



**KonaKart**  
Enterprise Java eCommerce

## KonaKart Shopping Widgets

3rd January 2018

DS Data Systems (UK) Ltd.,  
9 Little Meadow  
Loughton, Milton Keynes  
Bucks  
MK5 8EH  
UK

## Introduction

KonaKart ( [www.konakart.com](http://www.konakart.com) ) is a Java based eCommerce platform where all functionality is available through APIs. This makes it particularly suitable for integration into front and back end systems. In this document we'll concentrate on integrating shopping widgets into social networks and blogs in order to increase the visibility of your products to a wider audience. An affiliate code may be added to the widget in order to reward affiliate partners. When a customer checks out from the widget, the affiliate code is saved on the order.

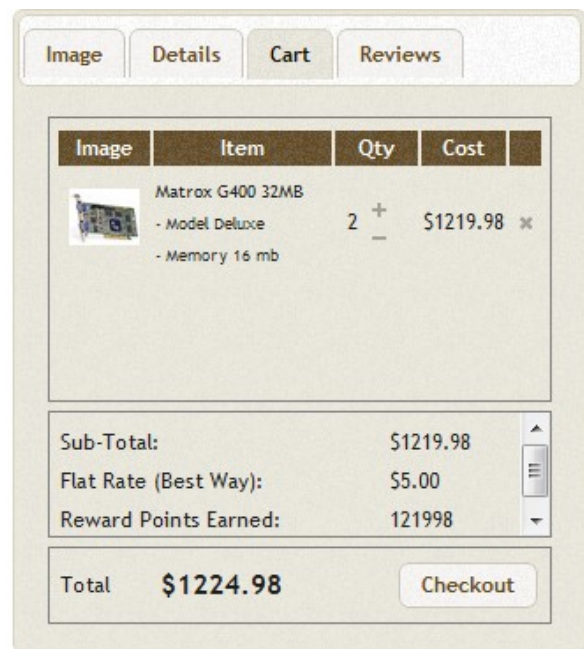
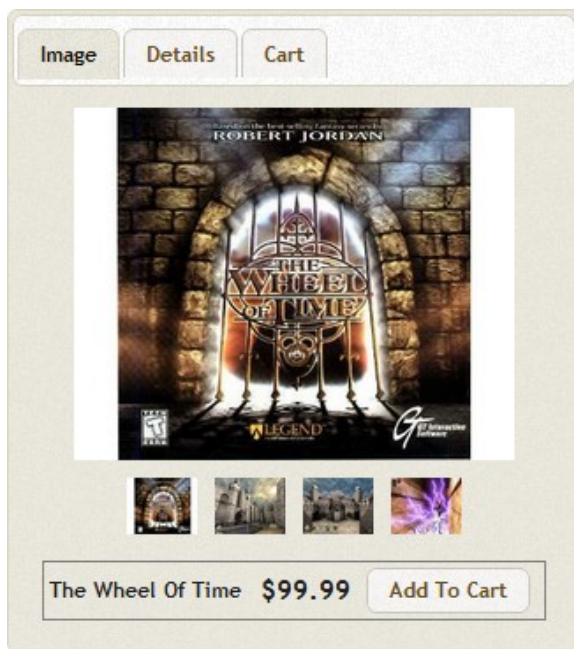
Note that shopping widgets are only available in the Enterprise Version of KonaKart.

## Prerequisite

It is suggested that you read the document called KKjQueryJSON.pdf before reading this document since it contains information explaining how the widget actually communicates with the KonaKart engine.

## The product widget

The product widget is a widget that can be easily embedded into social networks such as Facebook and blogs by inserting a few lines of JavaScript.



The widget uses live data by communicating with the KonaKart server from the customer's browser using the KonaKart JSON API. As can be seen from the image above, a page may contain multiple widgets. Each widget contains a number of tab folders:

**Image:** Displays a number of product images selectable by clicking on the image thumbnail. The price is shown and you can add the product to the cart.

**Details:** Contains a description of the product and allows you to configure the product by choosing for example the color and size before adding it to the cart. The price is shown and you can add the product to the cart.

**Cart:** This tab displays the products currently in the cart and allows you to manage the cart by modifying the quantity and removing products. The contents of the tab are identical for all widgets within a page. The tab also displays shipping charges and any promotion discounts and fidelity points earned etc. so that the customer has all the information he needs before deciding to checkout. When the checkout button is clicked, the customer is redirected to the KonaKart store to finalize the checkout process.

**Reviews:** If the product has any reviews, this tab will be visible and will show the last 10 reviews that were written.

