



**eBook Gratuit**

# APPRENEZ bigcommerce

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#bigcommer

ce

# Table des matières

<b>À propos</b> .....	<b>1</b>
<b>Chapitre 1: Démarrer avec bigcommerce</b> .....	<b>2</b>
Remarques.....	2
Exemples.....	2
Premiers pas avec BigCommerce.....	2
<b>Chapitre 2: Utiliser javascript pour personnaliser la vitrine</b> .....	<b>3</b>
Introduction.....	3
Exemples.....	3
Obtenir des données de produits et commander des détails de finishorder.php.....	3
Modifier l'image du produit en survol sur une page de catégorie dans Stencil.....	6
<b>Crédits</b> .....	<b>7</b>

---

# À propos

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](mailto:info@zzzprojects.com)

---

# Chapitre 1: Démarrer avec bigcommerce

## Remarques

Cette section fournit une vue d'ensemble de ce qu'est bigcommerce et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets au sein du bigcommerce, et établir des liens avec les sujets connexes. La documentation de bigcommerce étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

## Exemples

### Premiers pas avec BigCommerce

1. Accédez à la page d'accueil [BigCommerce](#) .
2. Entrez un nom de magasin dans le champ intitulé **Créez votre nom de magasin** , puis cliquez sur **Démarrer mon essai gratuit maintenant** .

Votre nom de magasin peut être modifié à tout moment après le début de votre essai.

3. Remplissez le nouveau formulaire de magasin d'essai, puis cliquez sur **Créer mon magasin maintenant** .

Une fois votre version d'évaluation créée, vous serez invité à vous connecter pour la première fois.

1. Entrez l'adresse e-mail et le mot de passe que vous avez définis lors de la création de votre version d'évaluation, puis cliquez sur **Connexion** .

Vous pouvez maintenant utiliser votre banque d'essai et votre nouveau compte pour commencer à utiliser l' [architecture frontale Stencil](#) ou commencer à [créer une application OAuth](#) .

Lorsque vous vous connectez au portail des développeurs pour obtenir des informations d'identification OAuth pour la première fois, vous êtes invité à autoriser l'application du portail de développeur.

1. Pour obtenir ces informations d'identification, cliquez sur **Mes applications** dans le coin supérieur droit de ce site, puis connectez-vous à votre compte BigCommerce.
2. Une fois dans le portail des développeurs, cliquez sur **Créer une application** . Vous pouvez fournir n'importe quel nom que vous souhaitez, tel que "Test".
3. Cliquez sur **Afficher l'ID client** pour obtenir votre **secret client** et votre **identifiant client** .

Lire [Démarrer avec bigcommerce en ligne](#):

<https://riptutorial.com/fr/bigcommerce/topic/8117/demarrer-avec-bigcommerce>

# Chapitre 2: Utiliser javascript pour personnaliser la vitrine

## Introduction

Exemples de Javascript pour interagir avec des variables de magasin existantes ou pour effacer des données qui ne sont pas facilement exposées

## Exemples

### Obtenir des données de produits et commander des détails de finishorder.php

Il analyse les données de la variable de modèle %% GLOBAL\_ConversionCode %% et, en tant que tel, ce script doit être inséré dans order.html immédiatement après la variable %% GLOBAL\_ConversionCode %%. Cela était à l'origine destiné à l'infrastructure de thème Blueprint et, en tant que tel, peut ne pas fonctionner sur 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;
}

```

## Modifier l'image du produit en survol sur une page de catégorie dans Stencil

Cela a été ajouté à `assets/js/theme/category.js` dans `loaded()` . Vous devrez également ajouter `{{inject "categoryProducts" category.products}}` à `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'));
        });
});

```

Lire Utiliser javascript pour personnaliser la vitrine en ligne:

<https://riptutorial.com/fr/bigcommerce/topic/8333/utiliser-javascript-pour-personnaliser-la-vitrine>



---

# Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec bigcommerce	<a href="#">Alyss</a> , <a href="#">Community</a>
2	Utiliser javascript pour personnaliser la vitrine	<a href="#">Alyss</a>