
bootstrap-datepicker Documentation

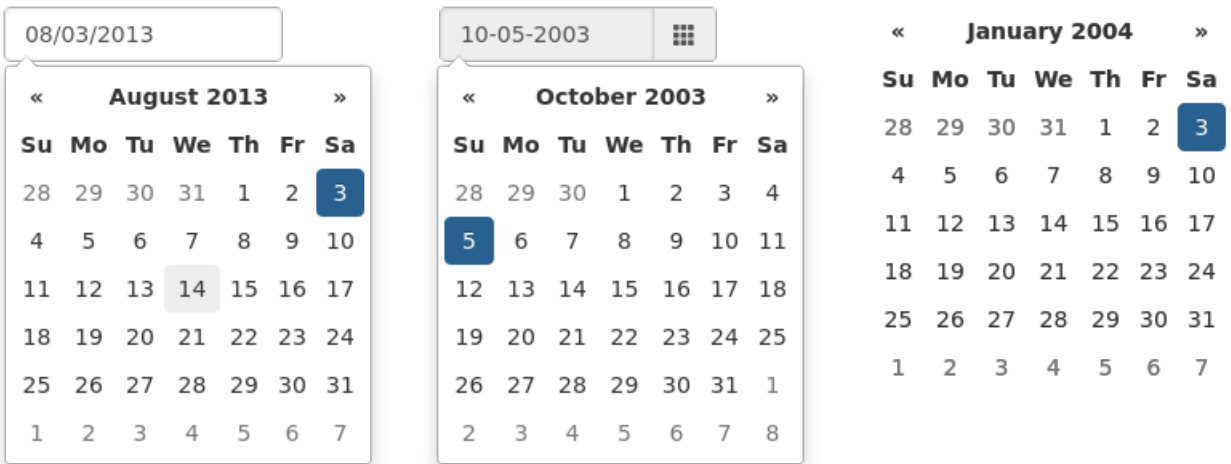
eternicode

Nov 12, 2019

Contents

1	Requirements	3
2	Dependencies	5
3	Usage	7
3.1	Data API	7
3.2	Configuration	7
3.3	Stylesheets	8
3.4	No Conflict mode	8
4	Table of Contents	9
4.1	Markup	9
4.2	Options	11
4.3	Quick reference	25
4.4	Methods	26
4.5	Events	29
4.6	Keyboard support	31
4.7	I18N	31

Bootstrap-datepicker provides a flexible datepicker widget in the Bootstrap style.



Versions are incremented according to [semver](#).

[Online Demo](#)

CHAPTER 1

Requirements

- Bootstrap 2.0.4+
- jQuery 1.7.1+

These are the specific versions bootstrap-datepicker is tested against (`js` files) and built against (`css` files). Use other versions at your own risk.

CHAPTER 2

Dependencies

Requires bootstrap's dropdown component (`dropdowns.less`) for some styles, and bootstrap's sprites (`sprites.less` and associated images) for arrows.

A standalone `.css` file (including necessary dropdown styles and alternative, text-based arrows) can be generated by running `build/build_standalone.less` through the `lessc` compiler:

```
$ lessc build/build_standalone.less datepicker.css
```


Call the datepicker via javascript:

```
$('.datepicker').datepicker();
```

3.1 Data API

As with bootstrap's own plugins, datepicker provides a data-api that can be used to instantiate datepickers without the need for custom javascript. For most datepickers, simply set `data-provide="datepicker"` on the element you want to initialize, and it will be initialized lazily, in true bootstrap fashion. For inline datepickers, use `data-provide="datepicker-inline"`; these will be immediately initialized on page load, and cannot be lazily loaded.

```
<input data-provide="datepicker">
```

Markup with component

```
<div class="input-group date" data-provide="datepicker">
  <input type="text" class="form-control">
  <div class="input-group-addon">
    <span class="glyphicon glyphicon-th"></span>
  </div>
</div>
```

You can disable datepicker's data-api in the same way as you would disable other bootstrap plugins:

```
$(document).off('.datepicker.data-api');
```

3.2 Configuration

Options are passed to the datepicker function via an options hash at instantiation:

```
$('.datepicker').datepicker({
  format: 'mm/dd/yyyy',
  startDate: '-3d'
});
```

Most options may be provided as data-attributes on the target element:

```
<input class="datepicker" data-date-format="mm/dd/yyyy">
```

```
$('.datepicker').datepicker({
  startDate: '-3d'
});
```

Defaults for all options can be modified directly by changing values in the `$.fn.datepicker.defaults` hash:

```
$.fn.datepicker.defaults.format = "mm/dd/yyyy";
$('.datepicker').datepicker({
  startDate: '-3d'
});
```

3.3 Stylesheets

There are a few different stylesheets included in the library. This is an overview of what each file is to be used for:

`bootstrap-datepicker.css` gives legacy support for twitter bootstrap v2, `bootstrap-datepicker3.css` is used for twitter bootstrap v3 support and `bootstrap-datepicker.standalone.css` can be used to include the datepicker without depending on the twitter bootstrap library.

3.4 No Conflict mode

`$.fn.datepicker.noConflict` provides a way to avoid conflict with other jQuery datepicker plugins:

```
var datepicker = $.fn.datepicker.noConflict(); // return $.fn.datepicker to
↳previously assigned value
$.fn.bootstrapDP = datepicker; // give $.fn.bootstrapDP the bootstrap-
↳datepicker functionality
```

4.1 Markup

The following are examples of supported markup. On their own, these will not provide a datepicker widget; you will need to instantiate the datepicker on the markup.

