23 Web API -1 DOTNET

The Web Api Assignment is titled "Purchase Order Processing System (POPS)".

The back-end is a small database with four tables containing the details of

Suppliers,

Items and

Purchase orders.

Web api consumed using two different ways:

1. Using J-query
2. Using Http Client

@section Scripts {

<script src="~/Scripts/jquery-1.10.2.min.js"></script>

@Scripts.Render("~/bundles/jqueryval")

<script type="text/javascript">

$(document).ready(function () {

var apiBaseUrl = "http://localhost:54204/";

//$('#btnGetData').click(function () {

$.ajax({

url: apiBaseUrl + 'api/PosService/GetOrderDetails',

type: 'GET',

dataType: 'json',

success: function (data) {

var $table = $('<table/>').addClass('dataTable table table-bordered table-striped');

var $header = $('<thead/>').html('<tr> <td>PO Number</td><td>PO Date</td><td>Supplier Name</td><td>Supplier Address</td><td>Item</td><td>Quantity</td><td>Rate</td><td>Total Amount</td></tr>');

$table.append($header);

$.each(data, function (i, val) {

var $row = $('<tr/>');

$row.append($('<td/>').html(val.PoNumber));

$row.append($('<td/>').html(val.PoDate));

$row.append($('<td/>').html(val.SupplierName));

$row.append($('<td/>').html(val.SupplierAddress));

$row.append($('<td/>').html(val.Item));

$row.append($('<td/>').html(val.Quantity));

$row.append($('<td/>').html(val.Rate));

$row.append($('<td/>').html(val.TotalAmount));

$table.append($row);

});

$('#updatePanel').html($table);

},

error: function () {

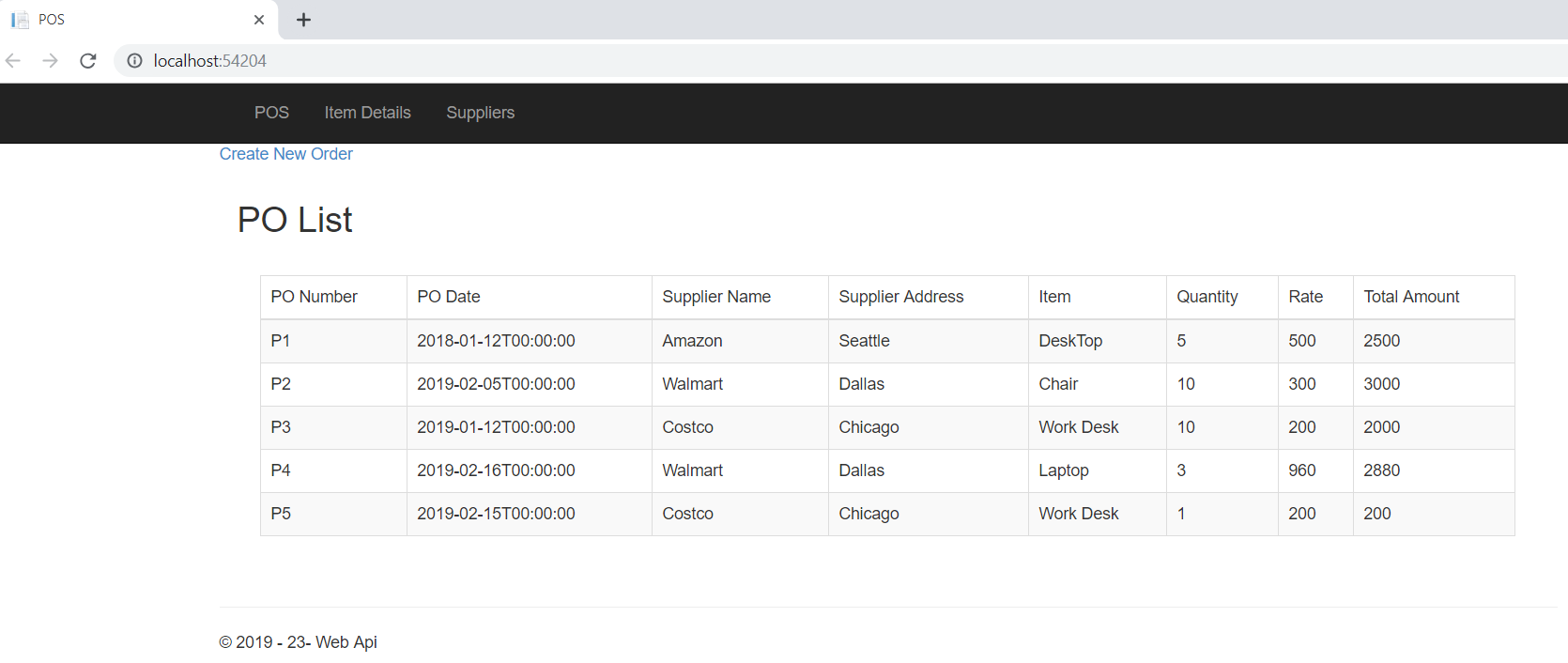
alert('Error!');

