



KonaKart
Enterprise Java eCommerce

KonaKart Tile Portlets for Liferay

24th January 2018

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

Table of Contents

KonaKart Tile Portlets.....	3
Creation of Portlets.....	3
Step 1 - Enable JSON.....	3
Step 2 - Build the Tiles.....	3
Step 3 - Build the Portlets.....	5
Step 4 - Deploy the portlets in Liferay.....	5
How to use the Portlets.....	6
Creating a Storefront.....	6
Single Portlet Store.....	6
Store created from multiple portlets.....	7
Product Portlet.....	9
Horizontal and Vertical Carousel Portlets.....	10
Customizing the portlet build.....	11

KonaKart Tile Portlets

KonaKart tile portlets are Liferay portlets created from KonaKart tiles. A prerequisite for this document is to read the KonaKart tiles documentation which can be downloaded from http://www.konakart.com/docs/KonaKart_Tiles.pdf. As explained in that document, tiles can be regarded as building blocks for creating an eCommerce application which can be easily integrated into a front end system such as a CMS (Content Management System) or portal. Each tile (or aggregation of similar tiles) has a template, a style sheet and a JavaScript file which control the look and feel as well as the functionality. The tiles communicate with a KonaKart server using the KonaKart JSON APIs called asynchronously from the JavaScript using AJAX.

Before the introduction of tile portlets it was possible to create a single portlet generated from the JSP / Struts2 KonaKart storefront application. By using KonaKart JavaScript tile technology it is now possible to create smaller more manageable portlets which can be added as modules to a standard Liferay 7 / DXP workspace and deployed from there or deployed as OSGi jar files.



Tile portlets are only available for Liferay 7 / DXP and future versions of Liferay that support OSGi.

Creation of Portlets

After an Enterprise installation of KonaKart you will find a directory under KonaKart/custom called `konakart_portlet_tiles` which contains the information required to build the portlets. The following steps must be taken:

Step 1 - Enable JSON

In order for the tiles to send and receive information from the KonaKart engine, you must ensure that the JSON APIs have been enabled. The process for enabling the APIs is explained in detail in the standard User Guide so here we will just give brief instructions.

A convenient way to enable JSON services is to run the *enable_JSON* ANT task provided in the `build.xml` file in the custom directory of the standard installation as follows:

```
C:\Program Files\KonaKart\custom>bin\kkant enable_JSON
Buildfile: build.xml
enable_JSON:
enable_JSON_warning:
enable_JSON_enterprise:
[echo] Fix konakart web.xml to start-up JSON
BUILD SUCCESSFUL
Total time: 0 seconds
```

Instructions for modifying the `web.xml` file manually are in the User Guide.

Step 2 - Build the Tiles

The Liferay portlets use the minimized version of the tiles which have to be created after installation. In order to do this you must run the ANT task provided in the `build.xml` file in the `KonaKart\webapps\konakart_tiles` directory, as follows:

```
C:\KonaKart\webapps\konakart_tiles>..\..\custom\bin\kkant
Buildfile: C:\KonaKart\webapps\konakart_tiles\build.xml

clean:
    [echo] Cleanup...

createGensrc:
    [echo] Creating directories for generated source...
    [mkdir] Created dir: C:\KonaKart\webapps\konakart_tiles\gensrc
    [mkdir] Created dir: C:\KonaKart\webapps\konakart_tiles\gensrc\script
    [mkdir] Created dir: C:\KonaKart\webapps\konakart_tiles\gensrc\html
    [mkdir] Created dir: C:\KonaKart\webapps\konakart_tiles\gensrc\styles
    [mkdir] Created dir: C:\KonaKart\webapps\konakart_tiles\gensrc\images
    [mkdir] Created dir: C:\KonaKart\webapps\konakart_tiles\gensrc\bin
    [mkdir] Created dir: C:\KonaKart\webapps\konakart_tiles\buildlog

compileBuildUtils:
    [echo] Compiling TileBuildUtils
    [javac] Compiling 1 source file to C:\KonaKart\webapps\konakart_tiles\gensrc\bin

minimizeJS:
    [echo] Creating JavaScript file from templates
    [echo] Creating single JavaScript file
    [echo] Minimizing JavaScript

minimizeCSS:
    [echo] Creating single CSS file
    [echo] Minimizing CSS

processHTML:
    [echo] Processing HTML files

copyJS:
    [echo] Copying JS files
    [copy] Copying 8 files to C:\KonaKart\webapps\konakart_tiles\gensrc\script

copyCSS:
    [echo] Copying CSS files
    [copy] Copying 34 files to C:\KonaKart\webapps\konakart_tiles\gensrc\styles

copyImages:
    [echo] Copying image files
    [copy] Copying 33 files to C:\KonaKart\webapps\konakart_tiles\gensrc\images

tidyUp:
    [echo] Tidy Up
    [delete] Deleting: C:\KonaKart\webapps\konakart_tiles\gensrc\script\kk-tile-gen.js
    [delete] Deleting: C:\KonaKart\webapps\konakart_tiles\gensrc\styles\kk-tile-gen.css

build:

BUILD SUCCESSFUL
Total time: 11 seconds
```

