ASP MVC .Net and Angular JS Assessment Documentation

Problem:

Create a Client Edit screen that will return a client's details from a Data Layer to a web frontend. The client's details must be editable and on clicking of a save button must update the client's details. When the site is refreshed the new updated values must be returned to the view.

The fields required are:

- Surname
- First name(s)
- Identity Type – Passport or Identity Document
- Identity Number
- Date of Birth

Only one client record is needed to be returned from the data layer with the following data:

- Surname – Bloggs
- First name(s) – Joe Peter
- Identity Type – Identity Document
- Identity Number – 7202025074084
- Date of Birth – 1972/02/02

Technical requirements

- MVC .Net 4, 5 or 6
- Angular JS (preferably latest version of 1 but can use Angular 2 or 4)
- Use any design patterns you would like to and where they would be appropriate
- How the client data is stored is not important, you can use a local mdf database or xml file, json file etc. as long as it is included in the source code.
- You do not need to go to further lengths like including validations etc. (unless you would like to) as we more importantly want to see how you would structure your code and apply good coding practices and standards.
- You can include a README file with any notes you'd like us to be aware of when examining or running your code.

Solution:

Step-1: CREATE Operation – Add new client data

First, Add new method Insert in ClientController which will add Client entity into SQL database. So add the following code in Controllers –> ClientController.cs:

```
// GET Client/GetClient
        [HttpGet]
        public JsonResult GetClient()
        {
            using (ClientDetailsEntities db = new ClientDetailsEntities())
            {
                List<Client> clientList = db.Clients.ToList();
                return Json(clientList, JsonRequestBehavior.AllowGet);
            }

        }
```

Above code will insert client details into database, which is returning a JSON result of boolean data type. If it's true then it indicate that data inserted to database successfully. Second, we need to add following code in AngularJS_CRUD_ASPNET–> Client–> Service.js file which call MVC ClientController's Insert method using $http.post method:

```
//POST client/AddClient
        [HttpPost]
        public JsonResult Insert(Client client)
        {
            if (client != null)
            {
                using (ClientDetailsEntities db = new ClientDetailsEntities())
                {
                    db.Clients.Add(client);
                    db.SaveChanges();
                    return Json(new { success = true });
                }
            }
            else
            {
                return Json(new { success = false });
            }
        }
```

Third, we need to add following code in AngularJSApp –> Client> Controller.js file which will call angular clientService –> save method. once save is successful we are resetting $scope objects and reloading client data to display new record in list.

```
//save clients data
    $scope.save = function () {
        var Client = {
            Surname: $scope.Surname,
            FirstName: $scope.FirstName,
            IdentityType: $scope.IdentityType,
            IdentityNumber: $scope.IdentityNumber,
            DateOfBirth: $scope.DateOfBirth
        };
        var saverecords = clientService.save(Client);
        saverecords.then(function (d) {
            if (d.data.success === true) {
                loadClients();
                alert("Client added successfully");
                $scope.resetSave();
            }
            else { alert("Client not added."); }
        },
        function () {
            alert("Error occurred while adding client.");
        });
    }

    //reset controls after save operation
    $scope.resetSave = function () {
        $scope.Surname = '';
        $scope.FirstName = '';
        $scope.IdentityType = '';
        $scope.IdentityNumber = '';
        $scope.DateOfBirth = '';
    }
```

