

Four51 OrderDetails Object

October 2014

Abstract

This article details the OrderDetails object that is available via JavaScript in the buyer interface. This object contains many details of the user’s current cart and can be used to provide unique functionality.

Table of Contents

- Abstract 1
- Overview..... 2
- Detailed Specifications 3
 - Example 4
- Troubleshooting and Support..... 5
 - Seller Support 5
 - Buyer Support..... 5
- References 5

Overview

The contents of this object are details of the user's current cart – order total, number of line items, and specifics of each line item, such as product ID, product name, quantity, and unit price. This object is available on every page of the buyer interface, so you can programmatically drive logic in places like the Shipping page, based on what is in the user's cart.

With this functionality and some creativity, you can create custom business logic or enhance the presentation of the buyer experience to suit your customers' needs.

The OrderDetails object is a property of the \$451 jQuery object that is exposed primarily to help Four51 developers accomplish tasks on the page easier, but can also be used by customers to enhance their buyer interfaces. You will need to have some knowledge of JavaScript in order to make use of this functionality.

It is important to note three things about this functionality:

1. Any logic built around it will be done through JavaScript, and therefore in the user's browser. It is not advisable to drive anything pricing related via this functionality, as a user with knowledge of debugging JavaScript can easily render the script useless if they wish.
2. Because it will be accessed with JavaScript, if there are other script errors on the page, there is the chance that the logic will never be performed since JavaScript is executed sequentially.
3. This functionality cannot add new items to the database for use. For example, you can have logic based on this object hide or display certain shippers on the Shipper page, but you cannot add new shippers that do not exist already in the Four51 application.

Detailed Specifications

The structure of the OrderDetails object is:

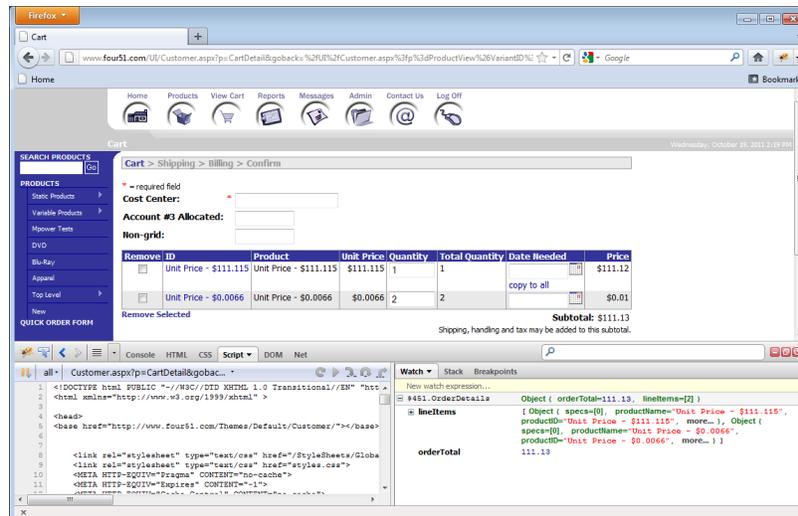
OrderDetails

- lineItems[n]
 - lineTotal
 - productID
 - productName
 - quantity
 - specs
 - unitPrice
- orderTotal

where n is the zero-based index of the line item you want to access.

For any JavaScript-related work, being familiar with your browser's debugging tools is extremely helpful. The Firebug extension for Firefox (see References) will be used to explore the `$451.OrderDetails` JSON object.

Log in as a buyer user, add some items to the cart, and open the Firebug pane. On the 'Script' tab, enter '`$451.OrderDetails`' as a new watch. After adding the watch, click on the '+' next to it to expand the object. Your screen should now look like this:



With the Firebug UI, examining the details of this object is very simple. You can use the 'watch' functionality to quickly test more complicated expressions. For example, if you wanted to know the product ID of the second line item, you could use the following watch expression:

`$451.OrderDetails.lineItems[1].productID`

Example

Some e-commerce platforms display the current cart details on every page – a piece of functionality that Four51 has not implemented yet. With the OrderDetails object, you can easily add your own cart status to every page of your buyer interface. **This is a basic example, and you may need to make further modifications for a production-ready version.**

1. Create a new custom theme. This needs to be done in a custom theme because you want the cart status to be displayed on every page.

