(');', '');
  data = data.replace('}', '');
  data = data.replace( /\n/g, ",").replaceAll( ",,","");
  data = data.replace(/\s/g, '');
  data = data.split(',');
  data = cleanArray(data); // Remove all empty values from array.
  return data;
}

/**
 * Removes all empty data from array.
 * @param array <mixed> - The array to clean.
 */
function cleanArray(array) {
  var newArray = new Array();
  for (var i = 0; i < array.length; i++) {
    if (array[i]) {
      newArray.push(array[i]);
    }
  }
}

```

```

    }
}
return newArray;
}

/**
 * Parse Analytics Data for Order Details
 * @param data <mixed> - The order analytics data.
 * @return <mixed>      - Object containing the order details.
 */
function getOrderDetails(data) {
    String.prototype.replaceAll = function(search, replacement) {
        var target = this;
        return target.split(search).join(replacement);
    };
    return {
        id            : parseFloat(data[0].replaceAll("'", '')),
        subtotal      : ( parseFloat(data[2].replaceAll("'", '')) -
(parseFloat(data[3].replaceAll("'", '')) + parseFloat(data[4].replaceAll("'", '')) ) ),
        total         : parseFloat(data[2].replaceAll("'", '')),
        tax           : parseFloat(data[3].replaceAll("'", '')),
        shipping      : parseFloat(data[4].replaceAll("'", '')),
        city          : data[5].replaceAll("'", ''),
        state         : data[6].replaceAll("'", ''),
        country       : data[7].replaceAll("'", '')
    }
}

/**
 * Parse Analytics Data for All Order Product Details.
 * @param data <mixed> - The order analytics data.
 * @return <mixed>      - Array containing individual product details.
 */
function getOrderProducts(data) {
    String.prototype.replaceAll = function(search, replacement) {
        var target = this;
        return target.split(search).join(replacement);
    };
    var counter = -1;           // Keep index of details per product.
    var productsArray = []; // Init empty array to hold all products.
    var product = {};         // Init empty object to hold single product data.
    /** Product data starts at index 8 */
    for (x=8; x<data.length; x++) {
        counter++;
        switch (counter) {
            case 1:
                product.id = parseFloat(data[x].replaceAll("'", ''));
                break;
            case 2:
                product.description = data[x].replaceAll("'", '');
                break;
            case 3:
                product.tax = parseFloat(data[x].replaceAll("'", ''));
                break;
            case 4:
                product.price = parseFloat(data[x].replaceAll("'", ''));
                break;
            case 5:
                product.qty = parseFloat(data[x].replaceAll("'", ''));
                counter = -1;           // reset counter
                productsArray.push(product); // push product to products array

```

```
        product = {};  
        break;  
    }  
}  
return productsArray;  
}
```

Change product image on hover on a category page in Stencil

This was added to `assets/js/theme/category.js` in `loaded()`. You will also need to add `{{inject "categoryProducts" category.products}}` to `templates/pages/category.html`

```
var mainImages = [];  
var rollOvers = [];  
this.context.categoryProducts.forEach(function(e, i) {  
    if (e.images[0]) {  
        mainImages[e.id] = e.images[0].data;  
    }  
    if (e.images[1]) {  
        rollOvers[e.id] = e.images[1].data;  
    }  
});  
  
rollOvers.forEach(function(image, id) {  
    image = image.replace('{:size}', '500x659');  
  
    $('a[data-product-id="' + id + '"]').closest('li.product').find('.card-image')  
        .on('mouseover', function() {  
            $(this).attr('src', image);  
        }).on('mouseout', function() {  
            $(this).attr('src', mainImages[id].replace('{:size}', '500x659'));  
        });  
});
```

Read [Using javascript to customize the storefront online](https://riptutorial.com/bigcommerce/topic/8333/using-javascript-to-customize-the-storefront):

<https://riptutorial.com/bigcommerce/topic/8333/using-javascript-to-customize-the-storefront>

Credits

S. No	Chapters	Contributors
1	Getting started with bigcommerce	Alyss , Community
2	Using javascript to customize the storefront	Alyss