<p>

So I am further along the SPA (Single Page Application) journey. What started out as a small grid is now a full fledged SPA done my way. Later on down the line, in the future, I will figure out how to port the code to do it the Microsoft way. Meantime here is the code.

</p>

Index.html

<pre>

&lt;!DOCTYPE html&gt;

&lt;html&gt;

&lt;head&gt;

&lt;meta charset="utf-8" /&gt;

&lt;title&gt;Star Cars&lt;/title&gt;

&lt;meta name="viewport" content="width=device-width, initial-scale=1" /&gt;

&lt;link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css" /&gt;

&lt;link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap-theme.min.css" /&gt;

&lt;script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.js"&gt;&lt;/script&gt;

&lt;script src="http://ajax.aspnetcdn.com/ajax/knockout/knockout-3.1.0.js"&gt;&lt;/script&gt;

&lt;script src="https://cdnjs.cloudflare.com/ajax/libs/knockout-validation/2.0.3/knockout.validation.js"&gt;&lt;/script&gt;

&lt;script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"&gt;&lt;/script&gt;

&lt;script src="https://cdnjs.cloudflare.com/ajax/libs/sammy.js/0.7.6/sammy.min.js"&gt;&lt;/script&gt;

&lt;script&gt;

$(document).ready(function () {

function toggleSPAPage(SPAPageName, targetElementID) {

ko.cleanNode(document.getElementById('main'));

$('#' + targetElementID).html($('#' + SPAPageName).html());

eval(SPAPageName + '();');

}

function Car(item) {

var self = this;

if (item) {

self.carStockID = ko.observable(item.ID);

self.carStockName = ko.observable(item.Name).extend({ required: true });

self.carStockPrice = ko.observable(item.Price).extend({ required: true, number: true });

self.carStockQuantity = ko.observable(item.Quantity).extend({ required: true, number: true, min: 1, max: 100000000 });

self.carStockImage = ko.observable(item.Image);

} else {

self.carStockID = ko.observable();

self.carStockName = ko.observable().extend({ required: true });

self.carStockPrice = ko.observable().extend({ required: true, number: true, min: 1 });

self.carStockQuantity = ko.observable().extend({ required: true, number: true, min: 1, max: 100000000 });

self.carStockImage = ko.observable('');

}

self.errors = ko.validation.group(self);

self.carStockFileUpload = function (data, e) {

var file = e.target.files[0];

var reader = new FileReader();

reader.onloadend = function (onloadend\_e) {

var result = reader.result;

self.carStockImage(result);

};

if (file) {

reader.readAsDataURL(file);

}

};

}

function Salesman(item) {

var self = this;

if (item) {

self.salesmanID = ko.observable(item.ID);

self.salesmanName = ko.observable(item.Name).extend({ required: true });

self.salesmanCellNumber = ko.observable(item.CellNumber).extend({ required: true, number: true });

self.salesmanEmailAddress = ko.observable(item.EmailAddress).extend({ required: true, email: true });

self.salesmanCommissionPercentage = ko.observable(item.CommissionPercentage).extend({ required: true, number: true, min: 0, max: 100 });

self.salesmanJoined = ko.observable(item.Joined.split('T')[0]).extend({ required: true });;

self.salesmanResigned = ko.observable(item.Resigned.split('T')[0]).extend({ ResignedDateMustBeGreaterThanJoinedDate: self.salesmanID() });

} else {

self.salesmanID = ko.observable();

self.salesmanName = ko.observable().extend({ required: true });

self.salesmanCellNumber = ko.observable().extend({ required: true, number: true });

self.salesmanEmailAddress = ko.observable().extend({ required: true, email: true });

self.salesmanCommissionPercentage = ko.observable().extend({ required: true, number: true, min: 1, max: 100 });

var today = new Date();

self.salesmanJoined = ko.observable(today.getFullYear() + '-' + (today.getMonth() + 1 &lt; 10 ? '0' + (today.getMonth() + 1) : today.getMonth() + 1) + '-' + (today.getDate() &lt; 10 ? '0' + today.getdate : today.getDate())).extend({ required: true });

self.salesmanResigned = ko.observable('2200-12-30').extend({ ResignedDateMustBeGreaterThanJoinedDate: 9999 });

}

self.errors = ko.validation.group(self);

}

function Customer(item) {

var self = this;

if (item) {

self.customerID = ko.observable(item.ID);

self.customerName = ko.observable(item.Name).extend({ required: true });

self.customerAddress = ko.observable(item.Address).extend({ required: true });

self.customerCity = ko.observable(item.City).extend({ required: true });

self.customerState = ko.observable(item.State).extend({ required: true });

self.customerCountry = ko.observable(item.Country).extend({ required: true });

self.customerZipCode = ko.observable(item.ZipCode).extend({ required: true });

self.customerCellNumber = ko.observable(item.CellNumber).extend({ number: true });

self.customerEmailAddress = ko.observable(item.EmailAddress).extend({ email: true });

self.customerAddedOn = ko.observable(item.AddedOn.split('T')[0]).extend({ required: true, });

} else {

self.customerID = ko.observable();

self.customerName = ko.observable().extend({ required: true });

self.customerAddress = ko.observable().extend({ required: true });

self.customerCity = ko.observable().extend({ required: true });

self.customerState = ko.observable().extend({ required: true });

self.customerCountry = ko.observable().extend({ required: true });

self.customerZipCode = ko.observable().extend({ required: true });

self.customerCellNumber = ko.observable().extend({ number: true });

self.customerEmailAddress = ko.observable().extend({ email: true });

var today = new Date();

self.customerAddedOn = ko.observable(today.getFullYear() + '-' + (today.getMonth() + 1 &lt; 10 ? '0' + (today.getMonth() + 1) : today.getMonth() + 1) + '-' + (today.getDate() &lt; 10 ? '0' + today.getdate : today.getDate())).extend({ required: true });

}

self.errors = ko.validation.group(self);

}

function Sale(item) {

var self = this;

if (item) {

self.saleID = ko.observable(item.ID);

self.saleSalesmanID = ko.observable(item.SalesmanID).extend({ required: true });

self.saleCustomerID = ko.observable(item.CustomerID).extend({ required: true });

self.saleCarID = ko.observable(item.CarID).extend({ required: true });

self.salePrice = ko.observable(item.Price).extend({ required: true, number: true });

self.saleSaleDate = ko.observable(item.SaleDate.split('T')[0]).extend({ required: true });

} else {

self.saleID = ko.observable();

self.saleSalesmanID = ko.observable().extend({ required: true });

self.saleCustomerID = ko.observable().extend({ required: true });

self.saleCarID = ko.observable().extend({ required: true });

self.salePrice = ko.observable().extend({ required: true, number: true });

var today = new Date();

self.saleSaleDate = ko.observable(today.getFullYear() + '-' + (today.getMonth() + 1 &lt; 10 ? '0' + (today.getMonth() + 1) : today.getMonth() + 1) + '-' + (today.getDate() &lt; 10 ? '0' + today.getdate : today.getDate())).extend({ required: true });

}

self.errors = ko.validation.group(self);

}

function HomePage() {

}

function CarStockPage() {

function CarStockViewModel() {

var self = this;

self.carStockAvailableCars = ko.observableArray();

self.carStockPageSize = ko.observable(5);

self.carStockPageIndex = ko.observable(0);

self.carStockSelectedItem = ko.observable();

self.carStockOldItem = {};

self.carStockPagedList = ko.dependentObservable(function () {

var size = self.carStockPageSize();

var start = self.carStockPageIndex() \* size;

return self.carStockAvailableCars.slice(start, start + size);

});

self.carStockMaxPageIndex = ko.dependentObservable(function () {

return Math.ceil(self.carStockAvailableCars().length / self.carStockPageSize()) - 1;

});

self.carStockAllPages = ko.dependentObservable(function () {

var carStockPages = [];

for (i = 0; i &lt;= self.carStockMaxPageIndex() ; i++) {

carStockPages.push({ carStockPageNumber: (i + 1) });

}

return carStockPages;

});

self.carStockIncreasePrices = function () {

for (var i = 0; i &lt; self.carStockAvailableCars().length; i++) {

self.carStockAvailableCars()[i].carStockPrice((self.carStockAvailableCars()[i].carStockPrice() \* 1.1).toFixed(2));

}

};

self.carStockEdit = function (item) {

item.carStockCurrentTemplate = 'carStockEditTemplate';

self.carStockOldItem.Name = item.carStockName();

self.carStockOldItem.Price = item.carStockPrice();

self.carStockOldItem.Quantity = item.carStockQuantity();

self.carStockOldItem.Image = item.carStockImage();

self.carStockSelectedItem(item);