Step 3 - Build the Portlets

To build the portlets you must run the ANT task provided in the build.xml file in the KonaKart\custom\konakart_portlet_tiles directory, as follows:

```
C:\KonaKart\custom\konakart_portlet_tiles>..\bin\kkant
Buildfile: C:\KonaKart\custom\konakart_portlet_tiles\build.xml
Trying to override old definition of task javac

debugenv:
  [echo] portlet.tiles.home           = C:\KonaKart\custom\konakart_portlet_tiles
  [echo] konakart.home                = C:\KonaKart\custom\konakart_portlet_tiles\../..
  [echo] host.name                    = my-host
  [echo] defaultPort                  = 8780
  [echo] konakart.version              = 8.7.0.0
  [echo] java.source                  = 1.7
  [echo] java.target                  = 1.7
  [echo] debug_javac                  = off
  [echo] JAVA_HOME                    = C:\jdk1.8.0_121
  [echo] konakart.java.home            = C:\jdk1.8.0_121
  [echo] konakart.Xlint.off            = :-path,-deprecation,-unchecked,-dep-ann
  [echo] konakart.Xlint.on             = :-path,deprecation,unchecked,dep-ann

clean:
  [echo] Cleanup...

compile:
  [echo] Compile the portlet tiles sources
  [mkdir] Created dir: C:\KonaKart\custom\konakart_portlet_tiles\classes
  [javac] Compiling 6 source files to C:\KonaKart\custom\konakart_portlet_tiles\classes
  [javac] warning: [options] bootstrap class path not set in conjunction with -source 1.7
  [javac] 1 warning










create_modules:
  [echo] Create the portlet modules under C:\KonaKart\custom\konakart_portlet_tiles\gensrc

build:

BUILD SUCCESSFUL
Total time: 6 seconds
```

This build creates the portlets in the KonaKart\custom\konakart_portlet_tiles\gensrc directory.

Step 4 - Deploy the portlets in Liferay

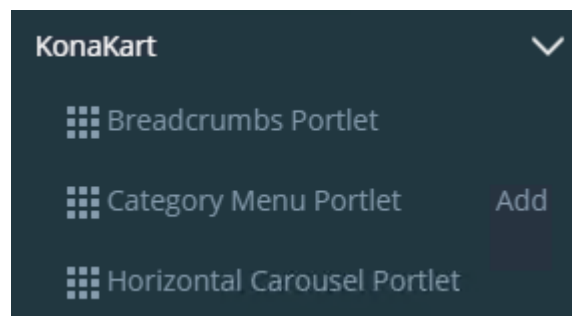
KonaKart > custom > konakart_portlet_tiles > gensrc >		
Name	Date modified	Type
 konakart.breadcrumbs	20-Feb-17 10:29	File folder
 konakart.category.menu	20-Feb-17 10:29	File folder
 konakart.horizontal.carousel	20-Feb-17 10:29	File folder
 konakart.product	20-Feb-17 10:29	File folder
 konakart.search	20-Feb-17 10:29	File folder
 konakart.store	20-Feb-17 10:29	File folder
 konakart.store.body	20-Feb-17 10:29	File folder
 konakart.top.menu	20-Feb-17 10:29	File folder
 konakart.vertical.carousel	20-Feb-17 10:29	File folder

