

# FileTreeWidget

## FileTreeWidget

```
new FileTreeWidget(x, y, path, fileManager) → {FileTreeWidget}
```

Source: [widgets/fileTreeWidget.js, line 16](#)

### 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>path</code>	<a href="#">String</a>	This is a string representing the path to the directory this file tree should display. Paths relative to the root folder are denoted with '~/folderName'.
<code>fileManager</code>	<a href="#">FileManager</a>	This is a reference to the fileManager class which provides this widget with access to the computers File System.

### Returns:

Returns a FileTreeWidget, an instance of the Widget class.

Type [FileTreeWidget](#)

## Extends

- [Widget](#)

## Methods

```
doubleClick(event, data)
```

Source: [widgets/fileTreeWidget.js, line 84](#)

This function is the callback method injected into this fileTreeWidget. Whenever a user double clicks on a file inside of the file tree, this function is called. This function should be modified to a specific developers needs.

## Parameters:

Name	Type	Description
<code>event</code>	<code>Object</code>	The event object contains information about what node was clicked as well as the specific DOM element that was interacted with.
<code>data</code>	<code>Object</code>	Data holds all of the data for that FileTree node.

`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()` → `{Element}`

Source: [widgets/widget.js, line 81](#)

Inherited From: [Widget#getElement](#)

Get the dom element that represents this widget.

### Returns:

element - The dom object for this widget.

Type      `Element`

`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}`

Source: [widgets/fileTreeWidget.js, line 33](#)

Overrides: [Widget#init](#)

This function recursively opens subdirectories from the given path, and then produces a file-tree object to be displayed within this file browser widget.

## Returns:

Returns an asynchronous promise that will resolve on tree generation.

Type **Promise**

## 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.

## subscribe(observer)

Source: [widgets/fileTreeWidget.js, line 95](#)

This function allows any class to pass functions into the observers object, when a file is clicked on, every observer will have their callback functions trigger.

## Parameters:

Name	Type	Description
<code>observer</code>	<b>function</b>	Observer is a callback function to execute when a file is double clicked on.

## translateRelative() → {String}

Source: [widgets/fileTreeWidget.js, line 71](#)

This function converts the '~' character into a path to the fileManger.PATH value. This value can be configured in the fileManager configuration file, therefore ~ will always point to that variable. This allows users to use realtive pathing by simply putting '~' in front of their path.

## Returns:

Returns the path to a folder relative to root.

Type

String

*Documentation generated by **JSDoc 3.5.5** on Sat Apr 13 2019 14:24:24 GMT-0400 (Eastern Daylight Time) using the **docdash** theme.*