Building a Flickr widget with Mobile Web Templates

This article shows the **building process of a Flickr photo search Web Runtime widget**, using the Templates for smartphones for the widget's User Interface.

Introduction

Flickr is a **photo sharing website**, that allows users to upload and share their favorite photos with other users.

Flickr REST API

Flickr offers a REST API, that can be used to build **external applications and clients** that access and use Flickr data in order to add, modify or retrieve uploaded photos.

The Flickr widget built in this article will use this REST API to perform the following actions:

- **search for photos**
- **get the latest uploaded photos**
- **retrieve details of an user**
- **get the comments for a single photo**

Mobile Web Templates

Templates for smartphones are a set of customizable templates, composed by **HTML, CSS and graphic resources**, built in order to display in an optimal manner on mobile devices' browsers.

This article uses the Templates for smartphones since Web Runtime widgets are currently compatible with S60 5th edition and S60 3rd edition FP2 devices (note: also some S60 3rd edition FP1 devices, via firmware upgrades, support Web Runtime widgets).

Setup the widget

This section shows the **initial setup of the widget**, describing the necessary resources and code to be added in order to properly **integrate the Mobile Web Templates**.

Copy the template resources

First step is to **create an empty widget**, and add the necessary **resources from Mobile Web Templates for High-End devices**.

These resources are contained into the **"resources/" folder** of the template, and are divided into these sub-folders:

- **images**: contains all the images used for the various interface components (e.g.: buttons, lists)
- **scripts**: contains the JavaScript code used to implement the dynamic user interface components (Slideshow, Accordion List, Toggle Switch) and to dynamically load CSS tweaks for specific platforms
- **styles**: contains the **main two CSS files** used from the template, and **a set of tweak CSS files** contained into the 'tweaks/' subfolder

These 3 folders can be copied into the widget main folder, in order to make them available to the widget itself. About the tweaks CSS files, it is enough to copy (and so to use) the **tweaks for S60 3rd edition FP2 and for 5th edition devices** (S603rdFP2.css and S605th.css, respectively).

Basic HTML code

Now that resources have been copied, the **basic HTML structure of the widget** can be defined, and resources appropriately included.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
  <title>Flickr - Mobile Web Templates Sample</title>

  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

  <link href="styles/reset.css" rel="stylesheet" type="text/css" />
  <link href="styles/baseStyles.css" rel="stylesheet" type="text/css" />

  <script type="text/javascript" src="scripts/main.js"></script>

 </head>
 <body onload="init()">
  <div id="wrap">
   <div id="header">
   </div>

   <h1>WRT Flickr</h1>

   <div id="content">


    <div id="some_screen" class="screen">
```


(C) Copyright Nokia 2013. All rights reserved.

```
      <!-- here the screen content goes -->
     </div>

     <!-- other screens go here -->


    </div>

    <div id="footer" class="ft-large">
     <p>&copy; 2009 Nokia Developer - &copy; 2009 Jappit.com</p>
    </div>
   </div>
  </body>
 </html>
```

The CSS files from the Mobile Web Templates, placed into the 'styles/' folder, are included with the appropriate LINK tags.

The widget **HTML structure** is composed by the following elements:

- a **header DIV**, defined by the "header" id
- a **title**, defined by the "h1" tag
- the **main content**, defined by the "content" id, that will contain all the widget views (defined by the "screen" CSS class)
- a **footer**, defined by the "footer" id

Then, in the main.js script, let's add the following init() function, that adds the necessary CSS tweaks:

```
function init()
{
 //init style tweaks

 var myStyleTweaks = new StyleTweaker();
 myStyleTweaks.add("Series60/5.0", "styles/tweaks/S605th.css");
 myStyleTweaks.add("Series60/3.2", "styles/tweaks/S603rdFP2.css");
 myStyleTweaks.tweak();
}
```

The widget's views

This section shows and described the different views of the widget, explaining how the Mobile Web Templates components were used.

Splash Screen

The first widget's view is represented by the splash screen, that **appears for some seconds and is then automatically dismissed**, bringing the user to the widget's main view. The splash screen HTML code is rather simple, as it's composed by a single <IMG> tag:

```
<div id="splash_screen" class="screen">
 <img id="splash" src="images/splash.png" />
</div>
```

The splash screen view is visible below.

Main Menu

The main menu is the **main widget's view**, and contains buttons that bring to the other widget's views. In this case, the views linked by the main view are:

- **Search for photos**

(C) Copyright Nokia 2013. All rights reserved.

- **View latest photos**
- **View favorite photos**
- **View widget's credits**

To create these 4 buttons, the **list component from Mobile Web Templates** is used, with the following HTML code:

```
<ul class="list">
 <li><a onclick="gotoScreen('search_screen')" href="#">Search</a></li>
 <li><a onclick="gotoLatestPhotos()" href="#">Latest</a></li>
 <li><a onclick="gotoFavoritePhotos()" href="#">Favorites</a></li>
 <li><a onclick="gotoScreen('credits_screen')" href="#">Credits</a></li>
</ul>
```

The main menu is visible below.

Search Form

The search form is the view that allow users to **search for specific photos**, by choosing the search criteria. In order to give users a good set of options to find the desired photos, **without using an overly populated form**, these fields will be used:

- a text field, to allow users to freely enter a **search term**
- a multi-option field, to choose which **kind of images** to look for (photos, screen-shots, other or all types)
- a multi-option field that allows to choose **how to sort results** (by newest, oldest, most interesting or most relevant)
- and a toggle that allows to **enable or disable safe search**, to discard unsafe results

The toggle input is implemented by using the **Toggle component from Mobile Web Templates**. This allows for a better look-n-feel of the search form, and for an **improved widget's usability**, given the large touchable area that the component offers.

```
<ul class="list-toggle stack">
 <li>
  <ul class="toggle">
   <li>
    <label>Safe search</label>
    <input id="safe_search" type="checkbox" name="safe_search" checked="checked" />
   </li>
  </ul>
 </li>
</ul>
```

The Toggle component also needs to be **initialized with this line of JavaScript code**:

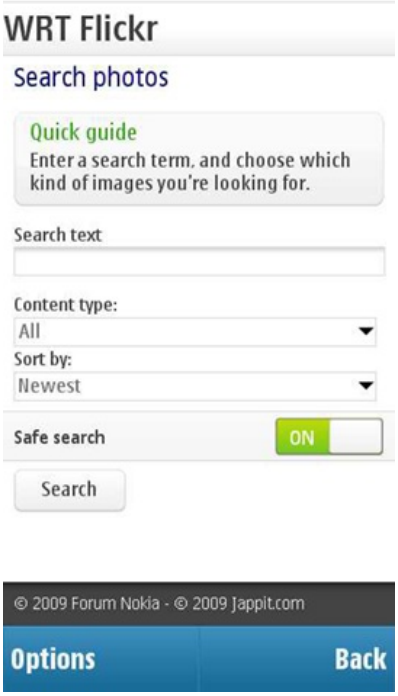
```
var myToggle = new ToggleSwitch(["safe_search"]);
```

The form submit button is styled by using the **rounded form submit button from Mobile Web Templates**, as shown in the following code snippet:

```
<button onclick="searchPhotos();return false;" value="submit" class="button-submit"><span>Search</span></button>
```

The search form view is visible below.

(C) Copyright Nokia 2013. All rights reserved.



Search Results

The search result is the view where the photos, found by using the search form, are presented. Here, it is possible to use **two different components** from the Mobile Web Templates, in order to **give users the ability to choose for their favorite results' visualization**.

Results' list

The first visualization is a typical **list that presents results showing the image thumbnail and its title**. Here, the **list with background image component from Mobile Web Templates** is used, as shown in the snippet below:

```
<ul class="list-graphical-inline stack">
 <li>
  <a onclick="viewPhotoDetail(0)" href="#">
   <div class="flickr_list_thumb"
   style="background-image: url(http://farm3.static.flickr.com/2553/4046017259_54a6ca3f78_t.jpg);" alt="image">
   </div>Devon Haskell looking relaxed.
  </a>
 </li>
 <li>
  <a onclick="viewPhotoDetail(1)" href="#">
   <div class="flickr_list_thumb"
   style="background-image: url(http://farm3.static.flickr.com/2455/4046017319_52bc556d1c_t.jpg);" alt="image">
   </div>photo
  </a>
 </li>
</ul>
```

The results' list visualization is visible below.

Results' grid

The **second visualization** consists in a **grid of images**: it shown only the image thumbnail, without its title, allowing users to have a **global overview of the images found**. Here, the **grid component from Mobile Web Templates** is used, as shown by the following code snippet:

```
<ul class="grid">
 <li>
  <a onclick="viewPhotoDetail(0)" href="#">
   <img class="flickr_grid_thumb" src="images/empty40x40.png"
   style="background-image: url(http://farm3.static.flickr.com/2553/4046017259_54a6ca3f78_t.jpg);" alt="">
  </a>
 </li>
 <li>
  <a onclick="viewPhotoDetail(1)" href="#">
   <img class="flickr_grid_thumb" src="images/empty40x40.png"
```


(C) Copyright Nokia 2013. All rights reserved.

```
    style="background-image: url(http://farm3.static.flickr.com/2455/4046017319_52bc556d1c_t.jpg);" alt="">
   </a>
  </li>
 </ul>
```

The grid visualization is visible below.

Visualization switch

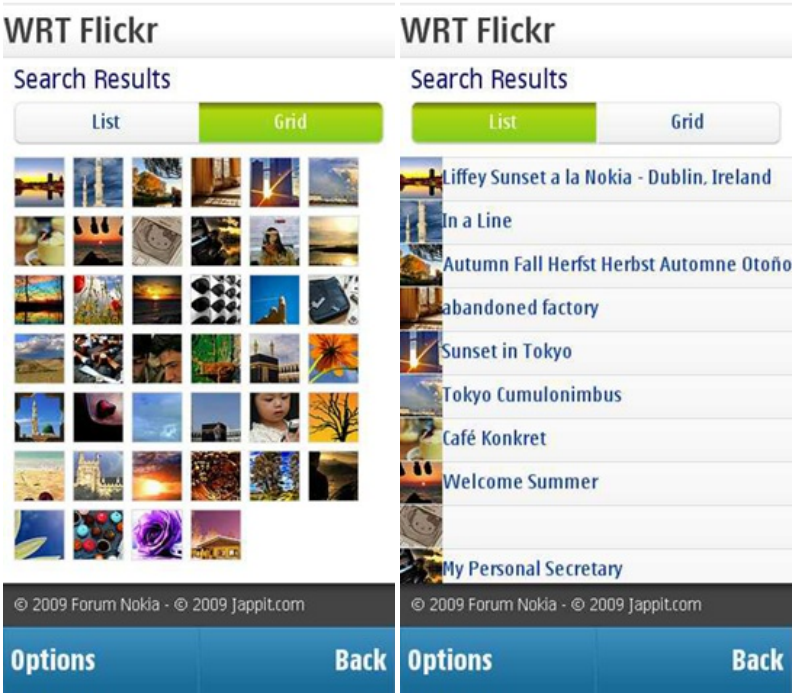
In order to allow users to **switch from a visualization to another**, the **horizontal button bar with rounded corners** is used:

```
<ul class="nav-horizontal-rounded stack">
 <li class="first two-piece"><a onclick="showResultsList()" href="#">List</a></li>
 <li class="last two-piece"><a onclick="showResultsGrid()" href="#">Grid</a></li>
</ul>
```

The button bar is visible below.

Complete view

The final, **complete search results view** is visible below.



Single photo

The photo detail is the view where a single photo is presented, along with details of the photo itself, and with commands that allow to:

- **navigate to the next and previous photos**
- **add or remove the photo from the favorite list**

The photo image

Again, Mobile Web Templates offer a **ready-to-use component: the Slideshow**, that can be easily used for the widget's purposes, and **automatically adds the navigation functionality without any programming efforts**. The HTML snippet below shows how the Slideshow in inserted in the widget's HTML code:

```
<div id="slideshow" class="slideshow">
 <a class="preview">
  <img src="images/photos/large/loading.png" alt="loading" />
  <span></span>
 </a>
```

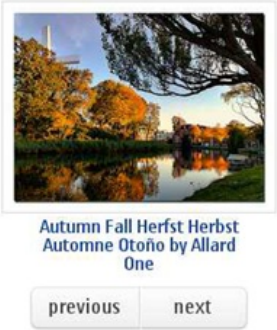

(C) Copyright Nokia 2013. All rights reserved.