};

self.carStockCancelEdit = function (item) {

self.carStockSelectedItem().carStockCurrentTemplate = 'carStockShowTemplate';

self.carStockSelectedItem().carStockName(self.carStockOldItem.Name);

self.carStockSelectedItem().carStockPrice(self.carStockOldItem.Price);

self.carStockSelectedItem().carStockQuantity(self.carStockOldItem.Quantity);

self.carStockSelectedItem().carStockImage(self.carStockOldItem.Image);

self.carStockOldItem = {};

self.carStockSelectedItem(null);

};

self.carStockCancelAdd = function (item) {

self.carStockSelectedItem().carStockCurrentTemplate = 'carStockShowTemplate';

self.carStockSelectedItem(null);

self.carStockAvailableCars.remove(item);

};

self.carStockAdd = function () {

var newItem = new Car();

newItem.carStockID(9999);

newItem.carStockImage('JUNK2');

newItem.carStockCurrentTemplate = 'carStockAddTemplate';

self.carStockAvailableCars.push(newItem);

self.carStockSelectedItem(newItem);

self.carStockMoveToPage(self.carStockMaxPageIndex());

};

self.carStockRemove = function (item) {

if (item.carStockID()) {

if (confirm('Are you sure you wish to delete this car?')) {

$.ajax({

type: 'DELETE', url: '/api/Cars/' + item.carStockID(), success:

function (result) {

self.carStockAvailableCars.remove(item);

if (self.carStockPageIndex() &gt; self.carStockMaxPageIndex()) {

self.carStockMoveToPage(self.carStockMaxPageIndex());

}

}

});

}

}

else {

self.carStockAvailableCars.remove(item);

if (self.carStockPageIndex() &gt; self.carStockMaxPageIndex()) {

self.carStockMoveToPage(self.carStockMaxPageIndex());

}

}

};

self.carStockSaveAdd = function () {

if (self.carStockSelectedItem().errors().length === 0) {

var item = self.carStockSelectedItem();

var dataItem = {

ID: item.carStockID(), Name: item.carStockName(), Price: item.carStockPrice(),

Quantity: item.carStockQuantity(), Image: item.carStockImage()

};

$.post('/api/Cars', dataItem, function (result) {

item.carStockID(result.ID);

item.carStockCurrentTemplate = null;

self.carStockSelectedItem(null);

});

} else {

self.carStockSelectedItem().errors.showAllMessages();

}

};

self.carStockSaveEdit = function () {

if (self.carStockSelectedItem().errors().length === 0) {

var item = self.carStockSelectedItem();

var dataItem = {

ID: item.carStockID(), Name: item.carStockName(), Price: item.carStockPrice(),

Quantity: item.carStockQuantity(), Image: item.carStockImage()

};

$.ajax({

type: 'PUT', url: '/api/Cars/' + item.carStockID(), data: dataItem,

contentType: "application/x-www-form-urlencoded;charset=UTF-8", success:

function (result) {

item.carStockCurrentTemplate = 'carStockShowTemplate';

self.carStockSelectedItem(null);

}

});

} else {

self.carStockSelectedItem().errors.showAllMessages();

}

}

self.carStockTemplateToUse = function (item) {

return self.carStockSelectedItem() === item &amp;&amp; item.carStockCurrentTemplate === 'carStockEditTemplate' ? 'carStockEditTemplate' :

self.carStockSelectedItem() === item &amp;&amp; item.carStockCurrentTemplate === 'carStockAddTemplate' ? 'carStockAddTemplate' : 'carStockShowTemplate';

};

self.carStockPreviousPage = function () {

if (self.carStockPageIndex() &gt; 0) {

self.carStockPageIndex(self.carStockPageIndex() - 1);

}

};

self.carStockNextPage = function () {

if (self.carStockPageIndex() &lt; self.carStockMaxPageIndex()) {

self.carStockPageIndex(self.carStockPageIndex() + 1);

}

};

self.carStockMoveToPage = function (index) {

self.carStockPageIndex(index);

};

$.get('/api/Cars', function (data, status) {

self.carStockAvailableCars($.map(data, function (item) {

return new Car(item);

}));

});

}

ko.validation.init({

registerExtenders: true,

messagesOnModified: true,

insertMessages: true,

parseInputAttributes: true,

messageTemplate: null

}, true);

var carStockViewModel = new CarStockViewModel();

ko.applyBindings(carStockViewModel, document.getElementById('main'));

}

function SalesmanPage() {

function SalesmanViewModel() {

var self = this;

self.salesmen = ko.observableArray();

self.salesmanPageSize = ko.observable(5);

self.salesmanPageIndex = ko.observable(0);

self.salesmanSelectedItem = ko.observable();

self.salesmanOldItem = {};

self.salesmanPagedList = ko.dependentObservable(function () {

var size = self.salesmanPageSize();

var start = self.salesmanPageIndex() \* size;

return self.salesmen.slice(start, start + size);

});

self.salesmanMaxPageIndex = ko.dependentObservable(function () {

return Math.ceil(self.salesmen().length / self.salesmanPageSize()) - 1;

});

self.salesmanAllPages = ko.dependentObservable(function () {

var salesmanPages = [];

for (i = 0; i &lt;= self.salesmanMaxPageIndex() ; i++) {

salesmanPages.push({ salesmanPageNumber: (i + 1) });

}

return salesmanPages;

});

self.salesmanEdit = function (item) {

item.salesmanCurrentTemplate = 'salesmanEditTemplate';

self.salesmanOldItem.Name = item.salesmanName();

self.salesmanOldItem.CellNumber = item.salesmanCellNumber();

self.salesmanOldItem.EmailAddress = item.salesmanEmailAddress();

self.salesmanOldItem.CommissionPercentage = item.salesmanCommissionPercentage();

self.salesmanOldItem.Joined = item.salesmanJoined();

self.salesmanOldItem.Resigned = item.salesmanResigned();

self.salesmanSelectedItem(item);

};

self.salesmanCancelEdit = function (item) {

self.salesmanSelectedItem().salesmanCurrentTemplate = 'salesmanShowTemplate';

self.salesmanSelectedItem().salesmanName(self.salesmanOldItem.Name);

self.salesmanSelectedItem().salesmanCellNumber(self.salesmanOldItem.CellNumber);

self.salesmanSelectedItem().salesmanEmailAddress(self.salesmanOldItem.EmailAddress);

self.salesmanSelectedItem().salesmanCommissionPercentage(self.salesmanOldItem.CommissionPercentage);

self.salesmanSelectedItem().salesmanJoined(self.salesmanOldItem.Joined);

self.salesmanSelectedItem().salesmanResigned(self.salesmanOldItem.Resigned);

self.salesmanOldItem = {};

self.salesmanSelectedItem(null);

};

self.salesmanCancelAdd = function (item) {

self.salesmanSelectedItem().salesmanCurrentTemplate = 'salesmanShowTemplate';

self.salesmanSelectedItem(null);

self.salesmen.remove(item);

};

self.salesmanAdd = function () {

var newItem = new Salesman();

newItem.salesmanID(9999);

newItem.salesmanCurrentTemplate = 'salesmanAddTemplate';

self.salesmen.push(newItem);

self.salesmanSelectedItem(newItem);

self.salesmanMoveToPage(self.salesmanMaxPageIndex());

};

self.salesmanRemove = function (item) {

if (item.salesmanID()) {

if (confirm('Are you sure you wish to delete this car?')) {

$.ajax({

type: 'DELETE', url: '/api/Salesmen/' + item.salesmanID(), success:

function (result) {

self.salesmen.remove(item);

if (self.salesmanPageIndex() &gt; self.salesmanMaxPageIndex()) {

self.salesmanMoveToPage(self.salesmanMaxPageIndex());

}

}

});

}

}

else {

self.salesmen.remove(item);

if (self.salesmanPageIndex() &gt; self.salesmanMaxPageIndex()) {

self.salesmanMoveToPage(self.salesmanMaxPageIndex());

}

}

};

self.salesmanSaveAdd = function () {

if (self.salesmanSelectedItem().errors().length === 0) {

var item = self.salesmanSelectedItem();

var dataItem = {

ID: item.salesmanID(), Name: item.salesmanName(), CellNumber: item.salesmanCellNumber(),

EmailAddress: item.salesmanEmailAddress(), CommissionPercentage: item.salesmanCommissionPercentage(),

Joined: item.salesmanJoined(), Resigned: item.salesmanResigned()

};

