/\* =========================================================

\* bootstrap-modal.js v1.4.0

\* http://twitter.github.com/bootstrap/javascript.html#modal

\* =========================================================

\* Copyright 2011 Twitter, Inc.

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* http://www.apache.org/licenses/LICENSE-2.0

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\* ========================================================= \*/

!function( $ ){

"use strict"

/\* CSS TRANSITION SUPPORT (https://gist.github.com/373874)

\* ======================================================= \*/

var transitionEnd

$(document).ready(function () {

$.support.transition = (function () {

var thisBody = document.body || document.documentElement

, thisStyle = thisBody.style

, support = thisStyle.transition !== undefined || thisStyle.WebkitTransition !== undefined || thisStyle.MozTransition !== undefined || thisStyle.MsTransition !== undefined || thisStyle.OTransition !== undefined

return support

})()

// set CSS transition event type

if ( $.support.transition ) {

transitionEnd = "TransitionEnd"

if ( $.browser.webkit ) {

transitionEnd = "webkitTransitionEnd"

} else if ( $.browser.mozilla ) {

transitionEnd = "transitionend"

} else if ( $.browser.opera ) {

transitionEnd = "oTransitionEnd"

}

}

})

/\* MODAL PUBLIC CLASS DEFINITION

\* ============================= \*/

var Modal = function ( content, options ) {

this.settings = $.extend({}, $.fn.modal.defaults, options)

this.$element = $(content)

.delegate('.close', 'click.modal', $.proxy(this.hide, this))

if ( this.settings.show ) {

this.show()

}

return this

}

Modal.prototype = {

toggle: function () {

return this[!this.isShown ? 'show' : 'hide']()

}

, show: function () {

var that = this

this.isShown = true

this.$element.trigger('show')

escape.call(this)

backdrop.call(this, function () {

var transition = $.support.transition && that.$element.hasClass('fade')

that.$element

.appendTo(document.body)

.show()

if (transition) {

that.$element[0].offsetWidth // force reflow

}

that.$element.addClass('in')

transition ?

that.$element.one(transitionEnd, function () { that.$element.trigger('shown') }) :

that.$element.trigger('shown')

})

return this

}

, hide: function (e) {

e && e.preventDefault()

if ( !this.isShown ) {

return this

}

var that = this

this.isShown = false

escape.call(this)

this.$element

.trigger('hide')

.removeClass('in')

$.support.transition && this.$element.hasClass('fade') ?

hideWithTransition.call(this) :

hideModal.call(this)

return this

}

}

/\* MODAL PRIVATE METHODS

\* ===================== \*/

function hideWithTransition() {

// firefox drops transitionEnd events :{o

var that = this

, timeout = setTimeout(function () {

that.$element.unbind(transitionEnd)

hideModal.call(that)

}, 500)

this.$element.one(transitionEnd, function () {

clearTimeout(timeout)

hideModal.call(that)

})

}

function hideModal (that) {

this.$element

.hide()

.trigger('hidden')

backdrop.call(this)

}

function backdrop ( callback ) {

var that = this

, animate = this.$element.hasClass('fade') ? 'fade' : ''

if ( this.isShown && this.settings.backdrop ) {

var doAnimate = $.support.transition && animate

this.$backdrop = $('<div class="modal-backdrop ' + animate + '" />')

.appendTo(document.body)

if ( this.settings.backdrop != 'static' ) {

this.$backdrop.click($.proxy(this.hide, this))

}

if ( doAnimate ) {

this.$backdrop[0].offsetWidth // force reflow

}

this.$backdrop.addClass('in')

doAnimate ?

this.$backdrop.one(transitionEnd, callback) :

callback()

} else if ( !this.isShown && this.$backdrop ) {

this.$backdrop.removeClass('in')

$.support.transition && this.$element.hasClass('fade')?

this.$backdrop.one(transitionEnd, $.proxy(removeBackdrop, this)) :

removeBackdrop.call(this)

} else if ( callback ) {

callback()

}

}

function removeBackdrop() {

this.$backdrop.remove()

this.$backdrop = null

}

function escape() {

var that = this

if ( this.isShown && this.settings.keyboard ) {

$(document).bind('keyup.modal', function ( e ) {

if ( e.which == 27 ) {

that.hide()

}

})

} else if ( !this.isShown ) {

$(document).unbind('keyup.modal')

}

}

/\* MODAL PLUGIN DEFINITION

\* ======================= \*/

$.fn.modal = function ( options ) {

var modal = this.data('modal')

if (!modal) {

if (typeof options == 'string') {

options = {

show: /show|toggle/.test(options)

}

}

return this.each(function () {

$(this).data('modal', new Modal(this, options))

})

}

if ( options === true ) {

return modal

}

if ( typeof options == 'string' ) {

modal[options]()

} else if ( modal ) {

modal.toggle()

}

return this

}

$.fn.modal.Modal = Modal

$.fn.modal.defaults = {

backdrop: false

, keyboard: false

, show: false

}

/\* MODAL DATA- IMPLEMENTATION

\* ========================== \*/

$(document).ready(function () {

$('body').delegate('[data-controls-modal]', 'click', function (e) {

e.preventDefault()

var $this = $(this).data('show', true)

$('#' + $this.attr('data-controls-modal')).modal( $this.data() )

})

})

}( window.jQuery || window.ender );