import $, { trim } from 'jquery'

const NAME = 'SidebarSearch'

const DATA\_KEY = 'lte.sidebar-search'

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

const CLASS\_NAME\_OPEN = 'sidebar-search-open'

const CLASS\_NAME\_ICON\_SEARCH = 'fa-search'

const CLASS\_NAME\_ICON\_CLOSE = 'fa-times'

const CLASS\_NAME\_HEADER = 'nav-header'

const CLASS\_NAME\_SEARCH\_RESULTS = 'sidebar-search-results'

const CLASS\_NAME\_LIST\_GROUP = 'list-group'

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

const SELECTOR\_SIDEBAR = '.main-sidebar .nav-sidebar'

const SELECTOR\_NAV\_LINK = '.nav-link'

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

const SELECTOR\_SEARCH\_INPUT = `${SELECTOR\_DATA\_WIDGET} .form-control`

const SELECTOR\_SEARCH\_BUTTON = `${SELECTOR\_DATA\_WIDGET} .btn`

const SELECTOR\_SEARCH\_ICON = `${SELECTOR\_SEARCH\_BUTTON} i`

const SELECTOR\_SEARCH\_LIST\_GROUP = `.${CLASS\_NAME\_LIST\_GROUP}`

const SELECTOR\_SEARCH\_RESULTS = `.${CLASS\_NAME\_SEARCH\_RESULTS}`

const SELECTOR\_SEARCH\_RESULTS\_GROUP = `${SELECTOR\_SEARCH\_RESULTS} .${CLASS\_NAME\_LIST\_GROUP}`

const Default = {

arrowSign: '->',

minLength: 3,

maxResults: 7,

highlightName: true,

highlightPath: false,

highlightClass: 'text-light',

notFoundText: 'No element found!'

}

const SearchItems = []

class SidebarSearch {

constructor(\_element, \_options) {

this.element = \_element

this.options = $.extend({}, Default, \_options)

this.items = []

}

// Public

\_init() {

if ($(SELECTOR\_DATA\_WIDGET).length === 0) {

return

}

if ($(SELECTOR\_DATA\_WIDGET).next(SELECTOR\_SEARCH\_RESULTS).length === 0) {

$(SELECTOR\_DATA\_WIDGET).after(

$('<div />', { class: CLASS\_NAME\_SEARCH\_RESULTS })

)

}

if ($(SELECTOR\_SEARCH\_RESULTS).children(SELECTOR\_SEARCH\_LIST\_GROUP).length === 0) {

$(SELECTOR\_SEARCH\_RESULTS).append(

$('<div />', { class: CLASS\_NAME\_LIST\_GROUP })

)

}

this.\_addNotFound()

$(SELECTOR\_SIDEBAR).children().each((i, child) => {

this.\_parseItem(child)

})

}

search() {

const searchValue = $(SELECTOR\_SEARCH\_INPUT).val().toLowerCase()

if (searchValue.length < this.options.minLength) {

$(SELECTOR\_SEARCH\_RESULTS\_GROUP).empty()

this.\_addNotFound()

this.close()

return

}

const searchResults = SearchItems.filter(item => (item.name).toLowerCase().includes(searchValue))

const endResults = $(searchResults.slice(0, this.options.maxResults))

$(SELECTOR\_SEARCH\_RESULTS\_GROUP).empty()

if (endResults.length === 0) {

this.\_addNotFound()

} else {

endResults.each((i, result) => {

$(SELECTOR\_SEARCH\_RESULTS\_GROUP).append(this.\_renderItem(escape(result.name), encodeURI(result.link), result.path))

})

}

this.open()

}

open() {

$(SELECTOR\_DATA\_WIDGET).parent().addClass(CLASS\_NAME\_OPEN)

$(SELECTOR\_SEARCH\_ICON).removeClass(CLASS\_NAME\_ICON\_SEARCH).addClass(CLASS\_NAME\_ICON\_CLOSE)

}

close() {

$(SELECTOR\_DATA\_WIDGET).parent().removeClass(CLASS\_NAME\_OPEN)

