
bootstrap-datepicker Documentation

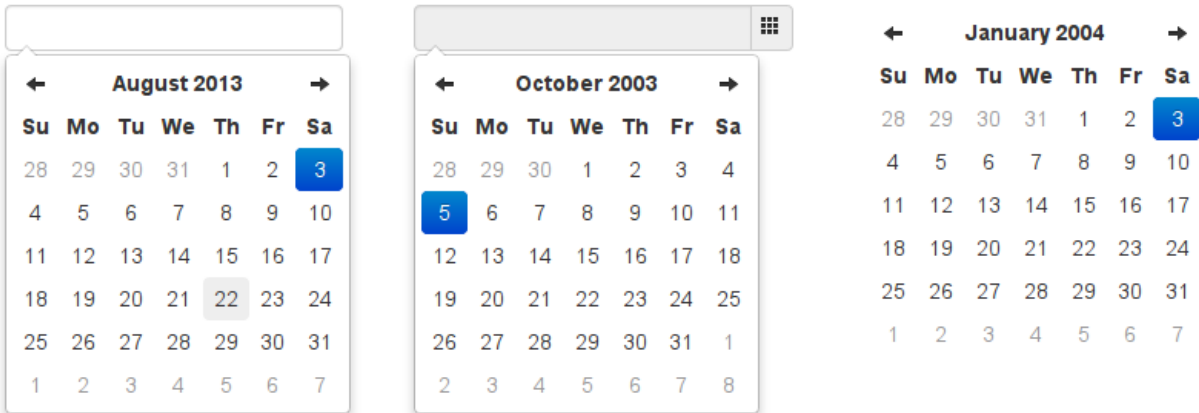
Release

eternicode

March 02, 2017

1	Requirements	3
2	Dependencies	5
3	Usage	7
3.1	Data API	7
3.2	Configuration	7
3.3	No Conflict mode	8
4	Table of Contents	9
4.1	Markup	9
4.2	Options	10
4.3	Methods	13
4.4	Events	16
4.5	Keyboard support	17
4.6	I18N	17
5	Indices and tables	19

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



This is a fork of Stefan Petre's [original code](#); thanks go to him for getting this thing started!

Please note that this fork is not used on Stefan's page at this time, nor is it maintained or contributed to by him.

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

[Online Demo](#)

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.

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
```

Usage

Call the datepicker via javascript:

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

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">
```

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

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

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'
})
```

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 function
```

Table of Contents

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.

input

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

```
<input value="02-16-2012">
```

component

Adding the date class to an `input-append` or `input-prepend` bootstrap component will allow the add-on elements to trigger the picker.

```
<div class="input-append date">  
  <input value="12-02-2012">  
  <span class="add-on"><i class="icon-th"></i></span>  
</div>
```

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-daterange">  
  <input value="2012-04-05" />  
  <span class="add-on">to</span>  
  <input value="2012-04-07" />  
</div>
```

inline or embedded

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

```
<div></div>
```

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 “y” (year).

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`.

autoclose

Boolean. Default: false

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

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

calendarWeeks

Boolean. Default: false

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

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.

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: ' 0, 6' or [0, 6].

endDate

Date. Default: End of time

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

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*.

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.

inputs

Array. 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.

keyboardNavigation

Boolean. Default: true

Whether or not to allow date navigation by arrow keys.

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](#).

minViewMode

Number, String. Default: 0, “days”

Set a limit for the view mode. Accepts: “days” or 0, “months” or 1, and “years” or 2. Gives the ability to pick only a month or an year. The day is set to the 1st for “months”, and the month is set to January for “years”.

multidate

Boolean, Number. Default: false

Enable multidate 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`.

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’).

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.

startDate

Date. Default: Beginning of time

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

startView

Number, String. Default: 0, “month”

The view that the datepicker should show when it is opened. Accepts values of 0 or “month” for month view (the default), 1 or “year” for the 12-month overview, and 2 or “decade” for the 10-year overview. Useful for date-of-birth datepickers.

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.

todayHighlight

Boolean. Default: false

If true, highlights the current date.

weekStart

Integer. Default: 0

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

Methods

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

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

remove

Arguments: None

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

show

Arguments: None

Show the picker.

hide

Arguments: None

Hide the picker.

update

Arguments:

- date (String|Date, optional)

Update the datepicker with given argument or the current input value.

If `date` is provided and is a `Date` object, it is assumed to be a “local” date object, and will be converted to UTC for internal use.

```
$('.datepicker').datepicker('update');
$('.datepicker').datepicker('update', '2011-03-05');
$('.datepicker').datepicker('update', new Date(2011, 2, 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.

setUTCDate

Arguments:

- date (Date)

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

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 multigate pickers.

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.

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.

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.

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.

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.

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.

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.

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.

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 multirate picker, this will be the latest date picked.
- *dates*: an Array of Date objects, in local timezone, when using a multirate 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.

show

Fired when the date picker is displayed.

hide

Fired when the date picker is hidden.

clearDate

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

changeDate

Fired when the date is changed.

changeYear

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

changeMonth

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

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.

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.

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.

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.

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", "Sunday"],
  daysShort: ["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"],
  daysMin: ["Su", "Mo", "Tu", "We", "Th", "Fr", "Sa", "Su"],
  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"
};
```

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

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

Indices and tables

- `genindex`
- `modindex`
- `search`