KonaKart
Enterprise Java eCommerce

KonaKartAdmin JavaScript Tiles

11th August 2016

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

Introduction

What are tiles?

KonaKartAdmin tiles are designed to enable you to easily integrate KonaKart administration functionality into any front end system such as the administration section of a content management system. Each tile 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 Admin server using the KonaKartAdmin JSON APIs called asynchronously from the JavaScript using AJAX.

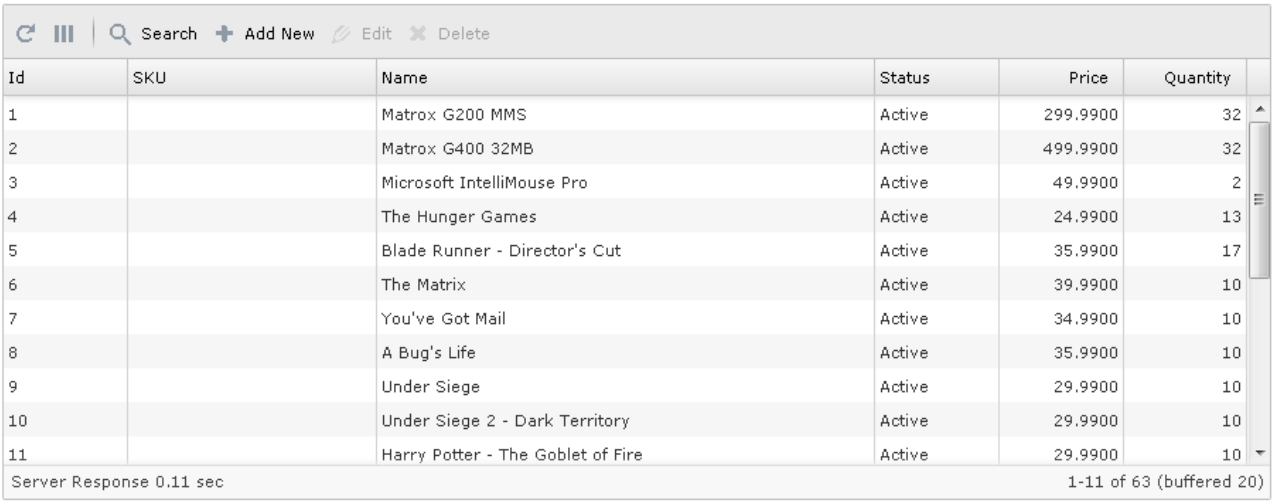
What's the purpose of administration tiles?

The standard KonaKart Administration application is designed to provide all of the functionality required to configure and maintain any KonaKart system. Although it's powerful, it may provide some irrelevant functionality for your business requirements which makes the UI more complicated to learn and to use, especially for a non technical user.

KonaKartAdmin tiles enable you to create a UI widget with business specific functionality and to easily integrate it into an existing system. For example your requirement may be for administrators to be able to easily and quickly change the price of an SKU from within the CMS or Liferay portal. A tile will allow you to do this in the preferred way of your business users in order to maximize productivity. You could achieve this directly using the APIs, but tiles greatly simplify this process because they provide an integration point at a higher level than the pure APIs. They provide functional widgets that already have a template based UI design and that autonomously capture events and communicate with the KonaKart Admin server.

How can tiles be used?