$.post('/api/Salesmen', dataItem, function (result) {

item.salesmanID(result.ID);

item.salesmanCommissionPercentage(result.CommissionPercentage);

item.salesmanCurrentTemplate = null;

self.salesmanSelectedItem(null);

});

} else {

self.salesmanSelectedItem().errors.showAllMessages();

}

};

self.salesmanSaveEdit = function () {

if (self.salesmanSelectedItem().errors().length === 0) {

var item = self.salesmanSelectedItem();

var dataItem = {

ID: item.salesmanID(), Name: item.salesmanName(), CellNumber: item.salesmanCellNumber(),

EmailAddress: item.salesmanEmailAddress(), CommissionPercentage: item.salesmanCommissionPercentage(),

Joined: item.salesmanJoined(), Resigned: item.salesmanResigned()

};

$.ajax({

type: 'PUT', url: '/api/Salesmen/' + item.salesmanID(), data: dataItem, success:

function (result) {

item.salesmanCurrentTemplate = 'salesmanShowTemplate';

self.salesmanSelectedItem(null);

}

});

} else {

self.salesmanSelectedItem().errors.showAllMessages();

}

}

self.salesmanTemplateToUse = function (item) {

return self.salesmanSelectedItem() === item &amp;&amp; item.salesmanCurrentTemplate === 'salesmanEditTemplate' ? 'salesmanEditTemplate' :

self.salesmanSelectedItem() === item &amp;&amp; item.salesmanCurrentTemplate === 'salesmanAddTemplate' ? 'salesmanAddTemplate' : 'salesmanShowTemplate';

};

self.salesmanPreviousPage = function () {

if (self.salesmanPageIndex() &gt; 0) {

self.salesmanPageIndex(self.salesmanPageIndex() - 1);

}

};

self.salesmanNextPage = function () {

if (self.salesmanPageIndex() &lt; self.salesmanMaxPageIndex()) {

self.salesmanPageIndex(self.salesmanPageIndex() + 1);

}

};

self.salesmanMoveToPage = function (index) {

self.salesmanPageIndex(index);

};

$.get('/api/Salesmen', function (data, status) {

self.salesmen($.map(data, function (item) {

return new Salesman(item);

}));

});

}

var salesmanViewModel = new SalesmanViewModel();

ko.validation.rules['ResignedDateMustBeGreaterThanJoinedDate'] = {

validator: function (val, id) {

var salesmen = salesmanViewModel.salesmen();

for (var i = 0; i &lt; salesmen.length; i++) {

if (salesmen[i].salesmanID() == id) {

return salesmen[i].salesmanResigned() &gt; salesmen[i].salesmanJoined();

}

}

return false;

},

message: 'The Resigned date must be greater than the Joined date.'

};

ko.validation.init({

registerExtenders: true,

messagesOnModified: true,

insertMessages: true,

parseInputAttributes: true,

messageTemplate: null

}, true);

ko.applyBindings(salesmanViewModel, document.getElementById('main'));

}

function CustomerPage() {

function CustomerViewModel() {

var self = this;

self.customers = ko.observableArray();

self.customerPageSize = ko.observable(5);

self.customerPageIndex = ko.observable(0);

self.customerSelectedItem = ko.observable();

self.customerOldItem = {};

self.customerPagedList = ko.dependentObservable(function () {

var size = self.customerPageSize();

var start = self.customerPageIndex() \* size;

return self.customers.slice(start, start + size);

});

self.customerMaxPageIndex = ko.dependentObservable(function () {

return Math.ceil(self.customers().length / self.customerPageSize()) - 1;

});

self.customerAllPages = ko.dependentObservable(function () {

var customerPages = [];

for (i = 0; i &lt;= self.customerMaxPageIndex() ; i++) {

customerPages.push({ customerPageNumber: (i + 1) });

}

return customerPages;

});

self.customerEdit = function (item) {

item.customerCurrentTemplate = 'customerEditTemplate';

self.customerOldItem.Name = item.customerName();

self.customerOldItem.Address = item.customerAddress();

self.customerOldItem.City = item.customerCity();

self.customerOldItem.State = item.customerState();

self.customerOldItem.Country = item.customerCountry();

self.customerOldItem.ZipCode = item.customerZipCode();

self.customerOldItem.CellNumber = item.customerCellNumber();

self.customerOldItem.EmailAddress = item.customerEmailAddress();

self.customerOldItem.AddedOn = item.customerAddedOn();

self.customerSelectedItem(item);

};

self.customerCancelEdit = function (item) {

self.customerSelectedItem().customerCurrentTemplate = 'customerShowTemplate';

self.customerSelectedItem().customerName(self.customerOldItem.Name);

self.customerSelectedItem().customerAddress(self.customerOldItem.Address);

self.customerSelectedItem().customerCity(self.customerOldItem.City);

self.customerSelectedItem().customerState(self.customerOldItem.State);

self.customerSelectedItem().customerCountry(self.customerOldItem.Country);

self.customerSelectedItem().customerZipCode(self.customerOldItem.ZipCode);

self.customerSelectedItem().customerCellNumber(self.customerOldItem.CellNumber);

self.customerSelectedItem().customerEmailAddress(self.customerOldItem.EmailAddress);

self.customerSelectedItem().customerAddedOn(self.customerOldItem.AddedOn);

self.customerOldItem = {};

self.customerSelectedItem(null);

};

self.customerCancelAdd = function (item) {

self.customerSelectedItem().customerCurrentTemplate = 'customerShowTemplate';

self.customerSelectedItem(null);

self.customers.remove(item);

};

self.customerAdd = function () {

var newItem = new Customer();

newItem.customerID(9999);

newItem.customerCurrentTemplate = 'customerAddTemplate';

self.customers.push(newItem);

self.customerSelectedItem(newItem);

self.customerMoveToPage(self.customerMaxPageIndex());

};

self.customerRemove = function (item) {

if (item.customerID()) {

if (confirm('Are you sure you wish to delete this car?')) {

$.ajax({

type: 'DELETE', url: '/api/Customers/' + item.customerID(), success:

function (result) {

self.customers.remove(item);

if (self.customerPageIndex() &gt; self.customerMaxPageIndex()) {

self.customerMoveToPage(self.customerMaxPageIndex());

}

}

});

}

}

else {

self.customers.remove(item);

if (self.customerPageIndex() &gt; self.customerMaxPageIndex()) {

self.customerMoveToPage(self.customerMaxPageIndex());

}

}

};

self.customerSaveAdd = function () {

if (self.customerSelectedItem().errors().length === 0) {

var item = self.customerSelectedItem();

var dataItem = {

ID: item.customerID(), Name: item.customerName(), Address: item.customerAddress(), City: item.customerCity(),

State: item.customerState(), Country: item.customerCountry(), ZipCode: item.customerZipCode(), CellNumber: item.customerCellNumber(),

EmailAddress: item.customerEmailAddress(), AddedOn: item.customerAddedOn()

};

$.post('/api/Customers', dataItem, function (result) {

item.customerID(result.ID);

item.customerCurrentTemplate = null;

self.customerSelectedItem(null);

});

} else {

self.customerSelectedItem().errors.showAllMessages();

}

};

self.customerSaveEdit = function () {

if (self.customerSelectedItem().errors().length === 0) {

var item = self.customerSelectedItem();

var dataItem = {

ID: item.customerID(), Name: item.customerName(), Address: item.customerAddress(), City: item.customerCity(),

State: item.customerState(), Country: item.customerCountry(), ZipCode: item.customerZipCode(), CellNumber: item.customerCellNumber(),

EmailAddress: item.customerEmailAddress(), AddedOn: item.customerAddedOn()

};

$.ajax({

type: 'PUT', url: '/api/Customers/' + item.customerID(), data: dataItem, success:

function (result) {

item.customerCurrentTemplate = 'customerShowTemplate';

self.customerSelectedItem(null);

}

});

} else {

self.customerSelectedItem().errors.showAllMessages();

}

}

self.customerTemplateToUse = function (item) {

return self.customerSelectedItem() === item &amp;&amp; item.customerCurrentTemplate === 'customerEditTemplate' ? 'customerEditTemplate' :

self.customerSelectedItem() === item &amp;&amp; item.customerCurrentTemplate === 'customerAddTemplate' ? 'customerAddTemplate' : 'customerShowTemplate';

};

self.customerPreviousPage = function () {

if (self.customerPageIndex() &gt; 0) {

self.customerPageIndex(self.customerPageIndex() - 1);

}

};

self.customerNextPage = function () {

if (self.customerPageIndex() &lt; self.customerMaxPageIndex()) {

self.customerPageIndex(self.customerPageIndex() + 1);