2. In the theme's *template.htm* file:

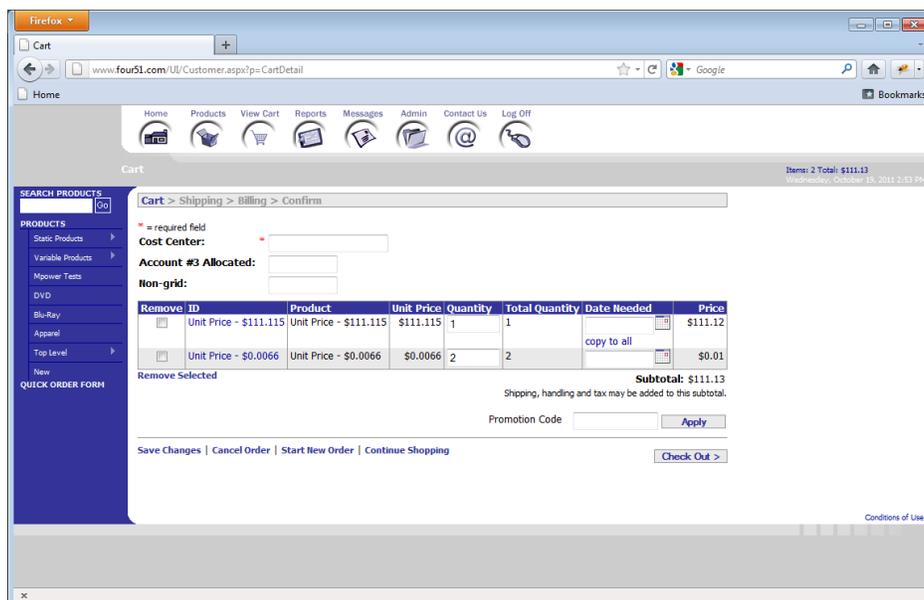
- a. Add the placeholder for the cart status. This example puts it next to the DateTime:

```
<div id="PageTitle">[[PageTitle]]</div><div id="DateTime"><div><a id="CartCount"
href="/UI/Customer.aspx?p=CartDetail">$CartCount $CartTotal</a></div><div>[[DateTime]]</div></div>
```

- b. Add the script:

```
<script type="text/javascript">
$(document).ready(function() {
    if ($451.OrderDetails != null)
    {
        var cartCount = $451.OrderDetails.lineItems.length;
        var cartTotal = $451.OrderDetails.orderTotal.toFixed(2);
        var oldText = $('#a#CartCount').text();
        var newText = oldText.replace('$CartCount', 'Items: ' + cartCount);
        newText = newText.replace('$CartTotal', 'Total: $' + cartTotal);
        $('#a#CartCount').text(newText);
    }
    else
    {
        $('#a#CartCount').text('Items: 0 Total: $0.00');
    }
});
</script>
```

The result is a link that displays the number of items in the cart, and the order total. Clicking the link will take you to the View Cart page:



Troubleshooting and Support

Disclaimer about custom HTML, CSS, or JavaScript - Four51 does not offer support for custom HTML, CSS, or JavaScript. Requests for assistance building a solution or troubleshooting an existing solution may be accepted as quotable projects depending on the nature of the project and available resources.

Seller Support

Questions can be submitted to Four51 by submitting a case in the Four51 Customer Resource Center. This can be accessed through the knowledgebase as follows:

- Logon to your admin interface on Four51 (<http://www.four51.com/ui/logon.aspx>)
- Select 'Help' in the upper right corner
- Click on 'Find Answers'

Please check the knowledge base first, as your question may have already been posted there. If you need additional help, the please select one of the following links:

- Click on the 'Submit A Case Here' link in the lower right-hand side of the panel; or
- Click on 'Click Here To Submit A Case' on the home page of the knowledgebase

Buyer Support

Please route all questions and issues through your Seller. Communication through e-mail or phone is possible, but this allows the Seller to be aware of all issues, and establishes a standard support process.

References

Firebug Web Development Tool - <http://getfirebug.com/>
jQuery – <http://www.jquery.com>
JSON - <http://www.json.org/>