

Tagger UI API

Tagger UI Overview

The fluid Tagger UI provides an easy to use component to plug in tagging features to a canvas element in web page. It provides functions to add and delete tags/annotations on images interactively. It uses the HTML5 canvas element to achieve tagging and therefore requires a browser that supports HTML5. Thus it will be delivered to modern, standards-compliant browsers, including:

- Firefox 3.6 and higher
- Safari 4 and higher
- Google Chrome

Creating a Tagger UI

To instantiate a new Tagger UI on your page:

```
var myTagger = fluid.TaggerUI(container, options);
```

Returns: A Tagger UI component object.

Parameters

container

The container is a CSS-based selector, single-element jQuery object, or DOM element that identifies the root DOM node of the Tagger UI markup.

options

The options object is an optional data structure that configures the Tagger UI, as described in the Options section below.

On This Page

- [Tagger UI Overview](#)
- [Creating a Tagger UI](#)
 - [Parameters](#)
 - [container](#)
 - [options](#)
- [Options](#)
- [Supported Events](#)
 - [Annotation Subcomponent](#)
- [Methods](#)
- [Dependencies](#)

See Also

- [Tutorial - CropperUI](#)
- [Image Editor](#)
- [Image Editor API](#)
- [Tutorial - Image Editor](#)

Still need help?

Join the [infusion-users mailing list](#) and ask your questions there.

Options

| Name | Description | Values | Default |
|-------------|-------------------------------------------------------------------------------------------------|--------------|------------------------------------------------------------------------------|
| styles | Specific class names used to achieve the look and feel of the different states of the Tagger UI | | <pre>styles: { remove: "fl-image-editor-annotation- action-remove" }</pre> |
| lineWidth | The width of line of tag box | 1, 2, 3, ... | lineWidth: 1 |
| strokeStyle | The style for borders of tags | CSS Colors | strokeStyle: 'white' |
| fillStyle | The style for area inside the tags | CSS Colors | fillStyle: 'rgba(255,255,255,0.4)' |
| blurColor | The color to be used for blurring the area outside the bounding box while tagging | CSS Colors | blurColor: 'rgba(255,255,255,0.4)' |

Supported Events

The Tagger UI fires the following events (for more information about events in the Fluid Framework, see [Events for Component Users](#)):

| Event | Type | Description | Parameters | Parameters Description |
|----------------------|---------|----------------------------------------------------------------------|------------------------------------|------------------------------------------------------------------------------------------------------------|
| onChangeWidth | default | This event is fired whenever the width of bounding box changes. | newWidth | newWidth : the new width of the bounding box |
| onChangeHeight | default | This event is fired whenever the height of bounding box changes. | newHeight | newHeight : the new height of the bounding box |
| onChangeLocationX | default | This event is fired whenever the x location of bounding box changes. | newLocationX | newLocationX : the new x location of the bounding box |
| onChangeLocationY | default | This event is fired whenever the y location of bounding box changes. | newLocationY | newLocationY : the new y location of the bounding box |
| onAnnotationNbChange | default | This event is fired whenever the number of annotations changes. | newNbAnnotations, oldNbAnnotations | newNbAnnotations : new number of annotations oldNbAnnotations : old number of annotations |
| OnAnnotationAdd | default | This event is fired whenever an annotation is added. | tag | tag : the text of the new annotation. |
| onAnnotationRemove | default | This event is fired whenever an annotation is removed. | index | index : the index of the removed annotation. |

Annotation Subcomponent

This sub-component stores the complete model for the bounding box for the Tagger UI component. The following properties are available with the object:

x : number, // x coordinate of the bounding box
y : number, // y coordinate of the bounding box
w : number, // width of the bounding box
h : number, // height of the bounding box
tag: string, //the text associated with the annotation

Methods

| Method | Description | Parameters |
|---------------------------------------------------|-------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| init(canvas, resizeFactor, image, imageX, imageY) | Begins the tagging and associates itself with the canvas element. | canvas : The canvas object with which the Tagger UI should associate. resizeFactor : The factor with which the image has been resized to fit in the canvas. e.g. if the original image width is 1000px and the canvas width is 500px, then the resize factor would be 2. image : The image to be drawn in canvas. imageX : The x coordinate for image to be drawn in canvas. imageY : The y coordinate for image to be drawn in canvas. |
| reset () | Completely resets the tagger UI component. Removes all the annotations. | <i>none</i> |
| activateKeyboardAccessibility () | Activate the keyboard accessibility of Tagger. | <i>none</i> |
| deactivateKeyboardAccessibility () | Deactivate the keyboard accessibility of Tagger. | <i>none</i> |
| confirmTagAdd(tagText) | Adds the current tag into the list. | tagText : The text to be associated with the tag. |
| deleteTag(index) | Deletes the tag at given index. | index : The index of the tag to be deleted. |