As can be seen from the above screen shot, each portlet consists of a directory structure which can be added to the modules directory of a Liferay workspace. The command “blade deploy” from the modules directory will build and deploy all of the portlets.

As an alternative you can simply copy the jar files for each module directly into the deploy directory of your Liferay installation.

How to use the Portlets

By clicking on Add >> Applications from within Liferay, you'll see a KonaKart section that contains all of the deployed portlets. The portlets can be added to your page by dragging them to the desired position.



Creating a Storefront

With the portlets provided in the download, there are two ways of creating a complete storefront.

Single Portlet Store

The portlet called “Store Single Portlet”, implements a complete storefront in a single portlet. The store is generated dynamically using a selection of KonaKart JavaScript tiles. An example of this portlet can be seen below:

KonaKart

[Computer Peripherals](#)[Games](#)[DVD Movies](#)[Software](#)[Electronics](#)[Home & Garden](#)[Gifts](#)

All-New Kindle fire HD

The worlds most advanced tablet

Stunning HD display
Exclusive Dolby audio
Fastest Wi-Fi

\$199.00



HARDWARE PRODUCTS



Matrox G200 MMS
★★★★★ (0 Reviews)
\$299.99



Matrox G400 32MB
★★★★★ (0 Reviews)
\$499.99

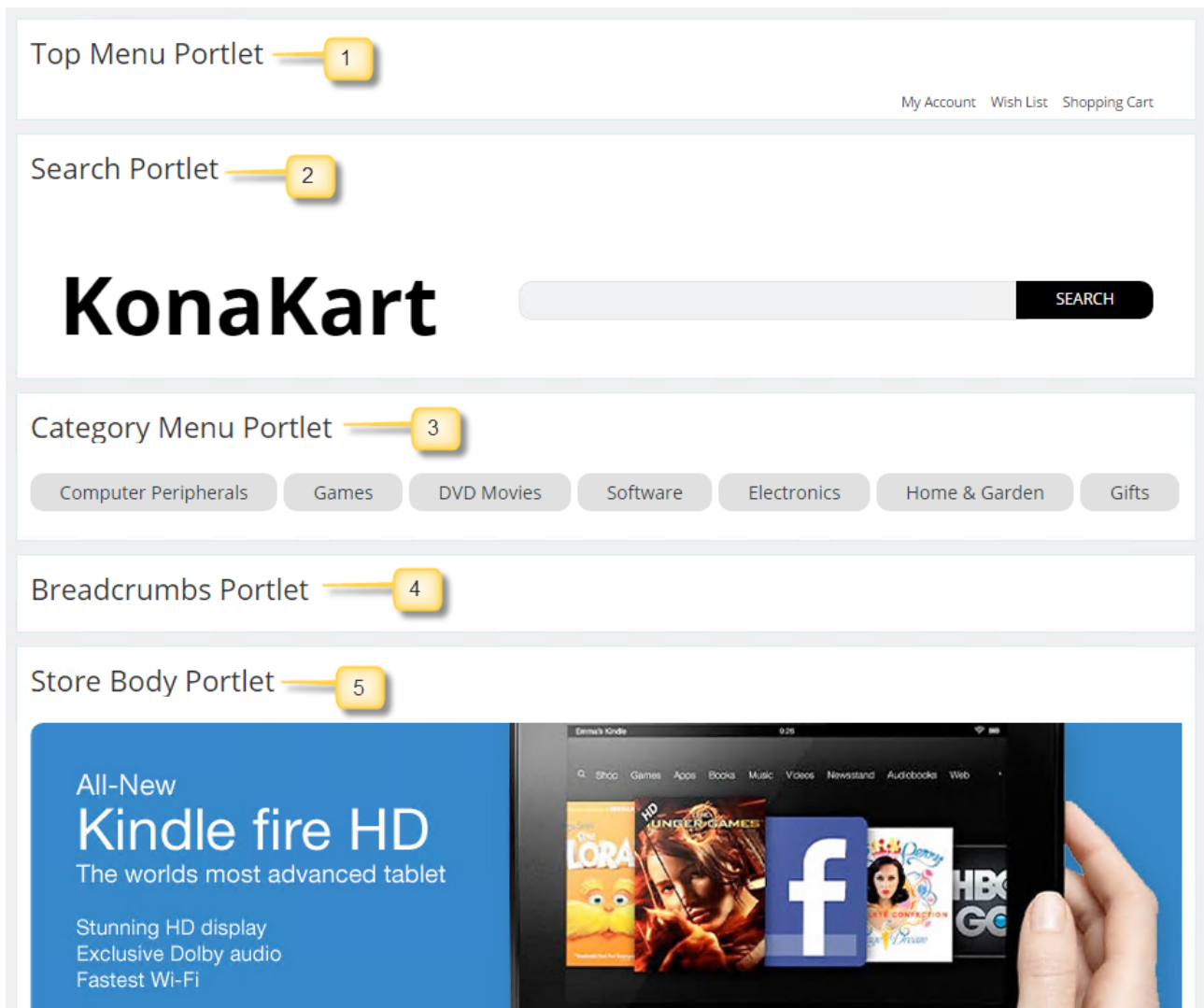


Microsoft
Microsoft IntelliMouse Pro
★★★★★ (0 Reviews)
~~\$49.99~~ \$39.99
Save \$10.00



Microsoft Internet Keyboard PS/2
★★★★★ (0 Reviews)
\$69.99

Store created from multiple portlets



As can be seen from the above image, to create a similar storefront using multiple portlets you should install the following portlets in this order:

1. Top Menu Portlet
2. Search Portlet
3. Category Menu Portlet
4. Breadcrumbs Portlet
5. Store Body Portlet

In both cases (single and multi-portlet stores), if you are logged into Liferay, then using an SSO mechanism you will be automatically logged into the storefront when clicking on "My Account" or when attempting to checkout. Note that in order for this to work you have to have installed the Liferay Login module using the KonaKart Admin App as shown below.

