

Join the Stack Overflow Community

Stack Overflow is a community of 6.8 million programmers, just like you, helping each other.
Join them; it only takes a minute:

[Sign up](#)

How to catch the bootstrap datepicker change event?

Now I am trying to do something after user change the date. but seems like my ng-change is ignored. Here is my code sample:

```
<input ng-change='changedate()'
      type="text"
      starting-day="2"
      show-button-bar="false"
      show-weeks="false"
      class="form-control addTicketDateInput"
      datepicker-popup="dd MMM"
      ng-model="startdate"
      is-open="openstart"
      datepicker-options="dateOptions"
      ng-required="true"
      close-text="Close" />

in my controller:
$scope.changedate=function(){
  console.log($scope.startdate);
}
```

Any idea?

[angularjs](#) [bootstrap-datepicker](#)

asked Apr 7 '15 at 17:23

 [linyuanxie](#)
262 4 7 22

1 I think it should work, plnkr.co/edit/QbtoD2ND46ceoDOXes9x?p=preview, click first datepicker and check console output – [ABOS](#) Apr 7 '15 at 19:56

2 Answers

You could keep an eye on when the `$scope` changed.

Something like this:

```
$scope.startdate;

$scope.$watch("startdate", function(newValue, oldValue) {
  console.log("I've changed : ", startdate);
});
```

answered Feb 3 '16 at 13:45

 [Michael Tot Korskgaard](#)
1,300 3 25 54

The way I've done this before is to wrap the datepicker jQuery stuff in an angular directive. This is rough but here's the idea:

```
app.directive('dateDirective', function() {
  return {
    restrict: 'A',
    scope: {
      onChange: '&'
    },
    link: function(scope, element, attrs) {
      element.datetimepicker({
        format: "MM-yyyy"
        //all your options here
      }).on('changeDate', function(e) {
```

```

    scope.$apply(function(scope) {
        scope.onChange(e.date);
    });
  });
}
});
});

```

You can capture the change event from the jQuery element and use it to call a function in your controller, or just use it to set scope values or whatever:

```

var app = angular.module('myApp', []);

app.controller('myAppCtrl', function($scope) {
  $scope.current = '';
  $scope.changedate = function(date){
    $scope.current = date;
  };
});

```

Then you just need to add the directive to your dom element:

```

<input date-directive
  type="text"
  starting-day="2"
  show-button-bar="false"
  show-weeks="false"
  class="form-control addTicketDateInput"
  datepicker-popup="dd MMM"
  ng-model="startdate"
  is-open="openstart"
  datepicker-options="dateOptions"
  ng-required="true"
  close-text="Close" />

```

Then tell it which scope function to call:

```

<input date-directive onChange='changedate(date)'
  type="text"
  starting-day="2"
  show-button-bar="false"
  show-weeks="false"
  class="form-control addTicketDateInput"
  datepicker-popup="dd MMM"
  ng-model="startdate"
  is-open="openstart"
  datepicker-options="dateOptions"
  ng-required="true"
  close-text="Close" />

```

Like I said this is rough just recalled it quickly. If this doesn't help I'll dig out a sample that is tested.

Hope that helps!

also, there's a datepicker directive here that might be helpful:

<https://angular-ui.github.io/bootstrap/>

edited Apr 7 '15 at 19:46

answered Apr 7 '15 at 19:12



Chris

48 1 7

Do you know where can I find a working example? – Tuco Mar 8 '16 at 14:45
