

# APP.JS



## ANGULAR JS APPLICATION

Set up AngularJS Application named "nuPlannerApp" with dependencies "ui.calendar" and "ui.bootstrap", part of the angular-ui suite. The ui.calendar is an encapsulation of FullCalendar while ui.bootstrap is obviously an encapsulation of the Twitter Bootstrap framework. The \$routeProvider sets the root URL of the application ("/") to use the "views/main.html" view with the "MainCtrl" controller.

```
angular.module('nuPlannerApp', ['ui.calendar', 'ui.bootstrap']).config(function
($routeProvider) {
  $routeProvider
  .when('/', {
    templateUrl: 'views/main.html',
    controller: 'MainCtrl'
  })
  .otherwise({ redirectTo: '/' });
});
```

## ANGULAR JS CONTROLLER

Create an AngularJS Controller named "MainCtrl". This controller handles all functionality within the Main view of the application (views/main.html).

```
angular.module('nuPlannerApp').controller('MainCtrl', function ($scope) {
```

The eventSource variable connects to the Play Framework's controller "/events" which provides a JSON feed of events with parameters to filter. Within ui.calendar (FullCalendar) two internal parameters are added to the "/events" URL. These are "start" and "end". The two parameters are unixtimestamps and change based on the current view's start and end dates (e.g. week, month, day). A third parameter, query is added to allow text based filtering of events based using the title.

```
$scope.eventSource = { url: "/events", currenttimezone: 'America/Chicago', data: {  
  query: '' } };
```

Since FullCalendar can use multiple eventSources, the eventSources variable adds the eventSource above to the array of eventSources that will be linked to FullCalendar.

```
$scope.eventSources = [$scope.eventSource];
```

This function changes a set of variables within the \$scope that represent the selected event and its attributes. It then shows displays the modal for the clicked Event.

```
$scope.alertOnEventClick = function( event, jsEvent, view ) {  
  $scope.$apply(function() {  
    $scope.selected_event_id = event.id;  
    $scope.selected_event_title = event.title;  
    $scope.selected_event_allDay = event.allDay;  
    $scope.selected_event_creator = event.creator;  
    $scope.selected_event_start = event.start;  
    $scope.selected_event_end = event.end;  
    $scope.selected_event_description = event.description;  
    $scope.selected_event_url = event.url;  
    if (event.pic_big!=NULL)  
      $scope.selected_event_pic = event.pic_big;  
    else  
      $scope.selected_event_pic = event.pic;  
  
    $('#eventModal').modal();  
  });  
  return false;  
};
```

This variable sets the configuration options for ui.calendar (FullCalendar). The left side of the calendar has buttons for month, week, and day. The center has the title. And lastly, the right side has buttons to navigate the weeks and return back to the current day.

Many of the events currently do not have corresponding functions because they are not needed. The eventClick handler however is functional.

```

$scope.uiConfig = {
  calendar: {
    height: 500,
    editable: false,
    theme: false,
    header: {
      left: 'month, basicWeek, basicDay',
      center: 'title',
      right: 'today prev,next'
    },
    },
  defaultView: 'week',
  aspectRatio: 1,
  dayClick: $scope.alertOnDayClick,
  eventDrop: $scope.alertOnDrop,
  eventResize: $scope.alertOnResize,
  eventClick: $scope.alertOnEventClick
}
};

```

This function enables text based filtering by updateing the eventSource's query parameter to the current query variable in the scope. It then refetches the events with the new or changed parameter.

```

$scope.updateQuery = function() {
  $scope.eventSource.data.query = $scope.query;
  $scope.myCalendar.fullCalendar('refetchEvents');
}
});

```

## ANGULAR JS DIRECTIVE

Creates an AngularJS directive named "keyup". As the name suggests, this directive, when applied as an attribute to an element, bind's the element's keyup event. When the keyup event is fired the function listed as an attribute to the keyup function in the HTML is used. In this application, that function is "updateQuery".

```

angular.module('nuPlannerApp').directive('keyup', function() {
  return function(scope, element, attrs, ctrl) {
    element.bind("keyup", function(event) { scope.$apply(attrs.keyup) });
  };
});

```