}

};

self.customerMoveToPage = function (index) {

self.customerPageIndex(index);

};

$.get('/api/Customers', function (data, status) {

self.customers($.map(data, function (item) {

return new Customer(item);

}));

});

}

ko.validation.init({

registerExtenders: true,

messagesOnModified: true,

insertMessages: true,

parseInputAttributes: true,

messageTemplate: null

}, true);

var customerViewModel = new CustomerViewModel();

ko.applyBindings(customerViewModel, document.getElementById('main'));

}

function SalesPage() {

function SalesViewModel() {

var self = this;

self.sales = ko.observableArray();

self.saleSalesmen = ko.observableArray();

self.saleCustomers = ko.observableArray();

self.saleCars = ko.observableArray();

self.salePageSize = ko.observable(5);

self.salePageIndex = ko.observable(0);

self.saleSelectedItem = ko.observable();

self.saleOldItem = {};

self.getSalesmanByID = function (salesmanID) {

for (var i = 0; i &lt; self.saleSalesmen().length; i++) {

if (self.saleSalesmen()[i].salesmanID() == salesmanID()) {

return self.saleSalesmen()[i];

}

}

return 'undefined';

}

self.getCustomerByID = function (customerID) {

for (var i = 0; i &lt; self.saleCustomers().length; i++) {

if (self.saleCustomers()[i].customerID() == customerID()) {

return self.saleCustomers()[i];

}

}

return 'undefined';

}

self.getCarByID = function (carID) {

for (var i = 0; i &lt; self.saleCars().length; i++) {

if (self.saleCars()[i].carStockID() == carID()) {

return self.saleCars()[i];

}

}

return 'undefined';

}

self.getSalesmanDisplayValue = function (salesman) {

return salesman.salesmanName() + ' [' + salesman.salesmanCellNumber() + ']';

}

self.getCustomerDispayValue = function (customer) {

return customer.customerName() + ' [' + customer.customerCellNumber() + ', ' + customer.customerEmailAddress() + ']';

}

self.salePagedList = ko.dependentObservable(function () {

var size = self.salePageSize();

var start = self.salePageIndex() \* size;

return self.sales.slice(start, start + size);

});

self.saleMaxPageIndex = ko.dependentObservable(function () {

return Math.ceil(self.sales().length / self.salePageSize()) - 1;

});

self.saleAllPages = ko.dependentObservable(function () {

var salePages = [];

for (i = 0; i &lt;= self.saleMaxPageIndex() ; i++) {

salePages.push({ salePageNumber: (i + 1) });

}

return salePages;

});

self.saleEdit = function (item) {

item.saleCurrentTemplate = 'saleEditTemplate';

self.saleOldItem.SalesmanID = item.saleSalesmanID();

self.saleOldItem.CustomerID = item.saleCustomerID();

self.saleOldItem.CarID = item.saleCarID();

self.saleOldItem.Price = item.salePrice();

self.saleOldItem.SaleDate = item.saleSaleDate();

self.saleSelectedItem(item);

};

self.saleCancelEdit = function (item) {

self.saleSelectedItem().saleCurrentTemplate = 'saleShowTemplate';

self.saleSelectedItem().saleSalesmanID(self.saleOldItem.SalesmanID);

self.saleSelectedItem().saleCustomerID(self.saleOldItem.CustomerID);

self.saleSelectedItem().saleCarID(self.saleOldItem.CarID);

self.saleSelectedItem().salePrice(self.saleOldItem.Price);

self.saleSelectedItem().saleSaleDate(self.saleOldItem.SaleDate);

self.saleOldItem = {};

self.saleSelectedItem(null);

};

self.saleCancelAdd = function (item) {

self.saleSelectedItem().saleCurrentTemplate = 'saleShowTemplate';

self.saleSelectedItem(null);

self.sales.remove(item);

};

self.saleAdd = function () {

var newItem = new Sale();

newItem.saleID(9999);

newItem.saleCurrentTemplate = 'saleAddTemplate';

self.sales.push(newItem);

self.saleSelectedItem(newItem);

self.saleMoveToPage(self.saleMaxPageIndex());

};

self.saleRemove = function (item) {

if (item.saleID()) {

if (confirm('Are you sure you wish to delete this car?')) {

$.ajax({

type: 'DELETE', url: '/api/Sales/' + item.saleID(), success:

function (result) {

self.sales.remove(item);

if (self.salePageIndex() &gt; self.saleMaxPageIndex()) {

self.saleMoveToPage(self.saleMaxPageIndex());

}

}

});

}

}

else {

self.sales.remove(item);

if (self.salePageIndex() &gt; self.saleMaxPageIndex()) {

self.saleMoveToPage(self.saleMaxPageIndex());

}

}

};

self.saleSaveAdd = function () {

if (self.saleSelectedItem().errors().length === 0) {

var item = self.saleSelectedItem();

var dataItem = {

ID: item.saleID(), SalesmanID: item.saleSalesmanID(), CustomerID: item.saleCustomerID(),

CarID: item.saleCarID(), Price: item.salePrice(), SaleDate: item.saleSaleDate()

};

$.post('/api/Sales', dataItem, function (result) {

item.saleID(result.ID);

item.saleCurrentTemplate = null;

self.saleSelectedItem(null);

});

} else {

self.saleSelectedItem().errors.showAllMessages();

}

};

self.saleSaveEdit = function () {

if (self.saleSelectedItem().errors().length === 0) {

var item = self.saleSelectedItem();

var dataItem = {

ID: item.saleID(), SalesmanID: item.saleSalesmanID(), CustomerID: item.saleCustomerID(),

CarID: item.saleCarID(), Price: item.salePrice(), SaleDate: item.saleSaleDate()

};

$.ajax({

type: 'PUT', url: '/api/Sales/' + item.saleID() + '?oldCarID=' + self.saleOldItem.CarID, data: dataItem, success:

function (result) {

item.saleCurrentTemplate = 'saleShowTemplate';

self.saleSelectedItem(null);

}

});

} else {

self.saleSelectedItem().errors.showAllMessages();

}

}

self.saleTemplateToUse = function (item) {

return self.saleSelectedItem() === item &amp;&amp; item.saleCurrentTemplate === 'saleEditTemplate' ? 'saleEditTemplate' :

self.saleSelectedItem() === item &amp;&amp; item.saleCurrentTemplate === 'saleAddTemplate' ? 'saleAddTemplate' : 'saleShowTemplate';

};

self.salePreviousPage = function () {

if (self.salePageIndex() &gt; 0) {

self.salePageIndex(self.salePageIndex() - 1);

}

};

self.saleNextPage = function () {

if (self.salePageIndex() &lt; self.saleMaxPageIndex()) {

self.salePageIndex(self.salePageIndex() + 1);

}

};

self.saleMoveToPage = function (index) {

self.salePageIndex(index);

};

$.get('/api/Salesmen', function (data, status) {

self.saleSalesmen($.map(data, function (item) {

return new Salesman(item);

}));

$.get('/api/Customers', function (data, status) {

self.saleCustomers($.map(data, function (item) {

return new Customer(item);

}));

$.get('/api/Cars', function (data, status) {

self.saleCars($.map(data, function (item) {

return new Car(item);

}));

$.get('/api/Sales', function (data, status) {

self.sales($.map(data, function (item) {

return new Sale(item);

}));

});

});

});

});

}

ko.validation.init({

registerExtenders: true,

messagesOnModified: true,

insertMessages: true,

parseInputAttributes: true,

messageTemplate: null

}, true);

var salesViewModel = new SalesViewModel();

ko.applyBindings(salesViewModel, document.getElementById('main'));

}

Sammy(function () {

this.get('#CarStockPage', function () {

toggleSPAPage('CarStockPage', 'main');

});

this.get('#SalesmenPage', function () {

toggleSPAPage('SalesmanPage', 'main');

});

this.get('#CustomersPage', function () {

toggleSPAPage('CustomerPage', 'main');

});

this.get('#SalesPage', function () {

toggleSPAPage('SalesPage', 'main');

});

this.get('', function () {

toggleSPAPage('HomePage', 'main');

});

}).run();

$('#copyrightYear').html((new Date()).getFullYear().toString());

});

&lt;/script&gt;

&lt;style type="text/css"&gt;

.txtboxStyle {

padding: 3px;

font-size: 16px;

border-style: solid;

border-radius: 5px;

border-color: #ebebeb;

color: #7a7a7a;

border-width: 1px;

}

.validationMessage {

color: red;