4.1.1 input

The simplest case: focusing the input (clicking or tabbing into it) will show the picker.

```
<input type="text" class="form-control" value="02-16-2012">
```

4.1.2 component

Adding the `date` class to an `input-group` bootstrap component will allow the `input-group-addon` elements to trigger the picker.

```
<div class="input-group date">
  <input type="text" class="form-control" value="12-02-2012">
  <div class="input-group-addon">
    <span class="glyphicon glyphicon-th"></span>
  </div>
</div>
```

4.1.3 date-range

Using the `input-daterange` construct with multiple child inputs will instantiate one picker per input and link them together to allow selecting ranges.



```
<div class="input-group input-daterange">
  <input type="text" class="form-control" value="2012-04-05">
  <div class="input-group-addon">to</div>
  <input type="text" class="form-control" value="2012-04-19">
</div>
```



Note that that `input-daterange` itself does not implement the `datepicker` methods. Methods should be directly called to the inputs. For example:

```
$('.input-daterange input').each(function() {
  $(this).datepicker('clearDates');
});
```

4.1.4 inline or embedded

Instantiating the datepicker on a simple div will give an embedded picker that is always visible.

```
<div data-date="12/03/2012"></div>
```

Example to save the embedded datepicker value to a hidden field

```
<div id="datepicker" data-date="12/03/2012"></div>
<input type="hidden" id="my_hidden_input">
```

```
$('#datepicker').datepicker();
$('#datepicker').on('changeDate', function() {
  $('#my_hidden_input').val(
    $('#datepicker').datepicker('getFormattedDate')
  );
});
```

4.2 Options

All options that take a “Date” can handle a `Date` object; a String formatted according to the given `format`; or a `timedelta` relative to today, eg “-1d”, “+6m +1y”, etc, where valid units are “d” (day), “w” (week), “m” (month), and

«	December 2012						»
Su	Mo	Tu	We	Th	Fr	Sa	
25	26	27	28	29	30	1	
2	3	4	5	6	7	8	
9	10	11	12	13	14	15	
16	17	18	19	20	21	22	
23	24	25	26	27	28	29	
30	31	1	2	3	4	5	

“y” (year). Use “0” as today. There are also aliases for the relative timedelta’s: “yesterday” equals “-1d”, “today” is equal to “+0d” and “tomorrow” is equal to “+1d”.

Most options can be provided via data-attributes. An option can be converted to a data-attribute by taking its name, replacing each uppercase letter with its lowercase equivalent preceded by a dash, and prepending “data-date-” to the result. For example, `startDate` would be `data-date-start-date`, `format` would be `data-date-format`, and `daysOfWeekDisabled` would be `data-date-days-of-week-disabled`.

See the [quick reference](#) for an overview of all options and their default values

4.2.1 autoclose

Boolean. Default: false

Whether or not to close the datepicker immediately when a date is selected.

4.2.2 assumeNearbyYear

Boolean or Integer. Default: false

If true, manually-entered dates with two-digit years, such as “5/1/15”, will be parsed as “2015”, not “15”. If the year is less than 10 years in advance, the picker will use the current century, otherwise, it will use the previous one. For example “5/1/15” would parse to May 1st, 2015, but “5/1/97” would be May 1st, 1997.

To configure the number of years in advance that the picker will still use the current century, use an Integer instead of the Boolean true. E.g. “assumeNearbyYear: 20”

4.2.3 beforeShowDay

Function(Date). Default: \$.noop

A function that takes a date as a parameter and returns one of the following values:

- undefined to have no effect
- A Boolean, indicating whether or not this date is selectable
- A String representing additional CSS classes to apply to the date's cell
- An object with the following properties:
 - `enabled`: same as the Boolean value above
 - `classes`: same as the String value above
 - `tooltip`: a tooltip to apply to this date, via the `title` HTML attribute
 - `content`: the content to display in the day cell, rather than the default (day of month as text)

4.2.4 beforeShowMonth

Function(Date). Default: \$.noop

A function that takes a date as a parameter and returns one of the following values:

- undefined to have no effect
- A Boolean, indicating whether or not this month is selectable
- A String representing additional CSS classes to apply to the month's cell
- An object with the following properties:
 - `enabled`: same as the Boolean value above
 - `classes`: same as the String value above
 - `tooltip`: a tooltip to apply to this date, via the `title` HTML attribute

4.2.5 beforeShowYear

Function(Date). Default: \$.noop

A function that takes a date as a parameter and returns one of the following values:

- undefined to have no effect
- A Boolean, indicating whether or not this year is selectable
- A String representing additional CSS classes to apply to the year's cell
- An object with the following properties:
 - `enabled`: same as the Boolean value above
 - `classes`: same as the String value above
 - `tooltip`: a tooltip to apply to this year, via the `title` HTML attribute

4.2.6 beforeShowDecade

Function(Date). Default: \$.noop

A function that takes a date as a parameter and returns one of the following values:

- undefined to have no effect
- A Boolean, indicating whether or not this year is selectable
- A String representing additional CSS classes to apply to the year's cell
- An object with the following properties:
 - `enabled`: same as the Boolean value above
 - `classes`: same as the String value above
 - `tooltip`: a tooltip to apply to this year, via the `title` HTML attribute

4.2.7 beforeShowCentury

Function(Date). Default: \$.noop

A function that takes a date as a parameter and returns one of the following values:

- undefined to have no effect
- A Boolean, indicating whether or not this year is selectable
- A String representing additional CSS classes to apply to the year's cell
- An object with the following properties:
 - `enabled`: same as the Boolean value above
 - `classes`: same as the String value above
 - `tooltip`: a tooltip to apply to this year, via the `title` HTML attribute

4.2.8 calendarWeeks

Boolean. Default: false

Whether or not to show week numbers to the left of week rows.

