Part 1 - How to Basic CRUD operations using Jquery ajax and modal popup in ASP.NET MVC4.

[Wednesday, May 20, 2015](http://www.dotnetawesome.com/2015/05/part-1-how-to-basic-crud-operations-using-jquery-ajax-model-popup-mvc4.html)[Sourav Mondal](https://draft.blogger.com/profile/14432473893011506876) [Ajax](http://www.dotnetawesome.com/search/label/Ajax?max-results=9) | [CRUD](http://www.dotnetawesome.com/search/label/CRUD?max-results=9) | [Jquery](http://www.dotnetawesome.com/search/label/Jquery?max-results=9) | [model popup](http://www.dotnetawesome.com/search/label/model%20popup?max-results=9) | [MVC](http://www.dotnetawesome.com/search/label/MVC?max-results=9) | [Retrive data](http://www.dotnetawesome.com/search/label/Retrive%20data?max-results=9)

Introduction

In this post, I am going to implement Basic CRUD (Create, Read, Update, Delete) operations using Jquery ajax and modal popup in ASP.NET MVC4 application.  
I have split the entire application split into following 3 parts for making things more simple and understandable specially for beginners.

* Part 1 : Perform read functionality of CRUD Operation.
* [Part 2 : Perform create and update functionality of CRUD Operation.](http://dotnetawesome.com/2015/05/part-2-how-to-basic-crud-operations-using-jquery-ajax-model-popup-mvc4.html)
* [Part 3 : Perform delete functionality of CRUD Operation.](http://www.dotnetawesome.com/2015/05/part-1-how-to-basic-crud-operations-using-jquery-ajax-model-popup-mvc4.html)

Step - 1 : Create New Project. Go to File > New > Project > Select asp.net MVC4 web application > Entry Application Name > Click OK > Select Basic Application > Select view engine Razor > OK

Step-2: Add a Database.

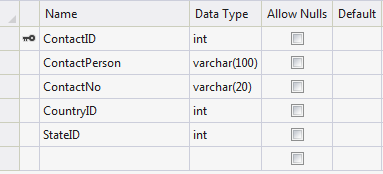
Go to Solution Explorer > Right Click on App\_Data folder > Add > New item > Select SQL Server Database Under Data > Enter Database name > Add.

Step-3: Create 3 tables for store Country, States and Contact information.

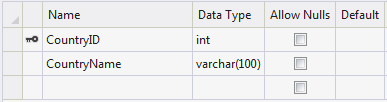
Open Database > Right Click on Table > Add New Table > Add Columns > Save > Enter table name > Ok.

In this example, I have used 3 tables as below

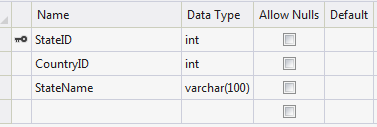
**Contacts**

[](http://4.bp.blogspot.com/-4AjoMwztoxs/VRy2Y0FtvBI/AAAAAAAABfs/jKhaSU-hk3I/s1600/ContactsDB.PNG)

**Countries**

[](http://2.bp.blogspot.com/-H5JVgEHd5OQ/VRy2pBB2wpI/AAAAAAAABf0/gAbEmrBy9Kk/s1600/CountryDB.PNG)

**States**

[](http://3.bp.blogspot.com/-ZvbU-NeNpDI/VRy2wbmw40I/AAAAAAAABf8/5yfO83I_6LM/s1600/StatesDB.PNG)

Step-4: Add Entity Data Model.

Go to Solution Explorer > Right Click on Project name form Solution Explorer > Add > New item > Select ADO.net Entity Data Model under data > Enter model name > Add.  
A popup window will come (Entity Data Model Wizard) > Select Generate from database > Next >  
Chose your data connection > select your database > next > Select tables > enter Model Namespace > Finish.

Step-5: Create a Folder

Here I have created a folder Named "ExtendedClass" in which we will create a class  
Go to Solution Explorer > Right Click on Project name form Solution Explorer > Add > New Folder > Rename Folder.

Step-6: Create a (partial) Class for extend 2 fileds CountryName and StateName. Here I have also added MetaData Class for Apply Validation on Contact Model.

