import $ from 'jquery'

const NAME = 'Fullscreen'

const DATA\_KEY = 'lte.fullscreen'

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

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

const SELECTOR\_ICON = `${SELECTOR\_DATA\_WIDGET} i`

const EVENT\_FULLSCREEN\_CHANGE = 'webkitfullscreenchange mozfullscreenchange fullscreenchange MSFullscreenChange'

const Default = {

minimizeIcon: 'fa-compress-arrows-alt',

maximizeIcon: 'fa-expand-arrows-alt'

}

class Fullscreen {

constructor(\_element, \_options) {

this.element = \_element

this.options = \_options

}

// Public

toggle() {

if (document.fullscreenElement ||

document.mozFullScreenElement ||

document.webkitFullscreenElement ||

document.msFullscreenElement) {

this.windowed()

} else {

this.fullscreen()

}

}

toggleIcon() {

if (document.fullscreenElement ||

document.mozFullScreenElement ||

document.webkitFullscreenElement ||

document.msFullscreenElement) {

$(SELECTOR\_ICON).removeClass(this.options.maximizeIcon).addClass(this.options.minimizeIcon)

} else {

$(SELECTOR\_ICON).removeClass(this.options.minimizeIcon).addClass(this.options.maximizeIcon)

}

}

fullscreen() {

if (document.documentElement.requestFullscreen) {

document.documentElement.requestFullscreen()

} else if (document.documentElement.webkitRequestFullscreen) {

document.documentElement.webkitRequestFullscreen()

} else if (document.documentElement.msRequestFullscreen) {

document.documentElement.msRequestFullscreen()

}

}

windowed() {

if (document.exitFullscreen) {

document.exitFullscreen()

} else if (document.webkitExitFullscreen) {

document.webkitExitFullscreen()

} else if (document.msExitFullscreen) {

document.msExitFullscreen()

}

}

// 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 Fullscreen($(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\_DATA\_WIDGET, function () {

Fullscreen.\_jQueryInterface.call($(this), 'toggle')

})

$(document).on(EVENT\_FULLSCREEN\_CHANGE, () => {

Fullscreen.\_jQueryInterface.call($(SELECTOR\_DATA\_WIDGET), 'toggleIcon')

})

$.fn[NAME] = Fullscreen.\_jQueryInterface

$.fn[NAME].Constructor = Fullscreen

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

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

return Fullscreen.\_jQueryInterface

}

export default Fullscreen

Footer