4.2.9 clearBtn

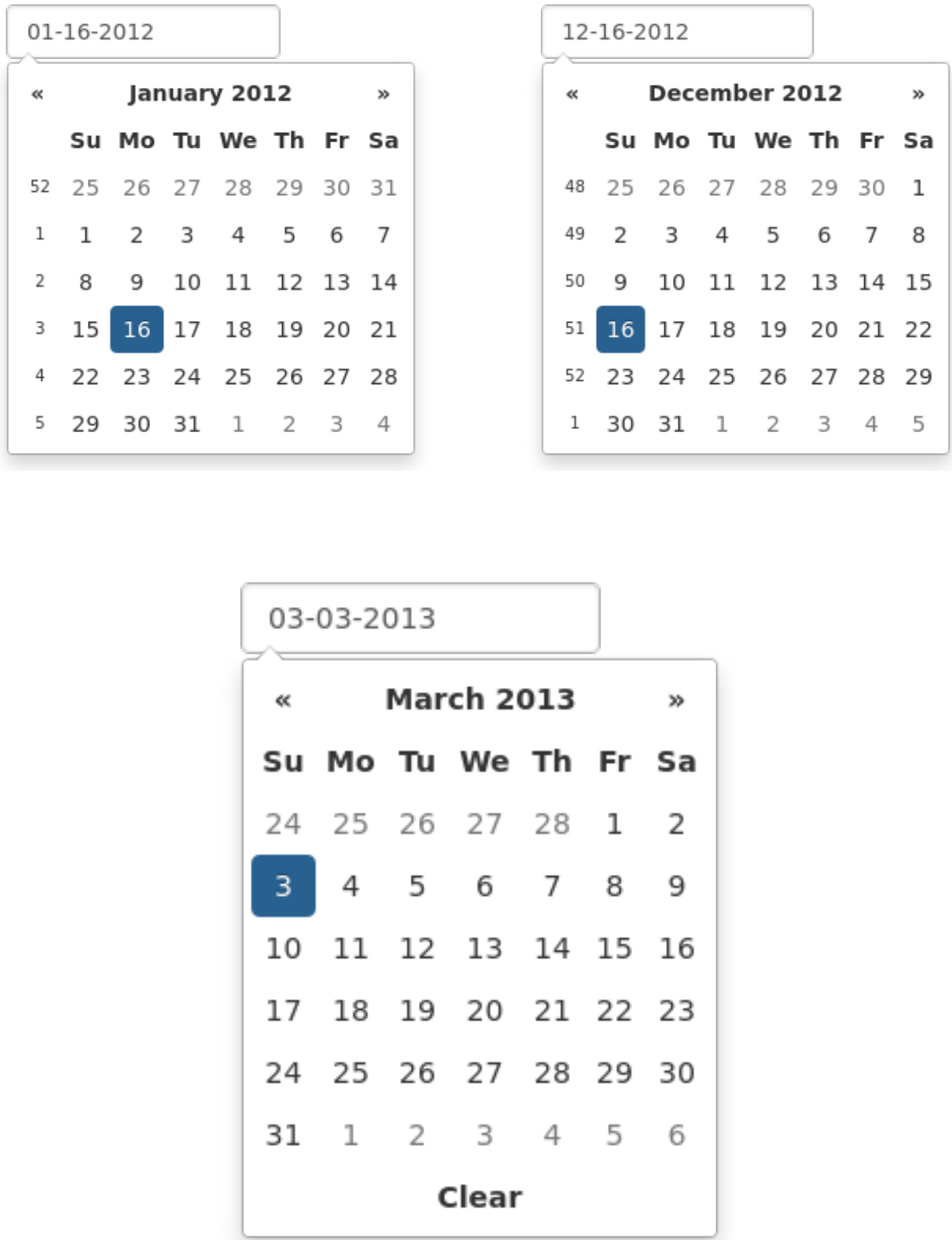
Boolean. Default: false

If true, displays a “Clear” button at the bottom of the datepicker to clear the input value. If “autoclose” is also set to true, this button will also close the datepicker.