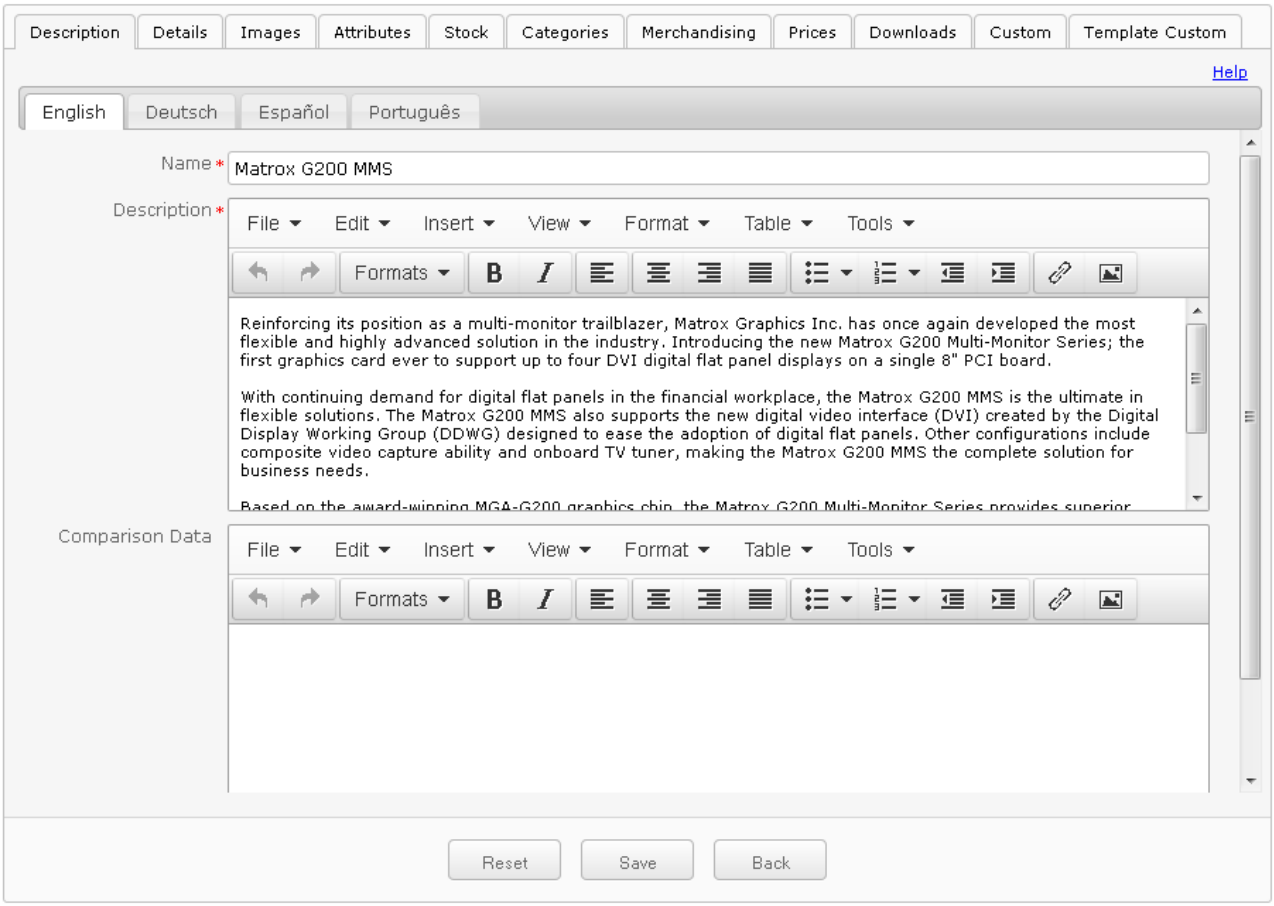
In the case of a CMS, the tiles may be placed anywhere on a page in order to provide seamless integration with the other administration functionality on the page. For example, you could integrate a grid (see below) allowing you to search, filter, select, edit and delete products without forcing an administrator to open and log into the KonaKart Admin app



Architecture

The tiles are designed so that they can be used independently allowing you to integrate only the eCommerce administration functionality that you require anywhere on the page. Each tile may be positioned separately.

The tiles delivered out of the box allow you to manage products, customers and orders and new tiles are continually being developed.



Each tile has an underscore template (http://underscorejs.org/) that allows you to easily modify the layout of the tile without touching any of the JavaScript code. It also has a CSS file containing the styles and a JavaScript file that controls the rendering and the functionality, so that whenever a button is clicked, an action is performed and if present, other tiles are invoked.