Here I have created a (partial) class named "Contact"  
Go to Solution Explorer > Right Click on the Folder > Add > Class > Enter class name > Add.  
  
*[N:B: I have changed namespace of the class "MVCPopupCRUD.ExtendedClass" to "MVCPopupCRUD" and make it partial class]*

2. using System;
3. using System.Collections.Generic;
4. using System.ComponentModel.DataAnnotations;
5. using System.Linq;
6. using System.Web;
8. namespace MVCPopupCRUD
9. {
10. [MetadataType(typeof(ContactMetaData))]
11. public partial class Contact
12. {
13. public string CountryName { get; set; }
14. public string StateName { get; set; }
15. }
17. public class ContactMetaData
18. {
19. [Required(ErrorMessage= "Contact Name required", AllowEmptyStrings = false)]
20. [Display(Name="Contact Person")]
21. public string ContactPerson { get; set; }
23. [Required(ErrorMessage="Contact No required", AllowEmptyStrings=false)]
24. [Display(Name="Contact No")]
25. public string ContactNo { get; set; }
27. [Required(ErrorMessage="Country required")]
28. [Display(Name="Country")]
29. public int CountryID { get; set; }
31. [Required(ErrorMessage="State required")]
32. [Display(Name="State")]
33. public int StateID { get; set; }
34. }
35. }

Step-7: Add a javascript file.

Go to Solution Explorer > Right Click on Scripts folder form Solution Explorer > Add new item...>Select JavaScript File> Enter file name (here "dotnetawesome.CRUDContacts.js") > Add.  
Here we will write js code for open popup, create new contact, read, update and delete existing contact(s)

Step-8: Add a javascript function for show contact list . (read operation of CRUD)

2. $(document).ready(function () {
4. //Populate Contact
5. LoadContacts();
6. });
7. function LoadContacts() {
8. $('#update\_panel').html('Loading Data...');
10. $.ajax({
11. url: '/home/GetContacts',
12. type: 'GET',
13. dataType: 'json',
14. success: function (d) {
15. if (d.length > 0) {
16. var $data = $('<table></table>').addClass('table table-responsive table-striped');
17. var header = "<thead><tr><th>Contact Person</th><th>Contact No</th><th>Country</th><th>State</th><th></th></tr></thead>";
18. $data.append(header);
20. $.each(d, function (i, row) {
21. var $row = $('<tr/>');
22. $row.append($('<td/>').html(row.ContactPerson));
23. $row.append($('<td/>').html(row.ContactNo));
24. $row.append($('<td/>').html(row.CountryName));
25. $row.append($('<td/>').html(row.StateName));
26. $row.append($('<td/>').html("<a href='/home/Save/" + row.ContactID + "' class='popup'>Edit</a>&nbsp;|&nbsp;<a href='/home/Delete/" + row.ContactID + "' class='popup'>Delete</a>"));
27. $data.append($row);
28. });
30. $('#update\_panel').html($data);
31. }
32. else {
33. var $noData = $('<div/>').html('No Data Found!');
34. $('#update\_panel').html($noData);
35. }
36. },
37. error: function () {
38. alert('Error! Please try again.');
39. }
40. });
42. }

Step-9: Add a new Controller.

Go to Solution Explorer > Right Click on Controllers folder form Solution Explorer > Add > Controller > Enter Controller name (Home) > Select Templete "empty MVC Controller"> Add.

Step-10: Add new action into your controller for get the view for show contact list.

Here I have used "Index" Action for show data. Please write this following code

2. public ActionResult Index()
3. {
4. return View();
5. }

Step-11: Add view for the Action & design.