4.2.10 container

String. Default: “body”

Appends the date picker popup to a specific element; eg: container: ‘#picker-container’ (will default to “body”)



4.2.11 datesDisabled

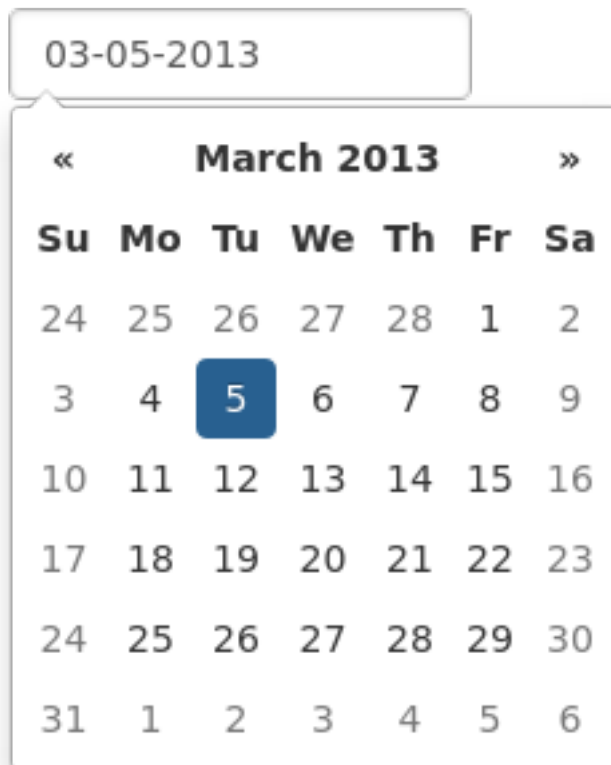
String, Array. Default: []

Array of date strings or a single date string formatted in the given date format

4.2.12 daysOfWeekDisabled

String, Array. Default: []

Days of the week that should be disabled. Values are 0 (Sunday) to 6 (Saturday). Multiple values should be comma-separated. Example: disable weekends: '06' or '0,6' or [0,6].



4.2.13 daysOfWeekHighlighted

String, Array. Default: []

Days of the week that should be highlighted. Values are 0 (Sunday) to 6 (Saturday). Multiple values should be comma-separated. Example: highlight weekends: '06' or '0,6' or [0,6].

4.2.14 defaultViewDate

Date, String or Object with keys year, month, and day. Default: today

Date to view when initially opening the calendar. The internal value of the date remains today as default, but when the datepicker is first opened the calendar will open to defaultViewDate rather than today. If this option is not used, “today” remains the default view date.

This option can be:

- A date, which should be in local timezone.
- A string which must be parsable with `format`.
- An object with keys `year`, `month` and `day` (can't be set from a data attribute). If the given object is missing any of the required keys, their defaults are:
 - `year`: the current year
 - `month`: 0 (Note that it starts with 0 for January)
 - `day`: 1

4.2.15 disableTouchKeyboard

Boolean. Default: false

If true, no keyboard will show on mobile devices

4.2.16 enableOnReadOnly

Boolean. Default: true

If false the datepicker will not show on a readonly datepicker field.

4.2.17 endDate

Date or String. Default: End of time

The latest date that may be selected; all later dates will be disabled.

Date should be in local timezone. String must be parsable with `format`.

```
<input type="text" class="form-control" data-date-end-date="0d">
```

Will disable all dates after today.

4.2.18 forceParse

Boolean. Default: true

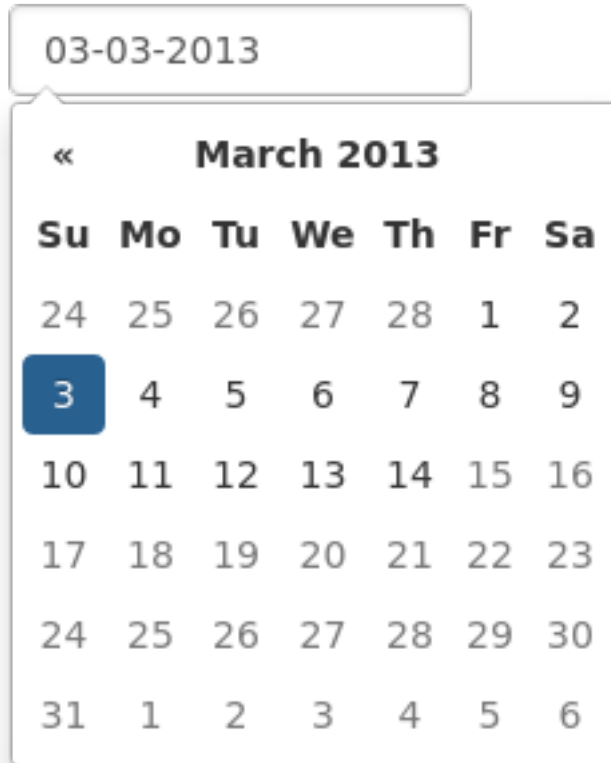
Whether or not to force parsing of the input value when the picker is closed. That is, when an invalid date is left in the input field by the user, the picker will forcibly parse that value, and set the input's value to the new, valid date, conforming to the given *format*.

4.2.19 format

String. Default: "mm/dd/yyyy"

The date format, combination of d, dd, D, DD, m, mm, M, MM, yy, yyyy.

- d, dd: Numeric date, no leading zero and leading zero, respectively. Eg, 5, 05.
- D, DD: Abbreviated and full weekday names, respectively. Eg, Mon, Monday.



- m, mm: Numeric month, no leading zero and leading zero, respectively. Eg, 7, 07.
- M, MM: Abbreviated and full month names, respectively. Eg, Jan, January
- yy, yyyy: 2- and 4-digit years, respectively. Eg, 12, 2012.

Object.