}

&lt;/style&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;div class="container-fluid"&gt;

&lt;div class="jumbotron page-header" style="background-color: black; color: white"&gt;

&lt;table&gt;

&lt;tr&gt;

&lt;td&gt;

&lt;img src="Images/audi.png" height="150" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;h1&gt;&amp;nbsp;&amp;nbsp;&amp;nbsp;&amp;nbsp;Star Cars&lt;/h1&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/table&gt;

&lt;/div&gt;

&lt;div&gt;

&lt;nav class="navbar navbar-inverse"&gt;

&lt;div class="container-fluid"&gt;

&lt;div class="navbar-header"&gt;

&lt;a class="navbar-brand" href="#"&gt;Star Cars&lt;/a&gt;

&lt;/div&gt;

&lt;div&gt;

&lt;ul class="nav navbar-nav"&gt;

&lt;li class="active"&gt;&lt;a href="#"&gt;Home&lt;/a&gt;&lt;/li&gt;

&lt;li&gt;&lt;a href="#CarStockPage"&gt;Cars in Stock&lt;/a&gt;&lt;/li&gt;

&lt;li&gt;&lt;a href="#SalesmenPage"&gt;Salesmen&lt;/a&gt;&lt;/li&gt;

&lt;li&gt;&lt;a href="#CustomersPage"&gt;Customers&lt;/a&gt;&lt;/li&gt;

&lt;li&gt;&lt;a href="#SalesPage"&gt;Sales&lt;/a&gt;&lt;/li&gt;

&lt;/ul&gt;

&lt;/div&gt;

&lt;/div&gt;

&lt;/nav&gt;

&lt;/div&gt;

&lt;div id="main"&gt;

&lt;/div&gt;

&lt;footer&gt;

&lt;p&gt;&amp;copy; &lt;span id="copyrightYear"&gt;&lt;/span&gt; - Star Cars&lt;/p&gt;

&lt;/footer&gt;

&lt;/div&gt;

&lt;script id="HomePage" type="text/html"&gt;

&lt;h1&gt;Home Page Under Construction&lt;/h1&gt;

&lt;/script&gt;

&lt;script id="CarStockPage" type="text/html"&gt;

&lt;div&gt;

&lt;h2&gt;Cars In Stock&lt;/h2&gt;

&lt;p&gt;

&lt;a class="btn btn-primary" data-bind="click: $root.carStockAdd" href="#" title="add"&gt;&lt;i class="glyphicon glyphicon-plus"&gt;&lt;/i&gt;Add Car to Stock&lt;/a&gt;

&lt;/p&gt;

&lt;table class="table table-striped"&gt;

&lt;thead&gt;

&lt;tr&gt;

&lt;th&gt;ID&lt;/th&gt;

&lt;th&gt;Name&lt;/th&gt;

&lt;th&gt;Price&lt;/th&gt;

&lt;th&gt;Quantity&lt;/th&gt;

&lt;th&gt;Image&lt;/th&gt;

&lt;th style="width: 100px; text-align: right;" /&gt;

&lt;/tr&gt;

&lt;/thead&gt;

&lt;tbody data-bind="template: { name: carStockTemplateToUse, foreach: carStockPagedList }"&gt;&lt;/tbody&gt;

&lt;/table&gt;

&lt;ul class="pagination"&gt;

&lt;li data-bind="css: { disabled: carStockPageIndex() === 0 }"&gt;

&lt;a href="#" data-bind="click: carStockPreviousPage"&gt;Previous&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;ul class="pagination" data-bind="foreach: carStockAllPages"&gt;

&lt;li data-bind="css: { active: $data.carStockPageNumber === ($root.carStockPageIndex() + 1) }"&gt;

&lt;a href="#" data-bind="text: $data.carStockPageNumber, click: function () { $root.carStockMoveToPage($data.carStockPageNumber - 1); }"&gt;&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;ul class="pagination"&gt;

&lt;li data-bind="css: { disabled: carStockPageIndex() === carStockMaxPageIndex() }"&gt;

&lt;a href="#" data-bind="click: carStockNextPage"&gt;Next&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;p&gt;

&lt;button class="btn btn-default" data-bind="click: carStockIncreasePrices"&gt;Increase Prices&lt;/button&gt;

&lt;/p&gt;

&lt;/div&gt;

&lt;/script&gt;

&lt;script id="carStockShowTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td data-bind="text: carStockID"&gt;&lt;/td&gt;

&lt;td data-bind="text: carStockName"&gt;&lt;/td&gt;

&lt;td data-bind="text: 'Rs.' + carStockPrice"&gt;&lt;/td&gt;

&lt;td data-bind="text: carStockQuantity"&gt;&lt;/td&gt;

&lt;td&gt;&lt;img src="Images/orderedList0.png" data-bind="attr: { src: $data.carStockImage() }" /&gt;&lt;/td&gt;

&lt;td class="buttons"&gt;

&lt;a class="btn btn-success" data-bind="click: $root.carStockEdit" href="#" title="edit"&gt;&lt;i class="glyphicon glyphicon-edit"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.carStockRemove" href="#" title="remove"&gt;&lt;i class="glyphicon glyphicon-remove"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;script id="carStockEditTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td data-bind="text: carStockID" /&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: carStockName" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: carStockPrice" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: carStockQuantity" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input data-bind="event: { change: carStockFileUpload }" type="file" accept="image/\*" class="fileChooser" /&gt;

&lt;/td&gt;

&lt;td class="buttons"&gt;

&lt;a class="btn btn-success" data-bind="click: $root.carStockSaveEdit" href="#" title="save"&gt;&lt;i class="glyphicon glyphicon-ok"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.carStockCancelEdit" href="#" title="cancel"&gt;&lt;i class="glyphicon glyphicon-trash"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;script id="carStockAddTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td&gt;&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: carStockName" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: carStockPrice" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: carStockQuantity" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input data-bind="event: { change: $data.carStockFileUpload }" type="file" accept="image/\*" class="fileChooser" /&gt;

&lt;/td&gt;

&lt;td class="buttons"&gt;

&lt;a class="btn btn-success" data-bind="click: $root.carStockSaveAdd" href="#" title="save"&gt;&lt;i class="glyphicon glyphicon-ok"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.carStockCancelAdd" href="#" title="cancel"&gt;&lt;i class="glyphicon glyphicon-trash"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;script id="SalesmanPage" type="text/html"&gt;

&lt;div&gt;

&lt;h2&gt;Salesmen&lt;/h2&gt;

&lt;p&gt;

&lt;a class="btn btn-primary" data-bind="click: $root.salesmanAdd" href="#" title="add"&gt;&lt;i class="glyphicon glyphicon-plus"&gt;&lt;/i&gt;Add Salesman&lt;/a&gt;

&lt;/p&gt;

&lt;table class="table table-striped"&gt;

&lt;thead&gt;

&lt;tr&gt;

&lt;th&gt;ID&lt;/th&gt;

&lt;th&gt;Name&lt;/th&gt;

&lt;th&gt;Cell Number&lt;/th&gt;

&lt;th&gt;Email Address&lt;/th&gt;

&lt;th&gt;Commission Percentage&lt;/th&gt;

&lt;th&gt;Joined (MM/DD/YYYY)&lt;/th&gt;

&lt;th&gt;Resigned (MM/DD/YYYY)&lt;/th&gt;

&lt;th style="width: 100px; text-align: right;" /&gt;

&lt;/tr&gt;

&lt;/thead&gt;

&lt;tbody data-bind="template: { name: salesmanTemplateToUse, foreach: salesmanPagedList }"&gt;&lt;/tbody&gt;

&lt;/table&gt;

&lt;ul class="pagination"&gt;

&lt;li data-bind="css: { disabled: salesmanPageIndex() === 0 }"&gt;

&lt;a href="#" data-bind="click: salesmanPreviousPage"&gt;Previous&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;ul class="pagination" data-bind="foreach: salesmanAllPages"&gt;

&lt;li data-bind="css: { active: $data.salesmanPageNumber === ($root.salesmanPageIndex() + 1) }"&gt;

&lt;a href="#" data-bind="text: $data.salesmanPageNumber, click: function () { $root.salesmanMoveToPage($data.salesmanPageNumber - 1); }"&gt;&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;ul class="pagination"&gt;

&lt;li data-bind="css: { disabled: salesmanPageIndex() === salesmanMaxPageIndex() }"&gt;

&lt;a href="#" data-bind="click: salesmanNextPage"&gt;Next&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;/div&gt;

&lt;/script&gt;

&lt;script id="salesmanShowTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td data-bind="text: salesmanID"&gt;&lt;/td&gt;

