import $ from 'jquery'

const NAME = 'Toasts'

const DATA\_KEY = 'lte.toasts'

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

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

const EVENT\_INIT = `init${EVENT\_KEY}`

const EVENT\_CREATED = `created${EVENT\_KEY}`

const EVENT\_REMOVED = `removed${EVENT\_KEY}`

const SELECTOR\_CONTAINER\_TOP\_RIGHT = '#toastsContainerTopRight'

const SELECTOR\_CONTAINER\_TOP\_LEFT = '#toastsContainerTopLeft'

const SELECTOR\_CONTAINER\_BOTTOM\_RIGHT = '#toastsContainerBottomRight'

const SELECTOR\_CONTAINER\_BOTTOM\_LEFT = '#toastsContainerBottomLeft'

const CLASS\_NAME\_TOP\_RIGHT = 'toasts-top-right'

const CLASS\_NAME\_TOP\_LEFT = 'toasts-top-left'

const CLASS\_NAME\_BOTTOM\_RIGHT = 'toasts-bottom-right'

const CLASS\_NAME\_BOTTOM\_LEFT = 'toasts-bottom-left'

const POSITION\_TOP\_RIGHT = 'topRight'

const POSITION\_TOP\_LEFT = 'topLeft'

const POSITION\_BOTTOM\_RIGHT = 'bottomRight'

const POSITION\_BOTTOM\_LEFT = 'bottomLeft'

const Default = {

position: POSITION\_TOP\_RIGHT,

fixed: true,

autohide: false,

autoremove: true,

delay: 1000,

fade: true,

icon: null,

image: null,

imageAlt: null,

imageHeight: '25px',

title: null,

subtitle: null,

close: true,

body: null,

class: null

}

class Toasts {

constructor(element, config) {

this.\_config = config

this.\_prepareContainer()

$('body').trigger($.Event(EVENT\_INIT))

}

// Public

create() {

const toast = $('<div class="toast" role="alert" aria-live="assertive" aria-atomic="true"/>')

toast.data('autohide', this.\_config.autohide)

toast.data('animation', this.\_config.fade)

if (this.\_config.class) {

toast.addClass(this.\_config.class)

}

if (this.\_config.delay && this.\_config.delay != 500) {

toast.data('delay', this.\_config.delay)

}

const toastHeader = $('<div class="toast-header">')

if (this.\_config.image != null) {

const toastImage = $('<img />').addClass('rounded mr-2').attr('src', this.\_config.image).attr('alt', this.\_config.imageAlt)

if (this.\_config.imageHeight != null) {

toastImage.height(this.\_config.imageHeight).width('auto')

}

toastHeader.append(toastImage)

}

if (this.\_config.icon != null) {

toastHeader.append($('<i />').addClass('mr-2').addClass(this.\_config.icon))

}

if (this.\_config.title != null) {

toastHeader.append($('<strong />').addClass('mr-auto').html(this.\_config.title))

}

if (this.\_config.subtitle != null) {

toastHeader.append($('<small />').html(this.\_config.subtitle))

}

if (this.\_config.close == true) {

const toastClose = $('<button data-dismiss="toast" />').attr('type', 'button').addClass('ml-2 mb-1 close').attr('aria-label', 'Close').append('<span aria-hidden="true">&times;</span>')

if (this.\_config.title == null) {

toastClose.toggleClass('ml-2 ml-auto')

}

toastHeader.append(toastClose)

}

toast.append(toastHeader)

if (this.\_config.body != null) {

toast.append($('<div class="toast-body" />').html(this.\_config.body))

}

$(this.\_getContainerId()).prepend(toast)

const $body = $('body')

$body.trigger($.Event(EVENT\_CREATED))

toast.toast('show')

if (this.\_config.autoremove) {

toast.on('hidden.bs.toast', function () {

$(this).delay(200).remove()

$body.trigger($.Event(EVENT\_REMOVED))

})

}

}

// Static

\_getContainerId() {

if (this.\_config.position === POSITION\_TOP\_RIGHT) {

return SELECTOR\_CONTAINER\_TOP\_RIGHT

}

if (this.\_config.position === POSITION\_TOP\_LEFT) {

return SELECTOR\_CONTAINER\_TOP\_LEFT

}

if (this.\_config.position === POSITION\_BOTTOM\_RIGHT) {

return SELECTOR\_CONTAINER\_BOTTOM\_RIGHT

}

if (this.\_config.position === POSITION\_BOTTOM\_LEFT) {

return SELECTOR\_CONTAINER\_BOTTOM\_LEFT

}

}

\_prepareContainer() {

if ($(this.\_getContainerId()).length === 0) {

const container = $('<div />').attr('id', this.\_getContainerId().replace('#', ''))

if (this.\_config.position == POSITION\_TOP\_RIGHT) {

container.addClass(CLASS\_NAME\_TOP\_RIGHT)

} else if (this.\_config.position == POSITION\_TOP\_LEFT) {

container.addClass(CLASS\_NAME\_TOP\_LEFT)

} else if (this.\_config.position == POSITION\_BOTTOM\_RIGHT) {

container.addClass(CLASS\_NAME\_BOTTOM\_RIGHT)

} else if (this.\_config.position == POSITION\_BOTTOM\_LEFT) {

container.addClass(CLASS\_NAME\_BOTTOM\_LEFT)

}

$('body').append(container)

}

if (this.\_config.fixed) {

$(this.\_getContainerId()).addClass('fixed')

} else {

$(this.\_getContainerId()).removeClass('fixed')

}

}

// Static

static \_jQueryInterface(option, config) {

return this.each(function () {

const \_options = $.extend({}, Default, config)

const toast = new Toasts($(this), \_options)

if (option === 'create') {

toast[option]()

}

})

}

}

$.fn[NAME] = Toasts.\_jQueryInterface

$.fn[NAME].Constructor = Toasts

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

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

return Toasts.\_jQueryInterface

}

export default Toasts

Footer