

# Keyboard Accessibility Plugin API

The jQuery Keyboard Accessibility Plugin makes it easy for developers to add keyboard handlers to their code without a lot of extra overhead.

**Please note** that, when used as a jQuery plugin, the Keyboard Accessibility API currently handles only two parameters. This is a bug in the current implementation that will be addressed in a future release. If more than two parameters are required, any parameters beyond the function name must be enclosed in an array.

## activatable

```
jQuery().fluid("activatable", customHandler);
jQuery().fluid("activatable", parameterArray); // parameterArray
= [customHandler, options]
```

Makes all matched elements activatable with the Spacebar and Enter keys. A handler function may be provided to trigger custom behaviour.

### Arguments:

Name	Description
customHandler	A function that will be called when the element is activated. <b>In v1.0:</b> The function should accept the element to be activated as a parameter: <code>myHandler(element);</code> <b>In v1.1 and beyond:</b> The function should accept the browser event as a parameter: <code>myHandler(event);</code>
options	(Optional) A collection of name-value pairs that allow you to active elements using other keystrokes (see below).

### Example:

```
var handler = function (evt) {
    $(evt.target).addClass("activated");
};
menu.items.fluid("activatable", handler);
```

### options:

Name	Description	Values	Default
additionalBindings	An array of keycode options to use for activation.	An array of objects of the form <pre>{   modifier: &lt;a modifier keycode&gt;, // e.g. \$.ui. keyCode.SHIFT   key: &lt;a keycode&gt;, // e. g. \$.ui.keyCode.DOWN   activateHandler: &lt;a function&gt; }</pre>	none

### Example with options:

```
var handler = function (evt) {
    $(evt.target).addClass("activated");
};
// Bind the Space, Enter, and Down Arrow keys to the activate
event.
// Notice the use of "additionalBindings," which is required only
for the Down Arrow Key.
// By default, Space and Enter are set up for you.
var opts = {
    additionalBindings: {
        modifier: null,
        key: $.ui.keyCode.DOWN,
        activateHandler: alternateActivate
    }
};
menu.items.fluid("activatable", [handler, opts]);
```

## On This Page

- [activatable](#)
- [activate](#)
- [selectable](#)
  - [currentSelection](#)
  - [select](#), [selectNext](#), [selectPrevious](#)
- [tabbable](#)
- [tabindex](#)

[Dependencies](#)

## See Also

- [Tutorial - Keyboard Accessibility Plugin](#)

## Still need help?

Join the [infusion-users mailing list](#) and ask your questions there, or hang out in our [IRC Channel](#).

## activate

```
jQuery().fluid("activate");
```

Activates all matched elements using the default activation handler. Note that if the user has made the elements "activatable" using `activatable()`, that is the handler that will be used.

### Example:

```
menu.items[selectedIndex].fluid("activate");
```

## selectable

```
jQuery().fluid("selectable", options)
```

Invoked on a JQuery which is a "container" node: makes all matched elements descended from the container selectable with the arrow keys. Custom handlers can be provided to inject custom styling and logic.

NOTE: The container element must be keyboard focusable. If necessary, add `tabindex="0"` to the container.

**Arguments:**

Name	Description
options	(Optional) A collection of name-value pairs that let you modify the plugin's default selection behaviour (see below).

**options:**

Name	Description	Values	Default
selectableElements	The set of nodes (a JQuery) which are to form the selectable set (these should all be nested within the DOM element representing the "container")	jQuery	selectableElements: null
selectableSelector	A selector which will be used to locate the selectableElements if they are not set manually, by matching nodes within the container	string	selectableSelector: ".selectable"
onSelect	A function to be invoked when an element is selected	function	onSelect: null
onUnselect	A function to be invoked when an element is unselected	function	onUnselect: null
onLeaveContainer	A function to be invoked when focus leaves the container	function	onLeaveContainer: null
direction	Indicates the orientation of the selectable elements. This value will be used to determine the appropriate mapping for next and previous.	fluid.ally.orientation.HORIZONTAL fluid.ally.orientation.VERTICAL	direction: fluid.ally.orientation.VERTICAL
autoSelectFirstItem	Indicates whether or not focus should be moved to the first child of the container (or to the last selected child, depending on rememberSelectionState)	boolean	autoSelectFirstItem: true
rememberSelectionState	Indicates whether or not to keep track of which child element was most recently selected.	boolean	rememberSelectionState: true
<b>New in v1.3:</b> noBubbleListeners	Useful in cases where the selectable items may have keystrokes swallowing default handlers. If this option is true, the arrow key handler will be bound to the selectables themselves instead of their container.	boolean	noBubbleListeners: false

**Example:**

```
menu.container.fluid("selectable", {
  selectableSelector: ".menuItem",
  onSelect: function (evt) {
    <function to show sub menu>
  },
  onUnselect: function (evt) {
    <function to hide sub menu>
  },
  autoSelectFirstItem: false,
  rememberSelectionState: false
});
```

**currentSelection**

```
jQuery().fluid("selectable.currentSelection");
```

Gets the currently selected element (invoked on the container element)

**Example:**

```
var currentMenuItem = menu.container.fluid("selectable.currentSelection");
```

**select, selectNext, selectPrevious**

```
jQuery().fluid("selectable.selectNext");
jQuery().fluid("selectable.selectPrevious");
jQuery().fluid("selectable.select", elementToSelect);
```

Selects the next, previous, or specified element in the list of selectables (invoked on the jQuery for the container element)

**Arguments:**

Name	Description
elementToSelect	The element to select

**Examples:**

```
menu.container.fluid("selectable.select", secondMenuElement);
menu.container.fluid("selectable.selectNext"); // will select third item
menu.container.fluid("selectable.selectPrevious"); // will select second item again
```

## tabbable

```
jQuery().fluid("tabbable")
```

Adds all matched elements to the tab order by giving them a tabindex attribute of "0." Note that if a matched element already has a tabindex value that places it in the tab order, the value is unchanged.

**Example:**

```
menu.container.fluid("tabbable");
```

## tabindex

Deprecated. This code is now available in jQuery core. Use `jQuery.attr("tabindex")` instead.

```
jQuery().fluid("tabindex")
```

Gets the tabindex value of the first matched element. If the element doesn't have a tabindex attribute, `undefined` is returned. The value returned is normalized to a Number, since browser implementations vary.

```
jQuery().fluid("tabindex", value)
```

Sets the tabindex value on all matched elements. `value` can be either a String or a Number, but valid tabindex attributes need to be integers.

**Arguments:**

Name	Description
value	The desired integer tabindex value. Can be either a String or a Number.

## Dependencies

The Keyboard Accessibility Plugin dependencies can be met by including the minified `InfusionAll.js` file in the header of the HTML file:

```
<script type="text/javascript" src="InfusionAll.js"></script>
```

Alternatively, the individual file requirements are:

In v1.2 and earlier:

```
<script type="text/javascript" src="lib/jquery/core/js/jquery.js"></script>
<script type="text/javascript" src="lib/jquery/ui/js/jquery.ui.core.js"></script>
<script type="text/javascript" src="framework/core/js/jquery.keyboard-ally.js"></script>
```

As of v1.3:

```
<script type="text/javascript" src="lib/jquery/core/js/jquery.js"></script>
<script type="text/javascript" src="lib/jquery/ui/js/jquery.ui.core.js"></script>
<script type="text/javascript" src="framework/core/js/FluidDocument.js"></script> <!-- New in v1.3 -->
<script type="text/javascript" src="framework/core/js/jquery.keyboard-ally.js"></script>
```