Custom formatting options

- toDisplay: function (date, format, language) to convert date object to string, that will be stored in input field
- toValue: function (date, format, language) to convert string object to date, that will be used in date selection

```

$('.datepicker').datepicker({
  format: {
    /*
     * Say our UI should display a week ahead,
     * but textbox should store the actual date.
     * This is useful if we need UI to select local dates,
     * but store in UTC
     */
    toDisplay: function (date, format, language) {
      var d = new Date(date);
      d.setDate(d.getDate() - 7);
      return d.toISOString();
    },
    toValue: function (date, format, language) {
      var d = new Date(date);
      d.setDate(d.getDate() + 7);
    }
  }
});

```

(continues on next page)

(continued from previous page)

```

        return new Date(d);
    }
}
});

```

4.2.20 immediateUpdates

Boolean. Default: false

If true, selecting a year or month in the datepicker will update the input value immediately. Otherwise, only selecting a day of the month will update the input value immediately.

4.2.21 inputs

Array, jQuery. Default: None

A list of inputs to be used in a range picker, which will be attached to the selected element. Allows for explicitly creating a range picker on a non-standard element.

```

<div id="event_period">
  <input type="text" class="actual_range">
  <input type="text" class="actual_range">
</div>

```

```

$('#event_period').datepicker({
  inputs: $('.actual_range')
});

```

4.2.22 keepEmptyValues

Boolean. Default: false

Only effective in a range picker. If true, the selected value does not get propagated to other, currently empty, pickers in the range.

4.2.23 keyboardNavigation

Boolean. Default: true

Whether or not to allow date navigation by arrow keys.

Keyboard navigation is not supported at all for embedded / inline mode. Also it's not working if input element hasn't focus. This could be an issue if used as component or if opened by *show* method.

4.2.24 language

String. Default: "en"

The IETF code (eg "en" for English, "pt-BR" for Brazilian Portuguese) of the language to use for month and day names. These will also be used as the input's value (and subsequently sent to the server in the case of form submissions). If a full code (eg "de-DE") is supplied the picker will first check for an "de-DE" language and if not found will fallback and check for a "de" language. If an unknown language code is given, English will be used. See [I18N](#).

« **March 2013** »

Su	Mo	Tu	We	Th	Fr	Sa
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

« **2013** »

Ene	Feb	Mar	Abr
May	Jun	Jul	Ago
Sep	Oct	Nov	Dic

« **mars 2013** »

l	ma	me	j	v	s	d
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

« **2013** »

1月	2月	3月	4月
5月	6月	7月	8月
9月	10月	11月	12月

« **2013年03月** »

日	月	火	水	木	金	土
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

« **2013** »

Янв	Фев	Мар	Апр
Май	Июн	Июл	Авг
Сен	Окт	Ноя	Дек

4.2.25 maxViewMode

Number, String. Default: 4, “centuries”

Set a maximum limit for the view mode. Accepts: 0 or “days” or “month”, 1 or “months” or “year”, 2 or “years” or “decade”, 3 or “decades” or “century”, and 4 or “centuries” or “millenium”. Gives the ability to pick only a day, a month, a year or a decade. The day is set to the 1st for “months”, the month is set to January for “years”, the year is set to the first year from the decade for “decades”, and the year is set to the first from the millennium for “centuries”.

4.2.26 minViewMode

Number, String. Default: 0, “days”

Set a minimum limit for the view mode. Accepts: 0 or “days” or “month”, 1 or “months” or “year”, 2 or “years” or “decade”, 3 or “decades” or “century”, and 4 or “centuries” or “millenium”. Gives the ability to pick only a month, a year or a decade. The day is set to the 1st for “months”, and the month is set to January for “years”, the year is set to the first year from the decade for “decades”, and the year is set to the first from the millennium for “centuries”.

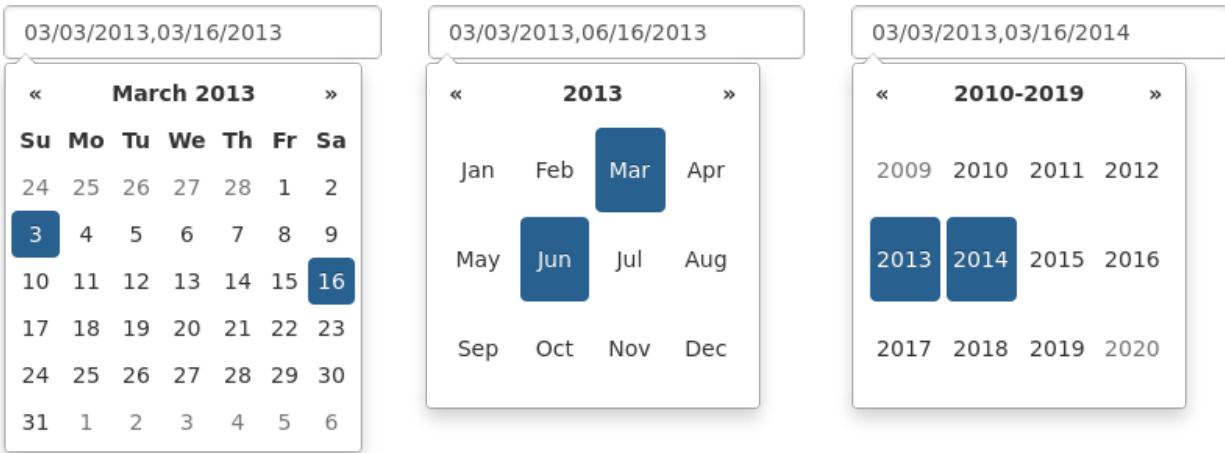
4.2.27 multigate

Boolean, Number. Default: false

Enable multigate picking. Each date in month view acts as a toggle button, keeping track of which dates the user has selected in order. If a number is given, the picker will limit how many dates can be selected to that number, dropping

the oldest dates from the list when the number is exceeded. `true` equates to no limit. The input's value (if present) is set to a string generated by joining the dates, formatted, with `multidateSeparator`.