```
  <ul class="controls nav-horizontal-rounded stack">
   <li class="first two-piece"><a class="previous">previous</a></li>
   <li class="last two-piece"><a class="next">next</a></li>
  </ul>
  <ol class="data" id="slideshow_list">
   <!-- images will be inserted dynamically -->
  </ol>

 </div>
```

The "data" part of the slideshow is left empty: this because the **images will be dynamically added** when needed (e.g.: when the user performs a search). As for the Toggle component, also the **Slideshow needs to be initialized via JavaScript**, and this can be easily done with this line of code:

```
var mySlideshow = new Slideshow("slideshow", photoIndex, true);
```

The photoIndex variable will be used to let the slideshow start from the photo selected by the user.



The photo details

Each photo has a set of associated details that can be visualized within the photo detail view. These details can be grouped in the 4 following groups:

- **tags** associated with the photo
- date and **details of the original photo** (e.g.: original resolution)
- **details about the user** that has uploaded the photo
- **comments** that other users made about the photo

This data can take up some space when represented on the display of a mobile device, so an optimal choice would be to use the **Accordion component shipped with Mobile Web Templates**, that allow users to **quickly visualize the main 4 details' groups**, and to choose which one to expand or collapse. The following code shows how the Accordion is integrated in the widget's HTML code:

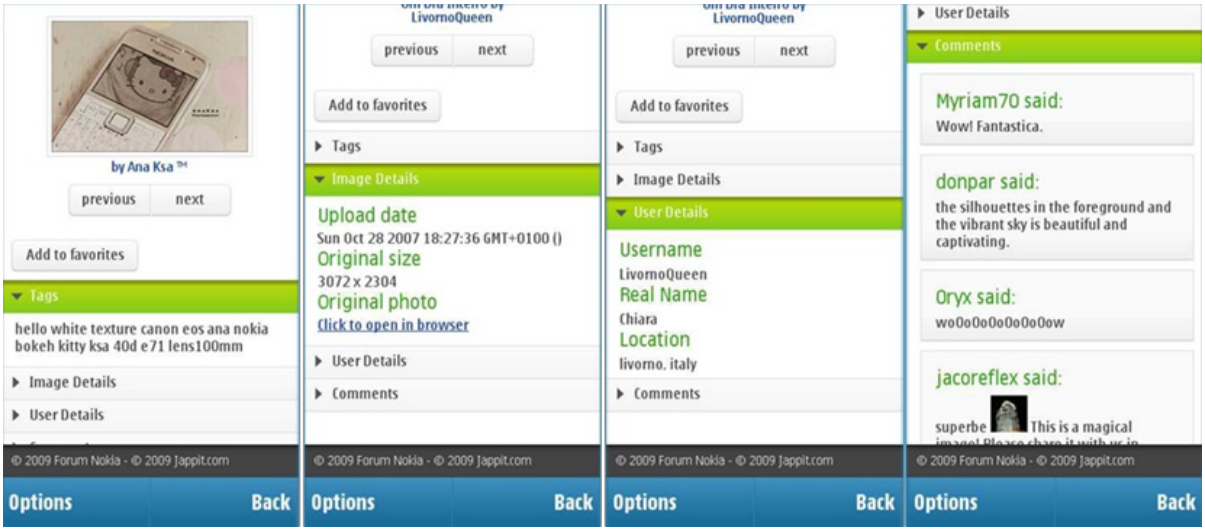
```
<dl id="accordion" class="list-accordion">

 <dt><span></span>Tags</dt>
 <dd><p id="photo_tags"></p></dd>

 <dt class="collapsed"><span></span>Image Details</dt>
 <dd><p id="photo_details"></p></dd>

 <dt><span></span>User Details</dt>
 <dd>
  <div id="user_loader" class="loader">
   <img src="images/ajax-loader.gif" />
   <br/>
   Loading user details..
  </div>

  <div id="user_container">

  </div>
 </dd>

 <dt class="collapsed"><span></span>Comments</dt>
 <dd>
  <div id="comments_loader" class="loader">
   <img src="images/ajax-loader.gif" />
   <br/>
   Loading comments..
  </div>

  <div id="comments_container">

  </div>
 </dd>
</dl>
```

Since both the user details and the photo's comments have to be loaded with separate XMLHttpRequests, 2 loading indicators are used in the respective sections, in order to display the loading action to the user. Also, the **Accordion component needs a JavaScript initialization** as the following:

```
var myAccordionList = new AccordionList("accordion");
```

The Accordion sections are visible below.

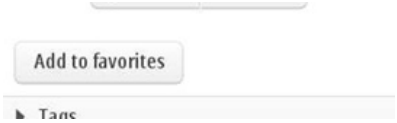
(C) Copyright Nokia 2013. All rights reserved.



Adding to favorites

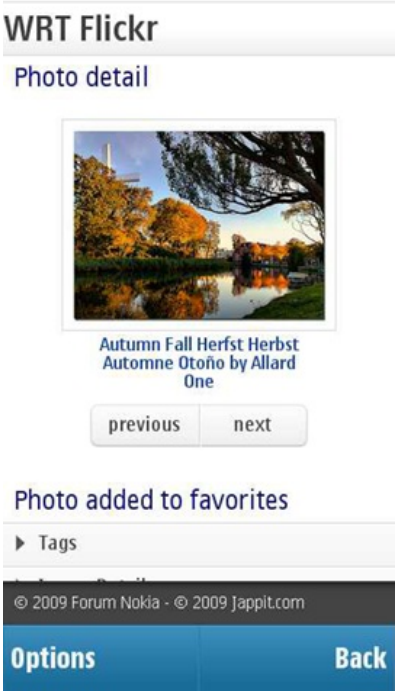
Final functionality to add to the photo detail view, is the "Add to favorite" one. In order to implement this **a rounded button is used**:

```
<ul class="button-rounded">
 <li><a id="favorites_add" href="#" onclick="addPhotoToFavorites()">Add to favorites</a></li>
</ul>
```



Complete view

The **complete photo detail view** is visible below.

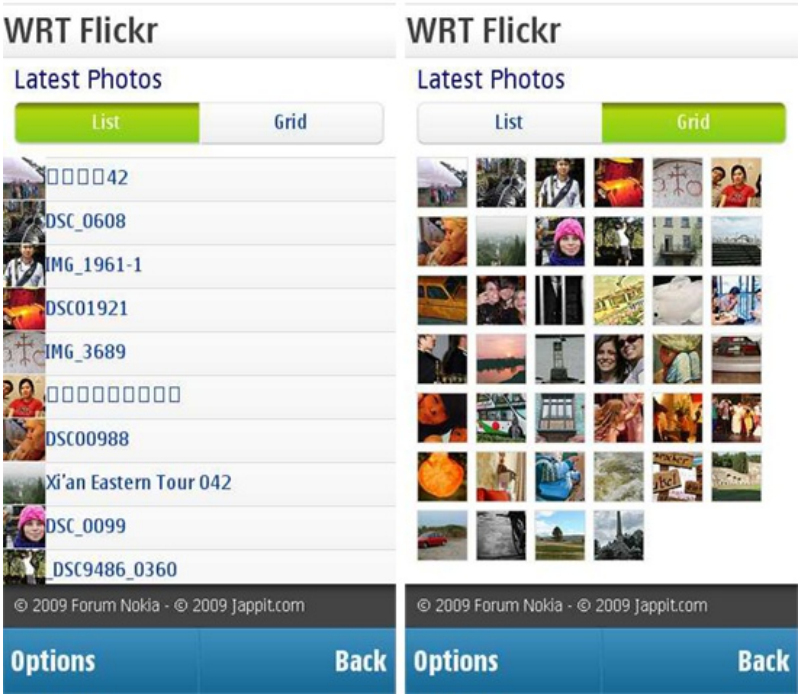


Latest photos

The latest photos view retrieves and **displays the latest photos uploaded on Flickr**.

Since the information to be visualized is the same, this view can be easily **built by using the same layout and structure of the search results view**, as shown below.

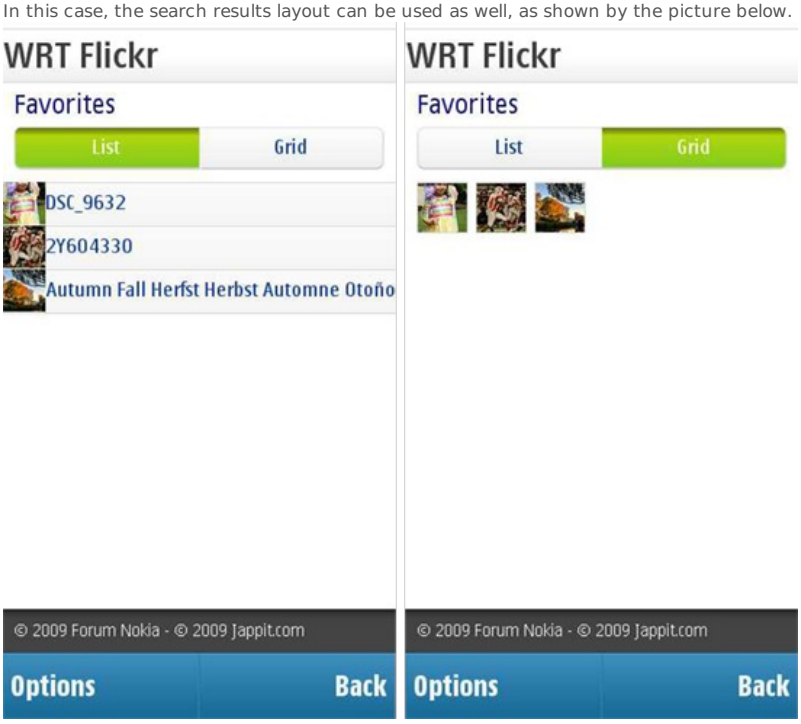
(C) Copyright Nokia 2013. All rights reserved.



Favorite photos

The favorite photos view shows the photos marked as "favorite" by the widget's user.

In this case, the search results layout can be used as well, as shown by the picture below.

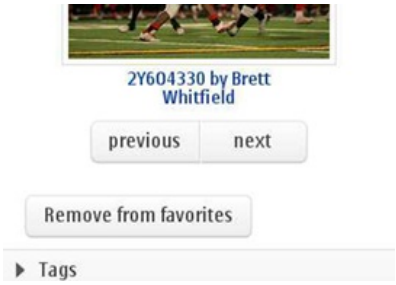


Removing from favorites

The user, when in the favorites section, should be able to **remove a photo previously marked as "favorite"**. In order to do this, a rounded button is added, in a similar manner to the "Add to favorites" one:

```
<ul class="button-rounded">
 <li><a id="favorites_remove" href="#" onclick="removePhotoFromFavorites()">Remove from favorites</a></li>
</ul>
```


(C) Copyright Nokia 2013. All rights reserved.



Credits

The credits view has to display some **information about the widget** (e.g.: its creator). In order to add useful functionalities to the view, some rounded buttons are added to **open related websites directly from the widget**, as the Nokia Developer Website:

```
<ul class="button-rounded">
 <li><a href="#" onclick="widget.openURL('http://www.forum.nokia.com')">Visit Nokia Developer</a></li>
</ul>
```

The simple credits view of this widget is shown below.

Template Customizations

Some **customizations have been done** to the Mobile Web Templates for High-End devices as shipped by Nokia Developer. These include:

- **fixed footer**: in order to have a fixed footer at the bottom of the device display, leaving scrollable the rest of the widget, a **fixed CSS positioning** is used for the footer itself:

```
#footer
{
 position: fixed;
 bottom: 0px;
 width: 100%;
}
```

- **slideshow callback**: when the user navigates from a photo to another, the associated details have to accordingly change. In order to do this, some changes to the Slideshow JavaScript code have been performed, in order to allow this kind of behavior. These changes also affect the Slideshow constructor, that accepts as fourth argument the callback:

```
var mySlideshow = new Slideshow("slideshow", photoIndex, true, onPhotoUpdate);

function onPhotoUpdate(index)
{
//user has navigated to photo at index "index"
}
```

Downloads

The Web Runtime widget developed in this article is available for download here: WRT Flickr download.

Conclusions

(C) Copyright Nokia 2013. All rights reserved.

This article shows **how a complete widget can be developed with the help of Mobile Web Templates for High-End devices**.

The **many ready-to-use and tested components** allow for quicker development, **focusing the developer efforts on widget's functionality** rather than on CSS and layout issues.

As shown, both the **CSS and JavaScript code can easily be adapted** in order to suit the needs of different widgets, without the need to rewrite whole components from scratch.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.