1. Install-Package Bootstrap

2. Insert references to index.html:

<script src="Scripts/jquery-1.9.1.js"></script>

<link href="Content/bootstrap.css" rel="stylesheet" />

3. Insert meta tags (just below <head>):

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

4. Change <li> item to <input/> element:

<li ng-repeat="sport in sportList">

<input type="text" ng-blur="onBlur(sport)" ng-model="sport.Caption" />

</li>

5. Add functions to sportsController.js:

var patchOk = function () {

alert('Product successfully updated.')

};

$scope.onBlur = function(sport) {

console.log(sport);

$http({ method: 'PATCH', url: '/odata/Sports/(' + sport.Id + ')', data: angular.toJson({ Caption: sport.Caption }) }).error(error).success(patchOk);

}

6. Add delete button to <li> item:

<li ng-repeat="sport in sportList">

<input type="text" ng-blur="onBlur(sport)" ng-model="sport.Caption" />

<span class="glyphicon glyphicon-remove" title="&times;" data-ng-click="deleteSport(sport)"></span>

</li>

7. Add functions to sportsController.js:

var deleteOk = function () {

alert('Product successfully deleted.')

loadSports();

};

$scope.deleteSport = function(sport) {

$http({ method: 'DELETE', url: '/odata/Sports/(' + sport.Id + ')'}).error(error).success(deleteOk);

}

8. Add query elements:

<p>

<input type="text" ng-model="sportQuery" size="30"/><button class="btn-default" ng-click="getSports()" >Query</button>

</p>

9. Add functions to sportsController.js:

$scope.sportQuery = '';

$scope.getSports = function () {

var queryPart = $scope.sportQuery ? '?' + $scope.sportQuery : ''

$http.get('/odata/Sports/' + queryPart).success(sportsLoaded).error(error);

}

10. Show queries:

a) $filter=Caption eq 'Sport 0'

b) $filter=startswith(Caption,'Sport')

c) $filter=substringof('S',Caption)

d) $filter=Caption eq 'Sport 0'or Caption eq 'Sport 1'

e) $skip=1&$top=2

f) $skip=1&$top=2&$select=Caption (*entity properties returned only!*)

11. *Explanation*: Queries being executed on the (database) server side.

12. Insert section just before </configuration> end element:

<system.diagnostics>

<trace autoflush="true" indentsize="4">

<listeners>

<add name="myListener" type="System.Diagnostics.TextWriterTraceListener" initializeData="C:\Temp\demo.log" />

</listeners>

</trace>

</system.diagnostics>

13. Add public ctor to SportsController to log EF into a trace file:

public SportsController()

: base()

{

db.Database.Log = x => { System.Diagnostics.Trace.Write(x); };

}

14. Define action "ChangeCaption" in webApiConfig.cs

var builder = new ODataConventionModelBuilder();

builder.EntitySet<Sport>("Sports");

ActionConfiguration changeCaption = builder.Entity<Sport>().Action("ChangeCaption");

changeCaption.Parameter<string>("Caption");

changeCaption.Returns<string>();

config.Routes.MapODataServiceRoute("odata", "odata", builder.GetEdmModel());

15. Add ChangeAction method to SportsController:

[HttpPost]

public async Task<IHttpActionResult> ChangeCaption([FromODataUri] int key, ODataActionParameters parameters)

{

if (!ModelState.IsValid)

{

return BadRequest();

}

string caption = (string)parameters["Caption"];

Sport sport = await db.Sports.FindAsync(key);

if (sport == null)

{

return NotFound();

}

var currentCaption = sport.Caption;

sport.Caption = caption;

db.SaveChanges();

return Ok(currentCaption + ":" + caption);

}

17. Show action (FunctionImport) in service metadata at http://localhost:60580/odata/$metadata

16. Update SportsController.js:

var changeActionOk = function (data) {

alert('Response: ' + data.value);

}

$scope.onBlur = function(sport) {

console.log(sport);

// comment out original PATCH call..

// $http({ method: 'PATCH', url: '/odata/Sports/(' + sport.Id + ')', data: angular.toJson({ Caption: sport.Caption }) }).error(error).success(patchOk);

// and call Sport.ChangeCaption action via POST...

$http({ method: 'POST', url: '/odata/Sports/(' + sport.Id + ')/ChangeCaption', data: angular.toJson({ Caption: sport.Caption }) }).error(error).success(changeActionOk);

}

17. Update ChangeAction to be a transient action:

ActionConfiguration changeCaption = builder.Entity<Sport>().TransientAction("ChangeCaption");

changeCaption.Parameter<string>("Caption");

changeCaption.Returns<string>();

changeCaption.HasActionLink(ctx => {

var sport = ctx.EntityInstance as Sport;

if (sport.Id % 2 == 0)

{

return new System.Uri(ctx.Url.CreateODataLink(

new System.Web.Http.OData.Routing.EntitySetPathSegment(ctx.EntitySet),

new System.Web.Http.OData.Routing.KeyValuePathSegment(sport.Id.ToString()),

new System.Web.Http.OData.Routing.ActionPathSegment(changeCaption.Name)));

}

else

{

return null;

}

}, followsConventions: true);

18. Show generated serialized Sport entity(ies) at http://localhost:60580/odata/Sports