

# tabWidget

## tabWidget

```
new tabWidget(x, y, fileTreeWidget) → {canvasWidget}
```

Source: [widgets/tabWidget.js, line 19](#)

### Parameters:

Name	Type	Description
<code>x</code>	<a href="#">Integer</a>	This is the Column that this widget should be added to.
<code>y</code>	<a href="#">Integer</a>	This is the row of of the Column that this widget should be added to.
<code>fileTreeWidget</code>	<a href="#">FileTreeWidget</a>	Reference to a FileTree.

### Returns:

Returns a tabWidget, an instance of the Widget class which creates a new tab every time a file in the file tree is double clicked on.

Type [canvasWidget](#)

## Extends

- [Widget](#)

## Methods

```
getCol() → {Integer}
```

Source: [widgets/widget.js, line 89](#)

Inherited From: [Widget#getCol](#)

Get the position in columns(Left = 0 to Right = n) of this widget.

### Returns:

colIndex - The column index of this Widget.

Type [Integer](#)

```
getElement() → {HTMLElement|*}
```

Source: [widgets/tabWidget.js, line 106](#)

Overrides: [Widget#getElement](#)

Getter for this widgets tab element. Used to append child nodes to the base tab bar element.

Returns:

Type `HTMLElement | *`

```
getPath(node) → {String}
```

Source: [widgets/tabWidget.js, line 147](#)

This function is used to get an absolute path for a specific file from a FancyTree node element.

Parameters:

Name	Type	Description
<code>node</code>	<code>FancyTreeNode</code>	This is a fancy tree node, a path is generated from this call.

Returns:

The file path to the root directory of this node.

Type `String`

```
getRow() → {Integer}
```

Source: [widgets/widget.js, line 97](#)

Inherited From: [Widget#getRow](#)

Get the position in rows(Top = 0 to Bottom = n) of this widget.

Returns:

rowIndex - The row index of this Widget.

Type `Integer`

```
(async) init() → {Promise.<any>}
```

Source: [widgets/tabWidget.js, line 39](#)

Overrides: [Widget#init](#)

This function overrides the parent widget initialize function and creates a tab bar to be displayed within this widget.

Returns:

Type      `Promise.<any>`

```
openFile(filePath)
```

Source:      [widgets/tabWidget.js, line 68](#)

This function takes a filePath, and adds a new tab to the tab bar, as well as calls the callback function defined for this file type if it is known.

Parameters:

Name	Type	Description
<code>filePath</code>	<code>String</code>	The Path to a file to open. Perform the callback function registered for this file type.

```
performCallbackForFileType(title)
```

Source:      [widgets/tabWidget.js, line 126](#)

This function determines if any callbacks for the passed file extension are known. If they are known they will be performed.

Parameters:

Name	Type	Description
<code>title</code>	<code>String</code>	This is the name of a file. The file extension is stripped from the file.

```
registerFiletype(extension, callback)
```

Source:      [widgets/tabWidget.js, line 57](#)

This lets you register a callback to trigger when a file of a specific type is opened. Callback must contain {extension:"the file extension", callback:function()}

Parameters:

Name	Type	Description
<code>extension</code>	<code>String</code>	This is the file extension that you want to register a callback for, ie '.png'
<code>callback</code>	<code>function</code>	This is the function you want to execute when a file of type 'extension' is clicked on.

## resizeTabs()

Source: [widgets/tabWidget.js, line 113](#)

This function is called whenever this widget is resized, it will automatically set each widget to be the correct size.

## save() → {Object}

Source: [widgets/tabWidget.js, line 162](#)

This is simply a wrapper to the parent widget save function.

Returns:

Type      **Object**

## setElement(element)

Source: [widgets/widget.js, line 70](#)

Inherited From: [Widget#setElement](#)

Set the dom element that represents this widget.

Parameters:

Name	Type	Description
<code>element</code>	<b>Element</b>	The dom object for this widget.

*Documentation generated by **JSDoc 3.5.5** on Sat Apr 13 2019 15:26:24 GMT-0400 (Eastern Daylight Time) using the **docdash** theme.*