Finally, we need to add following html code in Views–> Home–> Index.cshtml file which will help us to submit form data using ng-controller and ng-model binding and later on this $scope object values will be used by $scope.save method to save the data:

```
@*New record Modal addition..*@
    <div class="modal" id="AddNew" role="dialog">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <button type="button" class="close" data-
dismiss="modal">&times;</button>
                    <h3 class="text-info">Add New Client</h3>
                </div>
                <div class="modal-body" style="margin-left:20px">
                    @*Add New Client form starts here...*@
                    <form class="form-horizontal" name="AddNewForm">
                        <div class="form-group">
                            <label class="control-label"> Surname</label>
                            <input class="form-control"  name="Surname" ng-
model="Surname" type="text" placeholder="Client surname" />
                        </div>
                        <div class="form-group">
                            <label class="control-label"> First Name</label>
                            <input class="form-control" name="FirstName" ng-
model="FirstName" type="text" placeholder="client Name" />
```

```
                        </div>
                        <div class="form-group">
                            <label class="control-label"> Identity Type</label>
                            <input class="form-control" name="IdentityType" ng-
model="IdentityType" type="text" placeholder="ID or Passport" />

                        </div>
                        <div class="form-group">
                            <label class="control-label"> Identity Number</label>
                            <input class="form-control" name="IdentityNumber" ng-
model="IdentityNumber" type="text" placeholder="Identity Number" />

                        </div>
                        <div class="form-group">
                            <label class="control-label"> Date Of Birth</label>
                            <input class="form-control" data-date-format="DD MMMM
YYYY" value="1900-08-09" name="DateOfBirth" ng-model="DateOfBirth" type="date"
placeholder="Date of Birth" />

                        </div>
                    </form>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-primary" id="btnSave" data-
dismiss="modal" ng-click="save()">Save</button>
                    <button type="button" class="btn btn-default" data-dismiss="modal"
ng-click="resetSave()">Close</button>
                </div>
            </div>
        </div>
    </div>
```

Next, let us add code for update operation.

Step-2: UPDATE Operation – Update existing client date

First, we need to add new method Update in ClientController which will update client entity into SQL database. So Let us add following code in Controllers –> ClientController.cs:

```
//POST Client/Update
        [HttpPost]
        public JsonResult Update(Client updatedClient)
        {
            using (ClientDetailsEntities db = new ClientDetailsEntities())
            {
                Client existingClient = db.Clients.Find(updatedClient.Id);
                if (existingClient == null)
                {
                    return Json(new { success = false });
                }
                else
                {
                    existingClient.Surname = updatedClient.Surname;
                    existingClient.FirstName = updatedClient.FirstName;
                    existingClient.IdentityType = updatedClient.IdentityType;
                    existingClient.IdentityNumber = updatedClient.IdentityNumber;
                    existingClient.DateOfBirth = updatedClient.DateOfBirth;
```

```
                db.SaveChanges();
                return Json(new { success = true });
            }
        }
    }
```

Above code will update existing Client details in database that is selected to update, which is also returning a JSON result of boolean data type. As previously If it's true then it indicate that data updated to database successfully.

Second, we need to add following code in AngularJSApp–> Client> Service.js file which call MVC ClientController's Update method using angular $http.post:

```
    //update Client records
    this.update = function (Client) {
        var updaterequest = $http({
            method: 'post',
            url: '/Client/Update',
            data: Client
        });
        return updaterequest;
    }
```

Third, we need to add following code in AngularJSApp–> Client–> Controller.js file which will call angular clientService->update method. once data updated successfully we are resetting $scope objects and reloading client data to display updated record in list.

```
//update Client data
    $scope.update = function () {
        var Client = {
            Surname: $scope.UpdateSurname,
            FirstName: $scope.UpdateFirstName,
            IdentityType: $scope.UpdateIdentityType,
            IdentityNumber: $scope.UpdateIdentityNumber,
            DateOfBirth: $scope.UpdateDateOfBirth
        };
        var updaterecords = clientService.update(Client);
        updaterecords.then(function (d) {
            if (d.data.success === true) {
                loadClients();
                alert("Client updated successfully");
                $scope.resetUpdate();
            }
            else {
                alert("Client not updated.");
            }
        },
        function () {
            alert("Error occured while updating client record");
        });
    }

    //reset controls after update
    $scope.resetUpdate = function () {
        $scope.UpdateSurname = '';
        $scope.UpdateFirstName = '';
```

```
        $scope.UpdateIdentityType = '';
        $scope.UpdateIdentityNumber = '';
        $scope.UpdateDateOfBirth = '';
    }
```