$(SELECTOR\_SEARCH\_ICON).removeClass(CLASS\_NAME\_ICON\_CLOSE).addClass(CLASS\_NAME\_ICON\_SEARCH)

}

toggle() {

if ($(SELECTOR\_DATA\_WIDGET).parent().hasClass(CLASS\_NAME\_OPEN)) {

this.close()

} else {

this.open()

}

}

// Private

\_parseItem(item, path = []) {

if ($(item).hasClass(CLASS\_NAME\_HEADER)) {

return

}

const itemObject = {}

const navLink = $(item).clone().find(`> ${SELECTOR\_NAV\_LINK}`)

const navTreeview = $(item).clone().find(`> ${SELECTOR\_NAV\_TREEVIEW}`)

const link = navLink.attr('href')

const name = navLink.find('p').children().remove().end().text()

itemObject.name = this.\_trimText(name)

itemObject.link = link

itemObject.path = path

if (navTreeview.length === 0) {

SearchItems.push(itemObject)

} else {

const newPath = itemObject.path.concat([itemObject.name])

navTreeview.children().each((i, child) => {

this.\_parseItem(child, newPath)

})

}

}

\_trimText(text) {

return trim(text.replace(/(\r\n|\n|\r)/gm, ' '))

}

\_renderItem(name, link, path) {

path = path.join(` ${this.options.arrowSign} `)

name = unescape(name)

link = decodeURI(link)

if (this.options.highlightName || this.options.highlightPath) {

const searchValue = $(SELECTOR\_SEARCH\_INPUT).val().toLowerCase()

const regExp = new RegExp(searchValue, 'gi')

if (this.options.highlightName) {

name = name.replace(

regExp,

str => {

return `<strong class="${this.options.highlightClass}">${str}</strong>`

}

)

}

if (this.options.highlightPath) {

path = path.replace(

regExp,

str => {

return `<strong class="${this.options.highlightClass}">${str}</strong>`

}

)

}

}

const groupItemElement = $('<a/>', {

href: decodeURIComponent(link),

class: 'list-group-item'

})

const searchTitleElement = $('<div/>', {

class: 'search-title'

}).html(name)

const searchPathElement = $('<div/>', {

class: 'search-path'

}).html(path)

groupItemElement.append(searchTitleElement).append(searchPathElement)

return groupItemElement

}

\_addNotFound() {

$(SELECTOR\_SEARCH\_RESULTS\_GROUP).append(this.\_renderItem(this.options.notFoundText, '#', []))

}

// Static

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 SidebarSearch($(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()

}

})

}

}

$(document).on('click', SELECTOR\_SEARCH\_BUTTON, event => {

event.preventDefault()

SidebarSearch.\_jQueryInterface.call($(SELECTOR\_DATA\_WIDGET), 'toggle')

})

$(document).on('keyup', SELECTOR\_SEARCH\_INPUT, event => {

if (event.keyCode == 38) {

event.preventDefault()

$(SELECTOR\_SEARCH\_RESULTS\_GROUP).children().last().focus()

return

}

if (event.keyCode == 40) {

event.preventDefault()

$(SELECTOR\_SEARCH\_RESULTS\_GROUP).children().first().focus()

return

}

setTimeout(() => {

SidebarSearch.\_jQueryInterface.call($(SELECTOR\_DATA\_WIDGET), 'search')

}, 100)

})

$(document).on('keydown', SELECTOR\_SEARCH\_RESULTS\_GROUP, event => {

const $focused = $(':focus')

if (event.keyCode == 38) {

event.preventDefault()

if ($focused.is(':first-child')) {

$focused.siblings().last().focus()

} else {

$focused.prev().focus()

}

}

if (event.keyCode == 40) {

event.preventDefault()

if ($focused.is(':last-child')) {

$focused.siblings().first().focus()

} else {

$focused.next().focus()

}

}

})

$(window).on('load', () => {

SidebarSearch.\_jQueryInterface.call($(SELECTOR\_DATA\_WIDGET), 'init')

})

$.fn[NAME] = SidebarSearch.\_jQueryInterface

$.fn[NAME].Constructor = SidebarSearch

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

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

return SidebarSearch.\_jQueryInterface

}

export default SidebarSearch

Footer