For selecting 2 dates as a range please see [date-range](#)



4.2.28 multidateSeparator

String. Default: “,”

The string that will appear between dates when generating the input's value. When parsing the input's value for a multidate picker, this will also be used to split the incoming string to separate multiple formatted dates; as such, it is highly recommended that you not use a string that could be a substring of a formatted date (eg, using ‘-’ to separate dates when your format is ‘yyyy-mm-dd’).

4.2.29 orientation

String. Default: “auto”

A space-separated string consisting of one or two of “left” or “right”, “top” or “bottom”, and “auto” (may be omitted); for example, “top left”, “bottom” (horizontal orientation will default to “auto”), “right” (vertical orientation will default to “auto”), “auto top”. Allows for fixed placement of the picker popup.

“orientation” refers to the location of the picker popup's “anchor”; you can also think of it as the location of the trigger element (input, component, etc) relative to the picker.

“auto” triggers “smart orientation” of the picker. Horizontal orientation will default to “left” and left offset will be tweaked to keep the picker inside the browser viewport; vertical orientation will simply choose “top” or “bottom”, whichever will show more of the picker in the viewport.

4.2.30 showOnFocus

Boolean. Default: true

If false, the datepicker will be prevented from showing when the input field associated with it receives focus.

4.2.31 startDate

Date or String. Default: Beginning of time

The earliest date that may be selected; all earlier dates will be disabled.

Date should be in local timezone. String must be parsable with `format`.



4.2.32 startView

Number, String. Default: 0, “days”

The view that the datepicker should show when it is opened. Accepts: 0 or “days” or “month”, 1 or “months” or “year”, 2 or “years” or “decade”, 3 or “decades” or “century”, and 4 or “centuries” or “millenium”. Useful for date-of-birth datepickers.

4.2.33 templates

Object. Default:

```
{
  leftArrow: '&laquo;',
  rightArrow: '&raquo;'
}
```

The templates used to generate some parts of the picker. Each property must be a string with only text, or valid html. You can use this property to use custom icons libs. for example:

```
{
  leftArrow: '<i class="fa fa-long-arrow-left"></i>',
  rightArrow: '<i class="fa fa-long-arrow-right"></i>'
}
```

4.2.34 showWeekDays

Boolean. Default: true

If false, the datepicker will not append the names of the weekdays to its view. Default behavior is appending the weekdays.

4.2.35 title

String. Default: ""

The string that will appear on top of the datepicker. If empty the title will be hidden.

4.2.36 todayBtn

Boolean, "linked". Default: false

If true or "linked", displays a "Today" button at the bottom of the datepicker to select the current date. If true, the "Today" button will only move the current date into view; if "linked", the current date will also be selected.

4.2.37 todayHighlight

Boolean. Default: false

If true, highlights the current date.

4.2.38 toggleActive

Boolean. Default: false

If true, selecting the currently active date in the datepicker will unset the respective date. This option is always true when the multiselect option is being used.

4.2.39 updateViewDate

Boolean. Default: true

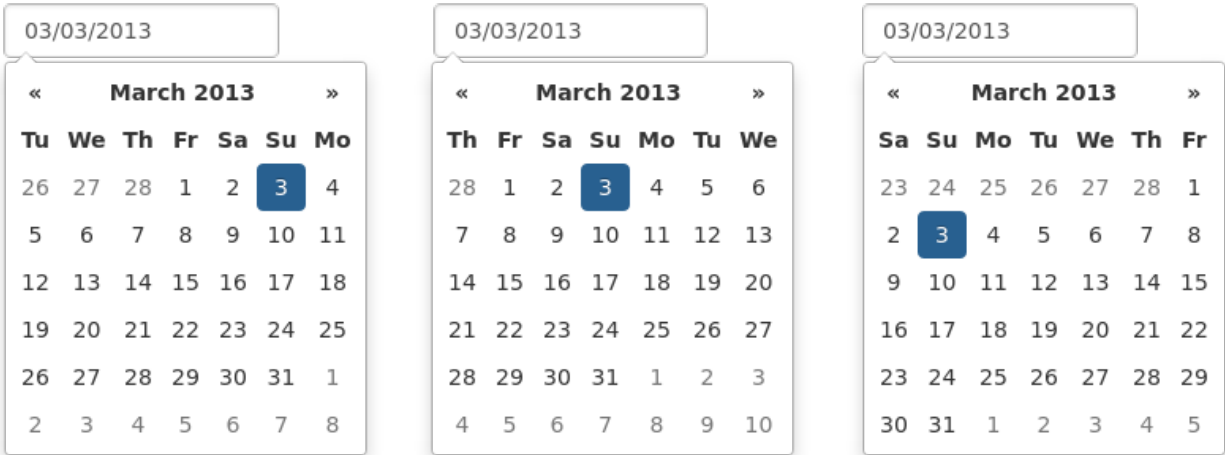
If false viewDate is set according to *value* on initialization and updated * if a day in last oder next month is selected or * if dates are changed by *setDate*, *setDates*, *setUTCDate* and *setUTCDates* methods. If *multiselect* option is *true* the last selected date or the last date in array passed to *setDates* or *setUTCDates* is used.

