

```

app.controller('TeamDetailController', ['$scope', '$rootScope', '$location', '$http',
'$routeParams', '$window', function ($scope, $rootScope, $location, $http, $routeParams,
$window) {
    $scope.changeName = false;
    $scope.orderByField = 'Number';
    $scope.reverseSort = false;

    //Getting all the data for teams from the backend.
    $scope.getTeamData = function() {
        $http.get($rootScope.apiUrl + "/Team/Details?id=" + $routeParams.teamId)
        .success(function(data)
        {
            if(data.status === "success"){
                $scope.team = data;
                //Rest is used for timeintervals.
                $scope.dateArray = [];
                $scope.startTimes = [];
                $scope.endTimes = [];

                $scope.startDate = new Date(parseInt(data.TimeIntervals[0].StartTime.substr(6)));
                $scope.startDate.setHours(0);
                $scope.startDate.setSeconds(0);
                $scope.startDate.setMinutes(0);
                $scope.startDate.setMilliseconds(0);
                $scope.endDate = new Date(parseInt(data.TimeIntervals[data.TimeIntervals.length-
1].EndTime.substr(6)));
                $scope.endDate.setHours(0);
                $scope.endDate.setSeconds(0);
                $scope.endDate.setMinutes(0);
                $scope.endDate.setMilliseconds(0);
                $scope.dateRange = ($scope.endDate - $scope.startDate) / (1000 * 60 * 60 * 24);

                for (var index = 0; index < data.TimeIntervals.length; index++) {
                    var date = new Date($scope.startDate.getTime());
                    date.setDate(date.getDate() + index);
                    $scope.dateArray.push(date);
                    $scope.startTimes.push(new
Date(parseInt(data.TimeIntervals[index].StartTime.substr(6))));
                    $scope.endTimes.push(new
Date(parseInt(data.TimeIntervals[index].EndTime.substr(6))));
                }
                for(var i = 0; i < $scope.team.Matches.length; i++)
                {
                    $scope.team.Matches[i].StartTime = new
Date(parseInt($scope.team.Matches[i].StartTime.substr(6)));
                }
            } else {
                $scope.error = "Kunne ikke læse hold";
            }
        }).error(function(err)
        {
            $scope.error = err;
        })
    }

    $scope.getTeamData();

    $scope.changeTeamNameFunc = function() {

```

```

    $scope.changeName = !$scope.changeName;
}

//This function is used to change the name of the team. Done by making a post request
//to the backend which calls the Edit function
//in TeamController, and sends the required parameters with it.
$scope.changeNewTeamNameFunc = function(newName) {
    $http.post($rootScope.apiUrl + "/Team/Edit", { id: $routeParams.teamId, name:
newName, poolId: $routeParams.poolId, startTimes: $scope.startTimes, endTimes:
$scope.endTimes})
        .success(function(data){
            if(data.status === "success"){
                $scope.team.Name = data.newName;
                $scope.changeTeamNameFunc();
                $scope.getTeamData();
            } else {
                $scope.error = "Kunne ikke ændre hold-navn";
            }
        }).error(function(err){
            $scope.editErr = err;
        })
}

//The remove function is used to delete a specific team with its id as a parameter. The
function is making a post request to the backend
//calling the Delete function
$scope.remove = function() {
    var deleteTeam = $window.confirm('Er du sikker på at du vil slette holdet? \nHvis
kampprogrammet har været planlagt, vil det nu blive slettet og kan tage lidt tid');

    if(deleteTeam) {
        $http.post($rootScope.apiUrl + "/Team/Delete", { id: $routeParams.teamId })
            .success(function(data) {
                if(data.status === "success"){
                    $location.path("/tournament/" + $routeParams.tournamentId + "/division/" +
$routeParams.divisionId + "/pool/" + $routeParams.poolId);
                } else {
                    $scope.error = "Kunne ikke slette hold";
                }
            }).error(function(data) {
                $scope.deleteErr = data;
            })
        }
    }

    $scope.gotoDivision = function() {
        $location.url("/tournament/" + $routeParams.tournamentId + "/division/" +
$routeParams.divisionId);
    }

    $scope.gotoTournament = function() {
        $location.url("/tournament/" + $routeParams.tournamentId);
    }

    $scope.gotoPool = function() {
        $location.url("/tournament/" + $routeParams.tournamentId + "/division/" +
$routeParams.divisionId + "/pool/" + $routeParams.poolId);
    }