Right Click on Action Method (here right click on index action) > Add View... > Enter View Name > Select View Engine (Razor) > Add. **Complete View**

2. @{
3. ViewBag.Title = "Contact List";
4. }
6. <h2>Contact List</h2>
7. @Html.ActionLink("Add New Contact","Save","Home",null,new{@style="font-size:22px;", @class="popup"})
9. <div id="update\_panel">
11. </div>
12. @\* Add Jquery UI Css \*@
13. <link rel="stylesheet" href="//[code.jquery.com/ui/1.11.4/themes/smoothness/jquery-ui.css](http://code.jquery.com/ui/1.11.4/themes/smoothness/jquery-ui.css)" />
14. <style>
15. /\*This is optional, I have added this for make looks good\*/
16. html, body,footer, #body {
17. background-color:#fff;
18. }
19. .ui-widget-header {
20. border: none !important;
21. background: none !important;
22. color: #222222;
23. font-weight: bold;
24. }
25. .ui-state-default, .ui-state-hover {
26. border: none !important;
27. background: none !important;
28. }
29. .ui-dialog{
30. webkit-box-shadow: 0 5px 15px rgba(0,0,0,.5);
31. box-shadow: 0 5px 15px rgba(0,0,0,.5);
32. }
33. h2 {
34. margin-top:0px;
35. font-size:30px;
36. }
38. .success {
39. color:green;
40. }
41. .failed {
42. color:red;
43. }
45. </style>
46. @section Scripts{
47. @\*//Add Jquery UI JS\*@
48. <script src="//[code.jquery.com/ui/1.11.4/jquery-ui.js](http://code.jquery.com/ui/1.11.4/jquery-ui.js)"></script>
49. <script src="~/Scripts/dotnetawesome.CRUDContacts.js"></script>
50. }

Optional: Here I have added Bootstrap css in the layout page for Responsive design.

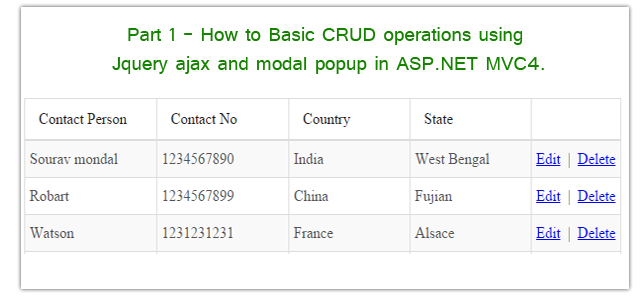
Step-12: Add another action into your controller for return contact list as Json Data

Here I have used "GetContacts" Action for fetch data. Please write this following code

2. public JsonResult GetContacts()
3. {
4. List<Contact> all = null;
6. using (MyDatabaseEntities dc = new MyDatabaseEntities())
7. {
8. var contacts = (from a in dc.Contacts
9. join b in dc.Countries on a.CountryID equals b.CountryID
10. join c in dc.States on a.StateID equals c.StateID
11. select new
12. {
13. a,
14. b.CountryName,
15. c.StateName
16. });
17. if (contacts != null)
18. {
19. all = new List<Contact>();
20. foreach (var i in contacts)
21. {
22. Contact con = i.a;
23. con.CountryName = i.CountryName;
24. con.StateName = i.StateName;
25. all.Add(con);
26. }
27. }
28. }
30. return new JsonResult { Data = all, JsonRequestBehavior = JsonRequestBehavior.AllowGet };
31. }

Step-13: Run Application.

[DOWNLOAD](http://demo.dotnetawesome.com/mvc/part-1-how-to-basic-crud-operations-using-jquery-ajax-model-popup-mvc4)    [LIVE DEMO](http://demo.dotnetawesome.com/mvc/part-1-how-to-basic-crud-operations-using-jquery-ajax-model-popup-mvc4)

[](http://3.bp.blogspot.com/-hBTxNhBXCe4/VVzJOiJI69I/AAAAAAAABiQ/epl6eGfSwkw/s1600/Part1CurdMVCPopup.png)

**Next Part :**  
[Part 2 : Perform create and update functionality of CRUD Operation.](http://dotnetawesome.com/2015/05/part-2-how-to-basic-crud-operations-using-jquery-ajax-model-popup-mvc4.html)  
  
**Related Post :**

* [Part 1 - Implement basic CRUD operation in ASP.NET MVC](http://dotnetawesome.com/2015/05/part-1-how-to-implement-basic-crud-functionality-with-EF-and-MVC4-application.html)
* [Part 1 - Implement basic CRUD operation in ASP.NET webform](http://dotnetawesome.com/2015/04/how-to-implement-basic-crud-functionality-entity-framework-aspnet-webforms.html)