}

});

//});

});



1. Items – written api for GET/PUT/PSOT/DELETE

Sd

1. Created Supplier api to get GET/PUT/POST/DELETE Supplier information.

GET Sample –

// GET: api/Suppliers

public IEnumerable<SupplierModel> GetSuppliers()

{

var suppliersList = \_dbcontext.SUPPLIERs

.Select(i => new SupplierModel

{

SupplierNumber = i.SUPLNO,

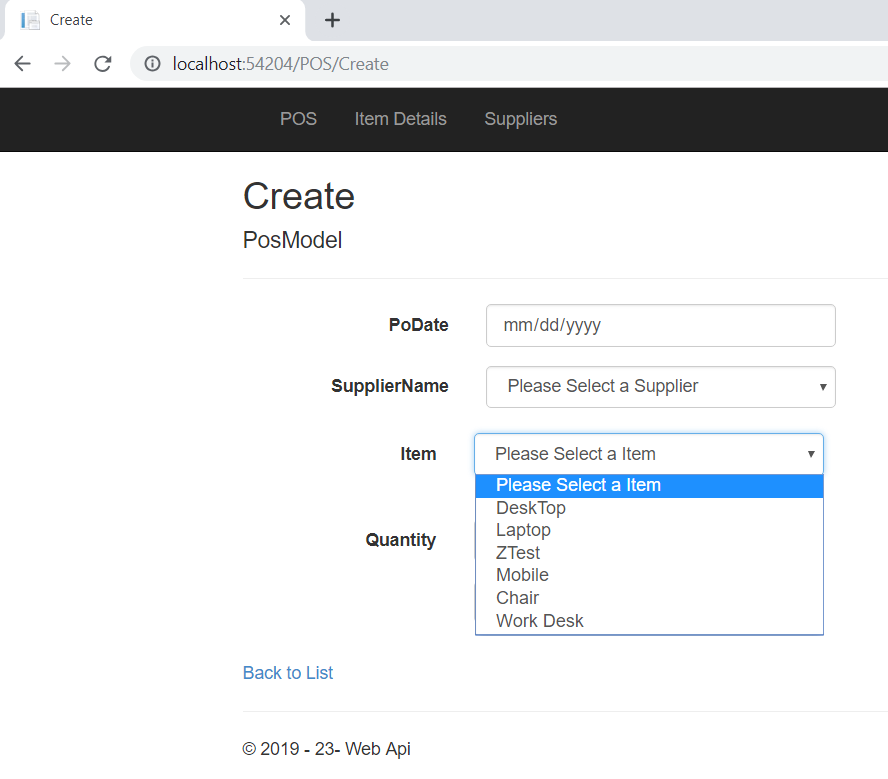
Name = i.SUPLNAME,

Address = i.SUPLADDR

});

return suppliersList;

