

Adding a Timepicker to jQuery UI Datepicker

The timepicker addon adds a timepicker to jQuery UI Datepicker, thus the datepicker and slider components (jQueryUI) are required for using any of these. In addition all datepicker options are still available through the timepicker addon.

If you are interested in contributing to Timepicker Addon please [check it out on GitHub](#). If you do make additions please keep in mind I enjoy tabs over spaces,... But contributions are welcome in any form.

[Back to Blog](#) or [Follow on Twitter](#)

Donation

Has this Timepicker Addon been helpful to you?

[Donate](#)

Getting Started Options Formatting Localization Examples

Examples

- [Basic Initializations](#)
- [Using Timezones](#)
- [Slider Modifications](#)
- [Alternate Fields](#)
- [Time Input](#)
- [Time Restraints](#)
- [Time Ranges](#)
- [Utilities](#)

Basic Initializations

Add a simple datetimepicker to jQuery UI's datepicker

```
$('#basic_example_1').datetimepicker();
```

Add only a timepicker:

```
$('#basic_example_2').timepicker();
```

Format the time:

```
$('#basic_example_3').datetimepicker({  
    timeFormat: "hh:mm tt"  
});
```

Timepicker comes with a collection of localization files and one combined file with all available localizations. `$.timepicker.regional["your localization code here"]` is a simple object with preset options:

```
$('#basic_example_4').timepicker(  
    $.timepicker.regional['es']  
);
```

Using Timezones

Simplest timezone usage:

```
$('#timezone_example_1').datetimepicker({  
    timeFormat: 'hh:mm tt z'  
});
```

Define your own timezone options:

```
$('#timezone_example_2').datetimepicker({  
    timeFormat: 'HH:mm z',  
    timeZoneList: [  
        { value: -300, label: 'Eastern'},  
        { value: -360, label: 'Central' },  
        { value: -420, label: 'Mountain' },  
        { value: -480, label: 'Pacific' }  
    ]  
});
```

Slider Modifications

Add a grid to each slider:

```
$('#slider_example_1').timepicker({  
    hourGrid: 4,  
    minuteGrid: 10,  
    timeFormat: 'hh:mm tt'  
});
```

Set the interval step of sliders:

```
$('#slider_example_2').datetimepicker({  
    timeFormat: 'HH:mm:ss',  
    stepHour: 2,
```

```
        stepMinute: 10,  
        stepSecond: 10  
    });
```

Add sliderAccess plugin for touch devices:

```
$('#slider_example_3').datetimepicker({  
    addSliderAccess: true,  
    sliderAccessArgs: { touchonly: false }  
});
```

Use dropdowns instead of sliders. By default if slider is not available dropdowns will be used.

```
$('#slider_example_4').datetimepicker({  
    controlType: 'select',  
    timeFormat: 'hh:mm tt'  
});
```

Uses one line dropdowns instead of sliders.

```
$('#slider_example_4andHalf').datetimepicker({  
    controlType: 'select',  
    oneLine: true,  
    timeFormat: 'hh:mm tt'  
});
```

Create your own control by implementing the create, options, and value methods. If you want to use your new control for all instances use the \$.timepicker.setDefaults({controlType:myControl}). Here we implement jQueryUI's spinner control (jQueryUI 1.9+).

```
var myControl= {  
    create: function(tp_inst, obj, unit, val, min, max, step){  
        $('<input class="ui-timepicker-input" value="'+val+'" style="width:50%">')  
            .appendTo(obj)  
            .spinner({  
                min: min,  
                max: max,  
                step: step,  
                change: function(e,ui){ // key events  
                    // don't call if api was used and not key press  
                    if(e.originalEvent !== undefined)  
                        tp_inst._onTimeChange();  
                    tp_inst._onSelectHandler();  
                },  
                spin: function(e,ui){ // spin events
```

```

        tp_inst.control.value(tp_inst, obj, unit, ui.value);
        tp_inst._onTimeChange();
        tp_inst._onSelectHandler();
    }
    });
    return obj;
},
options: function(tp_inst, obj, unit, opts, val){
    if(typeof(opts) == 'string' && val !== undefined)
        return obj.find('.ui-timepicker-input').spinner(opts, val);
    return obj.find('.ui-timepicker-input').spinner(opts);
},
value: function(tp_inst, obj, unit, val){
    if(val !== undefined)
        return obj.find('.ui-timepicker-input').spinner('value', val);
    return obj.find('.ui-timepicker-input').spinner('value');
}
});

$('#slider_example_5').datetimepicker({
    controlType: myControl
});

```

Alternate Fields

Alt field in the simplest form:

09/15/2012	10:15
------------	-------

```

$('#alt_example_1').datetimepicker({
    altField: "#alt_example_1_alt"
});

```

With datetime in both:

--	--

```

$('#alt_example_2').datetimepicker({
    altField: "#alt_example_2_alt",
    altFieldTimeOnly: false
});

```

Format the altField differently:

--	--

```

$('#alt_example_3').datetimepicker({
    altField: "#alt_example_3_alt",
    altFieldTimeOnly: false,
    altFormat: "yy-mm-dd",
    altTimeFormat: "h:m t",
    altSeparator: " @ "
});

```

With inline mode using altField:

03/15/2017 00:00

March 2017

Su	Mo	Tu	We	Th	Fr	Sa
			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	

Time 00:00

Hour

Minute

Now

```
$('#alt_example_4').datetimepicker({  
  altField: "#alt_example_4_alt",  
  altFieldTimeOnly: false  
});
```

Time Input

Allows time displayed inside the picker to allow being typed in.