My Store Status

Configuration

Products

Modules

Payment

Shipping

Order Totals

Other Modules

Customers

Orders

Marketing

Locations/Taxes

Localizations

Tools

Reports

Scheduler

Audit Data

Licensing

Custom

Other Modules

Module Name

Sort Order

Facebook Login	
Google+ Login	
Liferay Login	0
LDAP	
USPS Address Validation	

INSTALL

REMOVE

Liferay Login Status

☒ true
☐ false

Sort order of display

0

Liferay Login Module Class Name

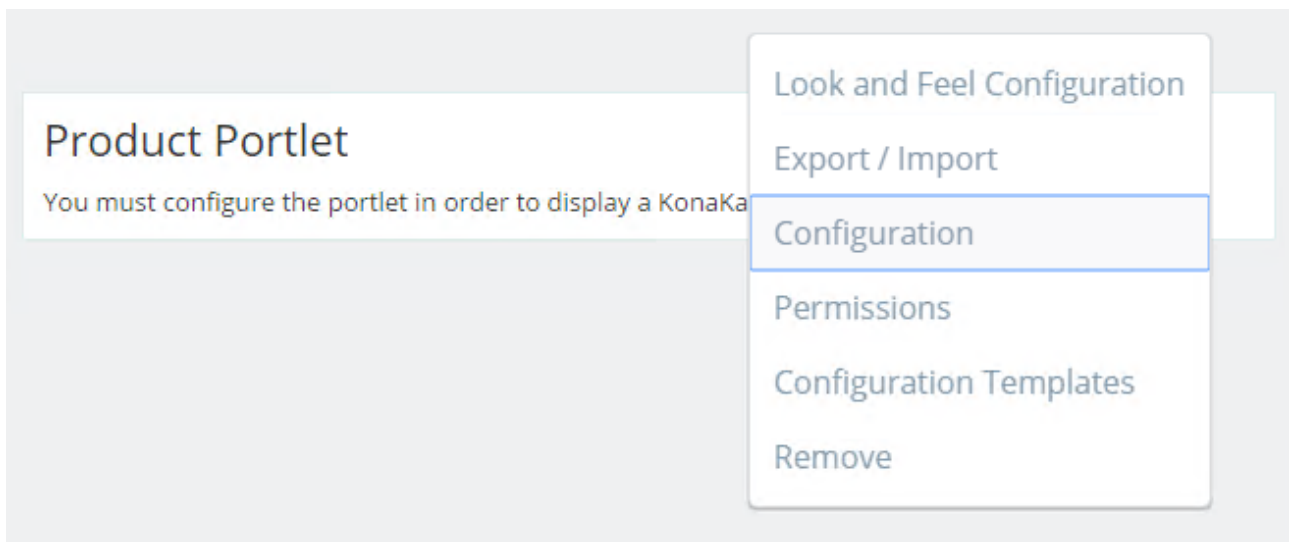
com.konakart.bl.modules.others.liferaylogin.LiferayLog

SAVE

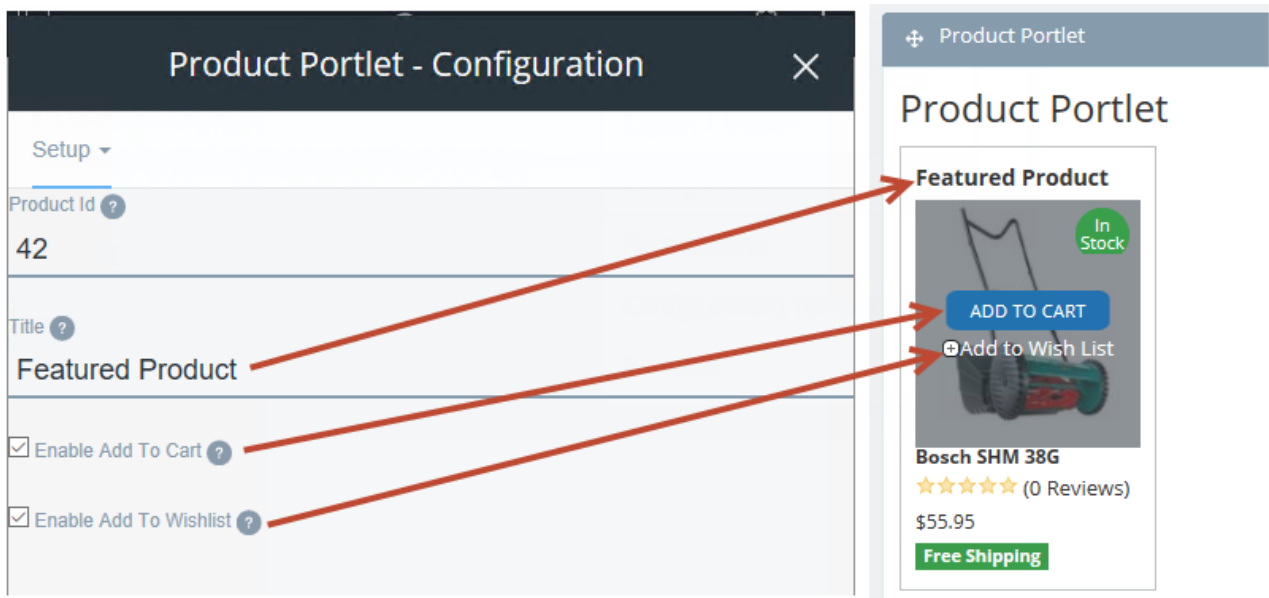
CANCEL

Product Portlet

The Product Portlet allows you to display a single product tile with Add to Cart and Add to Wishlist commands. The portlet must be configured after installation to choose the product id of the product that needs to be displayed, the title and whether to enable the Add to Cart and / or Add to Wishlist commands.



