import $ from 'jquery'

const NAME = 'Dropdown'

const DATA\_KEY = 'lte.dropdown'

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

const SELECTOR\_NAVBAR = '.navbar'

const SELECTOR\_DROPDOWN\_MENU = '.dropdown-menu'

const SELECTOR\_DROPDOWN\_MENU\_ACTIVE = '.dropdown-menu.show'

const SELECTOR\_DROPDOWN\_TOGGLE = '[data-toggle="dropdown"]'

const CLASS\_NAME\_DROPDOWN\_RIGHT = 'dropdown-menu-right'

const CLASS\_NAME\_DROPDOWN\_SUBMENU = 'dropdown-submenu'

const Default = {}

class Dropdown {

constructor(element, config) {

this.\_config = config

this.\_element = element

}

toggleSubmenu() {

this.\_element.siblings().show().toggleClass('show')

if (!this.\_element.next().hasClass('show')) {

this.\_element.parents(SELECTOR\_DROPDOWN\_MENU).first().find('.show').removeClass('show').hide()

}

this.\_element.parents('li.nav-item.dropdown.show').on('hidden.bs.dropdown', () => {

$('.dropdown-submenu .show').removeClass('show').hide()

})

}

fixPosition() {

const $element = $(SELECTOR\_DROPDOWN\_MENU\_ACTIVE)

if ($element.length === 0) {

return

}

if ($element.hasClass(CLASS\_NAME\_DROPDOWN\_RIGHT)) {

$element.css({

left: 'inherit',

right: 0

})

} else {

$element.css({

left: 0,

right: 'inherit'

})

}

const offset = $element.offset()

const width = $element.width()

const visiblePart = $(window).width() - offset.left

if (offset.left < 0) {

$element.css({

left: 'inherit',

right: offset.left - 5

})

} else if (visiblePart < width) {

$element.css({

left: 'inherit',

right: 0

})

}

}

// 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 Dropdown($(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]()

}

})

}

}

$(`${SELECTOR\_DROPDOWN\_MENU} ${SELECTOR\_DROPDOWN\_TOGGLE}`).on('click', function (event) {

event.preventDefault()

event.stopPropagation()

Dropdown.\_jQueryInterface.call($(this), 'toggleSubmenu')

})

$(`${SELECTOR\_NAVBAR} ${SELECTOR\_DROPDOWN\_TOGGLE}`).on('click', event => {

event.preventDefault()

if ($(event.target).parent().hasClass(CLASS\_NAME\_DROPDOWN\_SUBMENU)) {

return

}

setTimeout(function () {

Dropdown.\_jQueryInterface.call($(this), 'fixPosition')

}, 1)

})

$.fn[NAME] = Dropdown.\_jQueryInterface

$.fn[NAME].Constructor = Dropdown

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

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

return Dropdown.\_jQueryInterface

}

export default Dropdown

Footer