

# Tree View Plugin jquery

## identification :

This plugin takes a nested list of type ul or ol and converts it into a hierarchical tree that is collapsible, expandable, and drag-and-drop. In other words, it supports nested lists that allows to collapse / expand list items like a tree structure.

All operations of adding, deleting, modifying, dragging, dropping, folding and expanding can be done, such as adding a new child, modifying his name or deleting him from the tree.

The jQuery plugin is available for use in all personal or commercial projects.

It is possible to make more than one tree in the same HTML page, and none of them will work independently of the other.



## How to use it :

- Load the Bootstrap's stylesheet and `SmartTree.css` in the `head` section of the document.

```
<link rel="stylesheet" href="/path/to/bootstrap.min.css">
<link rel="stylesheet" href="/path/to/SmartTree.css">
```

- Create a nested list of type ol or ul in the html file as follows:

```
<ul class="smart">
  <li>Beverages
    <ul>
      <li>Water</li>
      <li>Coffee</li>
      <li>Tea
        <ul>
          <li>Black Tea</li>
          <li>Whit Tea</li>
          <li>Green Tea
            <ul>
              <li>Sencha</li>
              <li>Gyokuro</li>
            </ul>
          </li>
        </ul>
      </li>
    </ul>
  </li>
</ul>
```

- Download and load the plugin `SmartTree.js` script file in the html after jQuery.

```
<script src="/path/to/jquery.min.js"></script>
<script src="/path/to/smartTree.js"></script>
```

- Initialize the plugin on the HTML list.

```
$(document).ready(function(){
  $('.smart').SmartTree();

  $('.smart').SmartTree({
    // .....
  });
  $('.treeview').SmartTree();
```

```
});
```

- Apply the available options to the tree if desired. All available options with their default values are as follows :

```
$('.smart').SmartTree({  
  
    // all options with their default values  
  
    rtl: "",  
    active: "",  
    draggable: true,  
  
    editicons : true,  
    addicons : true,  
    removeicons : true,  
  
    selectColor : "",  
    selectedAttr : "",  
    textColor : "",  
    nodeColor : "",  
    selectednodetextColor : "",  
  
    fontSize : "",  
    fontType : "",  
});
```

**rtl** : When given the value rtl=true, the tree side will be from right to left. In the default case, its side will be from left to right.

**active** : If the tree is given active = true then the tree will appear in its expanded form (you will be able to see all the branches of the tree). In the default case, only the root of the tree will be shown. That is, the tree will be in the case of folding.

**draggable** : If the value is set to false. draggable =false, the tree will become not draggable. In the default case, it will be draggable.

**editicons** : When passing false to this option.editicons =false the edit icon will not appear and you will not be able to edit the name of any node in the tree.

**addicons** : When you pass the value false to this option. addicons = false, the add icon will not appear and you will not be able to add any new node to the tree.

**removeicons:** When the value false is passed to this option. removeicons = false , the delete icon will not appear and you will not be able to delete any node from the tree.

**selectColor :** When you click on any node, it will be colored with a specific color to indicate that it is selected. You can choose that color yourself and pass it to this option. For example I want to make it red so I will write selectColor="red". Knowing that the default color is light green.

**selectedAttr :** When the li element is given the data-select property with the value selected as data-select="selected", all the nodes will be colored in a specific color that the programmer can specify. Knowing that the default color is light pink.

**textColor :** This option is used to specify the color of the names of the nodes to which the property "data-select="selected ." has been added

**nodeColor :** The color of the nodes in the tree can be specified with this option. For example, I want to make the color of the nodes blue, so I will write nodeColor="blue".

**fontSize:** Determining the font size of the node names can be given values between 12 and 20. For example, I want the font size to be large, so I will write fontSize = 18

**fontType :** This option is used to specify the font type for node names For example  
fontType:'Franklin Gothic Medium'

- Available callback functions.

```
$( '.smart' ).SmartTree({
```

```
//The function that is called after the delete operation is
//executed. The parameter named element is the li element that was
//deleted.
```

```
    remove : function(element){
```

```
    },
```

```
// The function that is called after a new node has been added
// The parameter named element is the li element that was added. The
// parameter called text is the text that was written inside the
// element that was added
```

```
    add : function(element , text){
```

```

    },
    //The function that will be called after a modification to the name
    //of any node. The element named parameter is the li element whose
    // text has been modified. The parameter called text is the
    // modified text

    edit : function(element , text){

    },

    //The function that will be called after dragging and dropping any
    //node from its place. The element named parameter is the li of
    // element that was checked out. The parameter old_parent is the
    // old parent the dragged element. The new_parent parameter is
    // the new parent

    move : function(element , old_parent ,new_parent){

    }

});

```