



# View License

 User: fossy  
[logout](#)
**Folder: [Software Repository/](#)**
**[bootstrap-daterangepicker-2.1.13.zip/](#) [bootstrap-daterangepicker-2.1.13/](#) [daterangepicker.js](#)**
[One-Shot Copyright/Email/URL](#) | [One-Shot License](#) | [License Browser](#) | [Audit](#) | [Bucket Browser](#) | [View](#) | [Info](#) | [View Copyright/Email/Url](#) • [Hex](#) | [Text](#) | [Formatted](#) • [Refresh](#)
[Edit concluded license](#)

## Scanner results

The newest version of nomos license scanner found:

**[MIT \(#1\)](#)**
**[MIT-style \(#1\)](#)**
[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [\[Next\]](#)[\[Last\]](#)

```
/**
 * @version: 2.1.13
 * @author: Dan Grossman http://www.dangrossman.info/
 * @copyright: Copyright (c) 2012-2015 Dan Grossman. All rights reserved.
 * @license: Licensed under the MIT license. See http://www.opensource.org/licenses
 /mit-license.php
 * @website: https://www.improvely.com/
 */

(function(root, factory) {

  if (typeof define === 'function' && define.amd) {
    define(['moment', 'jquery', 'exports'], function(momentjs, $, exports) {
      root.daterangepicker = factory(root, exports, momentjs, $);
    });

  } else if (typeof exports !== 'undefined') {
    var momentjs = require('moment');
    var jQuery = (typeof window !== 'undefined') ? window.jQuery : undefined; //isomorphic issue
    if (!jQuery) {
      try {
        jQuery = require('jquery');
        if (!jQuery.fn) jQuery.fn = {}; //isomorphic issue
      } catch (err) {}
      if (!jQuery) throw new Error('jQuery dependency not found');
    }

  }

  factory(root, exports, momentjs, jQuery);

  // Finally, as a browser global.
} else {
  root.daterangepicker = factory(root, {}, root.moment || moment, (root.jQuery || root.Zenite || root.$) || root.$);
}
```

**Legend:**

license relevant text

```
}(this || {}, function(root, daterangepicker, moment, $) { // 'this' doesn't exist on a server
```

```
var DateRangePicker = function(element, options, cb) {
```

```
//default settings for options
```

```
this.parentEl = 'body';
```

```
this.element = $(element);
```

```
this.startDate = moment().startOf('day');
```

```
this.endDate = moment().endOf('day');
```

```
this.minDate = false;
```

```
this.maxDate = false;
```

```
this.dateLimit = false;
```

```
this.autoApply = false;
```

```
this.singleDatePicker = false;
```

```
this.showDropdowns = false;
```

```
this.showWeekNumbers = false;
```

```
this.timePicker = false;
```

```
this.timePicker24Hour = false;
```

```
this.timePickerIncrement = 1;
```

```
this.timePickerSeconds = false;
```

```
this.linkedCalendars = true;
```

```
this.autoUpdateInput = true;
```

```
this.ranges = {};
```

```
this.opens = 'right';
```

```
if (this.element.hasClass('pull-right'))
```

```
this.opens = 'left';
```

```
this.drops = 'down';
```

```
if (this.element.hasClass('dropup'))
```

```
this.drops = 'up';
```

```
this.buttonClasses = 'btn btn-sm';
```

```
this.applyClass = 'btn-success';
```

```
this.cancelClass = 'btn-default';
```

```
this.locale = {
```

```
format: 'MM/DD/YYYY',
```

```
separator: ' - ',
```

```
applyLabel: 'Apply',
```

```
cancelLabel: 'Cancel',
```

```
weekLabel: 'W',
```

```
customRangeLabel: 'Custom Range',
```

```
daysOfWeek: moment.weekdaysMin(),
```

```
monthNames: moment.monthsShort(),
```

```
firstDay: moment.localeData().firstDayOfWeek()
```

```
};
```

```
this.callback = function() { };
```

```
//some state information
```

```
this.isShowing = false;
```

```
this.leftCalendar = {};
```

```
this.rightCalendar = {};
```