Finally, we need to add following html code in Views–> Home–> Index.cshtml file which will help us to submit form data using ng-controller and ng-model binding and later on this $scope object values will be used by $scope.update method to update the client data:

```
@*Update Client records*@
    <div class="modal" id="Update" role="dialog">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <button type="button" class="close" data-
dismiss="modal">&times;</button>
                    <h3 class="text-info">Update Existing Client</h3>
                </div>
                <div class="modal-body" style="margin-left:20px">
                    @*Update Client form starts here...*@
                    <form class="form-horizontal" name="UpdateClientForm">
                        <div class="form-group">
                            <label class="text-info"> ClientID</label>
                            <input class="form-control" readonly="readonly" name="Id"
ng-model="UpdateId" type="text" placeholder="ClientId" />
                        </div>
                        <div class="form-group">
                            <label class="text-info"> Surname</label>
                            <input class="form-control" name="Surname" ng-
model="UpdateSurname" type="text" placeholder="Surname" />
                        </div>
                        <div class="form-group">
                            <label class="text-info"> First Name</label>
                            <input class="form-control" name="FirstName" ng-
model="UpdateFirstName" type="text" placeholder="First Name" />
                        </div>
                        <div class="form-group">
                            <label class="text-info"> Identity Type</label>
                            <input class="form-control" name="IdentityType" ng-
model="UpdateIdentityType" type="text" placeholder="ID or Passport" />
                        </div>
                        <div class="form-group">
                            <label class="text-info"> Identity Number</label>
                            <input class="form-control" name="IdentityNumber" ng-
model="UpdateIdentityNumber" type="text" placeholder="Identity Number" />
                        </div>
                        <div class="form-group">
                            <label for="exampleInput" class="text-info"> Date of
birth</label>
                            <input class="form-control" id="exampleInput"
name="DateOfBirth" data-date-format="DD MMMM YYYY" value="1900-08-09" ng-
model="UpdateDateOfBirth" type="date" placeholder="Date Of Birth" />
                        </div>

                    </form>
                </div>
                <div class="modal-footer">
```

```
                    <button type="button" class="btn btn-primary" id="btnUpdate" data-
dismiss="modal" ng-click="update()">
                        Update
                    </button>
                    <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
                </div>
            </div>
        </div>
    </div>
```

Next, let us add code for delete operation.

Step-3: DELETE Operation – Delete client data

First, we need to add new method Delete in ClientController which will Delete client entity. So Let us add following code in Controllers –> ClientController.cs:

```
//POST Client/Delete/1
        [HttpPost]
        public JsonResult Delete(int Id)
        {
            using (ClientDetailsEntities db = new ClientDetailsEntities())
            {
                Client client = db.Clients.Find(Id);
                if (client == null)
                {
                    return Json(new { success = false });
                }
                db.Clients.Remove(client);
                db.SaveChanges();
                return Json(new { success = true });
            }

        }
```

Above code will delete existing client details from database that is selected to delete, which is also returning a JSON result of boolean data type. If it's true then it indicate that data deleted from database successfully.

Second, we need to add following code in AngularJSApp–> Client–> Service.js file which call MVC ClientController's Delete method using angular $http.post method:

```
    //delete record
    this.delete = function (UpdateId) {
        return $http.post('/Client/Delete/' + UpdateId);
    }
```

Third, we need to add following code in AngularJSApp–> C–> Controller.js file which will call angular clientService –> delete method. once delete is successful we are reloading client data to display updated list.

```
//POST Client/Delete/1
        [HttpPost]
        public JsonResult Delete(int Id)
```

```
        {
            using (ClientDetailsEntities db = new ClientDetailsEntities())
            {
                Client client = db.Clients.Find(Id);
                if (client == null)
                {
                    return Json(new { success = false });
                }
                db.Clients.Remove(client);
                db.SaveChanges();
                return Json(new { success = true });
            }

        }
```

Finally, we need to add following html code in Views–> Home–> Index.cshtml file which will bind $scope variables using angular expression {{variable}}.

```
@*Delete Client record*@
    <div class="modal" role="dialog" id="deleteDialog">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <button type="button" class="close" data-
dismiss="modal">&times;</button>
                    <h3 class="text-info">Are you sure you want to delete the
record?</h3>
                </div>
                <div class="modal-body">
                    <div>
                        <p>Client ID : {{UpdateId}}</p>
                        <p>Surname : {{UpdateSurname}}</p>
                    </div>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-danger" data-dismiss="modal"
ng-click="delete(UpdateId)">Delete Record</button>
                    <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
                </div>
            </div>
        </div>
    </div>
```

Next, let us update and add code to call ng-click events for Delete/Update operation from buttons.

