import $ from 'jquery'

const NAME = 'IFrame'

const DATA\_KEY = 'lte.iframe'

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

const SELECTOR\_DATA\_TOGGLE = '[data-widget="iframe"]'

const SELECTOR\_DATA\_TOGGLE\_CLOSE = '[data-widget="iframe-close"]'

const SELECTOR\_DATA\_TOGGLE\_SCROLL\_LEFT = '[data-widget="iframe-scrollleft"]'

const SELECTOR\_DATA\_TOGGLE\_SCROLL\_RIGHT = '[data-widget="iframe-scrollright"]'

const SELECTOR\_DATA\_TOGGLE\_FULLSCREEN = '[data-widget="iframe-fullscreen"]'

const SELECTOR\_CONTENT\_WRAPPER = '.content-wrapper'

const SELECTOR\_CONTENT\_IFRAME = `${SELECTOR\_CONTENT\_WRAPPER} iframe`

const SELECTOR\_TAB\_NAV = `${SELECTOR\_CONTENT\_WRAPPER}.iframe-mode .nav`

const SELECTOR\_TAB\_NAVBAR\_NAV = `${SELECTOR\_CONTENT\_WRAPPER}.iframe-mode .navbar-nav`

const SELECTOR\_TAB\_NAVBAR\_NAV\_ITEM = `${SELECTOR\_TAB\_NAVBAR\_NAV} .nav-item`

const SELECTOR\_TAB\_NAVBAR\_NAV\_LINK = `${SELECTOR\_TAB\_NAVBAR\_NAV} .nav-link`

const SELECTOR\_TAB\_CONTENT = `${SELECTOR\_CONTENT\_WRAPPER}.iframe-mode .tab-content`

const SELECTOR\_TAB\_EMPTY = `${SELECTOR\_TAB\_CONTENT} .tab-empty`

const SELECTOR\_TAB\_LOADING = `${SELECTOR\_TAB\_CONTENT} .tab-loading`

const SELECTOR\_TAB\_PANE = `${SELECTOR\_TAB\_CONTENT} .tab-pane`

const SELECTOR\_SIDEBAR\_MENU\_ITEM = '.main-sidebar .nav-item > a.nav-link'

const SELECTOR\_SIDEBAR\_SEARCH\_ITEM = '.sidebar-search-results .list-group-item'

const SELECTOR\_HEADER\_MENU\_ITEM = '.main-header .nav-item a.nav-link'

const SELECTOR\_HEADER\_DROPDOWN\_ITEM = '.main-header a.dropdown-item'

const CLASS\_NAME\_IFRAME\_MODE = 'iframe-mode'

const CLASS\_NAME\_FULLSCREEN\_MODE = 'iframe-mode-fullscreen'

const Default = {

onTabClick(item) {

return item

},

onTabChanged(item) {

return item

},

onTabCreated(item) {

return item

},

autoIframeMode: true,

autoItemActive: true,

autoShowNewTab: true,

autoDarkMode: false,

allowDuplicates: false,

allowReload: true,

loadingScreen: true,

useNavbarItems: true,

scrollOffset: 40,

scrollBehaviorSwap: false,

iconMaximize: 'fa-expand',

iconMinimize: 'fa-compress'

}

class IFrame {

constructor(element, config) {

this.\_config = config

this.\_element = element

this.\_init()

}

// Public

onTabClick(item) {

this.\_config.onTabClick(item)

}

onTabChanged(item) {

this.\_config.onTabChanged(item)

}

onTabCreated(item) {

this.\_config.onTabCreated(item)

}

createTab(title, link, uniqueName, autoOpen) {

let tabId = `panel-${uniqueName}`

let navId = `tab-${uniqueName}`

if (this.\_config.allowDuplicates) {

tabId += `-${Math.floor(Math.random() \* 1000)}`

navId += `-${Math.floor(Math.random() \* 1000)}`

}

const newNavItem = `<li class="nav-item" role="presentation"><a href="#" class="btn-iframe-close" data-widget="iframe-close" data-type="only-this"><i class="fas fa-times"></i></a><a class="nav-link" data-toggle="row" id="${navId}" href="#${tabId}" role="tab" aria-controls="${tabId}" aria-selected="false">${title}</a></li>`

$(SELECTOR\_TAB\_NAVBAR\_NAV).append(unescape(escape(newNavItem)))

const newTabItem = `<div class="tab-pane fade" id="${tabId}" role="tabpanel" aria-labelledby="${navId}"><iframe src="${link}"></iframe></div>`

