

Adding HTML widgets in your vision Layer

In this tutorial we will show how to add HTML Widgets in a Vision layer. We will create an interactive carousel (slideshow) and a twitter timeline widget and explain how you can add them to your Page. These example widgets are build-in features in the Creator tool.

You can see this example in action by scanning the image below (available in the Page folder included in the attachment) using latest Layar build for iPhone or Android. The layer name is HTML widgets.



Image 1. The page

Requirements:

In order to create your own layer with HTML widget you need to have a server that is publicly accessible. Basic knowledge of Web development is also required.

If you have no previous experience with Vision layers please make sure to check the links bellow before you proceed with this tutorial

How does Vision layer work:

<http://www.layar.com/documentation/browser/howtos/layer-vision-doc/>

How to enable vision layer:

<http://www.layar.com/documentation/browser/howtos/layer-vision-doc/enable-layer-vision-layer/>

Create a simple vision layer tutorial:

<http://www.layar.com/documentation/browser/tutorials-tools/create-simple-vision-layer-copy/>

Create the HTML file containing the image carousel (slideshow)



The first step to create the image carousel widget for our Vision layer is to create the HTML file that will actually show one. In the attached folder you will find the 3 files needed to create the image carousel as shown in our example:

- carousel.html
- carousel.js
- carousel.css

In order to have your own images shown in the carousel you only need to adjust the carousel.html file.

```
<div id="carousel">
<ul>
<!--Add the images and define the width and height, images will show in the order placed bellow--&gt;
    &lt;li&gt; &lt;img src=" http://www.example.com /carousel/images/first.png" width="262"
height="260" /&gt; &lt;/li&gt;
    &lt;li&gt; &lt;img src=" http://www.example.com /carousel/images/png2.png" width="262"
height="260" /&gt; &lt;/li&gt;
    &lt;li&gt; &lt;img src=" http://www.example.com /carousel/images/png4.png" width="262"
height="260" /&gt; &lt;/li&gt;
    &lt;li&gt; &lt;img src=" http://www.example.com /carousel/images/png5.png" width="262"
height="260" /&gt; &lt;/li&gt;
    &lt;li&gt; &lt;img src=" http://www.example.com /carousel/images/png13.png" width="262"
height="260" /&gt; &lt;/li&gt;</pre>
```

In the part of the code shown above insert the URL of your images and their height and width. Keep in mind that the minimum values of the width and height should be 100 pixels. If you want the images to show smaller on your page you can define that later in your JSON file. You will get the best result when all images have the same size. You can find the example images used in this tutorial under folder images.

Once you make the necessary changes save the page and upload all three files in the same folder on your server. There is no need to change any other files. Do not forget to place the images on the server as well.

Prepare the JSON file

Once the HTML file with the image carousel is ready we can create our JSON file so that it represents the HTML panel. We will use an object of type text/HTML to do so. You can find the complete JSON file in the attached zip in folder JSON. For more information please do check the API documentation:

<http://www.layar.com/documentation/browser/api/changes-71>

```
"object": {  
    "url": "http://www.example.com/folder/carousel/carousel.HTML", //url pointing to  
    the carousel.HTML  
    "contentType": "text/HTML",  
    "viewport": {  
        "height": 500, //define the height of viewport  
        "width": 500, //define the width of the viewport. Width and height define how  
        much of our page will be shown  
        "interactive": true, //users can swipe the carousel  
        "scrollable": false //users cannot scroll  
    },  
    "size": 0.3604497354497354 //the size of the widget. This is calculated relative to  
    the reference image/Page. The real world size for our page was set to 1.  
},
```

“Interactive” parameter is set to “true” because we want the user to swipe and navigate through images. “Scrollable” is set to “false” because there is no need to scroll to see the whole image.

The width and height of the viewport should not be less than the height and width set for the images in the HTML file because in that case only part of images will show. In order to configure the size of the image relative to the page you can change the size parameter.

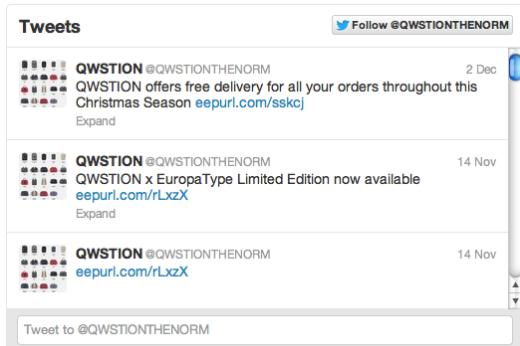
Since “interactive” is set to “true” we cannot add any actions to it. In order to place it in the part of the page we user the transform parameter below:

```
"transform": {  
    "translate": {  
        "y": -0.3,  
        "x": 0.19,  
        "z": 0.0001  
    },  
    "scale": 1  
},
```

Creating a Twitter feed HTML file

In this example we are going to create a user Twitter feed. Our Twitter feed widget will not be interactive but clickable. Once a user clicks the widget a webpage will be launched.