08/20/2014 01:22 pm

```
$('#input_example_1').datetimepicker({  
  timeInput: true,  
  timeFormat: "hh:mm tt"  
});
```

Don't show any sliders, only the time input.

08/20/2014 01:22 pm

```
$('#input_example_2').datetimepicker({  
  timeInput: true,  
  timeFormat: "hh:mm tt",  
  showHour: false,  
  showMinute: false  
});
```

Time Restraints

Set the min/max hour of every date:

```
$('#rest_example_1').timepicker({  
  hourMin: 8,
```

```
hourMax: 16
});
```

Set the min/max date numerically:

```
$('#rest_example_2').datetimepicker({
  numberOfMonths: 2,
  minDate: 0,
  maxDate: 30
});
```

Set the min/max date and time with a Date object:

```
$('#rest_example_3').datetimepicker({
  minDate: new Date(2010, 11, 20, 8, 30),
  maxDate: new Date(2010, 11, 31, 17, 30)
});
```

Time Ranges

Restrict a start and end date by using onSelect and onClose events for more control over functionality:

For more examples and advanced usage grab the [Handling Time eBook](#).

08/20/2014 09:22 -0400	08/21/2014 08:00 -0400
------------------------	------------------------

```
var startDateTextBox = $('#range_example_1_start');
var endDateTextBox = $('#range_example_1_end');

startDateTextBox.datetimepicker({
  timeFormat: 'HH:mm z',
  onClose: function(dateText, inst) {
    if (endDateTextBox.val() != '') {
      var testStartDate = startDateTextBox.datetimepicker('getDate');
      var testEndDate = endDateTextBox.datetimepicker('getDate');
      if (testStartDate > testEndDate)
        endDateTextBox.datetimepicker('setDate', testStartDate);
    }
    else {
      endDateTextBox.val(dateText);
    }
  },
  onSelect: function (selectedDateTime){
    endDateTextBox.datetimepicker('option', 'minDate', startDateTextBox.datetimepicker('getDate') );
  }
});
endDateTextBox.datetimepicker({
  timeFormat: 'HH:mm z',
  onClose: function(dateText, inst) {
    if (startDateTextBox.val() != '') {
      var testStartDate = startDateTextBox.datetimepicker('getDate');
      var testEndDate = endDateTextBox.datetimepicker('getDate');
      if (testStartDate > testEndDate)
        startDateTextBox.datetimepicker('setDate', testEndDate);
    }
  }
});
```

```

    }
    else {
        startDateTextBox.val(dateText);
    }
},
onSelect: function (selectedDateTime){
    startDateTextBox.datetimepicker('option', 'maxDate', endDateTextBox.datetimepicker('getDate') );
}
});

```

Timepicker also includes some shortcut methods for ranges:

```

var startDateTextBox = $('#range_example_2_start');
var endDateTextBox = $('#range_example_2_end');

$.timepicker.datetimeRange(
    startDateTextBox,
    endDateTextBox,
    {
        minInterval: (1000*60*60), // 1hr
        dateFormat: 'dd M yy',
        timeFormat: 'HH:mm',
        start: {}, // start picker options
        end: {} // end picker options
    }
);

```

To use only times for a time range use \$.timepicker.timeRange():

09:16

10:16

```

var startTimeTextBox = $('#range_example_3_start');
var endTimeTextBox = $('#range_example_3_end');

$.timepicker.timeRange(
    startTimeTextBox,
    endTimeTextBox,
    {
        minInterval: (1000*60*60), // 1hr
        timeFormat: 'HH:mm',
        start: {}, // start picker options
        end: {} // end picker options
    }
);

```

Even though this plugin focuses on datetime, it also provides a dateRange function:

```

var startDateTextBox = $('#range_example_4_start');
var endDateTextBox = $('#range_example_4_end');

$.timepicker.dateRange(
    startDateTextBox,
    endDateTextBox,

```

```

    {
      minInterval: (1000*60*60*24*4), // 4 days
      maxInterval: (1000*60*60*24*8), // 8 days
      start: {}, // start picker options
      end: {} // end picker options
    }
  );

```

Utilities

Get and Set Datetime with the getDate and setDate methods. This example uses timezone to demonstrate the timepicker regionalizes the timezones and computes the offsets when getting and setting.

Set Datetime
Get Datetime

```

var ex13 = $('#utility_example_1');

ex13.datetimepicker({
  timeFormat: 'hh:mm tt z',
  separator: ' @ ',
  showTimezone: true
});

$('#utility_example_1_setdt').click(function(){
  ex13.datetimepicker('setDate', (new Date()) );
});

$('#utility_example_1_getdt').click(function(){
  alert(ex13.datetimepicker('getDate'));
});

```

Use the utility function to format your own time. \$.datepicker.formatTime(format, time, options)

format	required - string representation of the time format to use
time	required - hash: { hour, minute, second, millisecond, timezone }
options	optional - hash of any options in regional translation (ampm, amNames, pmNames..)

Returns a time string in the specified format.

14:36 +2000

```

$('#utility_example_2').text(
  $.datepicker.formatTime('HH:mm z', { hour: 14, minute: 36, timezone: '+2000' }, {})
);

```

Use the utility function to parse a formatted time. \$.datepicker.parseTime(format, timeString, options)

format	required - string representation of the time format to use
time	required - time string matching the format given in parameter 1
options	optional - hash of any options in regional translation (ampm, amNames, pmNames..)

Returns an object with hours, minutes, seconds, milliseconds, timezone.

```

{"hour":14,"minute":36,"second":21,"millisec":765,"microsec":0,"timezone":1200}

```



```
$('#utility_example_3').text(JSON.stringify(  
    $.datepicker.parseTime('HH:mm:ss:1 z', "14:36:21:765 +2000", {})  
));
```