}



@section Scripts {

@\*// @Scripts.Render("~/bundles/jqueryval")\*@

<script src="~/Scripts/jquery-1.10.2.min.js"></script>

<script type="text/javascript">

$(document).ready(function () {

var apiBaseUrl = "http://localhost:54204/";

$.ajax({

type: "GET",

url: apiBaseUrl + "api/SupplierService/GetSuppliers",

data: "{}",

success: function (data) {

var s = '<option value="-1">Please Select a Supplier</option>';

for (var i = 0; i < data.length; i++) {

s += '<option value="' + data[i].SupplierNumber + '">' + data[i].Name + '</option>';

}

$("#SupplierDropdown").html(s);

}

});

$.ajax({

type: "GET",

url: apiBaseUrl + "api/ItemService/GetItems",

data: "{}",

success: function (data) {

var s = '<option value="-1">Please Select a Item</option>';

for (var i = 0; i < data.length; i++) {

s += '<option value="' + data[i].ItemCode + '">' + data[i].Description + '</option>';

}

$("#ItemDropdown").html(s);

}

});

$("#btnPost").click(function () {

var pos = new Object();

pos.PoDate = $('#PoDate').val();

pos.Quantity = $('#Quantity').val();

pos.ItemCode = $('#ItemDropdown').val();

pos.SupplierNo = $('#SupplierDropdown').val();

if (pos != null) {

//alert($('#SupplierDropdown').val())

$.ajax({

type: "POST",

url: apiBaseUrl+"api/PosService/AddOrder",

data: JSON.stringify(pos),

contentType: "application/json; charset=utf-8",

dataType: "json",

success: function (response) {

if (response != null) {

alert("PoNumber : " + response.PoNumber);

} else {

alert("Something went wrong");

}

},

failure: function (response) {

alert(response.responseText);

},

error: function (response) {

alert(response.responseText);

}

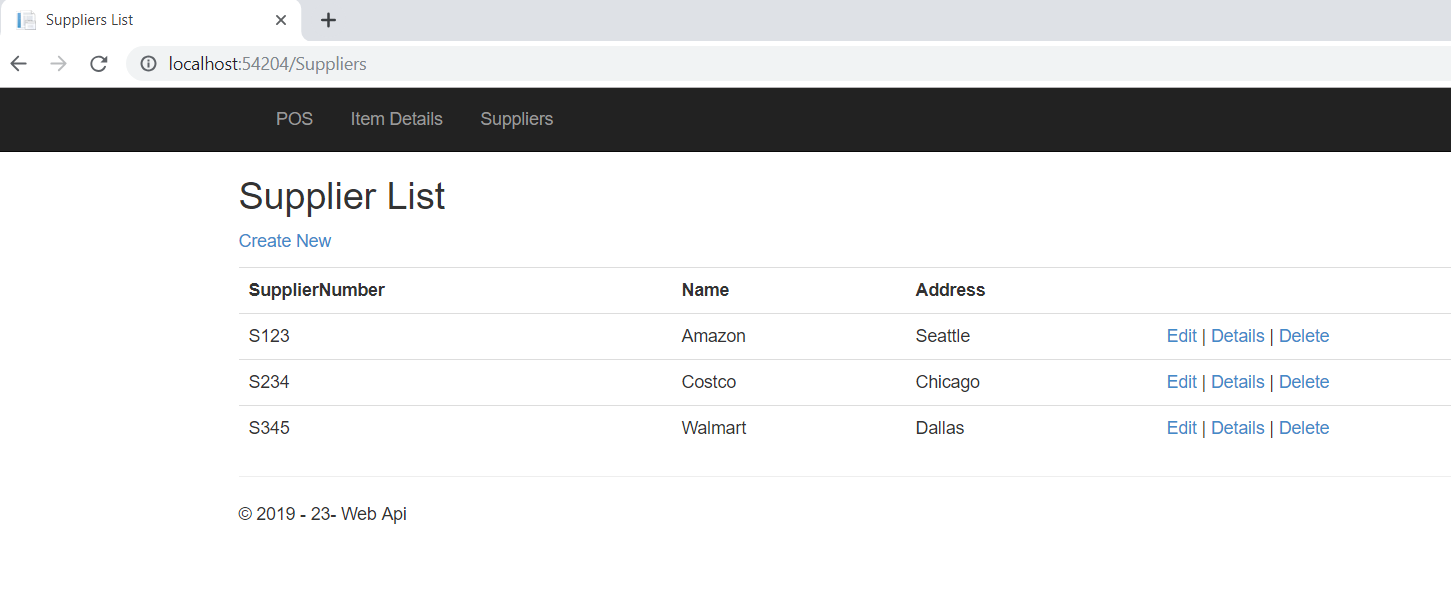
});