Create the HTML file for the Twitter feed



We created our twitter feed using the widget tool on Twitter page. Using the Twitter widget tool you can embed a timeline for Tweets from an individual user, a user's favorites, Twitter lists, or any search query or hashtag and customize it. In order to use this tool you need to have a Twitter account.

To create a timeline you must sign in to twitter.com and visit the [widgets section of your settings page](#). From this page you can see a list of the timelines you've already configured and create new timelines.

Login on Twitter and select settings -> Widgets. In the page that will appear (image 2) select create new.

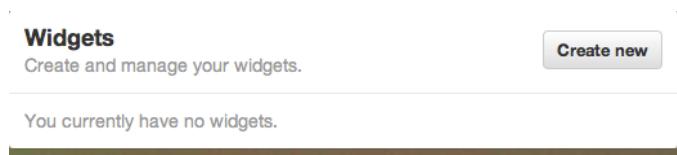


image 2

A new page is shown (image 3) and you can choose the twitter widget you would like to create

The screenshot shows the Twitter configuration interface. On the left, there's a form with fields for 'Username' (@QWSTIONTHENORM), 'Height' (400), 'Theme' (Light), and 'Link color' (Default (blue)). It also includes a 'Domains' field with 'custom.layar.nl' and a note about embedding domains. There are checkboxes for 'Auto expand photos', 'Opt-out of tailoring Twitter', and 'Save changes' (which is highlighted in blue). On the right, a preview window shows a timeline of tweets from QWSTION and Selectism.

Image 3. Twitter configuration

For the scope of this demonstration we are going to create a user timeline. On search box we insert the name of the user you would like to use and complete the fields in the form. Make sure to use the same domain name of the server you are going to host the HTML file that will actually show your twitter feed. For example, if you upload your files in a server with domain name: www.example.com you should fill in this domain name.

Once you save the changes a code snippet will appear (image 4)

A screenshot of a code editor showing a snippet of JavaScript code. The code starts with `href="https://twitter.com/QWSTIONTHENORM" data-widget-id="27854712"`. Below the code, there's a button labeled 'Copy' and a note saying 'Copy and paste the code into the HTML of your site.'

Image 4. Code

You can open a text editor ,copy paste this code. The default dimensions for a timeline are 520×600px, which can be overridden to fit the dimension requirements of your page. We will add `width="300"` `height="300"` in the “`a`” tag, so it fits nicely on our page later, you can adjust yours accordingly. Save the file as `twitter.html` and place it on your server.

You can find more information on Twitter widgets in this page:
<https://dev.twitter.com/docs/embedded-timelines>.

Prepare the JSON file

Now that you have your HTML file with twitter timeline you are ready to create the HTML widget for your Vision layer.

We will add an object in our JSON file to represent the Twitter timeline widget

```
{ "object": {  
    "url": " http://www.example.com/twitter/twitter.html",//url  
    pointing to twitter widget  
    "contentType": "text/HTML",  
    "viewport": {  
        "height": 300,//same as the one set in our HTML file  
        "width": 300,//same as the one set in our HTML file  
        "interactive": false,//users cannot interact with the widget  
        "scrollable": true // this parameter is also considered as false now  
    },  
    "size": 0.30
```

Since “interactive” is set to false, the value of “scrollable” parameter is overridden and considered as false. Therefore, users cannot scroll the Twitter widget or interact with the posts (answer, retweet etc.)

According to documentation once interactivity is set to false we can add actions to the HTML widgets. For instance users will click the twitter widget and the Layar website will appear.

```
"actions": [{  
    "uri": "http://www.layar.com ",  
    "contentType": "text/html "  
}],
```

This concludes our tutorial; feel free to download the attached Sample Code and give it a try. You can play with the values of size and viewport height and width to better understand how it works!

You can also add onCreate animation to your widgets so they appear nicely when they first load.

Good luck!

Helpful links:

<http://devsupport.layar.com/forums>

<http://devsupport.layar.com/entries/22561033-video-tutorial-8-how-to-use-dynamic-widgets>

<http://www.layar.com/documentation/browser/api/changes-71/>

<http://www.layar.com/documentation/browser/api/getpois-response/>

<http://devsupport.layar.com/forums/21561743-HTML-widgets>