Step-4: Integrate UPDATE and DELETE button for operations

First, we need to update the button html tags for ng-click events in Views–> Home–> Index.cshtml as per below:

```
<td style="width:200px;">
                        <a href="#"
                           class="btn btn-info"
                           data-toggle="modal"
                           data-target="#Update"
                           ng-click="getForUpdate(Client)">
                            Update
```

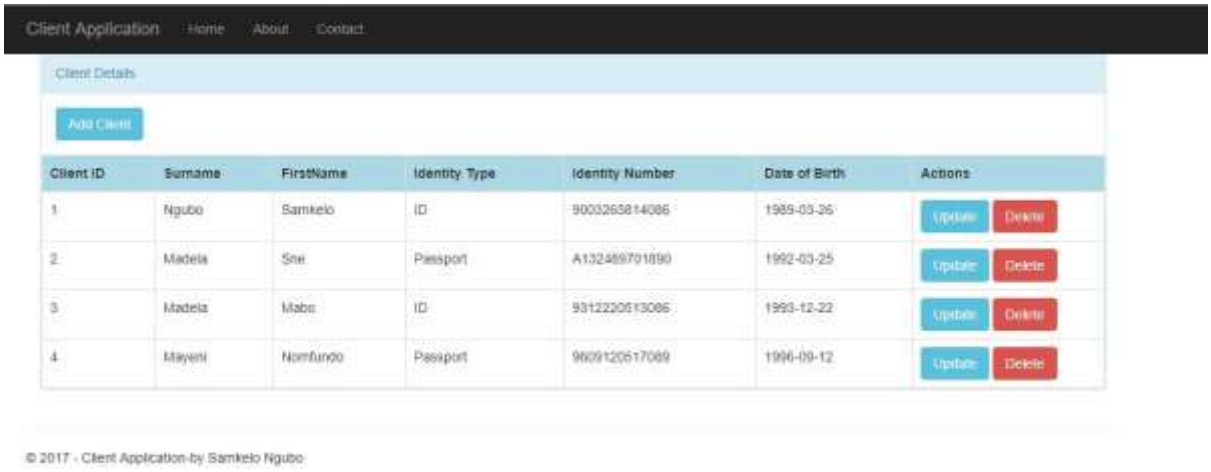
```
                </a>
                <a href="#" class="btn btn-danger"
                   id="btnDelete"
                   data-toggle="modal"
                   data-target="#deleteDialog"
                   ng-click="getForDelete(Client)">
                    Delete
                </a>
            </td>
```

Then add following angular events (getForUpdate & getForDelete) for Update/Delete button click events in AngularJSApp–> Client–> Controller.js file:

```
//get single record by ID
    $scope.getForUpdate = function (Client) {
        $scope.UpdateSurname = Client.Surname;
        $scope.UpdateFirstName = Client.FirstName;
        $scope.UpdateIdentityType = Client.IdentityType;
        $scope.UpdateIdentityNumber = Client.IdentityNumber;
        $scope.UpdateDateOfBirth = Client.DateOfBirth;
    }

    //get data for delete confirmation
    $scope.getForDelete = function (Client) {
        $scope.UpdateId = Client.Id;
        $scope.UpdateFirstName = Client.FirstName;
        $scope.UpdateSurname = Client.Surname;
    }
```

Above $scope variables are used during data-binding of client properties when model dialog open for Update/Delete operation.That's all. we are all set to run the application now.Step-5: Run the web application – F5