}

});

});

</script>}



Supplier service consumed using Http Client

// GET: Suppliers/Details/5

public ActionResult Details(string id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

string operation = baseAddress + "api/SupplierService/GetSupplier/" + id;

HttpResponseMessage response = client.GetAsync(operation).Result;

SupplierModel serviceResponse = null;

if (response.IsSuccessStatusCode)

{

serviceResponse = response.Content.ReadAsAsync<SupplierModel>().Result;

}

if (serviceResponse == null)

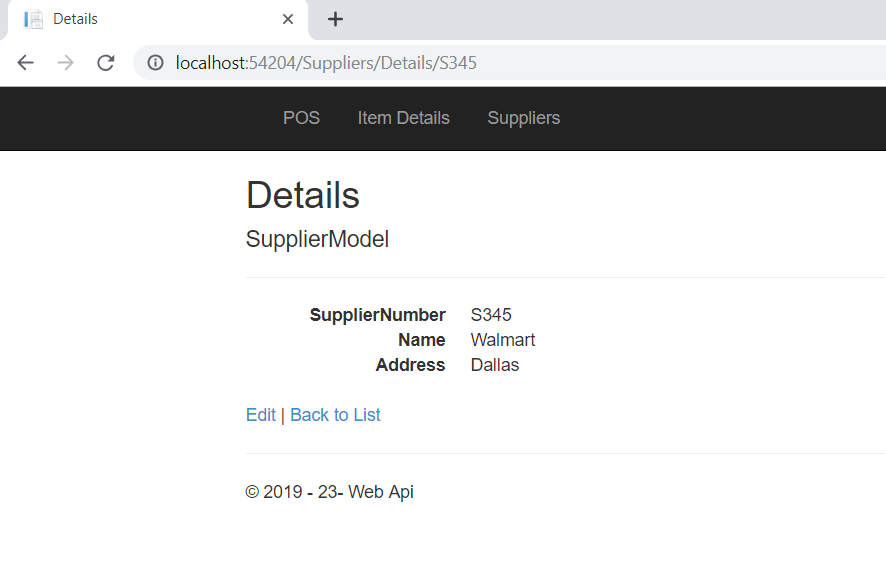
{

return HttpNotFound();

}

return View(serviceResponse);

}



Create supplier:

// POST: api/Suppliers

[ResponseType(typeof(SupplierModel))]

public IHttpActionResult AddSupplier(SupplierModel supplier)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

if (supplier != null)

{

var sp = new SUPPLIER()

{

SUPLNO=supplier.SupplierNumber,

SUPLNAME=supplier.Name,

SUPLADDR=supplier.Address

};

\_dbcontext.SUPPLIERs.Add(sp);

}

try

{

\_dbcontext.SaveChanges();

}

catch (DbUpdateException)

{

if (SupplierExists(supplier.SupplierNumber))

{

return Conflict();

}

else

{ throw;