**Legend:**

license relevant text

```
//custom options from user
if (typeof options !== 'object' || options === null)
options = {};

//allow setting options with data attributes
//data-api options will be overwritten with custom javascript options
options = $.extend(this.element.data(), options);

//html template for the picker UI
if (typeof options.template !== 'string')
options.template = '<div class="daterangepicker dropdown-menu">' +
'<div class="calendar left">' +
'<div class="daterangepicker_input">' +
'<input class="input-mini" type="text" name="daterangepicker_start" value="" />' +
'<i class="fa fa-calendar glyphicon glyphicon-calendar"></i>' +
'<div class="calendar-time">' +
'<div></div>' +
'<i class="fa fa-clock-o glyphicon glyphicon-time"></i>' +
'</div>' +
'</div>' +
'<div class="calendar-table"></div>' +
'</div>' +
'<div class="calendar right">' +
'<div class="daterangepicker_input">' +
'<input class="input-mini" type="text" name="daterangepicker_end" value="" />' +
'<i class="fa fa-calendar glyphicon glyphicon-calendar"></i>' +
'<div class="calendar-time">' +
'<div></div>' +
'<i class="fa fa-clock-o glyphicon glyphicon-time"></i>' +
'</div>' +
'</div>' +
'<div class="calendar-table"></div>' +
'</div>' +
'<div class="ranges">' +
'<div class="range_inputs">' +
'<button class="applyBtn" disabled="disabled" type="button"></button>' +
'<button class="cancelBtn" type="button"></button>' +
'</div>' +
'</div>' +
'</div>';

this.parentEl = (options.parentEl && $(options.parentEl).length) ? $(options.parentEl) :
$(this.parentEl);
this.container = $(options.template).appendTo(this.parentEl);

//
// handle all the possible options overriding defaults
//

if (typeof options.locale === 'object') {

if (typeof options.locale.format === 'string')
this.locale.format = options.locale.format;
```

<b>Legend:</b> license relevant text
---

```
if (typeof options.locale.separator === 'string')
this.locale.separator = options.locale.separator;

if (typeof options.locale.daysOfWeek === 'object')
this.locale.daysOfWeek = options.locale.daysOfWeek.slice();

if (typeof options.locale.monthNames === 'object')
this.locale.monthNames = options.locale.monthNames.slice();

if (typeof options.locale.firstDay === 'number')
this.locale.firstDay = options.locale.firstDay;

if (typeof options.locale.applyLabel === 'string')
this.locale.applyLabel = options.locale.applyLabel;

if (typeof options.locale.cancelLabel === 'string')
this.locale.cancelLabel = options.locale.cancelLabel;

if (typeof options.locale.weekLabel === 'string')
this.locale.weekLabel = options.locale.weekLabel;

if (typeof options.locale.customRangeLabel === 'string')
this.locale.customRangeLabel = options.locale.customRangeLabel;

}

if (typeof options.startDate === 'string')
this.startDate = moment(options.startDate, this.locale.format);

if (typeof options.endDate === 'string')
this.endDate = moment(options.endDate, this.locale.format);

if (typeof options.minDate === 'string')
this.minDate = moment(options.minDate, this.locale.format);

if (typeof options.maxDate === 'string')
this.maxDate = moment(options.maxDate, this.locale.format);

if (typeof options.startDate === 'object')
this.startDate = moment(options.startDate);

if (typeof options.endDate === 'object')
this.endDate = moment(options.endDate);

if (typeof options.minDate === 'object')
this.minDate = moment(options.minDate);

if (typeof options.maxDate === 'object')
this.maxDate = moment(options.maxDate);

// sanity check for bad options
if (this.minDate && this.startDate.isBefore(this.minDate))
this.startDate = this.minDate.clone();

// sanity check for bad options
```

<b>Legend:</b> license relevant text
---

```
if (this.maxDate && this.endDate.isAfter(this.maxDate))
```

```
this.endDate = this.maxDate.clone();
```

```
if (typeof options.applyClass === 'string')
```

```
this.applyClass = options.applyClass;
```

```
if (typeof options.cancelClass === 'string')
```

```
this.cancelClass = options.cancelClass;
```

```
if (typeof options.dateLimit === 'object')
```

```
this.dateLimit = options.dateLimit;
```

```
if (typeof options.opens === 'string')
```

```
this.opens = options.opens;
```

```
if (typ
```

**1** [2](#) [3](#) [4](#) [5](#) [6](#) [\[Next\]](#) [\[Last\]](#)

**Legend:**

license relevant text