The idea behind the open architecture and code is to allow you to modify the existing tiles to match your requirements and to develop new tiles. This type of development work is also offered as a service by KonaKart.

Installation

KonaKartAdmin tiles are automatically installed during the installation of the Enterprise Extensions. A konakartadmin_tiles webapp is installed under the webapps directory. The webapp contains the following directories:

html directory

This directory contains an HTML file (login.html) that invokes the login tile.

```
<body>
    <script>
    $(function() {
            kka.setLoginTileDivId("kka-body-tile");
            kka.setLogoutLinkDivId("kka-logout-link");
            kka.setProductsTileDivId("kka-body-tile");
            kka.setEditProductTileDivId("kka-body-tile");
            kka.setCustomersTileDivId("kka-body-tile");
            kka.setEditCustomerTileDivId("kka-body-tile");
            kka.setOrdersTileDivId("kka-body-tile");
            kka.setEditOrderTileDivId("kka-body-tile");
            kka.renderLoginTile();
    });
    </script>
    <div id="kka-page-container">
    <div id="kka-page">
        <div id="kka-top-bar-container">
            <div id="kka-top-bar">
                <div id="kka-options-container">
                    <div id="kka-options">
                        <a id="kka-logout-link"></a>
                    </div>
                </div>
            </div>
        </div>
        <div id="kka-body-tile"></div>
    </div>
    </div>
</body>
```

The script contained in the HTML file may be seen above. It defines the positions of the various tiles and then calls the function to render the login tile (kka.renderLoginTile()). This file is provided as an example for running the tiles as an independent application.

There are other examples (admin_app_products.html, admin_app_customers.html etc.) which are used to integrate the tiles into the standard KonaKart Admin App. After an installation of the Enterprise Extensions you can see these in action under the menu items:

- Products >> Product Tile Example
- Customers >> Customer Tile Example
- Orders >> Order Tile Example

In this case there is no login panel because the tiles use the session id being used by the Admin App. The language is also picked up from the Admin App although the tiles only contain a partial Spanish translation in order to demonstrate the functionality. This is a good example to follow in order to integrate the tiles into your own administration application. Note that the Enterprise installation does not enable the Admin JSON APIs. You must do this before using the Admin Tile Example from the Admin App.

html/template directory

This directory contains the tile underscore templates. More information regarding the templates may be found at the following link http://underscorejs.org/#template . Each tile has a template which is used to render the tile. The template is compiled and at render time, it is passed data which it may use to populate the generated HTML. These templates may be modified in order to customize the generated look and feel of each tile.

styles directory

This directory contains a number of CSS files that provide style information for the tiles. If the template of a tile is modified to include a new style, then this new style should be included in the appropriate CSS file. Some of the CSS files provide styles for more than one tile.

script directory

The script directory contains the JavaScript files that control the individual tiles. Each file may control one or more tiles. Amongst other things, the JavaScript is responsible for creating and populating the tiles with data from the KonaKartAdmin engine and also for managing events such as button clicks once the tiles have been rendered.

gensrc directory

All content within the gensrc directory is dynamically created by running the ant build file (build.xml) directly under the konakartadmin_tiles directory. It is also populated automatically after an Enterprise Installation. This directory contains script, html and styles sub directories although in this case all of the CSS and JavaScript files have been compressed and minimized and the templates have been added to the minimized JavaScript file. The files under gensrc are typically what you would use in a production environment.

Eclipse Project

The webapp includes a .classpath file and .project file so that it can easily be imported into Eclipse for development.

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_KKAdmin_JSON* ANT task provided in the build.xml file in the custom directory of the standard installation as follows:

```
C:\Program Files (x86)\KonaKart\custom>bin\kkant enable_KKAdmin_JSON
Buildfile: C:\Program Files (x86)\KonaKart\custom\build.xml

enable_KKAdmin_JSON:
enable_KKAdmin_JSON_warning:
enable_KKAdmin_JSON_enterprise:
     [echo] Fix konakartadmin web.xml to start-up KKAdmin JSON

BUILD SUCCESSFUL
Total time: 0 seconds
```

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

Try a tile

The URL for this the login tile after a standard install on port 8780, is

http://localhost:8780/konakartadmin_tiles/html/login.html . The same tile using the minified JavaScript and CSS can be found at http://localhost:8780/konakartadmin_tiles/gensrc/html/login.html .