4.2.40 weekStart

Integer. Default: 0

Day of the week start. 0 (Sunday) to 6 (Saturday)





4.2.41 zIndexOffset

Integer. Default: 10

The CSS z-index of the open datepicker is the maximum z-index of the input and all of its DOM ancestors *plus* the `zIndexOffset`.

4.3 Quick reference

This is a quick overview of all the options and their default values

Option	Default value
<code>autoclose</code>	<code>false</code>
<code>assumeNearbyYear</code>	<code>false</code>
<code>beforeShowDay</code>	
<code>beforeShowMonth</code>	
<code>beforeShowYear</code>	
<code>beforeShowDecade</code>	
<code>beforeShowCentury</code>	
<code>calendarWeeks</code>	<code>false</code>
<code>clearBtn</code>	<code>false</code>
<code>container</code>	<code>'body'</code>
<code>datesDisabled</code>	<code>[]</code>
<code>daysOfWeekDisabled</code>	<code>[]</code>
<code>daysOfWeekHighlighted</code>	<code>[]</code>
<code>defaultViewDate</code>	<code>today</code>
<code>disableTouchKeyboard</code>	<code>false</code>
<code>enableOnReadonly</code>	<code>true</code>
<code>endDate</code>	<code>Infinity</code>
<code>forceParse</code>	<code>true</code>
<code>format</code>	<code>'mm/dd/yyyy'</code>
<code>immediateUpdates</code>	<code>false</code>
<code>inputs</code>	
<code>keepEmptyValues</code>	<code>false</code>

Continued on next page

Table 1 – continued from previous page

Option	Default value
keyboardNavigation	true
language	'en'
maxViewMode	4 'centuries'
minViewMode	0 'days'
multidate	false
multidateSeparator	','
orientation	'auto'
showOnFocus	true
startDate	-Infinity
startView	0 'days' (current month)
templates	
title	''
todayBtn	false
todayHighlight	true
toggleActive	false
weekStart	0 (Sunday)
zIndexOffset	10

4.4 Methods

Methods are called on a datepicker by calling the `datepicker` function with a string first argument, followed by any arguments the method takes

```
$('.datepicker').datepicker('method', arg1, arg2);
```

4.4.1 destroy

Arguments: None

Remove the datepicker. Removes attached events, internal attached objects, and added HTML elements.

Alias: remove

4.4.2 show

Arguments: None

Show the picker.

4.4.3 hide

Arguments: None

Hide the picker.

4.4.4 update

Arguments:

- date (String|Date|Array, optional)
- date (String|Date, optional)
- ...

Update the datepicker with given arguments or the current input value. The arguments can be either an array of strings, an array of Date objects, multiples strings or multiples Date objects. If date arguments are provided and they are Date objects, it is assumed to be “local” Date objects, and will be converted to UTC for internal use.

```
$('.datepicker').datepicker('update');
$('.datepicker').datepicker('update', '2011-03-05');
$('.datepicker').datepicker('update', '2011-03-05', '2011-03-07');
$('.datepicker').datepicker('update', new Date(2011, 2, 5));
$('.datepicker').datepicker('update', [new Date(2011, 2, 5), new Date(2011, 2, 7)]);
```

To reset the datepicker and clear the selected date, pass an empty string with update:

```
$('.datepicker').datepicker('update', '');
```

4.4.5 setDate

Arguments:

- date (Date)

Sets the internal date. date is assumed to be a “local” date object, and will be converted to UTC for internal use.

4.4.6 setUTCDate

Arguments:

- date (Date)

Sets the internal date. date is assumed to be a UTC date object, and will not be converted.

4.4.7 setDates

Arguments:

- date[, date[, ...]] (Date)

or

- [date[, date[, ...]]] (Array)

Sets the internal date list; accepts multiple dates or a single array of dates as arguments. Each date is assumed to be a “local” date object, and will be converted to UTC for internal use. For use with multirate pickers.

4.4.8 clearDates

Arguments: None

Clear dates.

4.4.9 setUTCDates

Arguments:

- `date[, date[, ...]]` (Date)

or

- `[date[, date[, ...]]]` (Array)

Sets the internal date list. Each `date` is assumed to be a UTC date object, and will not be converted. For use with multigate pickers.

4.4.10 getDate

Arguments: None

Returns a localized date object representing the internal date object of the first datepicker in the selection. For multigate pickers, returns the latest date selected.

4.4.11 getUTCDate

Arguments: None

Returns the internal UTC date object, as-is and unconverted to local time, of the first datepicker in the selection. For multigate pickers, returns the latest date selected.

4.4.12 getDates

Arguments: None

Returns a list of localized date objects representing the internal date objects of the first datepicker in the selection. For use with multigate pickers.

4.4.13 getUTCDates

Arguments: None

Returns the internal list of UTC date objects, as they are and unconverted to local time, of the first datepicker in the selection. For use with multigate pickers.

4.4.14 getStartDate

Arguments: None

Returns the lower date limit on the datepicker.

4.4.15 getEndDate

Arguments: None

Returns the upper date limit on the datepicker.

4.4.16 `setStartDate`

Arguments:

- `startDate` (Date)

Sets a new lower date limit on the datepicker. See *startDate* for valid values.

Omit `startDate` (or provide an otherwise falsey value) to unset the limit.