$(SELECTOR\_TAB\_CONTENT).append(unescape(escape(newTabItem)))

if (autoOpen) {

if (this.\_config.loadingScreen) {

const $loadingScreen = $(SELECTOR\_TAB\_LOADING)

if (!$loadingScreen.is(':animated')) {

$loadingScreen.fadeIn()

}

$(`${tabId} iframe`).ready(() => {

if (typeof this.\_config.loadingScreen === 'number') {

this.switchTab(`#${navId}`)

setTimeout(() => {

$loadingScreen.fadeOut()

}, this.\_config.loadingScreen)

} else {

this.switchTab(`#${navId}`)

$loadingScreen.fadeOut()

}

})

} else {

this.switchTab(`#${navId}`)

}

}

this.onTabCreated($(`#${navId}`))

}

openTabSidebar(item, autoOpen = this.\_config.autoShowNewTab) {

let $item = $(item).clone()

if ($item.attr('href') === undefined) {

$item = $(item).parent('a').clone()

}

$item.find('.right, .search-path').remove()

let title = $item.find('p').text()

if (title === '') {

title = $item.text()

}

const link = $item.attr('href')

if (link === '#' || link === '' || link === undefined) {

return

}

const uniqueName = unescape(link).replace('./', '').replace(/["#&'./:=?[\]]/gi, '-').replace(/(--)/gi, '')

const navId = `tab-${uniqueName}`

if (!this.\_config.allowDuplicates && $(`#${navId}`).length > 0) {

return this.switchTab(`#${navId}`, this.\_config.allowReload)

}

if ((!this.\_config.allowDuplicates && $(`#${navId}`).length === 0) || this.\_config.allowDuplicates) {

this.createTab(title, link, uniqueName, autoOpen)

}

}

switchTab(item, reload = false) {

const $item = $(item)

const tabId = $item.attr('href')

$(SELECTOR\_TAB\_EMPTY).hide()

if (reload) {

const $loadingScreen = $(SELECTOR\_TAB\_LOADING)

if (this.\_config.loadingScreen) {

$loadingScreen.show(0, () => {

$(`${tabId} iframe`).attr('src', $(`${tabId} iframe`).attr('src')).ready(() => {

if (this.\_config.loadingScreen) {

if (typeof this.\_config.loadingScreen === 'number') {

setTimeout(() => {

$loadingScreen.fadeOut()

}, this.\_config.loadingScreen)

} else {

$loadingScreen.fadeOut()

}

}

})

})

} else {

$(`${tabId} iframe`).attr('src', $(`${tabId} iframe`).attr('src'))

}

}

$(`${SELECTOR\_TAB\_NAVBAR\_NAV} .active`).tab('dispose').removeClass('active')

this.\_fixHeight()

$item.tab('show')

$item.parents('li').addClass('active')

this.onTabChanged($item)

if (this.\_config.autoItemActive) {

this.\_setItemActive($(`${tabId} iframe`).attr('src'))

}

}

removeActiveTab(type, element) {

if (type == 'all') {

$(SELECTOR\_TAB\_NAVBAR\_NAV\_ITEM).remove()

$(SELECTOR\_TAB\_PANE).remove()

$(SELECTOR\_TAB\_EMPTY).show()

} else if (type == 'all-other') {

$(`${SELECTOR\_TAB\_NAVBAR\_NAV\_ITEM}:not(.active)`).remove()

$(`${SELECTOR\_TAB\_PANE}:not(.active)`).remove()

} else if (type == 'only-this') {

const $navClose = $(element)

const $navItem = $navClose.parent('.nav-item')

const $navItemParent = $navItem.parent()

const navItemIndex = $navItem.index()

const tabId = $navClose.siblings('.nav-link').attr('aria-controls')

$navItem.remove()

$(`#${tabId}`).remove()

if ($(SELECTOR\_TAB\_CONTENT).children().length == $(`${SELECTOR\_TAB\_EMPTY}, ${SELECTOR\_TAB\_LOADING}`).length) {

$(SELECTOR\_TAB\_EMPTY).show()

} else {

const prevNavItemIndex = navItemIndex - 1

this.switchTab($navItemParent.children().eq(prevNavItemIndex).find('a.nav-link'))

}

} else {

const $navItem = $(`${SELECTOR\_TAB\_NAVBAR\_NAV\_ITEM}.active`)

const $navItemParent = $navItem.parent()

const navItemIndex = $navItem.index()