Configuration

The JavaScript file used to configure the tiles is called kka-configure.js. The configuration variables are annotated and hopefully easy to understand.

```
/*
 * The root URL of the KonaKart admin server engine.
 * The value can be changed depending on where KonaKart is running.
 */
var kkaRoot = 'http://localhost:8780/konakartadmin/';

/*
 * Define the configuration object used to instantiate a KonaKart admin engine
 * which is used to make the JSON calls to the server engine. The mode in which
 * the server engine is running is defined by the properties file of that
 * engine.
 */
var KKA_MODE_SINGLE_STORE = 0;
var KKA_MODE_MULTI_STORE_NON_SHARED_DB = 1;
var KKA_MODE_MULTI_STORE_SHARED_DB = 2;

var kkaConf = new adminEngineConfig(kkaRoot + 'konakartadminjson');
kkaConf.storeId = "store1";
kkaConf.protocol = 'jsonp';
kkaConf.mode = KKA_MODE_SINGLE_STORE;
kkaConf.customersShared = true;
kkaConf.productsShared = false;
kkaConf.categoriesShared = false;
```

As can be seen above, you can configure the location of the KonaKart Admin engine where the JSON requests are sent and other parameters such as the mode in which the admin engine is running, the default locale and default currency etc. When running in multi-store mode the login tile allows you to choose the store for which to login. The Enterprise installer sets the correct engine mode.

There is also a section for configuring the number of images and their sizes that are created on the server when a product image is uploaded.

Development

As mentioned previously the konakartadmin_tiles webapp includes a .classpath file and .project file so that it can easily be imported into Eclipse for development.

Most of the JavaScript files control one or more tiles. At the top of each file there is a short description informing you which tiles it controls. e.g. for kka-editProductTile.js the comment is:

```
/**
 * JavaScript for the edit product tile using the template
 * adminEditProductTile.html
 */
```

Some exceptions are:

kka-tile.js

This contains code that is common to all tiles. The utility methods include methods to get a template, to manage the session, to retrieve messages from the message catalog and to create a common template context which is the data passed to all templates when they generate the tile HTML.

kka-formTiles.js

This file contains utility methods related to forms. It is used by the tiles that contain forms.

kka-configure.js

This file contains configuration variables that may be modified to configure your system.

I18N

Messages

All messages are retrieved from a JavaScript message catalog under script/i18n. The name of a message catalog is in the format kka-{locale}.js so for example could be kka-en_GB.js or kka-es_ES.js . The message catalog also defines the format for dates and time. The messages are grouped by tile.

```
// Locale of message catalog must set as the index of the array
kkaMsgMap["en_GB"] = {
    // Common messages for all tiles
    "exception.render.tile" : "%{0} must be called before calling %{1}",
    // Date format for displaying dates
    "dateformat.date.format" : "dd/mm/yyyy",
    // Date format for selecting and entering dates using the date picker
    "datepicker.date.format" : "dd/mm/yyyy",
    "datepicker.shortmonths"       : ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
```

A typical message catalog uses the format shown above. The locale of the catalog must be entered in the first line which could for example be kkaMsgMap[“es_ES”] for Spanish.

The default locale is set in kka-configure.js and there must be a matching message catalog for this locale. If a new language is added, the JavaScript file containing the message catalog must be added to login.html. e.g.

```
<script type="text/javascript" src="../script/i18n/kka-es_ES.js"></script>
```

KonaKart uses the Polyglot JavaScript library (http://airbnb.github.io/polyglot.js/) to manage the message

catalogs.