```

```

}

$scope.gotoTeam = function(currTeam) {
    $location.url("/tournament/" + $routeParams.tournamentId + "/division/" +
$routeParams.divisionId + "/pool/" + $routeParams.poolId + "/team/" + currTeam.Id);
}

/* Change time intervals start */
//All of the following are used to set the TimeIntervals for the team.
$scope.toggleMin = function () {
    $scope.minDate = $scope.minDate ? null : new Date();
};

$scope.toggleMin();

$scope.maxDate = new Date(2020, 5, 22);

$scope.openEndDate = function ($event) {
    $scope.statusEndDate.opened = true;
};

$scope.isChanged = function () {
    $scope.dateRange = ($scope.endDate - $scope.startDate) / (1000 * 60 * 60 * 24);
    $scope.dateArray = [];
    $scope.startTimes = [];
    $scope.endTimes = [];
    for (var i = 0; i <= $scope.dateRange; i++) {
        var date = new Date($scope.startDate.getTime());
        date.setDate(date.getDate() + i);
        $scope.dateArray.push(date);
        $scope.startTimes.push(date);
        $scope.endTimes.push(date);
    }
}

$scope.openStartDate = function ($event) {
    $scope.statusStartDate.opened = true;
};
$scope.dateOptions = {
    formatYear: 'yy',
    startingDay: 1
};
$scope.formats = ['dd-MMMM-yyyy', 'yyyy/MM/dd', 'dd.MM.yyyy', 'shortDate'];
$scope.format = $scope.formats[0];
$scope.statusStartDate = {
    opened: false
};
$scope.statusEndDate = {
    opened: false
};

$scope.errMsg = function () {
    $scope.error = !$scope.error;
}

$scope.successMsg = function () {
    $scope.isSuccess = !$scope.isSuccess;
}

```

```

$scope.uploadTeam = function () {
    if (!$scope.team.Name){
        $scope.error = "Navn er ikke sat";
    }else{
        $scope.startDateTimes = [];
        $scope.endDateTimes = [];
        $scope.error = false;
        for (var index = 0; index < $scope.startTimes.length; index++) {
            $scope.startDateTimes[index] = $scope.startTimes[index];
            $scope.endDateTimes[index] = $scope.endTimes[index];
        }
        if($scope.startDateTimes.length-1 !== $scope.dateRange &&
        $scope.endDateTimes.length-1 !== $scope.dateRange){
            $scope.error = "Fejl i start eller slut tidspunkt for en af dagene";
        }else{
            for(var i = 0; i <= $scope.dateRange; i++){
                if($scope.startDateTimes[i] >= $scope.endDateTimes[i]){
                    $scope.error = "Alle slut tidspunkter skal være senere end start
tidspunkter";
                }
            }
            if(!$scope.error){
                var teamData = {
                    id: $scope.team.Id,
                    name: $scope.team.Name,
                    poolId: $routeParams.poolId,
                    startTimes: $scope.startDateTimes,
                    endTimes: $scope.endDateTimes
                }
                $http.post($rootScope.apiUrl + "/Team/Edit/", teamData)
                .success(function(Data)
                {
                    if(Data.status === "success"){
                        $scope.isSuccess = true;
                        $scope.getTeamData();
                    } else {
                        $scope.error = "Kunne ikke opdatere hold";
                    }
                })
                .error(function(err)
                {
                    $scope.isSuccess = false;
                    $scope.error = err;
                    $scope.getTeamData();
                });
            }
        }
    }
}
}
}
}
});

```