4.4.17 `setEndDate`

Arguments:

- `endDate` (Date)

Sets a new upper date limit on the datepicker. See *endDate* for valid values.

Omit `endDate` (or provide an otherwise falsey value) to unset the limit.

4.4.18 `setDatesDisabled`

Arguments:

- `datesDisabled` (String!Array)

Sets the days that should be disabled. See *datesDisabled* for valid values.

Omit `datesDisabled` (or provide an otherwise falsey value) to unset the disabled days.

4.4.19 `setDaysOfWeekDisabled`

Arguments:

- `daysOfWeekDisabled` (String!Array)

Sets the days of week that should be disabled. See *daysOfWeekDisabled* for valid values.

Omit `daysOfWeekDisabled` (or provide an otherwise falsey value) to unset the disabled days of week.

4.4.20 `setDaysOfWeekHighlighted`

Arguments:

- `daysOfWeekHighlighted` (String!Array)

Sets the days of week that should be highlighted. See *daysOfWeekHighlighted* for valid values.

Omit `daysOfWeekHighlighted` (or provide an otherwise falsey value) to unset the highlighted days of week.

4.5 Events

DatePicker triggers a number of events in certain circumstances. All events have extra data attached to the event object that is passed to any event handlers

```
$('.datepicker').datepicker()  
  .on(picker_event, function(e) {  
    // `e` here contains the extra attributes  
  });
```

- `date`: the relevant Date object, in local timezone. For a multidecade picker, this will be the latest date picked.
- `dates`: an Array of Date objects, in local timezone, when using a multidecade picker.
- `format([ix], [format])`: a function to make formatting date easier. `ix` can be the index of a Date in the `dates` array to format; if absent, the last date selected will be used. `format` can be any format string that datepicker supports; if absent, the format set on the datepicker will be used. Both arguments are optional.

4.5.1 show

Fired when the date picker is displayed.

4.5.2 hide

Fired when the date picker is hidden.

4.5.3 clearDate

Fired when the date is cleared, normally when the “clear” button (enabled with the `clearBtn` option) is pressed.

4.5.4 changeDate

Fired when the date is changed.

4.5.5 changeMonth

Fired when the `view` month is changed from year view.

4.5.6 changeYear

Fired when the `view` year is changed from decade view.

4.5.7 changeDecade

Fired when the `view` decade is changed from century view.

4.5.8 changeCentury

Fired when the `view` century is changed from millennium view.

4.6 Keyboard support

The datepicker includes keyboard navigation. The “focused date” is kept track of and highlighted (as with mouse hover) during keyboard nav, and is cleared when a date is toggled or the picker is hidden.

4.6.1 up, down, left, right arrow keys

By themselves, left/right will move focus backward/forward one day, up/down will move focus back/forward one week.

With the shift key, up/left will move focus backward one month, down/right will move focus forward one month.

With the ctrl key, up/left will move focus backward one year, down/right will move focus forward one year.

Shift+ctrl behaves the same as ctrl – that is, it does not change both month and year simultaneously, only the year.

4.6.2 enter

When the picker is visible, enter will toggle the focused date (if there is one). When the picker is not visible, enter will have normal effects – submitting the current form, etc.

When the date is deselected, the `clearDate` event is triggered; otherwise, the `changeDate` event is triggered. If `autoclose` is enabled, the picker will be hidden after selection or deselection.

4.6.3 escape

The escape key can be used to clear the focused date and hide and re-show the datepicker; hiding the picker is necessary if the user wants to manually edit the value.

4.7 I18N

The plugin supports `i18n` for the month and weekday names and the `weekStart` option. The default is English (“en”); other available translations are available in the `js/locales/` directory, simply include your desired locale after the plugin. To add more languages, simply add a key to `$.fn.datepicker.dates`, before calling `.datepicker()`. Example

```
$.fn.datepicker.dates['en'] = {
  days: ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"],
  daysShort: ["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"],
  daysMin: ["Su", "Mo", "Tu", "We", "Th", "Fr", "Sa"],
  months: ["January", "February", "March", "April", "May", "June", "July", "August",
  "September", "October", "November", "December"],
  monthsShort: ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"],
  today: "Today",
  clear: "Clear",
  format: "mm/dd/yyyy",
  titleFormat: "MM yyyy", /* Leverages same syntax as 'format' */
  weekStart: 0
};
```

Right-to-left languages may also include `rtl: true` to make the calendar display appropriately.

If your browser (or those of your users) is displaying characters wrong, chances are the browser is loading the javascript file with a non-unicode encoding. Simply add `charset="UTF-8"` to your `script` tag:

```
<script src="bootstrap-datepicker.XX.js" charset="UTF-8"></script>
```

```
$('.datepicker').datepicker({  
  language: 'XX'  
});
```

« **March 2013** »

Su	Mo	Tu	We	Th	Fr	Sa
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

« **2013** »

Ene	Feb	Mar	Abr
May	Jun	Jul	Ago
Sep	Oct	Nov	Dic

« **mars 2013** »

l	ma	me	j	v	s	d
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

« **2013** »

1月	2月	3月	4月
5月	6月	7月	8月
9月	10月	11月	12月

« **2013年03月** »

日	月	火	水	木	金	土
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

« **2013** »

Янв	Фев	Мар	Апр
Май	Июн	Июл	Авг
Сен	Окт	Ноя	Дек