## ***How to insert a widget***

The process is very simple and involves the addition of a few lines of JavaScript:

```
<div kkProdId='1' kkAffiliateId='86754' class="kk-prod"/>

<script type="text/javascript">
  var url = "http://www.konakart.com/konakart/widgets/prodWidget.min.js";
  document.write(unescape("%3Cscript src=' " + url
    + "' type='text/javascript'%3E%3C/script%3E"));
</script>
```

The div tag is used to insert the widget and includes a number of attributes:

**kkProdId:** This is the numeric id of the product. To include multiple widgets, a number of these divs may be added to the page with different kkProdIds.

**kkAffiliateId:** This optional attribute can be used to add a code to identify the affiliate partner. When a customer starts the checkout process from the widget, the affiliate id is saved in the database as an attribute of the order.

**class:** This has to be set to "kk-prod". It is used by the JavaScript to determine where to insert the widget.

The script tag is used to add the KonaKart JavaScript. The url may be modified to point to the location where the script may be found.

## ***How to edit a widget***

The JavaScript that manages the widget needs to be configured in order to work correctly within your environment. The parameters commonly changed are all at the top of the file as shown on the next page.

The jsonProtocol variable can be either set to "json" or "jsonp". JSONP or "JSON with padding" is a complement to the base JSON data format. It provides a method to request data from a server in a different domain, something prohibited by typical web browsers because of the Same Origin Policy where a web page served from server1.example.com cannot normally connect to or communicate with a server other than server1.example.com.

When using JSONP, the browser does a GET rather than a POST. Some of the API calls send a large amount of data so you may have to modify the configuration of your web server to increase the maximum length of the HTTP GET request.

```

/*
 * Start of configuration parameters
 */
// Change the root depending on where KK is running
// Define the root URL
var kkRoot = 'http://localhost:8780/konakart/';
// var kkRoot = 'http://www.konakart.com/konakart/';

// Define the protocol: json or jsonp
var jsonProtocol = 'jsonp';

// Use the image base to display images from the KK store-front app
var kkImgBase = kkRoot + "images/";

// Use the script base to load the JavaScript
var kkScriptBase = kkRoot + "widgets/";

// Use the css base to load the css
var kkCSSBase = kkRoot + "widgets/";

// Define the libraries
var jqueryLib = 'jquery-1.7.2.min.js';
var jqueryUILib = 'jquery-ui-1.8.20.custom.min.js';
var jqueryJsonLib = 'jquery.json-2.3.js';
var jqueryKCLib = 'jquery.konakart-7.4.0.0.min.js';

// Define styles
var jqueryUICSS = 'jquery-ui-1.8.20.custom.css';
var kkCSS = 'kkWidget.css';

// Messages / Labels
var kkMsgCartEmpty = "Your Cart is Empty";
var kkMsgImage = 'Image';
var kkMsgDetails = 'Details';
var kkMsgCart = 'Cart';
var kkMsgReviews = 'Reviews';
var kkMsgItem = 'Item';
var kkMsgQty = 'Qty';
var kkMsgCost = 'Cost';
var kkMsgRemove = 'Remove';
var kkMsgCheckout = 'Checkout';
var kkMsgAddToCart = 'Add To Cart';
var kkMsgTotal = 'Total';
var kkMsgInc = 'Increase the quantity';
var kkMsgDec = 'Decrease the quantity';
var kkMsgRemove = 'Remove the item from the cart';

/*
 * End of configuration parameters
 */

```

## Packaging

The JavaScript, CSS, html and image files may be found under `../KonaKart/webapps/konakart/shoppingWidgets`. The shopping widget consists of a JavaScript file called `prodWidget-x.x.x.x.js` where `x.x.x.x` is the version of KonaKart that it is compatible with. In order to communicate with the KonaKart engine it makes use of the KonaKart jQuery plugin called `jquery.konakart-x.x.x.x.js`.

## ***How to Enable JSON for the KonaKart Engine***

By default the JSON Server servlet sections in the web.xml file for the konakart webapp are commented out so JSON services are disabled after a standard install. Full instructions for enabling JSON can be found in the KonaKart User Guide.

### ***Example Demo***

A example html file is supplied in order to demonstrate how to embed a widget using JavaScript. The file called kkProdWidget.html and can be found under the /KonaKart/webapps/konakart/shoppingWidgets/html directory. In order to run it after a standard install (and after having enabled the JSON APIs on the engine), you just have to enter this URL in a browser:

<http://localhost:8780/konakart/shoppingWidgets/html/kkProdWidget.html> .

In order to work correctly there must be a running KonaKart system which includes a product with id equivalent to the kkProdId defined in the HTML file.