KonaKart Admin Engine APIs

The JavaScript calls to the KonaKart Admin engine are identical to the standard KKAdminIf API calls. The Javadoc for these API calls may be found at http://www.konakart.com/javadoc/admin/com/konakartadmin/appif/KKAdminIf.html . When called from JavaScript they are asynchronous and use a callback function to return results.

Production

In development mode the tiles include many JavaScript files and many CSS files. Also the templates are read as files from disk when a tile is rendered.

Once you have finished customizing and developing your own tiles you can run an Ant task that performs the following:

1. Creates a gensrc directory under konakartadmin_tiles with all files required for production.
2. Creates a single minimized JavaScript file for all of the tile JavaScript files. All of the templates and message catalogs are included in this single file called kka-tile-gen.min.js.
3. Creates a single minimized CSS file for all of the tile CSS files. It is called kka-tile-gen.min.css.
4. When copied over to the gensrc area, all of the tile HTML files are modified to remove the old includes and include the new minimized includes.

```
C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles>..\..\custom\bin\kkant
Buildfile: C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\build.xml

clean:
     [echo] Cleanup...

createGensrc:
     [echo] Creating directories for generated source...
    [mkdir] Created dir: C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc
    [mkdir] Created dir: C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\script
    [mkdir] Created dir: C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\html
    [mkdir] Created dir: C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\styles
    [mkdir] Created dir: C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\images
    [mkdir] Created dir: C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\bin
    [mkdir] Created dir: C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\tinymce

compileBuildUtils:
     [echo] Compiling TileBuildUtils
    [javac] Compiling 1 source file to C:\Program Files (x86)\KonaKart\webapps\konakartadmin_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 6 files to C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\script
     [copy] Copying 81 files to C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\tinymce

copyCSS:
     [echo] Copying CSS files
     [copy] Copying 7 files to C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\styles

copyImages:
     [echo] Copying image files

tidyUp:
     [echo] Tidy Up
   [delete] Deleting: C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\script\kka-template-
gen.js
   [delete] Deleting: C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\script\kka-tile-gen.js
   [delete] Deleting: C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\styles\kka-tile-gen.css
   [delete] Deleting directory C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles\gensrc\bin

build:

BUILD SUCCESSFUL
Total time: 3 seconds

C:\Program Files (x86)\KonaKart\webapps\konakartadmin_tiles>
```

Part of the build is performed by a Java class called TileBuildUtils.java found under konakartadmin_tiles/src/com/dsdata/util . It has several methods called by Ant during the build. The

JavaScript and CSS files that are added to the minimized files are defined in Static variables within TileBuildUtils.java as shown below:

```
/*
 * Update to add all javaScript files that should be added to the minimized Konakartadmin
 * javaScript file. Don't remove ../gensrc/script/kka-template-gen.js since this is a generated
 * file containing all of the templates.
 */
private static final String[] jsFileArray = new String[]
{ "jquery.konakartadmin.js", "../gensrc/script/kka-template-gen.js", "kk-dateFormat.js",
        "kka-configure.js", "kka-tile.js", "i18n/kka-en_GB.js", "i18n/kka-es_ES.js",
        "kka-formTiles.js", "kka-w2ui-1.4.2.js", "kka-loginTile.js", "kka-productsTile.js",
        "kka-customersTile.js", "kka-ordersTile.js", "kka-editProductTile.js",
        "kka-editCustomerTile.js", "kka-editOrderTile.js" };

/*
 * Update to add all css files that should be added to the minimized KonaKart css file
 */
private static final String[] cssFileArray = new String[]
{ "kka-tile.css", "kka-w2ui-1.4.2.css", "kka-formTiles.css", "kka-loginTile.css",
        "kka-productsTile.css", "kka-customersTile.css", "kka-ordersTile.css",
        "kka-editProductTile.css", "kka-editCustomerTile.css", "kka-editOrderTile.css" };
```

You may modify the included files depending on how many tiles you are using and whether you have created any of your own tiles or added new message catalogs etc.