The image below displays the configuration parameters (on left) and the portlet produced (on right) that displays the product id = 42 (demo storefront) with the title “Featured Product” and active commands.



Horizontal and Vertical Carousel Portlets

In a similar fashion to the Product Portlet, the carousel portlets must be configured to select the category from which the products will be selected, the title of the carousel and the maximum number of products.

The image shows a 'Horizontal Carousel Portlet - Configuration' window. It has two tabs: 'Setup' and 'Sharing'. The 'Setup' tab is selected. It contains three input fields: 'Category Id' with the value '24', 'Title' with the value 'For your Home and Garden', and 'Limit' with the value '15'. Each field has a question mark icon to its right.

The above configuration creates the horizontal carousel portlet below:

Horizontal Carousel Portlet

FOR YOUR HOME AND GARDEN



De'Longhi BCO 410
★★★★★ (4 Reviews)
\$159.99



Cadiz Clock
★★★★★ (3 Reviews)
\$19.99



Milan 93 Clock
★★★★★ (0 Reviews)
\$12.99



Acctim Metal Clock
★★★★★ (0 Reviews)
\$21.99

Customizing the portlet build

Part of the build is performed by a Java class called `LiferayBuildTiles.java` found under `KonaKart\custom\konakart_portlet_tiles\src\com\dsdata\util`. This class is called from the ANT task defined in `build.xml` under `KonaKart\custom\konakart_portlet_tiles`.

From the Java source you can see that the class accepts a set of parameters, the first four of which are mandatory.

Usage: `LiferayBuildTiles`

- i portletTilesBaseDir
- i2 tilesBaseDir
- i3 tileGensrcURL
- i4 KKVersion
- i5 tileSrcURL
- i6 debug (true or false)
- i7 genSourceCopyDir

First 4 parameters are mandatory

At run time the portlets use JavaScript which is at the location defined by `i3 - tileGensrcURL`. The build sets this to the URL of the current installation. You may wish to modify this parameter to point to a different KonaKart installation.

By default `i6` is set to false. If set to true then the `tileSrcURL` is used instead of the `tileGensrcURL`. This is useful for debugging the portlets because the KonaKart tile JavaScript is not minimized.

Tiles are considered to be building blocks that can be easily customized by modifying the tile templates and JavaScript code. It's also possible for you to create your own tiles to match your eCommerce requirements. In order to create a portlet from your own tile, you need to add it to `LiferayBuildTiles.java`.

```

/**
 * Defines an array of liferay portlets to be created. The information required for each portlet
 * is the following:
 * <ul>
 * <li>File Name</li>
 * <li>Directory Name which is also the name of the generated jar file once the version has been
 * added</li>
 * <li>Friendly Name</li>
 * <li>Java Class Name</li>
 * </ul>
 */
private static String[][] tileInfoArray = new String[][]
{
    new String[]
    { "breadcrumbs.html", "konakart.breadcrumbs", "Breadcrumbs Portlet", "KKBreadcrumbs" },
    new String[]
    { "categoryMenu.html", "konakart.category.menu", "Category Menu Portlet",
      "KKCategoryMenu" },
    new String[]
    { "horizontalCarousel.html", "konakart.horizontal.carousel",
      "Horizontal Carousel Portlet", "KKHorizontalCarousel" },

```

Each tile included in the build has an entry in a static array as shown above. In order to add your own tile, you need to add an entry to the array that has the Java class name of your tile along with the other information described in the Javadoc.