&lt;td data-bind="text: salesmanName"&gt;&lt;/td&gt;

&lt;td data-bind="text: salesmanCellNumber"&gt;&lt;/td&gt;

&lt;td data-bind="text: salesmanEmailAddress"&gt;&lt;/td&gt;

&lt;td data-bind="text: salesmanCommissionPercentage"&gt;&lt;/td&gt;

&lt;td data-bind="text: (new Date(salesmanJoined()).getMonth() + 1) + '/' + new Date(salesmanJoined()).getDate() + '/' + new Date(salesmanJoined()).getFullYear()"&gt;&lt;/td&gt;

&lt;td data-bind="text: new Date(salesmanResigned()) &gt; new Date() ? 'Not Resigned' : (new Date(salesmanResigned()).getMonth() + 1) + '/' + new Date(salesmanResigned()).getDate() + '/' + new Date(salesmanResigned()).getFullYear()"&gt;&lt;/td&gt;

&lt;td class="buttons"&gt;

&lt;a class="btn btn-success" data-bind="click: $root.salesmanEdit" href="#" title="edit"&gt;&lt;i class="glyphicon glyphicon-edit"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.salesmanRemove" href="#" title="remove"&gt;&lt;i class="glyphicon glyphicon-remove"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;script id="salesmanEditTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td data-bind="text: salesmanID" /&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanName" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanCellNumber" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanEmailAddress" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanCommissionPercentage" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanJoined" type="date" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanResigned" type="date" /&gt;

&lt;/td&gt;

&lt;td class="buttons"&gt;

&lt;a class="btn btn-success" data-bind="click: $root.salesmanSaveEdit" href="#" title="save"&gt;&lt;i class="glyphicon glyphicon-ok"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.salesmanCancelEdit" href="#" title="cancel"&gt;&lt;i class="glyphicon glyphicon-trash"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;script id="salesmanAddTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td&gt;&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanName" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanCellNumber" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanEmailAddress" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanCommissionPercentage" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanJoined" type="date" /&gt;

&lt;/td&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: salesmanResigned" type="date" /&gt;

&lt;/td&gt;

&lt;td class="buttons"&gt;

&lt;a class="btn btn-success" data-bind="click: $root.salesmanSaveAdd" href="#" title="save"&gt;&lt;i class="glyphicon glyphicon-ok"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.salesmanCancelAdd" href="#" title="cancel"&gt;&lt;i class="glyphicon glyphicon-trash"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;script id="CustomerPage" type="text/html"&gt;

&lt;div&gt;

&lt;h2&gt;Customers&lt;/h2&gt;

&lt;p&gt;

&lt;a class="btn btn-primary" data-bind="click: $root.customerAdd" href="#" title="add"&gt;&lt;i class="glyphicon glyphicon-plus"&gt;&lt;/i&gt;Add Customer&lt;/a&gt;

&lt;/p&gt;

&lt;table class="table table-striped"&gt;

&lt;thead&gt;

&lt;tr&gt;

&lt;th&gt;ID&lt;/th&gt;

&lt;th&gt;Name&lt;/th&gt;

&lt;th&gt;Address&lt;/th&gt;

&lt;th&gt;City&lt;/th&gt;

&lt;th&gt;State&lt;/th&gt;

&lt;th&gt;Country&lt;/th&gt;

&lt;th&gt;Zip Code&lt;/th&gt;

&lt;th&gt;Cellphone Number&lt;/th&gt;

&lt;th&gt;Email Address&lt;/th&gt;

&lt;th&gt;Added On (MM/DD/YYYY)&lt;/th&gt;

&lt;th style="width: 100px; text-align: right;" /&gt;

&lt;/tr&gt;

&lt;/thead&gt;

&lt;tbody data-bind="template: { name: customerTemplateToUse, foreach: customerPagedList }"&gt;&lt;/tbody&gt;

&lt;/table&gt;

&lt;ul class="pagination"&gt;

&lt;li data-bind="css: { disabled: customerPageIndex() === 0 }"&gt;

&lt;a href="#" data-bind="click: customerPreviousPage"&gt;Previous&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;ul class="pagination" data-bind="foreach: customerAllPages"&gt;

&lt;li data-bind="css: { active: $data.customerPageNumber === ($root.customerPageIndex() + 1) }"&gt;

&lt;a href="#" data-bind="text: $data.customerPageNumber, click: function () { $root.customerMoveToPage($data.customerPageNumber - 1); }"&gt;&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;ul class="pagination"&gt;

&lt;li data-bind="css: { disabled: customerPageIndex() === customerMaxPageIndex() }"&gt;

&lt;a href="#" data-bind="click: customerNextPage"&gt;Next&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;/div&gt;

&lt;/script&gt;

&lt;script id="customerShowTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td data-bind="text: customerID"&gt;&lt;/td&gt;

&lt;td data-bind="text: customerName"&gt;&lt;/td&gt;

&lt;td data-bind="text: customerAddress"&gt;&lt;/td&gt;

&lt;td data-bind="text: customerCity"&gt;&lt;/td&gt;

&lt;td data-bind="text: customerState"&gt;&lt;/td&gt;

&lt;td data-bind="text: customerCountry"&gt;&lt;/td&gt;

&lt;td data-bind="text: customerZipCode"&gt;&lt;/td&gt;

&lt;td data-bind="text: customerCellNumber"&gt;&lt;/td&gt;

&lt;td data-bind="text: customerEmailAddress"&gt;&lt;/td&gt;

&lt;td data-bind="text: ((new Date(customerAddedOn())).getMonth() + 1) + '/' + (new Date(customerAddedOn())).getDate() + '/' + (new Date(customerAddedOn())).getFullYear()"&gt;&lt;/td&gt;

&lt;td class="buttons"&gt;

&lt;a class="btn btn-success" data-bind="click: $root.customerEdit" href="#" title="edit"&gt;&lt;i class="glyphicon glyphicon-edit"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.customerRemove" href="#" title="remove"&gt;&lt;i class="glyphicon glyphicon-remove"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;script id="customerEditTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td colspan="10"&gt;

&lt;table style="width:100%"&gt;

&lt;tr&gt;

&lt;th style="width:150px"&gt;ID:&lt;/th&gt;

&lt;td data-bind="text: customerID" /&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Name:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerName" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Address:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerAddress" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;City:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerCity" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;State:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerState" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Country:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerCountry" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Zip Code:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerZipCode" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Cellphone Number:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerCellNumber" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Email Address:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerEmailAddress" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Added On:&lt;/th&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: customerAddedOn" type="date" /&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;td&gt;

&lt;a class="btn btn-success" data-bind="click: $root.customerSaveEdit" href="#" title="save"&gt;&lt;i class="glyphicon glyphicon-ok"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.customerCancelEdit" href="#" title="cancel"&gt;&lt;i class="glyphicon glyphicon-trash"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/table&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;script id="customerAddTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td colspan="10"&gt;

&lt;table style="width:100%"&gt;

&lt;tr&gt;

&lt;th style="width:150px"&gt;Name:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerName" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Address:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerAddress" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;City:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerCity" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;State:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerState" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Country:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerCountry" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Zip Code:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerZipCode" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Cellphone Number:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerCellNumber" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Email Address:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: customerEmailAddress" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Added On:&lt;/th&gt;

&lt;td&gt;

&lt;input class="txtboxStyle" data-bind="value: customerAddedOn" type="date" /&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;td&gt;

&lt;a class="btn btn-success" data-bind="click: $root.customerSaveAdd" href="#" title="save"&gt;&lt;i class="glyphicon glyphicon-ok"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.customerCancelAdd" href="#" title="cancel"&gt;&lt;i class="glyphicon glyphicon-trash"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/table&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;script id="SalesPage" type="text/html"&gt;

&lt;div&gt;

&lt;h2&gt;Sales&lt;/h2&gt;

&lt;p&gt;

&lt;a class="btn btn-primary" data-bind="click: $root.saleAdd" href="#" title="add"&gt;&lt;i class="glyphicon glyphicon-plus"&gt;&lt;/i&gt;Add Sale&lt;/a&gt;

&lt;/p&gt;

&lt;table class="table table-striped"&gt;

&lt;thead&gt;

&lt;tr&gt;

&lt;th&gt;ID&lt;/th&gt;

&lt;th&gt;Salesman Name&lt;/th&gt;

&lt;th&gt;Customer Name&lt;/th&gt;

&lt;th&gt;Customer Cell Number&lt;/th&gt;

&lt;th&gt;Customer Email Address&lt;/th&gt;