$navItem.remove()

$(`${SELECTOR\_TAB\_PANE}.active`).remove()

if ($(SELECTOR\_TAB\_CONTENT).children().length == $(`${SELECTOR\_TAB\_EMPTY}, ${SELECTOR\_TAB\_LOADING}`).length) {

$(SELECTOR\_TAB\_EMPTY).show()

} else {

const prevNavItemIndex = navItemIndex - 1

this.switchTab($navItemParent.children().eq(prevNavItemIndex).find('a.nav-link'))

}

}

}

toggleFullscreen() {

if ($('body').hasClass(CLASS\_NAME\_FULLSCREEN\_MODE)) {

$(`${SELECTOR\_DATA\_TOGGLE\_FULLSCREEN} i`).removeClass(this.\_config.iconMinimize).addClass(this.\_config.iconMaximize)

$('body').removeClass(CLASS\_NAME\_FULLSCREEN\_MODE)

$(`${SELECTOR\_TAB\_EMPTY}, ${SELECTOR\_TAB\_LOADING}`).height('100%')

$(SELECTOR\_CONTENT\_WRAPPER).height('100%')

$(SELECTOR\_CONTENT\_IFRAME).height('100%')

} else {

$(`${SELECTOR\_DATA\_TOGGLE\_FULLSCREEN} i`).removeClass(this.\_config.iconMaximize).addClass(this.\_config.iconMinimize)

$('body').addClass(CLASS\_NAME\_FULLSCREEN\_MODE)

}

$(window).trigger('resize')

this.\_fixHeight(true)

}

// Private

\_init() {

const usingDefTab = ($(SELECTOR\_TAB\_CONTENT).children().length > 2)

this.\_setupListeners()

this.\_fixHeight(true)

if (usingDefTab) {

const $el = $(`${SELECTOR\_TAB\_PANE}`).first()

const uniqueName = $el.attr('id').replace('panel-', '')

const navId = `#tab-${uniqueName}`

this.switchTab(navId, true)

}

}

\_initFrameElement() {

if (window.frameElement && this.\_config.autoIframeMode) {

const $body = $('body')

$body.addClass(CLASS\_NAME\_IFRAME\_MODE)

if (this.\_config.autoDarkMode) {

$body.addClass('dark-mode')

}

}

}

\_navScroll(offset) {

const leftPos = $(SELECTOR\_TAB\_NAVBAR\_NAV).scrollLeft()

$(SELECTOR\_TAB\_NAVBAR\_NAV).animate({ scrollLeft: (leftPos + offset) }, 250, 'linear')

}

\_setupListeners() {

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

setTimeout(() => {

this.\_fixHeight()

}, 1)

})

if ($(SELECTOR\_CONTENT\_WRAPPER).hasClass(CLASS\_NAME\_IFRAME\_MODE)) {

$(document).on('click', `${SELECTOR\_SIDEBAR\_MENU\_ITEM}, ${SELECTOR\_SIDEBAR\_SEARCH\_ITEM}`, e => {

e.preventDefault()

this.openTabSidebar(e.target)

})

if (this.\_config.useNavbarItems) {

$(document).on('click', `${SELECTOR\_HEADER\_MENU\_ITEM}, ${SELECTOR\_HEADER\_DROPDOWN\_ITEM}`, e => {

e.preventDefault()

this.openTabSidebar(e.target)

})

}

}

$(document).on('click', SELECTOR\_TAB\_NAVBAR\_NAV\_LINK, e => {

e.preventDefault()

this.onTabClick(e.target)

this.switchTab(e.target)

})

$(document).on('click', SELECTOR\_TAB\_NAVBAR\_NAV\_LINK, e => {

e.preventDefault()

this.onTabClick(e.target)

this.switchTab(e.target)

})

$(document).on('click', SELECTOR\_DATA\_TOGGLE\_CLOSE, e => {

e.preventDefault()

let { target } = e

if (target.nodeName === 'I') {

target = e.target.offsetParent

}

this.removeActiveTab(target.attributes['data-type'] ? target.attributes['data-type'].nodeValue : null, target)

})

$(document).on('click', SELECTOR\_DATA\_TOGGLE\_FULLSCREEN, e => {

e.preventDefault()

this.toggleFullscreen()

