import $ from 'jquery'

const NAME = 'Treeview'

const DATA\_KEY = 'lte.treeview'

const EVENT\_KEY = `.${DATA\_KEY}`

const JQUERY\_NO\_CONFLICT = $.fn[NAME]

const EVENT\_EXPANDED = `expanded${EVENT\_KEY}`

const EVENT\_COLLAPSED = `collapsed${EVENT\_KEY}`

const EVENT\_LOAD\_DATA\_API = `load${EVENT\_KEY}`

const SELECTOR\_LI = '.nav-item'

const SELECTOR\_LINK = '.nav-link'

const SELECTOR\_TREEVIEW\_MENU = '.nav-treeview'

const SELECTOR\_OPEN = '.menu-open'

const SELECTOR\_DATA\_WIDGET = '[data-widget="treeview"]'

const CLASS\_NAME\_OPEN = 'menu-open'

const CLASS\_NAME\_IS\_OPENING = 'menu-is-opening'

const CLASS\_NAME\_SIDEBAR\_COLLAPSED = 'sidebar-collapse'

const Default = {

trigger: `${SELECTOR\_DATA\_WIDGET} ${SELECTOR\_LINK}`,

animationSpeed: 300,

accordion: true,

expandSidebar: false,

sidebarButtonSelector: '[data-widget="pushmenu"]'

}

class Treeview {

constructor(element, config) {

this.\_config = config

this.\_element = element

}

\_init() {

$(`${SELECTOR\_LI}${SELECTOR\_OPEN} ${SELECTOR\_TREEVIEW\_MENU}${SELECTOR\_OPEN}`).css('display', 'block')

this.\_setupListeners()

}

expand(treeviewMenu, parentLi) {

const expandedEvent = $.Event(EVENT\_EXPANDED)

if (this.\_config.accordion) {

const openMenuLi = parentLi.siblings(SELECTOR\_OPEN).first()

const openTreeview = openMenuLi.find(SELECTOR\_TREEVIEW\_MENU).first()

this.collapse(openTreeview, openMenuLi)

}

parentLi.addClass(CLASS\_NAME\_IS\_OPENING)

treeviewMenu.stop().slideDown(this.\_config.animationSpeed, () => {

parentLi.addClass(CLASS\_NAME\_OPEN)

$(this.\_element).trigger(expandedEvent)

})

if (this.\_config.expandSidebar) {

this.\_expandSidebar()

}

}

collapse(treeviewMenu, parentLi) {

const collapsedEvent = $.Event(EVENT\_COLLAPSED)

parentLi.removeClass(`${CLASS\_NAME\_IS\_OPENING} ${CLASS\_NAME\_OPEN}`)

treeviewMenu.stop().slideUp(this.\_config.animationSpeed, () => {

$(this.\_element).trigger(collapsedEvent)

treeviewMenu.find(`${SELECTOR\_OPEN} > ${SELECTOR\_TREEVIEW\_MENU}`).slideUp()

treeviewMenu.find(SELECTOR\_OPEN).removeClass(`${CLASS\_NAME\_IS\_OPENING} ${CLASS\_NAME\_OPEN}`)

})

}

toggle(event) {

const $relativeTarget = $(event.currentTarget)

const $parent = $relativeTarget.parent()

let treeviewMenu = $parent.find(`> ${SELECTOR\_TREEVIEW\_MENU}`)

if (!treeviewMenu.is(SELECTOR\_TREEVIEW\_MENU)) {

if (!$parent.is(SELECTOR\_LI)) {

treeviewMenu = $parent.parent().find(`> ${SELECTOR\_TREEVIEW\_MENU}`)

}

if (!treeviewMenu.is(SELECTOR\_TREEVIEW\_MENU)) {

return

}

}

event.preventDefault()

const parentLi = $relativeTarget.parents(SELECTOR\_LI).first()

const isOpen = parentLi.hasClass(CLASS\_NAME\_OPEN)

if (isOpen) {

this.collapse($(treeviewMenu), parentLi)

} else {

this.expand($(treeviewMenu), parentLi)

}

}

\_setupListeners() {

const elementId = this.\_element.attr('id') !== undefined ? `#${this.\_element.attr('id')}` : ''

$(document).on('click', `${elementId}${this.\_config.trigger}`, event => {

this.toggle(event)

})

}

\_expandSidebar() {

if ($('body').hasClass(CLASS\_NAME\_SIDEBAR\_COLLAPSED)) {

$(this.\_config.sidebarButtonSelector).PushMenu('expand')

}

}

static \_jQueryInterface(config) {

return this.each(function () {

let data = $(this).data(DATA\_KEY)

const \_config = $.extend({}, Default, typeof config === 'object' ? config : $(this).data())

if (!data) {

data = new Treeview($(this), \_config)

$(this).data(DATA\_KEY, data)

data.\_init()

} else if (typeof config === 'string') {

if (typeof data[config] === 'undefined') {

throw new TypeError(`No method named "${config}"`)

}

data[config]()

} else if (typeof config === 'undefined') {

data.\_init()

}

})

}

}

$(window).on(EVENT\_LOAD\_DATA\_API, () => {

$(SELECTOR\_DATA\_WIDGET).each(function () {

Treeview.\_jQueryInterface.call($(this), 'init')

})

})

$.fn[NAME] = Treeview.\_jQueryInterface

$.fn[NAME].Constructor = Treeview

$.fn[NAME].noConflict = function () {

$.fn[NAME] = JQUERY\_NO\_CONFLICT

return Treeview.\_jQueryInterface

}

export default Treeview

Footer