&lt;th&gt;Car Name&lt;/th&gt;

&lt;th&gt;Price&lt;/th&gt;

&lt;th&gt;Sale Date (MM/DD/YYYY)&lt;/th&gt;

&lt;th style="width: 100px; text-align: right;" /&gt;

&lt;/tr&gt;

&lt;/thead&gt;

&lt;tbody data-bind="template: { name: saleTemplateToUse, foreach: salePagedList }"&gt;&lt;/tbody&gt;

&lt;/table&gt;

&lt;ul class="pagination"&gt;

&lt;li data-bind="css: { disabled: salePageIndex() === 0 }"&gt;

&lt;a href="#" data-bind="click: salePreviousPage"&gt;Previous&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;ul class="pagination" data-bind="foreach: saleAllPages"&gt;

&lt;li data-bind="css: { active: $data.salePageNumber === ($root.salePageIndex() + 1) }"&gt;

&lt;a href="#" data-bind="text: $data.salePageNumber, click: function () { $root.saleMoveToPage($data.salePageNumber - 1); }"&gt;&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;ul class="pagination"&gt;

&lt;li data-bind="css: { disabled: salePageIndex() === saleMaxPageIndex() }"&gt;

&lt;a href="#" data-bind="click: saleNextPage"&gt;Next&lt;/a&gt;

&lt;/li&gt;

&lt;/ul&gt;

&lt;/div&gt;

&lt;/script&gt;

&lt;script id="saleShowTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td data-bind="text: saleID"&gt;&lt;/td&gt;

&lt;td data-bind="text: $root.getSalesmanByID($data.saleSalesmanID).salesmanName"&gt;&lt;/td&gt;

&lt;td data-bind="text: $root.getCustomerByID($data.saleCustomerID).customerName"&gt;&lt;/td&gt;

&lt;td data-bind="text: $root.getCustomerByID($data.saleCustomerID).customerCellNumber"&gt;&lt;/td&gt;

&lt;td data-bind="text: $root.getCustomerByID($data.saleCustomerID).customerEmailAddress"&gt;&lt;/td&gt;

&lt;td data-bind="text: $root.getCarByID($data.saleCarID).carStockName"&gt;&lt;/td&gt;

&lt;td data-bind="text: 'Rs.' + salePrice"&gt;&lt;/td&gt;

&lt;td data-bind="text: (new Date(saleSaleDate()).getMonth() + 1) + '/' + new Date(saleSaleDate()).getDate() + '/' + new Date(saleSaleDate()).getFullYear()"&gt;&lt;/td&gt;

&lt;td class="buttons"&gt;

&lt;a class="btn btn-success" data-bind="click: $root.saleEdit" href="#" title="edit"&gt;&lt;i class="glyphicon glyphicon-edit"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.saleRemove" href="#" title="remove"&gt;&lt;i class="glyphicon glyphicon-remove"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;script id="saleEditTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td colspan="9"&gt;

&lt;table style="width:100%"&gt;

&lt;tr&gt;

&lt;th style="width:150px"&gt;ID:&lt;/th&gt;

&lt;td data-bind="text: saleID" /&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Salesman:&lt;/th&gt;

&lt;td&gt;&lt;select data-bind="options: $root.saleSalesmen, optionsText: $root.getSalesmanDisplayValue, optionsValue: 'salesmanID', value: $data.saleSalesmanID" style="width:100%"&gt;&lt;/select&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Customer:&lt;/th&gt;

&lt;td&gt;&lt;select data-bind="options: $root.saleCustomers, optionsText: $root.getCustomerDispayValue, optionsValue: 'customerID', value: $data.saleCustomerID" style="width:100%"&gt;&lt;/select&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Car:&lt;/th&gt;

&lt;td&gt;&lt;select data-bind="options: $root.saleCars, optionsText: 'carStockName', optionsValue: 'carStockID', value: $data.saleCarID" style="width:100%"&gt;&lt;/select&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Price:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: salePrice" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Sale Date:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: saleSaleDate" type="date" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;td&gt;

&lt;a class="btn btn-success" data-bind="click: $root.saleSaveEdit" href="#" title="save"&gt;&lt;i class="glyphicon glyphicon-ok"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.saleCancelEdit" href="#" title="cancel"&gt;&lt;i class="glyphicon glyphicon-trash"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/table&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;script id="saleAddTemplate" type="text/html"&gt;

&lt;tr&gt;

&lt;td colspan="9"&gt;

&lt;table style="width:100%"&gt;

&lt;tr&gt;

&lt;th&gt;Salesman:&lt;/th&gt;

&lt;td&gt;&lt;select data-bind="options: $root.saleSalesmen, optionsText: $root.getSalesmanDisplayValue, optionsValue: 'salesmanID', value: saleSalesmanID, optionsCaption: 'Select a Salesman.'" style="width:100%"&gt;&lt;/select&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Customer:&lt;/th&gt;

&lt;td&gt;&lt;select data-bind="options: $root.saleCustomers, optionsText: $root.getCustomerDispayValue, optionsValue: 'customerID', value: saleCustomerID, optionsCaption: 'Select a Customer.'" style="width:100%"&gt;&lt;/select&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Car:&lt;/th&gt;

&lt;td&gt;&lt;select data-bind="options: $root.saleCars, optionsText: 'carStockName', optionsValue: 'carStockID', value: saleCarID, optionsCaption: 'Select a car model.'" style="width:100%"&gt;&lt;/select&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Price:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: salePrice" style="width:100%" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;th&gt;Sale Date:&lt;/th&gt;

&lt;td&gt;&lt;input class="txtboxStyle" data-bind="value: saleSaleDate" type="date" /&gt;&lt;/td&gt;

&lt;/tr&gt;

&lt;tr&gt;

&lt;td&gt;

&lt;a class="btn btn-success" data-bind="click: $root.saleSaveAdd" href="#" title="save"&gt;&lt;i class="glyphicon glyphicon-ok"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;a class="btn btn-danger" data-bind="click: $root.saleCancelAdd" href="#" title="cancel"&gt;&lt;i class="glyphicon glyphicon-trash"&gt;&lt;/i&gt;&lt;/a&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/table&gt;

&lt;/td&gt;

&lt;/tr&gt;

&lt;/script&gt;

&lt;/body&gt;

&lt;/html&gt;

</pre>

The Car.cs Model C# file.

<pre>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace CarDealership.Models

{

public class Car

{

public int ID { get; set; }

public string Name { get; set; }

public float Price { get; set; }

public int Quantity { get; set; }

public string Image { get; set; }

}

}

</pre>

The Customer.cs Model C# file.

<pre>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace CarDealership.Models

{

public class Customer

{

public int ID { get; set; }

public string Name { get; set; }

public string Address { get; set; }

public string City { get; set; }

public string State { get; set; }

public string Country { get; set; }

public string ZipCode { get; set; }

public string CellNumber { get; set; }

public string EmailAddress { get; set; }

public DateTime AddedOn { get; set; }

}

}

</pre>

The Sale.cs Model C# file.

<pre>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace CarDealership.Models

{

public class Sale

{

public int ID { get; set; }

public int SalesmanID { get; set; }

public int CustomerID { get; set; }

public int CarID { get; set; }

public float Price { get; set; }

public DateTime SaleDate { get; set; }

}

}

</pre>

The Salesman.cs C# file.

<pre>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace CarDealership.Models

{

public class Salesman

{

public int ID { get; set; }

public string Name { get; set; }

public string CellNumber { get; set; }

public string EmailAddress { get; set; }

public float CommissionPercentage { get; set; }

public DateTime Joined { get; set; }

public DateTime Resigned { get; set; }

}

}

</pre>

<p>

Build the solution to register your model classes with MVC. Now go ahead and make controllers based on each of the model files using the Web API with Entity Framework option. After you are done you should have a CarsController.cs, CustomersController.cs, SalesController.cs and SalesmenController.cs files. Now that you have all your code setup you need to generate the model first database via entity framework. Build your solution to register all the classes. To do this go to Tools > NuGet Package Manager > Package Manager Console from the Visual Studio menu. In the Package Manager Console type in ‘enable-migrations –enableautomaticmigrations’ and run that command. When you created your controllers based on the models it generated a database context for you. So now type in add-migration command with the parameter as the name of the database context generated at the time of creating your controllers. Use just one database context for all your controllers. Then run the command update-database. Now you have the schema in SQL Server Express. If you want to run it in some other SQL Server other than the Visual Studio one you can go to the Web.config file and in the ConnectionStrings sections you can find a connection string named the same as the database context name and supply your own connection string with server name, custom database name, etc. One thing worth pointing out is when you are doing this custom approach delete the attach database and database name which is provided by default from the Visual Studio template. Otherwise you will get at run time errors that it does not have enough directory permissions etc.

