import $ from 'jquery'

const NAME = 'NavbarSearch'

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

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

const SELECTOR\_TOGGLE\_BUTTON = '[data-widget="navbar-search"]'

const SELECTOR\_SEARCH\_BLOCK = '.navbar-search-block'

const SELECTOR\_SEARCH\_INPUT = '.form-control'

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

const Default = {

resetOnClose: true,

target: SELECTOR\_SEARCH\_BLOCK

}

class NavbarSearch {

constructor(\_element, \_options) {

this.\_element = \_element

this.\_config = \_options

}

// Public

open() {

$(this.\_config.target).css('display', 'flex').hide().fadeIn().addClass(CLASS\_NAME\_OPEN)

$(`${this.\_config.target} ${SELECTOR\_SEARCH\_INPUT}`).focus()

}

close() {

$(this.\_config.target).fadeOut().removeClass(CLASS\_NAME\_OPEN)

if (this.\_config.resetOnClose) {

$(`${this.\_config.target} ${SELECTOR\_SEARCH\_INPUT}`).val('')

}

}

toggle() {

if ($(this.\_config.target).hasClass(CLASS\_NAME\_OPEN)) {

this.close()

} else {

this.open()

}

}

// 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 NavbarSearch($(this), \_config)

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

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

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

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

}

data[config]()

}

})

}

}

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

event.preventDefault()

let button = $(event.currentTarget)

if (button.data('widget') !== 'navbar-search') {

button = button.closest(SELECTOR\_TOGGLE\_BUTTON)

}

NavbarSearch.\_jQueryInterface.call(button, 'toggle')

})

$.fn[NAME] = NavbarSearch.\_jQueryInterface

$.fn[NAME].Constructor = NavbarSearch

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

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

return NavbarSearch.\_jQueryInterface

}

export default NavbarSearch

Footer