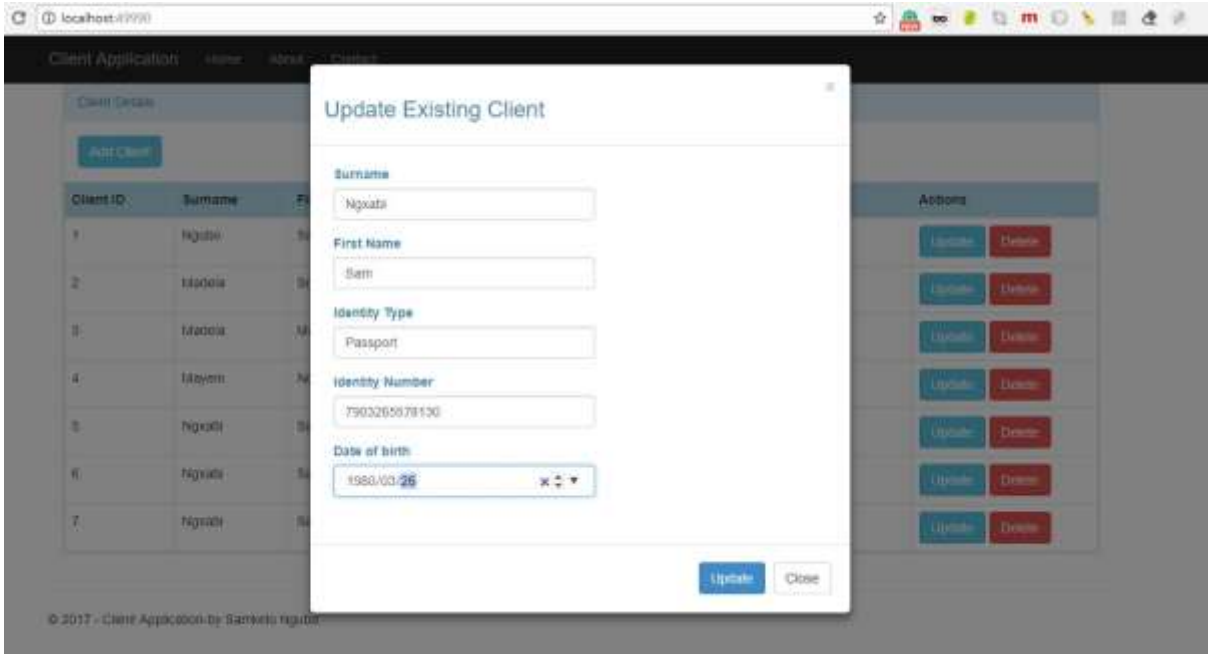


Press Add Client button, it will open model dialog.

Add the details for client and press save button. currently we are not using any data validation during insert/update operation but in next article we will see how to validation form data before saving it. Once data is saved, it will show alert of operation.

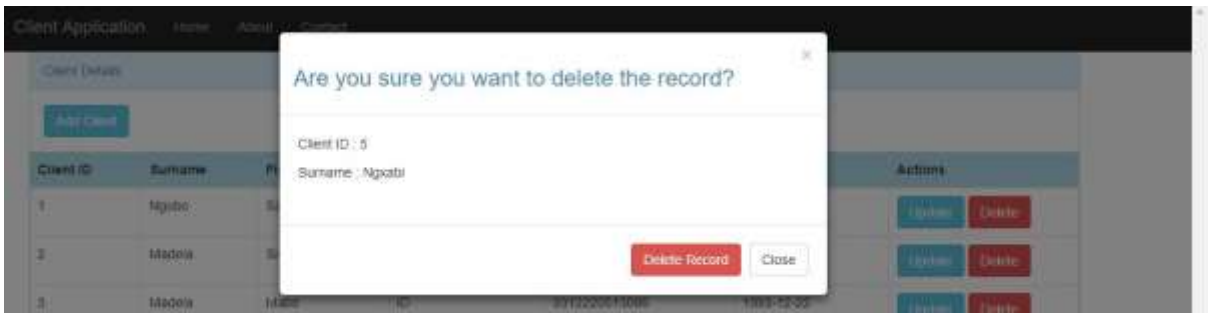
Currently we are using simple JavaScript alert to display notification messages for each operations. but in upcoming article we will add toaster notification for sexy and beautiful alerts.

