


editor.content.selection

The aim of this API is to provide detailed access to the editor selection while adding a layer of safety for developers. As a result it may be somewhat more limited than is convenient for developers accustomed to raw DOM access.

 This API was introduced in Textbox.io release 2.2.1

Methods

Name	Parameters	Returns	Details
<code>getText</code>	none	String	Returns the text content of the selection, removing all HTML tags. If the selection is collapsed an empty string (" ") is returned.
<code>getHtml</code>	none	String	Similar to <code>getText</code> but returns the complete HTML representation. Partially selected tags are automatically closed at the edge of the selection range.
<code>findTagAtCursor</code>	CSS3 selector (String)	Element or null	See below for Element API details. <code>Null</code> is returned if the cursor is not within a tag matching the selector.

API status

This API is fully supported but has been released with bare minimum functionality to gauge developer interest and gather feedback.

Please [contact us](#) if there is an addition to the selection API you would like to see in a future release.

Examples

`editor.content.selection.getText()`

```
// Retrieve the current editor selection as text
var selectedText = editor.content.selection.getText();
```

`editor.content.selection.getHtml()`

```
// Retrieve the current editor selection as HTML
var selectedHtml = editor.content.selection.getHtml();
```

`editor.content.selection.findTagAtCursor()`

```
// Retrieve a reference to a hyperlink surrounding the cursor
var selectedLink = editor.content.selection.findTagAtCursor('a[href]');
```

Element object


The Element API is a wrapper around DOM operations designed to smooth over differences between browsers and clean up internal editor attributes from returned data.

Do not retain references

The Element object retains a reference to a DOM node within the editor, which is easily replaced by user action (for example undo and redo). Retaining a reference to Element objects in your application will cause a memory leak.

Methods

Name	Parameters	Returns	Details
------	------------	---------	---------

getAttributes	none	Object	Returns a frozen object of key:value pairs matching the element attribute name:value pairs (see below). Tag name is not included.
setAttributes	Object	no return value	Given an object similar to the one returned by getAttributes, adds (or overwrites) attributes on the element. This method cannot be used to remove attributes.
getText	none	String	Identical to selection.getText() but returns content for this element only
getHtml	none	String	Identical to selection.getHtml() but returns content for this element only
setText	String	no return value	Updates the element text contents (using .textContent)
setHtml	String	no return value	Updates the element HTML contents (using .innerHTML)
replaceElement	String	new Element object	<p>Completely replaces the element in the document and handles cursor placement as it will be impacted by this change. This is equivalent to setting .outerHTML on the DOM node, except the old element reference is not retained.</p> <p>If parsing the string argument produces more than one HTML element, a JavaScript error is thrown.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"> <p> Replaced Element references</p> <p>Once an element has been replaced, any reference to the original element should be discarded. Continuing to use it will either have no effect or fail with a DOM exception.</p> </div>

Examples

attribute objects

```
// for a HTML element <a class="mylink" href="http://ephox.com" alt="alternative text for link" target="_blank"
>...</a>
var attributes = {
  href: 'http://ephox.com',
  alt: 'alternative text for link',
  target: '_blank',
  'class': 'mylink'
}
```

replaceElement

```
var newElement = element.replaceElement('<a href="http://google.com">new text content</a>');
```

Adjusting link attributes

```
// Retrieve a reference to a hyperlink surrounding the cursor - assuming not null for this example
var selectedLink = editor.content.selection.findTagAtCursor('a[href]');

// Retrieve the attributes of the link
var linkAttributes = selectedLink.getAttributes();

// Create new attributes. Mutating the linkAttributes variable will have no effect, it's a frozen object
var newAttributes = {
  href: linkAttributes.href + '?replaced=true',
  'data-link-details': 'replaced'
};

// Update the link attributes
selectedLink.setAttributes(newAttributes);
```

Updating a complete link

```
// Retrieve a reference to a hyperlink surrounding the cursor - assuming not null for this example
var selectedLink = editor.content.selection.findTagAtCursor('a[href]');

// Retrieve the HTML contents of the link
var linkHtml = selectedLink.getHtml();

// Replace the link completely without changing the contents
selectedLink.replaceElement('<a href="http://google.com">' + linkHtml + '</a>');
```