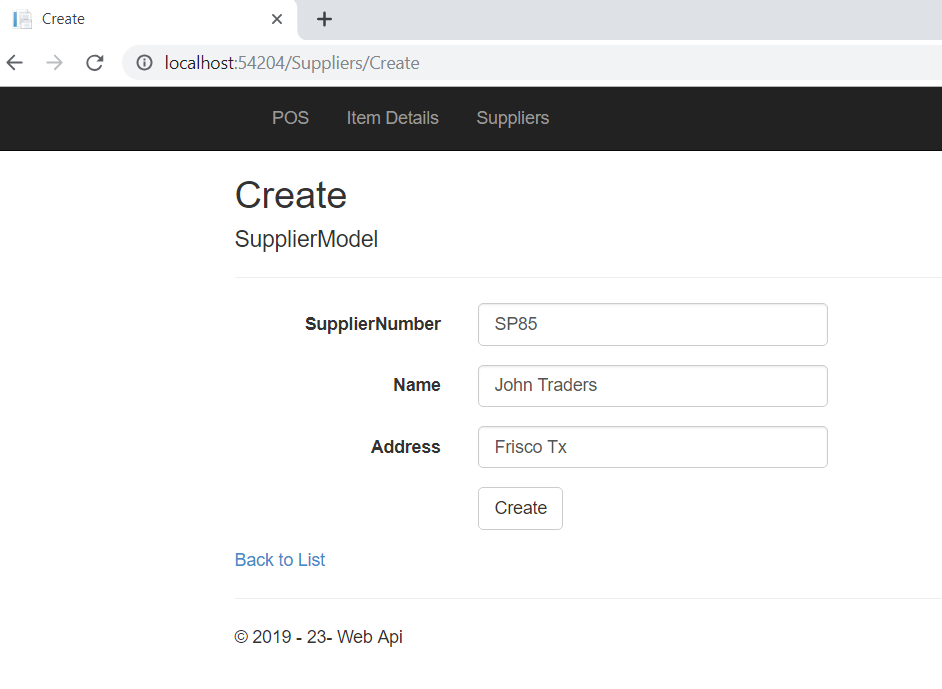
}

}

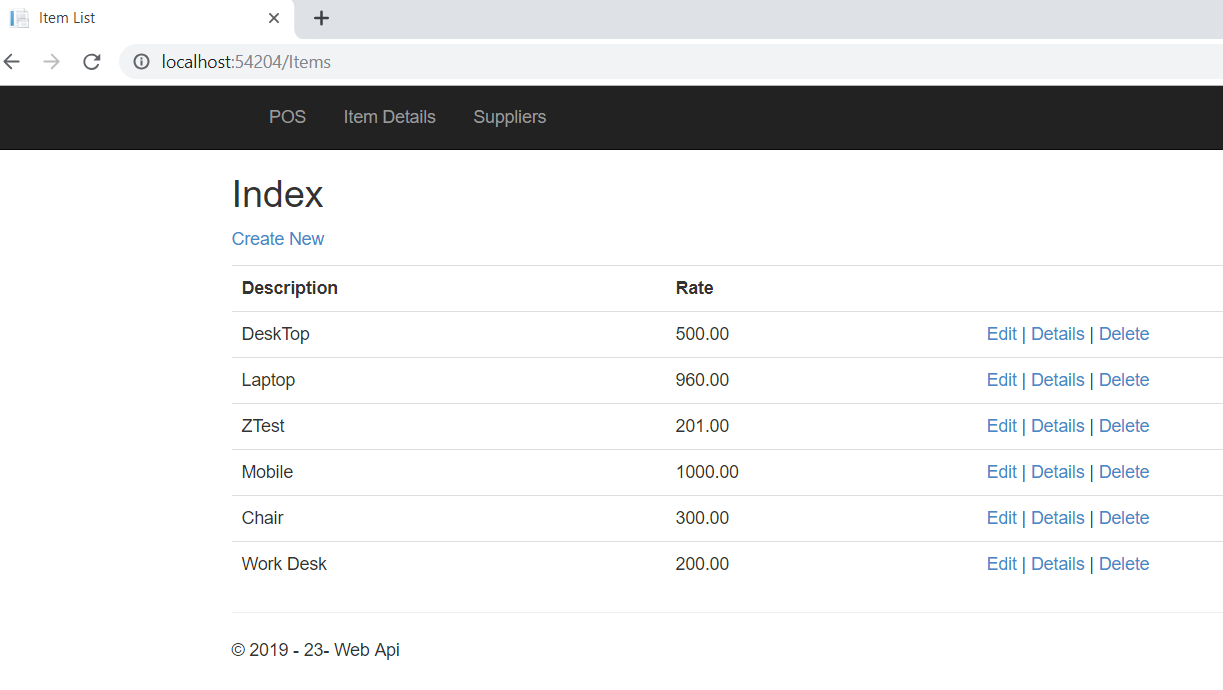
return CreatedAtRoute("PosApi",

new { id = supplier.SupplierNumber }, supplier);

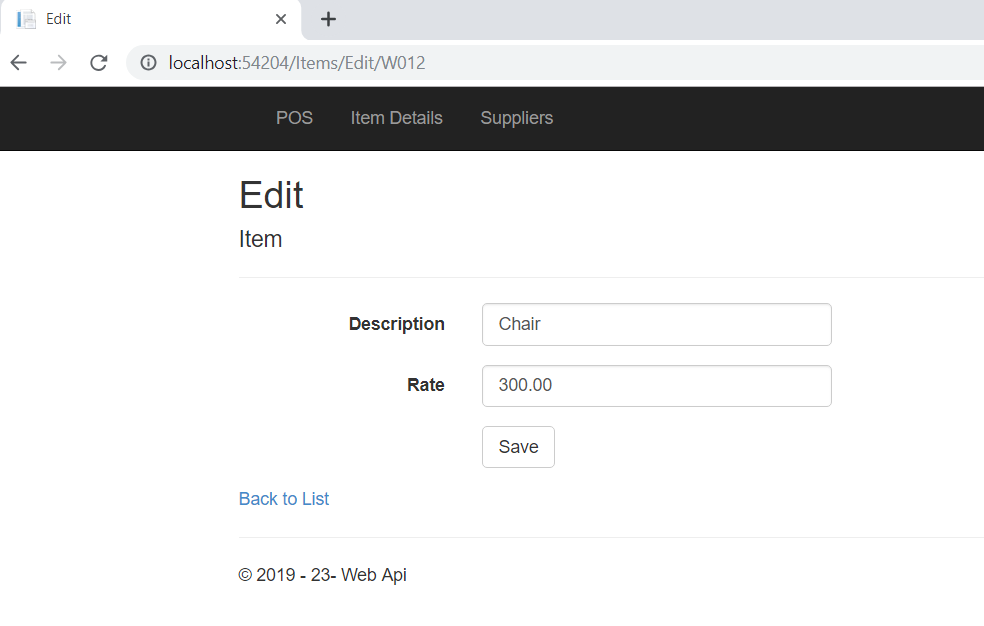
}



1. Item web api provides GET/PUT/POST and DELETE features for Items.



PUT:



[HttpPut]

[ResponseType(typeof(void))]

public IHttpActionResult UpdateItem(string id, ItemsModel updtItm)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

if (id != updtItm.ItemCode)

{

return BadRequest();

}

var filter = \_dbcontext.ITEMs

.Where(x => x.ITCODE.ToLower() == updtItm.ItemCode.ToLower())

.FirstOrDefault();

if (filter != null)

{

filter.ITCODE = updtItm.ItemCode;

filter.ITDESC = updtItm.Description;

filter.ITRATE = updtItm.Rate;

}

\_dbcontext.Entry(filter).State = EntityState.Modified;

try

{

\_dbcontext.SaveChanges();

}

catch (DbUpdateConcurrencyException)

{

if (!ItemExists(id))

{

return NotFound();

}

else

{

throw;

}

}

return StatusCode(HttpStatusCode.NoContent);

}

1. POS service will add new purchase order using existing supplier and Item details.

Note: No validations are implanted as this is implemented as concept.