| | | |
|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| doneTagging() | Indicate the completion of a tagging session. Doesn't affect the given tags. Just performs cleanup and redraws the background. If you want to reset the tagger component and delete all the existing tags, use reset instead. | <i>none</i> |
| showAnnotations() | Show all the annotations. Also associate the mouse handlers for annotations. | <i>none</i> |
| hideAnnotations() | Hides all the annotations and unbinds the mouse handlers. | <i>none</i> |
| highlightTag(index) | Highlight the annotation/tag at given index | index: The index of the tag to be highlighted. |
| removeHighlights() | Removes all existing highlights | <i>none</i> |
| getNbAnnotations() | Returns the number of annotations. | <i>none</i> |
| getTagList() | Returns the complete list of all the tags. | <i>none</i> |
| adjustTagsForResize(newW, newH, resizeFactor, image, imageX, imageY) | Function that automatically adjusts tags when the image has been resized. It automatically reduces/increases the size of the tags to reflect the new size. | newW: new width of the image newH: new height of the image resizeFactor: the ratio by which the image has been resized to fit the canvas image: the new resized image imageX: the new starting x coordinate of image imageY: the new starting y coordinate of image |
| adjustTagsForCrop(newW, newH, resizeFactor, image, imageX, imageY, croppingDimensions) | Function that automatically adjusts tags when the image has been cropped. It automatically discards all the tags that were outside the cropping dimensions. | newW: new width of the image newH: new height of the image resizeFactor: the ratio by which the image has been resized to fit the canvas image: the new resized image imageX: the new starting x coordinate of image imageY: the new starting y coordinate of image CroppingDimensions: A crop model object that stores dimensions of crop bounding box. |
| setLocationX(newLocationX) | Sets the x location of the crop bounding box. | newLocationX: The new x location of bounding box. |
| setLocationY(newLocationY) | Sets the y location of the crop bounding box. | newLocationY: The new y location of bounding box. |
| setWidth(newWidth) | Sets the width of the crop bounding box. | newWidth: The new width of the bounding box. |
| setHeight(newHeight) | Sets the height of the crop bounding box. | newHeight: The new height of the bounding box. |

Dependencies

The Tagger UI dependencies can be met by including in the header of the HTML file

- the minified InfusionAll.js file
- the Fluid layout CSS file
- the TaggerUI JS file
- the TaggerUI CSS file

as shown below:

```
<script type="text/javascript" src="infusion/InfusionAll.js"></script>
<link rel="stylesheet" type="text/css" href="infusion/src/webapp/framework/fss/css/fss-layout.css" />
<script type="text/javascript" src="infusion/src/webapp/components/TaggerUI/js/TaggerUI.js"></script>
<link rel="stylesheet" type="text/css" href="infusion/src/webapp/components/taggerUI/css/TaggerUI.css" />
```

Alternatively, the individual file requirements are:

```
<link rel="stylesheet" type="text/css" href="../../../framework/fss/css/fss-reset-global.css" />
<link rel="stylesheet" type="text/css" href="../../../framework/fss/css/fss-base-global.css" />
<link rel="stylesheet" type="text/css" href="../../../framework/fss/css/fss-text.css" />
<link rel="stylesheet" type="text/css" href="../../../framework/fss/css/fss-layout.css" />
<link rel="stylesheet" type="text/css" href="../../../inlineEdit/css/InlineEdit.css" />
<link rel="stylesheet" type="text/css" href="../../../taggerUI/css/TaggerUI.css" />

<!-- Fluid and jQuery Dependencies -->
<script type="text/javascript" src="infusion/src/webapp/lib/jquery/core/js/jquery.js"></script>
<script type="text/javascript" src="infusion/src/webapp/lib/jquery/ui/js/jquery.ui.core.js"></script>
<script type="text/javascript" src="infusion/src/webapp/framework/core/js/jquery.keyboard-ally.js"></script>
<script type="text/javascript" src="infusion/src/webapp/lib/jquery/plugins/scrollTo/js/jquery.scrollTo.js"></script>
<script type="text/javascript" src="infusion/src/webapp/lib/swfobject/js/swfobject.js"></script>
<script type="text/javascript" src="infusion/src/webapp/lib/swfupload/js/swfupload.js"></script>
<script type="text/javascript" src="infusion/src/webapp/framework/core/js/Fluid.js"></script>
<script type="text/javascript" src="infusion/src/webapp/framework/core/js/FluidDocument.js"></script>
<script type="text/javascript" src="infusion/src/webapp/framework/core/js/FluidView.js"></script>
<script type="text/javascript" src="infusion/src/webapp/framework/core/js/DataBinding.js"></script>
<script type="text/javascript" src="infusion/src/webapp/framework/core/js/FluidIoC.js"></script>
<script type="text/javascript" src="infusion/src/webapp/framework/enhancement/js/ProgressiveEnhancement.js"></script>

<!-- Tagger UI dependencies -->
<script type="text/javascript" src="infusion/src/webapp/components/taggerUI/js/TaggerUI.js"></script>
```