})

let mousedown = false

let mousedownInterval = null

$(document).on('mousedown', SELECTOR\_DATA\_TOGGLE\_SCROLL\_LEFT, e => {

e.preventDefault()

clearInterval(mousedownInterval)

let { scrollOffset } = this.\_config

if (!this.\_config.scrollBehaviorSwap) {

scrollOffset = -scrollOffset

}

mousedown = true

this.\_navScroll(scrollOffset)

mousedownInterval = setInterval(() => {

this.\_navScroll(scrollOffset)

}, 250)

})

$(document).on('mousedown', SELECTOR\_DATA\_TOGGLE\_SCROLL\_RIGHT, e => {

e.preventDefault()

clearInterval(mousedownInterval)

let { scrollOffset } = this.\_config

if (this.\_config.scrollBehaviorSwap) {

scrollOffset = -scrollOffset

}

mousedown = true

this.\_navScroll(scrollOffset)

mousedownInterval = setInterval(() => {

this.\_navScroll(scrollOffset)

}, 250)

})

$(document).on('mouseup', () => {

if (mousedown) {

mousedown = false

clearInterval(mousedownInterval)

mousedownInterval = null

}

})

}

\_setItemActive(href) {

$(`${SELECTOR\_SIDEBAR\_MENU\_ITEM}, ${SELECTOR\_HEADER\_DROPDOWN\_ITEM}`).removeClass('active')

$(SELECTOR\_HEADER\_MENU\_ITEM).parent().removeClass('active')

const $headerMenuItem = $(`${SELECTOR\_HEADER\_MENU\_ITEM}[href$="${href}"]`)

const $headerDropdownItem = $(`${SELECTOR\_HEADER\_DROPDOWN\_ITEM}[href$="${href}"]`)

const $sidebarMenuItem = $(`${SELECTOR\_SIDEBAR\_MENU\_ITEM}[href$="${href}"]`)

$headerMenuItem.each((i, e) => {

$(e).parent().addClass('active')

})

$headerDropdownItem.each((i, e) => {

$(e).addClass('active')

})

$sidebarMenuItem.each((i, e) => {

$(e).addClass('active')

$(e).parents('.nav-treeview').prevAll('.nav-link').addClass('active')

})

}

\_fixHeight(tabEmpty = false) {

if ($('body').hasClass(CLASS\_NAME\_FULLSCREEN\_MODE)) {

const windowHeight = $(window).height()

const navbarHeight = $(SELECTOR\_TAB\_NAV).outerHeight()

$(`${SELECTOR\_TAB\_EMPTY}, ${SELECTOR\_TAB\_LOADING}, ${SELECTOR\_CONTENT\_IFRAME}`).height(windowHeight - navbarHeight)

$(SELECTOR\_CONTENT\_WRAPPER).height(windowHeight)

} else {

const contentWrapperHeight = parseFloat($(SELECTOR\_CONTENT\_WRAPPER).css('height'))

const navbarHeight = $(SELECTOR\_TAB\_NAV).outerHeight()

if (tabEmpty == true) {

setTimeout(() => {

$(`${SELECTOR\_TAB\_EMPTY}, ${SELECTOR\_TAB\_LOADING}`).height(contentWrapperHeight - navbarHeight)

}, 50)

} else {

$(SELECTOR\_CONTENT\_IFRAME).height(contentWrapperHeight - navbarHeight)

}

}

}

// Static

// eslint-disable-next-line max-params

static \_jQueryInterface(config, name, link, id, reload) {

if ($(SELECTOR\_DATA\_TOGGLE).length > 0) {

let data = $(this).data(DATA\_KEY)

if (!data) {

data = $(this).data()

}

const \_options = $.extend({}, Default, typeof config === 'object' ? config : data)

localStorage.setItem('AdminLTE:IFrame:Options', JSON.stringify(\_options))

const plugin = new IFrame($(this), \_options)

window.iFrameInstance = plugin

$(this).data(DATA\_KEY, typeof config === 'object' ? config : { link, name, id, reload, ...data })

if (typeof config === 'string' && /createTab|openTabSidebar|switchTab|removeActiveTab/.test(config)) {

plugin[config](name, link, id, reload)

}

} else {

window.iFrameInstance = new IFrame($(this), JSON.parse(localStorage.getItem('AdminLTE:IFrame:Options'))).\_initFrameElement()

}

}

}

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

IFrame.\_jQueryInterface.call($(SELECTOR\_DATA\_TOGGLE))

})

$.fn[NAME] = IFrame.\_jQueryInterface

$.fn[NAME].Constructor = IFrame

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

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

return IFrame.\_jQueryInterface

}

export default IFrame