</p>

<p>

On a side note of setting up SQL Server Express 2016 CTP 2.2 on VirtualBox VM and connecting it from your host machine: When setting up networking for the VM in VirtualBox settings what I did was setup Adapter 1 to use NAT, this gives the guest VM internet access, and on Adapter 3 put in Host. Now your VS Studio IIS Express ASP.NET MVC Web API database context can connect to your SQL Server. With host networking you can provide the netbios name in the connection string of the guest VM machine and don’t have to worry about IP address. Also of interest SQL Server Express 2016 CTP 2.2 does not install on Windows 7 even with all updates so. So I am running Windows Server 2012 with full updates on a VirtualBox VM and then installed SQL Server 2016 CTP 2.2. Since I didn’t select evaluation at setup time and instead choose Express there are some problems regarding SQL Server listening on TCP/IP and Netbios. Also the SQL Browser service should be on Automatic start. So to fix this networking problem since I was connecting remotely from the host OS to guest VM you need to startup Configuration Manager and enable TCP/IP and Named Pipes. This not unusual as I selected Express in the setup and hence it is expecting whatever is accessing it would be local to the machine and uses memory based networking. So now you are set as far as the database backend is concerned.

</p>

<p>

Now click the play button in Visual Studio and see the working result in your web browser which looks like the following for me.

</p>

<a href="https://aviebris.files.wordpress.com/2015/08/carsstock1.png"><img class="alignnone size-medium wp-image-33" src="https://aviebris.files.wordpress.com/2015/08/carsstock1.png?w=225" alt="CarsStock1" width="225" height="300" /></a> <a href="https://aviebris.files.wordpress.com/2015/08/carsstock2.png"><img class="alignnone size-medium wp-image-34" src="https://aviebris.files.wordpress.com/2015/08/carsstock2.png?w=225" alt="CarsStock2" width="225" height="300" /></a> <a href="https://aviebris.files.wordpress.com/2015/08/carsstock3.png"><img class="alignnone size-medium wp-image-35" src="https://aviebris.files.wordpress.com/2015/08/carsstock3.png?w=300" alt="CarsStock3" width="300" height="278" /></a> <a href="https://aviebris.files.wordpress.com/2015/08/carsstock4.png"><img class="alignnone size-medium wp-image-36" src="https://aviebris.files.wordpress.com/2015/08/carsstock4.png?w=300" alt="CarsStock4" width="300" height="278" /></a> <a href="https://aviebris.files.wordpress.com/2015/08/customer1.png"><img class="alignnone size-medium wp-image-37" src="https://aviebris.files.wordpress.com/2015/08/customer1.png?w=300" alt="Customer1" width="300" height="218" /></a> <a href="https://aviebris.files.wordpress.com/2015/08/sale1.png"><img class="alignnone size-medium wp-image-38" src="https://aviebris.files.wordpress.com/2015/08/sale1.png?w=300" alt="Sale1" width="300" height="218" /></a> <a href="https://aviebris.files.wordpress.com/2015/08/sale2.png"><img class="alignnone size-medium wp-image-39" src="https://aviebris.files.wordpress.com/2015/08/sale2.png?w=300" alt="Sale2" width="300" height="218" /></a> <a href="https://aviebris.files.wordpress.com/2015/08/sale3.png"><img class="alignnone size-medium wp-image-40" src="https://aviebris.files.wordpress.com/2015/08/sale3.png?w=300" alt="Sale3" width="300" height="218" /></a> <a href="https://aviebris.files.wordpress.com/2015/08/sale4.png"><img class="alignnone size-medium wp-image-41" src="https://aviebris.files.wordpress.com/2015/08/sale4.png?w=300" alt="Sale4" width="300" height="218" /></a> <a href="https://aviebris.files.wordpress.com/2015/08/salesman1.png"><img class="alignnone size-medium wp-image-42" src="https://aviebris.files.wordpress.com/2015/08/salesman1.png?w=300" alt="Salesman1" width="300" height="273" /></a>

<p>

One thing the Increases Prices button on the Cars in Stock page is only a demonstration to show that when you update the underlying property values in code the UI is immediately updated. This is a little non trivial because the objects whose Price property value is being updated in a loop are ko.observable() but they are all sitting inside an array which is ko.observable() array and yet the UI gets updated when you change an object in the array. The normal flow is when you bind an input control and the users changes its value the value of the underlying ko.observable() property is changed.

</p>

<p>

For how Knockout.js works you can check out <a href=<http://learn.knockoutjs.com/>>“Learn Knockout.js”</a>. The knockout code I am using is pretty straightforward. For the how paging works check out this <a href=<http://blog.ryanvanderpol.com/knockout-pager/>>article</a> I found on the internet and based my code on. So what I am going to describe is the troubles I ran into when coding all this code and not waste time explaining every single line of code.

</p>

<p>

In my SalesController’s PUT and POST methods you can see there is some custom logic there. What I wanted is when a car is sold the corresponding car model in inventory has its quantity reduced by 1. Also further logic is when you edit and change the car model being sold the old car is put back into inventory and hence goes up by 1 in quantity too. Now I had written this code the intuitive way and ran into some scary complex errors in the service call. Basically I could not do a db.Sales.find() call before I did my db.Entry(sale) call. So after much playing around with this I realized the solution was to not do a find at all on the to find out what was the old car model to update its quantity. Instead I decided from the $.ajax PUT call I would pass in the old CarID as a parameter. Ok so this led to more experimenting on routes. So I added a route (I am using MVC 5 and Visual Studio 2015) as follows to the RouteConfig.cs file which sits in the App\_Start folder in your solution explorer tree (the pane on the right unless you changed its default position in the Visual Studio UI).

<pre>

routes.MapRoute(

name: "UpdateSale",

url: "{controller}/{action}/{id}/{oldCarID}",

defaults: new { controller = "Sales", Action="Sale", id = 0, oldCarID = 0 }

);

</pre>

Now I haven’t done any of this before so I very smartly thought I could call the web service like /1/5. Of course this doesn’t work so another point to note is that when you add parameters to the route when you call them from ajax all parameters other than id have to be URL encoded ex. ‘api/Sales/1?oldCarID=5’. So the resulting code in javascript was build the string accordingly.

</p>

<p>

Another problem that was very hard to debug/fix was the datepicker. By the way datepicker on edit and add is not working in IE last I checked but works fine in Google Chrome. So use Google Chrome to test datepicker functionality out. To cut to the chase the jQuery datepicker control wants the value of the bound (ko.observable()) to be in the ISO format prescribed by the W3C as ‘YYYY-MM-DD’ and this cannot change. Hence all the funky code building the date string in that format. Also on the Salesmen grid a fun thing I did was that if the date is greater than today then the Resigned date field would display Not Resigned. Of course internally the Resigned field is not null its just a far future date which you can see I set to 2200-12-30 in the Salesman model when null is passed in.

</p>

<p>

Another interesting name since my UI side model property names are different from the ones in the backend side I build the JSON data object which I pass to $.ajax manually. I do the same for unwrapping them too the code for which is in the model definition.

</p>

<p>

One of the fun things I did which you can see is you can submit a picture using the usual input file control and it gets stored in the database as base64 string and served up and displayed on get. The tricky bit is in the Car model definition where a function is defined CarStockFileUpload. So this instead of being super complicated in most examples I have seen on the net while researching how to do this my solution is simple and elegant. Don’t worry about binary data just think of it as a simple observable string property in base64 encoding.

</p>

<p>

The Knockout-Validation code is simple. I only create one custom validation which checks that the Salesman’s resignation date is not less than their joining date. The basic concept is that val will be in this case the date in ‘YYYY-MM-DD’ format which is useless. So the second parameter I have added is the id of the row and then I go through the collection looking for the corresponding joining date and compare it and return the result. Again simple logic and elegant code.

</p>

<p>

I will keep updating this post with the latest code as I add more bells and whistles to Star Cars demo website and share it with you my fellow netizens. Till then party on, have fun and most importantly be excellent to each other.

</p>

<em>

When I first saw a production SPA’s code I was shocked by its complexity. So I took it up as a challenge that not only will I crack SPA and make my own I will make the code super simple to boot. My philosophy is if you really understand something well then you can make complicated things with it in a simple elegant easy to understand maintain and extend fashion without being arcane in logic. So since I have uncomplicated SPA I can say I understand SPA well now.

</em>