Next, let us update any client data, press Update button for client you want to update.



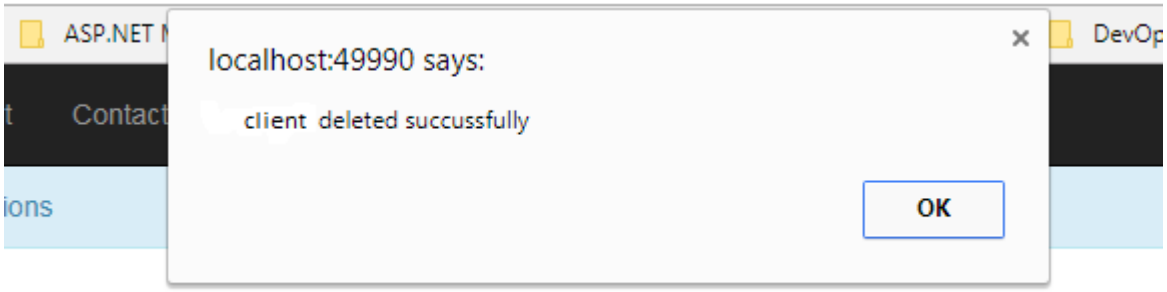
Update the field values and press Update button to save the data. It will display alert for successful update operation.

Now, let us delete client, it will ask for confirmation before deleting the record

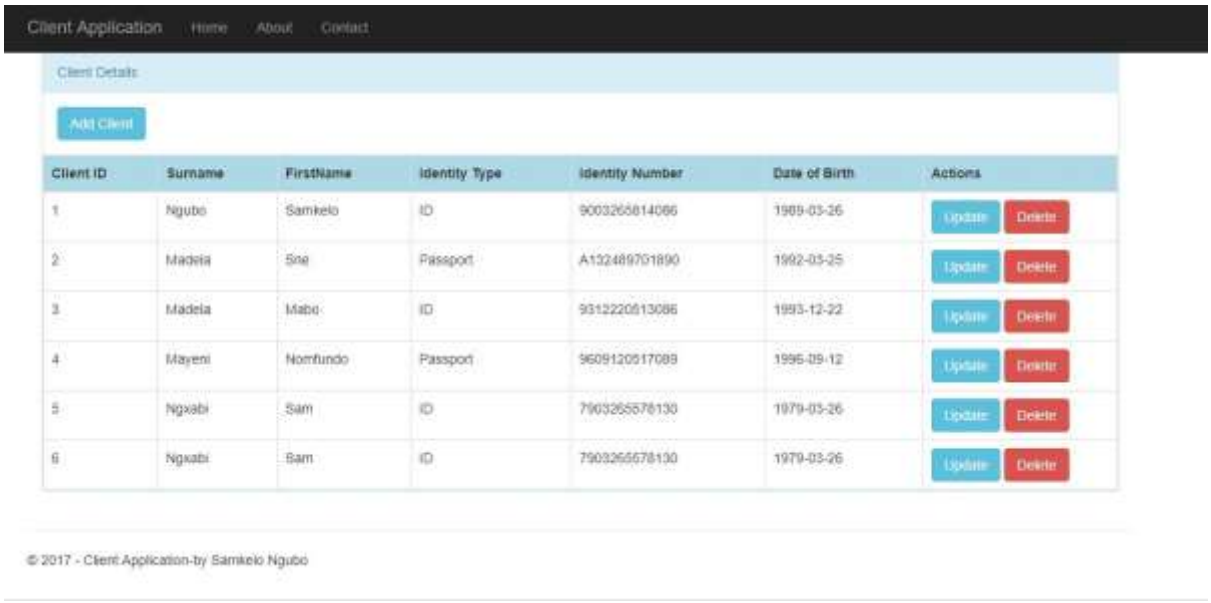


once you confirm, it will delete the record by showing alert.

Samkelo Ngubo Angular Js Solution



After deletion, client list will look like



Thank you.